

Université de Montréal

**Réutilisation des processus d'affaires pour le développement de  
systèmes d'information**

par

Guitta Bou Jaoude

Département d'informatique et de recherche opérationnelle  
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures  
en vue de l'obtention du grade de  
Maître ès Sciences (M.Sc.)  
en informatique

Juillet 2004

© Guitta Boujaoude, 2004



QA

76

U54

2004

V.048

**Direction des bibliothèques**

**AVIS**

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

**NOTICE**

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal  
Faculté des études supérieures

Ce mémoire intitulé :

Réutilisation des processus d'affaires pour le développement  
de systèmes d'information

présenté par:

Guitta Bou Jaoude

a été évalué par un jury composé des personnes suivantes:

Petko Valtchev

(Président-rapporteur)

---

El Mostapha Aboulhamid

(Directeur de recherche)

---

Hafedh Mili

(Codirecteur)

---

Houari Sahraoui

(Membre du jury)

Mémoire accepté le: 28 octobre 2004

## Sommaire

Pour comprendre le fonctionnement d'un système d'information, nous devons le placer dans le contexte des *processus d'affaires* qu'il doit supporter. En effet, les entreprises n'agissent pas de façon improvisée, mais font face à leur quotidien selon des processus bien établis et généralement bien suivis. Certains processus d'affaires ne dépendent pas du domaine d'affaires de l'industrie en question, et peuvent être applicables à différentes industries. Récemment, il y a eu beaucoup d'efforts, tant au niveau de la recherche qu'au niveau des comités de standardisation, visant à codifier et à réutiliser les processus d'affaires génériques [SAP, 2003], [ebXML, 2001], [Coad et al., 1999], [Malone et al., 1999]. Dans la mesure où les systèmes d'information reflètent les processus qu'ils supportent, plusieurs travaux ont tenté d'associer des composants logiciels génériques à des fragments de processus d'affaires [IBM, 2003]. Ces efforts rendent possible une approche de développement des systèmes d'informations basée sur la réutilisation des modèles de processus et des composants logiciels qui leurs sont associés. Une telle approche permettrait de réduire le temps de conception des systèmes d'information, d'en améliorer la qualité et d'en faciliter la maintenance.

Nous visons une méthodologie qui permet le passage du modèle générique d'un processus d'affaires à un modèle d'analyse indépendant des plate-formes. Cette recherche constitue un pas de plus vers l'élaboration d'une méthodologie de développement de systèmes d'information par réutilisation de composants d'affaires génériques. Pour ce faire, nous avons proposé une représentation des modèles de processus génériques qui prend en compte les variabilités. Nous avons aussi proposé une approche systématique pour la classification et la spécialisation de modèles de processus d'affaires. Le résultat de cette spécialisation est un modèle de processus spécifique, reconnu sous le nom de CIM (« *computation independent model* ») dans la méthodologie MDA (« *Model Driven Architecture* ») [MDA, 2001], qui peut supporter la génération du modèle d'analyse correspondant indépendamment des plate-formes. La génération en tant que telle dépasse le cadre de cette recherche. Une étude de cas est présentée pour illustrer notre méthodologie et pour mettre en évidence les problèmes de recherche ouverts.

**MOTS-CLEF:** Processus d'affaires, réutilisation, modèles génériques, spécialisation, MDA (Model driven architecture).

## Abstract

To understand what an information system is supposed to do, we need to put it into the context of the *business processes* that it is meant to support. Indeed, the enterprises do not act in an improvised way, but according to processes well built and generally strictly followed. Some business processes do not depend on the specific business area of the enterprise, and apply across industries. Recently, many research and standardization efforts have attempted to codify and reuse business processes [SAP, 2003], [ebXML, 2001], [Coad et al., 1999], [Malone et al., 1999]. To the extent that information systems embody the business processes that they support, a number of researchers have attempted to associate individual software components with the process fragments that they support [IBM, 2003]. This raises the possibility of developing information systems by first identifying the business processes that they are meant to support, and then by reusing the corresponding software components. Such a development methodology would minimize development time, increase system quality, and facilitate its maintenance.

Our research aims at developing a methodology for going from a generic, industry-wide business process model down to an enterprise-specific, platform independent analysis model. This methodology constitutes the first step towards an information system development method based on the reuse of generic business (software) components. To this end, we first proposed a representation for generic business processes that covers the various views of a business process and that takes into account concepts such as genericity and variabilities. We have also developed a systematic method for the classification and incremental specialization of business processes that would enable us to derive an enterprise and industry-specific business process from a generic business process. The specific process, referred to as *computation independent model* (CIM) in the Model Driven Architecture (MDA [MDA, 2001]), would be amenable to transformation into an enterprise-specific, but platform independent analysis model. The actual transformation is beyond the scope of this thesis. However, a case-study is used to illustrate our methodology and to highlight open research questions.

**KEYWORDS:** Business processes, reuse, generic models, specialization, the Model-Driven Architecture.

# Table des matières

Réutilisation des processus d'affaires pour le développement de systèmes d'information-----	i
<b>CHAPITRE 1. INTRODUCTION -----</b>	<b>1</b>
1.1 La réutilisation -----	2
1.1.1 Approche par composants -----	2
1.1.2 Approche par génération -----	3
1.1.3 Approche hybride-----	4
1.2 MDA -----	5
1.2.1 Principes-----	5
1.2.2 Les différents modèles-----	6
1.3 Objectif de notre travail -----	8
1.4 Organisation du mémoire -----	10
<b>CHAPITRE 2. LANGAGES DE MODÉLISATION DE PROCESSUS D'AFFAIRES-----</b>	<b>11</b>
2.1 Introduction-----	11
2.2 Les processus d'affaires -----	11
2.2.1 Qu'est ce qu'un processus d'affaires-----	12
2.3 Langages de modélisation des processus d'affaires-----	14
2.3.1 Critères de classification des langages de modélisation-----	14
2.3.2 Classification des langages de modélisation de processus d'affaires-----	16
2.4 Langages traditionnels de modélisation de processus -----	18
2.4.1 IDEF -----	18
2.4.2 Réseau de Petri -----	21
2.4.3 Role activity Diagram (RAD) -----	23
2.4.4 Event Process Chains (EPC)-----	25
2.4.5 Ressource Event Agent (REA)-----	26
2.4.6 BPML -----	27
2.5 Langage de modélisation de workflow (WPDL) -----	29
2.6 Langages d'intégration de processus-----	31
2.6.1 RosettaNet -----	31
2.6.2 Ebxml/BPSS -----	33
2.6.3 BPEL4WS-----	35
2.7 Langages de modélisation orientée objet-----	38

2.7.1	<i>Enterprise Distributed Object Computing (EDOC)</i>	39
2.7.2	<i>Langage Unifié de Modélisation version 2 (UML 2.0)</i>	43
2.8	Comparaison entre les langages de modélisation	45
2.9	Conclusion	47
<b>CHAPITRE 3. INITIATIVES DE MODÉLISATION DE PROCESSUS D’AFFAIRES GÉNÉRIQUES</b>		<b>48</b>
3.1	Introduction	48
3.2	Cadre de classification des initiatives de modélisation	48
3.3	Initiatives orientées description de processus d’affaires internes	51
3.3.1	<i>Initiative pour répertoire des processus d’affaires: MIT Process Handbook Project</i>	51
3.3.2	<i>Initiatives orientées répertoriage des composants d’affaires</i>	56
3.3.3	<i>Initiatives orientées cadre d’applications</i>	62
3.4	Initiatives orientées vers l’intégration des processus inter-entreprises	73
3.4.1	<i>L’initiative RosettaNet</i>	73
3.4.2	<i>Initiative EBXML</i>	75
3.5	Discussion	80
3.6	Conclusion	81
<b>CHAPITRE 4. APPROCHE PROPOSÉE</b>		<b>82</b>
4.1	Introduction	82
4.2	Une méthodologie d’adaptation de processus génériques	82
4.3	Classification descriptive	84
4.4	Représentation des processus génériques	86
4.4.1	<i>Critères de représentation du modèle générique</i>	87
4.4.2	<i>Méta-modèle élaboré du processus d’affaires</i>	87
4.4.3	<i>Vues de représentation du modèle générique.</i>	89
4.5	Procédure de spécialisation de processus génériques	95
4.5.1	<i>Approches par questions</i>	96
4.5.2	<i>Principes de notre approche</i>	98
4.5.3	<i>Les questions proposées</i>	101
4.6	Exemple de spécialisation	105
4.6.1	<i>Les questions appliquées au processus générique d’approvisionnement</i>	106
4.6.2	<i>Exemple de situation</i>	112
4.6.3	<i>Instantiation</i>	117
4.7	Conclusion	118
<b>CHAPITRE 5. CONCLUSION</b>		<b>119</b>



5.1	Résumé et conclusions	119
5.2	Directions futures	120
	<b>BIBLIOGRAPHIE</b>	<b>122</b>
	<b>ANNEXE 1</b>	<b>I</b>
	<b>ANNEXE 2</b>	<b>IV</b>

## Liste des tableaux

TABLEAU 2.1: COMPARAISON DE LANGAGES DE MODÉLISATION DE PROCESSUS EN TERME DE PORTÉE.....	46
TABLEAU 2.2: UTILISATION POTENTIELLE DES DIFFÉRENTS LANGAGES. ....	47
TABLEAU 4.1: TABLEAU DES QUESTIONS ET DES VUES .....	105
TABLEAU 4.2 : TABLEAU D'INTERPRÉTATION DE QUESTIONS SPÉCIALISÉES .....	114

## Liste des figures

FIGURE 1-1: ARCHITECTURE DE MDA.....	6
FIGURE 2-1: META MODÈLE PRIMAIRE DU PROCESSUS D’AFFAIRES.....	13
FIGURE 2-2: MODÉLISATION DU MONDE RÉEL ET PSEUDO- RÉEL (FIGURE TIRÉE DE [ISODA, 2001]).....	16
FIGURE 2-3 : ANATOMIE D’UNE FONCTION DANS IDEF0 .....	19
FIGURE 2-4: DIAGRAMME FONCTIONNEL AVEC IDEF0 .....	19
FIGURE 2-5: UN DIAGRAMME DE PROCESSUS AVEC IDEF3. ( EXEMPLE TIRÉ DE [MAYER ET AL. 1995]).....	20
FIGURE 2-6: SCHÉMA DE TRANSITION D’UN OBJET (DEMANDE ACHAT) AVEC IDEF3. ....	21
FIGURE 2-7 : UN PROCESSUS MODÉLISÉ AVEC LE RÉSEAU DE PETRI CLASSIQUE .....	22
FIGURE 2-8: NOTATION DE RAD [MURDOCH, 1998].....	24
FIGURE 2-9: UN PROCESSUS MODÉLISÉ AVEC EPC .....	25
FIGURE 2-10 : LE MÉTA MODÈLE DE REA ( TIRÉ DE [ MCCARTHY, 1982]).....	27
FIGURE 2-11: MÉTA- MODÈLE DE WFMS [WFMC, 1999A].....	30
FIGURE 2-12: STRUCTURE DU STANDARD ROSETTANET.....	32
FIGURE 2-13: PIP POUR LE CHANGEMENT DANS UNE COMMANDE EXISTANTE (TIRÉ DE [ROSETTANET, 2003]).....	33
FIGURE 2-14 : UNE COLLABORATION BINAIRE SIMPLE AVEC BPSS DE EBXML. ....	35
FIGURE 2-15: UNE COMPOSITION .....	40
FIGURE 2-16 : NOTATION D’UN COMPOSANT AVEC EDOC .....	40
FIGURE 2-17: NOTATION D’UNE ACTIVITÉ AVEC EDOC.....	42
FIGURE 2-18: STRUCTURE D’UN COMPOSANT ÉLABORÉ AVEC UML2.0 .....	44
FIGURE 2-19: UNE ACTIVITÉ COMPOSÉE DES ACTIONS AVEC UML 2.0.....	44
FIGURE 3-1: CLASSIFICATION DES INITIATIVES DE MODÉLISATION .....	50
FIGURE 3-2 : BOUSSOLE DE PROCESSUS (« PROCESS COMPASS »)(FIGURE EXTRAITE DE [MALON ET AL. 1999]) .....	52
FIGURE 3-3 : PROCESSUS DE VENTES ALTERNATIFS À DIFFÉRENTS NIVEAUX DE DÉCOMPOSITION ET DE SPÉCIALISATION [MALONE, 1999].....	53
FIGURE 3-4: LES SPÉCIALISATIONS DU PROCESSUS « VENTES » (TIRÉ DE [MIT, 2004]).....	54
FIGURE 3-5 : PLAN SIMPLIFIÉ DU MANUEL MIT .....	55
FIGURE 3-6 : ÉCHANTILLONS DES PATRONS FOURNIS PAR COAD [COAD, 1997].....	58
FIGURE 3-7: DIAGRAMME D’ACTIVITÉS ÉTENDUE PAR HANS ET AL. [HANS, 2000].....	60
FIGURE 3-8 : PATRON « CONTRAT » [HANS, 2000].....	61
FIGURE 3-9 : PATRON « INSTANCE PROCESSUS-PROCESSUS » [HANS, 2000].....	61
FIGURE 3-10 : COMPOSANT INDÉPENDANT DU DOMAINE [LEFEBVRE ET AL., 1999].....	64

FIGURE 3-11: LE COMPOSANT PROCESSUS DE VENTES (FIGURE EXTRAITE DE [COAD ET AL., 1999]) .....	65
FIGURE 3-12 : ARCHITECTURE GÉNÉRALE DE L'APPROCHE IBM SAN FRANCISCO. ....	66
FIGURE 3-13 : CLASSIFICATION DES CBO (FIGURE EXTRAITE DE [HENN, 1998]).....	68
FIGURE 3-14: LES COMPOSANTS DE IBM SAN FRANCISCO ( FIGURE EXTRAITE DE [HENN, 1998]) .....	68
FIGURE 3-15: CHAÎNE DE VALEUR DE PORTER ET LES MODÈLES DE SAP.....	71
FIGURE 3-16 : DOCUMENT COMMANDE (EBXML).....	76
FIGURE 3-17 : EXEMPLE D'UN COMPOSANT COMMUN D'EBXML.....	77
FIGURE 3-18 : ÉCHANTILLON DU CATALOGUE DE PROCESSUS D'AFFAIRES COMMUN D'EBXML [EBXML, 2001D] .....	78
FIGURE 4-1 : LES ÉTAPES DE L'APPROCHE .....	83
FIGURE 4-2 : EXEMPLE DE NAVIGATION.....	86
FIGURE 4-3 : MÉTA-MODÈLE ÉLABORÉ D'UN PROCESSUS D'AFFAIRES. ....	88
FIGURE 4-4: MODÉLISATION D'UN PROCESSUS RÉUTILISABLE À L'AIDE DE QUATRE VUES	89
FIGURE 4-5 : SPÉCIFICATION UML2.0 .....	94
FIGURE 4-6: QUESTIONS BIAIT [CARLSON, 1980].....	96
FIGURE 4-7 : VUE INFORMATIONNELLE SPÉCIFIQUE À UN DOMAINE. ....	98
FIGURE 4-8 : REPRÉSENTATION GÉNÉRIQUE DU MODÈLE INFORMATIONNEL D'UNE COMMANDE .....	99
FIGURE 4-9 : COMBINAISONS DES RÉPONSES .....	100
FIGURE 4-10: ZONES DES QUESTIONS DANS LE MÉTA-MODÈLE.....	101
FIGURE 4-11 : MODÈLE ORGANISATIONNEL GÉNÉRIQUE.....	106
FIGURE 4-12 : MODÈLE FONCTIONNEL D'UN PROCESSUS D'APPROVISIONNEMENT .....	108
FIGURE 4-13: MODÈLE COMPORTEMENTAL POUR TRAITER UNE COMMANDE .....	112
FIGURE 4-14 : MODÈLE ORGANISATIONNEL DU MAGASIN DE DISQUES .....	115
FIGURE 4-15 : MODÈLE FONCTIONNEL DU PROCESSUS D'APPROVISIONNEMENT D'UN MAGASIN DE DISQUES .....	115
FIGURE 4-16 : MODÈLE COMPORTEMENTAL DU PROCESSUS D'APPROVISIONNEMENT D'UN MAGASIN DE DISQUES .....	116
FIGURE 4-17 : MODÈLE INFORMATIONNEL DU PROCESSUS D'APPROVISIONNEMENT D'UN MAGASIN DE DISQUES .....	117

*À la mémoire de mon père ...*

*À ma famille ...*

## Remerciements

Mes plus grands remerciements reviennent à Monsieur Hafedh Mili, professeur à l'Université de Québec à Montréal, pour m'avoir donné l'occasion de faire partie du laboratoire LATECE (LABoratoire de recherche sur les TEchnologies du Commerce Electronique), ainsi que pour avoir supervisé cette recherche. Sa grande disponibilité, sa persévérance, son souci du détail ainsi que ses encouragements m'ont beaucoup aidée tout au long de ce travail. Que ce mémoire soit le modeste témoignage de ma reconnaissance et de mon admiration.

Je remercie vivement Monsieur El Mostapha Aboulhamid, professeur à l'Université de Montréal, de m'avoir co-dirigée durant ma recherche.

Je remercie tous mes collègues du groupe LATECE, en particulier Céline, Ghizlane et Radhouane, qui ont supporté mes petites angoisses et mes remises en question, mes amis de l'université de Montréal pour leur patience et leur gentillesse surtout: Amine, Charles, Kamal, Luc, Nourchène, et spécialement Sarita. À tous mes amis, je tiens à vous remercier pour les bons moments que j'ai pu partager avec vous. Je pense plus particulièrement à Rima, Ghada et Imad.

Enfin, je tiens à exprimer ma profonde gratitude à toute ma famille, qui m'a toujours supportée, surtout dans les moments les moins faciles. Merci à vous tous, particulièrement Aldo et Maha.

Joseph, merci pour ton extrême patience et pour ton soutien indéfectible dans tous les moments. Réna, tu es une sœur formidable, je ne sais pas comment t'exprimer ma profonde gratitude mais je peux juste te dire « c'est toi le soleil ☺!!! ». Comme je ne peux pas oublier de remercier vos enfants (Mike, Georgio et Jonathan) qui m'entourent avec un amour sincère et profond.

# Chapitre 1. Introduction

Durant les trente dernières années, la pratique du développement de logiciels a évolué de façon remarquable, mais elle continue toujours à faire face à des défis aussi importants. Des progrès ont été réalisés dans plusieurs domaines, dont les langages de programmation, les techniques d'analyse et de conception, les infrastructures de déploiement d'applications, mais aussi les pratiques de gestion de projets informatiques. Si l'on continue encore de parler de dépassement de coût et de délais de livraison, de faible qualité, ou encore de difficultés d'entretien des logiciels aujourd'hui, c'est parce que les logiciels que l'on développe aujourd'hui sont nettement plus complexes que ceux que la technologie des décennies précédentes permettaient d'envisager.

McIlroy, le père des termes « génie logiciel » et « crise du logiciel », a depuis longtemps rêvé d'une industrie de logiciels qui comporterait, à la base, une industrie de composants logiciels réutilisables et inter-opérables que l'on puisse assembler pour construire des logiciels sur mesure. Un grand nombre d'avancées dans le domaine du génie logiciel peuvent être vues sous cet angle là. Les premières librairies mathématiques du langage FORTRAN n'en étaient que la première étape. Avec la complexité grandissante des applications, les chercheurs se sont vite rendus compte que ces librairies étaient « trop peu, trop tard ». D'abord, la taille des logiciels est telle que ces librairies n'en représentaient qu'une infime partie. Mais aussi, le codage représente une partie décroissante du cycle de vie du logiciel—et donc du coût des logiciels. Cette poussée vers des composants réutilisables de plus grande taille, mais aussi plus tôt dans le cycle de développement, nous a donné les patrons de conception (réutilisation de la *conception* d'un petit *agrégat logique* de classes), les cadres d'applications (réutilisation de la *conception* et du *codage* d'un *agrégat fonctionnel* de classes), les patrons d'analyse, et les modèles génériques d'analyse (produits par l'*analyse de domaines*).

Nous croyons que la réutilisation doit commencer par l'analyse des besoins. En même temps, nous devrions être capables de relier facilement et systématiquement l'expression des besoins aux artefacts logiciels des étapes subséquentes, à commencer par les modèles d'analyse. Or, pour décrire les besoins fonctionnels que doit remplir un système d'information d'entreprise, nous devons placer ce dernier dans le contexte du processus d'affaires qu'il doit supporter. Ainsi, notre

recherche vise la génération de modèles d'analyse spécifiques à une entreprise donnée, à partir de modèles de processus d'affaires génériques.

La section 1.1 présente brièvement les grandes familles des approches de réutilisation de logiciel. La section 1.2 présente la méthodologie MDA (Model-Driven Architecture) de l'OMG, qui essaie de promouvoir un modèle de développement de logiciels basé sur la transformation de modèles. La section 1.3 présente de façon plus détaillée l'objectif de cette recherche. L'organisation du mémoire est présentée dans la section 1.4.

## **1.1 La réutilisation**

La réutilisation est le fait de réutiliser des éléments déjà existants, facile à récupérer et modifiable en fonction de nouveaux besoins que l'on souhaite traiter avec leurs réutilisations. En fait, la réutilisation constitue le paradigme de résolution de problème par défaut : on essaie de résoudre les problèmes auxquels on fait face d'abord en repérant dans notre « banque » de paires <problème, solution>, celle dont le problème se rapproche le plus du problème en question, dans l'espoir de pouvoir en adapter la solution, à moindre frais. La réutilisation de logiciel permet de minimiser les coûts de développement tout en augmentant la fiabilité des systèmes conçus. Elle permet de réaliser a) des gains en productivité, b) des gains en qualité et c) des gains en inter-opérabilité. Cependant, la réutilisation peut être coûteuse et risquée comme elle n'est pas rentable à court termes. Pour être efficace elle ne doit pas être occasionnelle et doit structurer le projet de développement [Mili et al., 2002].

La réutilisation de logiciel est catégorisée en trois grandes familles que nous allons présenter dans ce qui suit: approche par composants, approche par génération et approche hybride.

### **1.1.1 Approche par composants**

Le principe qui sous-tend cette approche est la construction de logiciels en réutilisant des artefacts (produits) logiciels déjà développés. Une première génération de cette approche s'est centrée sur la réutilisation du code source [McIlory, 1969], mais cette génération a révélé différents problèmes. Le problème majeur résidait en l'intégration du code réutilisé au sein du code existant. Ce problème a conduit plus tard à la proposition du concept d'unités de code réutilisables (ex unités Pascal) [Sodhi, 1998]. Dans un deuxième temps, avec l'arrivée massive du concept objet, la réutilisation de code



logiciel a évoluée, avec les mécanismes d'interfaçage et d'héritage (ex: bibliothèque Java). Cependant, la réutilisation au niveau codage uniquement a révélé une large insuffisance, de par le fait que celui-ci représente juste une partie infime du coût de développement. Différentes recherches ont tenté de réutiliser les livrables des étapes qui précèdent le codage. Avec les progrès dans la matière, on s'est intéressé à la réutilisation tout au long du cycle de vie d'un logiciel et principalement au début du cycle [Mili et al., 1995]. Ainsi, différents types d'artéfacts réutilisables dans les différents niveaux conceptuels existent actuellement. Notons entre autres les patrons de conceptions et d'analyses, les modèles de spécification des besoins et les composants exécutables. Aujourd'hui, le développement des logiciels à base de composants (Component-Based Software Engineering) évolue rapidement surtout avec les technologies web et les langages objets qui ont facilité la distribution, la recherche, l'interopérabilité des composants sur internet et en particulier des COTS (Commercial Off The Shelf). Le nombre et la diversité de ces technologies (ex CORBA, J2EE, ActiveX, COM/DCOM, etc.) sont un signe de l'effervescence des travaux dans le domaine mais aussi de leur disparité.

### **1.1.2 Approche par génération**

Avec cette approche, au lieu de réutiliser le *produit (artéfacts)* de développement, l'emphase est mise sur la codification du *processus-même* de développement dans des outils que l'on applique à différentes exigences [Ellis et al., 1984]. L'approche transformationnelle et la programmation automatique sont deux techniques qui s'inscrivent dans cette approche.

L'approche transformationnelle est le développement d'un programme en appliquant une suite de transformations à une spécification. Dans un premier temps, avec l'approche transformationnelle, l'objectif était d'appliquer des transformations formelles en se basant sur des spécifications et des opérations mathématiques. Peu de grands systèmes ont été développés en utilisant cette approche. Une des raisons est que les langages de spécifications sont à base mathématique et non à base de langages humains (naturels). Avec le langage mathématique l'ambiguïté n'est pas permise tandis qu'avec le langage naturel une expression peut prendre différents sens en fonction du contexte, de l'environnement. Par conséquent, tous les participants au développement du logiciel doivent avoir une base mathématique et logique forte pour comprendre et créer des spécifications mathématiques formelles, ce qui n'est pas réaliste [Sommerville, 1996]. Avec l'évolution de ce domaine, différentes recherches ont utilisé les mêmes principes de cette approche mais en effectuant quelques amendements. Par exemple, en fonction de l'objectif de l'application, le langage de spécification

peut aller d'un langage naturel structuré à un langage purement descriptif tel que les notations mathématiques ou la logique des prédicats.

La programmation automatique (aussi reconnue sous le nom de « Metaprogramming » [Levy, 1986]) ou l'approche de générateur d'application est la construction d'un système logiciel à partir d'un ensemble réduit de paramètres, d'un outil (le générateur) ou un ensemble d'outils intégrés et d'un langage particulier de haut niveau. Ces types de générateurs doivent être faciles à utiliser, compréhensibles par des non-programmeurs et rapides à générer des applications [Martin, 1985]. Ces conditions font que les générateurs dépendent fortement des domaines d'applications. Par conséquent ils ont été appliqués à de petits problèmes jouets ou bien à des domaines restreints. Parmi les différents travaux à petite échelle identifiés, notons: a) La génération de compilateurs, par exemple, « yacc » dans l'environnement Unix permet de générer un compilateur suivant une définition formelle du langage à traduire [Yacc], b) la génération des interfaces graphiques en utilisant les éditeurs graphiques (GUI) c) et la génération de scripts de BD.

### **1.1.3 Approche hybride**

Cette approche est une synthèse pragmatique des deux précédentes. En effet, les deux approches par composant ou par génération constituent les deux extrémités pour le développement d'un logiciel. Entre ces deux extrémités nous trouvons des situations où les logiciels sont créés par un mélange de transformations et d'assemblages de composants réutilisables. Parmi ces situations nous pouvons avoir pour l'ingénierie d'une application, une architecture fonctionnelle, une infrastructure de traitement disponibles en avance et un grand nombre de composants à ajouter au besoin à l'infrastructure ou l'architecture [Mili et al., 2002]. Souvent, les cadres d'application (frameworks) et « les lignes de produits » utilisent une approche hybride. Les cadres d'applications pour les interfaces graphiques sont des exemples typiques de cette approche. Avec les approches hybrides nous pouvons réduire le coût et la durée des premières phases de développement d'une application. Par exemple, dans le cas d'une ligne de produits, avec la phase d'analyse et des exigences, nous pouvons juste spécifier ce qui distingue un produit d'un autre. Quant à l'architecture logicielle de ce produit, elle va être réutilisée tel quelle, en permettant l'ajout ou la création des composants manquants [Mili et al. 2002].

## 1.2 MDA (Model Driven Architecture)

MDA (« Model Driven Architecture ») est une approche hybride, présentée par l'OMG (« Object Management Group ») [MDA, 2003]. La motivation réelle de cette approche était l'observation qu'aucune technologie de middleware (ex. CORBA, EJB, .NET, COM) n'allait dominer et que les ponts entre ces technologies ne faisaient que ralentir davantage les applications [Pool, 2001]. Il fallait donc avoir une façon pour déployer les applications d'une façon efficace sur différentes plates-formes. Un autre constat simple d'OMG est que seuls les modèles conceptuels, tels que les modèles UML produits lors des étapes de spécification ou de conception, constituent des productions pérennes et capitalisables dans le cycle de développement d'un logiciel ou d'un système d'information. Ce sont donc ces modèles conceptuels qui doivent constituer les véritables composants réutilisables d'un logiciel [MDA, 2003].

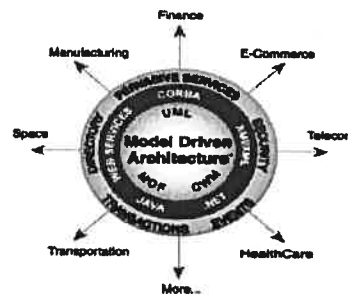
L'approche MDA pousse à résoudre ces problèmes d'interopérabilité et de portabilité dès le niveau modélisation en utilisant une approche hybride. MDA se veut donc indépendante de toute plate-forme et de tout système. Cette indépendance totale doit permettre de changer d'infrastructure sans perdre ce qui a déjà été conçu. Elle permet ainsi de capitaliser le travail effectué pendant les phases d'analyse et de conception pour une réutilisation ultérieure.

### 1.2.1 Principes

Le principe de base de MDA est de séparer les spécifications fonctionnelles d'un système des spécifications de son implémentation sur une plate-forme donnée. L'intégration des logiciels n'est alors plus conduite par l'intégration de composants logiciels interopérables mais par l'intégration de modèles conceptuels, décrits dans un langage de modélisation standard et technologiquement neutre (model driven). A cette fin, MDA définit une architecture de spécifications structurées en modèles indépendants des traitements (Computation Independent Model CIM), en modèles indépendants des plates-formes (Platform Independent Model PIM) et en modèles spécifiques aux plates-formes (Platform Specific Model PSM) [MDA, 2003]. Le passage entre ces modèles se fait par des transformations spécifiées par des correspondances. En effet, il existe différents types de techniques de transformation dont la transformation en associant des profils aux modèles ou par marquages des modèles [MDA, 2003]. Les composants logiciels implémentant ces modèles conceptuels, en principe, seront alors automatiquement obtenus à l'aide de générateurs de codes fournis par la plate-forme cible, ce qui permettra de faire migrer à moindre coût les composants vers de nouvelles

plates-formes. Ainsi, le passage d'une architecture .NET à une architecture J2EE, par exemple, ne consiste qu'à reprendre la logique métier spécifiée en MDA et à redemander une génération, mais cette fois, avec une cible différente.

Il faut noter que MDA se base sur des standards d'OMG très largement adoptés comme références (Figure 1-1). Notons entre autres UML (« Unified Modeling Language »), le MOF (« Meta Object Facility ») et le CWM (« Common Warehouse Metamodel »). Ces standards définissent l'infrastructure du MDA. Ils devront être complétés par des règles de transformation de modèles qui seront standardisées par l'OMG.



*Figure 1-1: Architecture de MDA*

### 1.2.2 Les différents modèles

La démarche MDA supporte toutes les phases du cycle de vie d'un logiciel et standardise le passage de l'une à l'autre. Elle est formée des étapes suivantes [Siegel, 2001] [MDA, 2003].

- La réalisation d'un modèle indépendant du calcul/traitement « Computation Independent Model » (CIM) qui présente les exigences métiers du domaine d'application.
- La réalisation d'un modèle indépendant de toute plate-forme appelé « Platform Independent Model » (PIM). Ce dernier permet de présenter les aspects fonctionnels et computationnels du système. Un enrichissement de ce modèle est fait par des étapes successives.
- Une génération du modèle spécifique appelé « Platform Specific Model » (PSM) correspondant à une plate-forme choisie. Un raffinement du modèle s'en suit jusqu'à obtention d'une implémentation exécutable.

**CIM** : L'objectif du CIM (Computation Independent Model) est de représenter ce qui reste stable à travers différentes mises en oeuvre informatiques en supposant que le cœur de métier n'évolue pas [Bézivin, 2002]. En d'autres termes, le CIM décrit les situations dans lesquelles le système en

construction doit être utilisé. Un tel modèle est parfois appelé modèle d'affaires ou modèle de domaine de l'entreprise. Le CIM correspond à la modélisation de l'entreprise sans parler encore de système informatique. Il montre le système au sein de l'environnement dans lequel il opérera. Ce modèle utilise un vocabulaire compréhensible par les experts du domaine qui n'ont pas de connaissances importantes en technologie logicielle. Il est considéré comme un pont entre les développeurs et les experts du domaine [MDA, 2003]. Il faut noter que l'OMG vise à fournir des profils UML pour différents modèles de domaines par exemple l'aérospatial, les télécommunications, la mécanique, etc. (Figure 1-1).

**PIM :** Le PIM se concentre sur la modélisation de la logique de métier du système en construction indépendamment de toutes plates-formes. On peut avoir des niveaux de PIM multiples pour le raffinement du modèle. Il faut que le PIM de base représente exclusivement les capacités métiers fonctionnelles et le comportement du système visé. Les PIM ultérieurs adoptent des aspects technologiques et architecturaux toujours sans détails propres à une plate-forme. Ces PIM peuvent utiliser des profils UML pour les besoins fonctionnels et les besoins non fonctionnels. Le PIM va être adapté pour un ou plusieurs styles architecturaux particuliers [MDA 2003].

**PSM:** Une fois le PIM mis en place, il est transformé en un modèle spécifique à une plate-forme (PSM). Les caractéristiques et les informations de configuration qui ont été définies de façon générique sont converties pour prendre en compte les spécificités de la plate-forme. La transformation est basée sur les caractéristiques de la plate-forme choisie. Ces caractéristiques doivent être décrites à l'aide d'UML et éventuellement d'un profil. Comme pour les PIM, il existe plusieurs niveaux de PSM.

Le MDA est un cadre de développement. Parmi ses avantages notons la standardisation des livrables, l'utilisation des profils UML pour les plates-formes cibles (ex. EJB, Web services, .Net), et la définition d'un cadre de travail qui permettra la complémentarité et l'interopérabilité des outils conformes à MDA. De plus, la divergence des modèles technologiques et d'affaires permet une évolution indépendante, améliore la productivité et diminue les coûts de développement et de gestion [Brassard, 2002]. Cependant, MDA ne résout pas les problèmes techniques inhérents à l'approche transformationnelle. Le marquage et les transformations actuelles des PIM vers les PSM restent à définir. De plus, le passage de CIM à PIM n'est pas trivial car le modèle de l'entreprise fait une abstraction de l'utilisation ou non des systèmes d'information pour le fonctionnement de l'entreprise.

### 1.3 Objectif de notre travail

*Observation:* Il existe deux types de processus d'affaires: des processus spécifiques à un métier donné et d'autres qui ne dépendent pas du domaine d'affaires (du métier) en question. Par exemple dans le domaine pharmaceutique, nous avons les processus de recherches qui sont propres aux études pharmacologiques. Ces processus sont des processus spécifiques au domaine, qui sont différents des protocoles de recherche dans d'autres domaines biomédicaux, ou encore, de la recherche en informatique! Par contre, d'autres processus dépendent très peu ou pas du tout du domaine d'affaires, et peuvent s'appliquer tels quels dans différentes industries. La plupart des processus de chaînes de valeur (voir [Porter, 1985]) sont de ce type là, et dépendent plus du *modèle d'affaires* que du domaine d'affaires. Les modèles de ces processus là seront alors réutilisables à travers les industries. Dans le contexte du MDA, de tels modèles de processus sont encore plus abstraits et plus généralement réutilisables que les « computation independent models » (CIM).

*But:* Le but de notre recherche est de produire des modèles d'analyse de systèmes d'information, spécifiques à une organisation donnée, et opérant dans une industrie donnée, à partir des modèles de processus d'affaires génériques. Ces modèles de processus génériques vont être réutilisables dans différents domaines. L'objectif ultime est de pouvoir développer des composants logiciels par transformations successives de composants génériques d'affaires. Ce faisant, nous étendons le modèle de développement de MDA en commençant en amont de CIM.

*Problématiques:* sur le plan théorique, les différentes problématiques que nous devons traiter sont:

1. Le choix d'une notation pour la représentation et la description de processus d'affaires sous forme d'artefacts réutilisables. En effet, une panoplie de notations a été proposée pour différentes utilisations. Laquelle choisir ?
2. Faire ressortir la distinction entre les parties communes et les parties variables de ces processus-là.
3. Développer une technique systématique pour la spécialisation de processus génériques en fonction des caractéristiques spécifiques à une organisation ou à une industrie. Cette procédure devra inclure la spécification ou spécialisation des parties variables des processus génériques,
4. Établir les relations entre les modèles de processus d'affaires et les composants logiciels correspondants.

Sur le plan pratique, pour que cette approche soit utile dans l'immédiat, il faut disposer d'un catalogue initial de processus et de composants logiciel existants. Alors, il faut identifier des modèles des processus candidats à la réutilisation, soit par l'étude de systèmes existants, soit par l'analyse des domaines. Il faut avoir une notation qui facilite la transition graduelle vers une automatisation totale. Il faut aussi disposer de mécanismes de recherche et de sélection de modèles.

*Résultats:* Concernant la notation, nous avons évalué une douzaine de langages de modélisation de processus d'affaires, provenant de traditions scientifiques différentes, et visant des utilisations différentes. Nous avons dressé une liste de critères de comparaison entre ces langages. Plusieurs raisons nous ont amené à retenir le langage UML 2—moyennant quelques amendements—dont le niveau de formalité, supporte certaines analyses, sans nuire à la communicabilité des modèles, et parce qu'il incarne aussi la synthèse de plusieurs initiatives visant la modélisation de processus d'affaires.

Nous avons développé une procédure systématique pour la spécialisation et l'instanciation de processus génériques d'affaires. La procédure de spécialisation, inspirée de la classification de processus par l'approche par questions, développée dans le cadre de la méthode BIAT [Carlson, 1980] et reprise et raffinée par plusieurs auteurs (voir ex. [Lefebvre, 1996], [MIT, 2004]), identifie de façon systématique les points de variation « intéressants » des différents processus, indépendamment du processus, et du domaine d'application. L'instanciation à proprement parler consiste principalement en la substitution de paramètres (ou variables) par des constantes. Une étude de cas a été utilisée pour illustrer notre méthodologie.

Quant à l'établissement de liens entre les modèles de processus et les composants logiciels correspondants, les initiatives de catalogage de composants logiciels réutilisables que nous avons étudiées se servent de fragments de modèles de processus principalement pour documenter les composants, mais n'établissent *pas* de liens entre processus et composants logiciels de façon *systématique*. Nous croyons qu'il est possible de générer totalement ou partiellement des modèles d'analyse—ou, tout au moins, une esquisse—à partir des modèles de processus. Cet objectif soulève plusieurs problèmes de recherche que nous avons esquissés, mais qui n'ont pas été abordés dans ce travail.

## 1.4 Organisation du mémoire

Dans le *deuxième* chapitre, nous définirons les concepts de base des processus d'affaires. Par la suite, nous présenterons une grille d'analyse qui nous permettra de comparer les langages de modélisation. Cette grille est utilisée pour comparer une douzaine de langages.

Dans le *troisième* chapitre, nous présenterons différentes initiatives de catalogage de processus d'affaires ou de composants logiciels métiers. Le but de cette étude étant de nous inspirer des procédures de classification et de spécialisation utilisées dans ces initiatives là, mais aussi de nous en servir comme données pour la validation de notre approche. Les différentes initiatives ont été organisées selon une grille d'analyse.

Dans le *quatrième* chapitre, nous décrirons notre approche pour la présentation des modèles génériques des processus d'affaires, et nous présenterons différents mécanismes de spécialisation. Notre approche est illustrée par une étude de cas qui traite le processus d'approvisionnement d'une entreprise particulière. La totalité des modèles pertinents à l'étude de cas est montrée dans l'appendice.

Nous terminerons ce mémoire par une conclusion générale où nous ferons une synthèse des travaux réalisés à date, et nous présenterons les suites qu'il nous semble opportun de donner à nos travaux.



# **Chapitre 2. Langages de modélisation de processus d'affaires**

## **2.1 Introduction**

L'identification des exigences et des besoins demeure la phase de développement la plus difficile et la moins formalisée, malgré le progrès dans l'ingénierie des logiciels [Phalp et Shepperd, 2000]. En effet, pour comprendre ce qu'un système logiciel est censé faire, nous devons le mettre dans le contexte des processus d'affaires qu'il doit supporter. La modélisation et l'analyse de ces processus d'affaires permettent de comprendre l'entreprise, pour ensuite la rendre plus efficace et améliorer la qualité du travail qui y est effectué. L'apparition de nombreux outils de modélisation tels que les systèmes logiciels de simulation et d'exécution automatique des processus, et les progiciels qui utilisent des modèles de processus prédéfinis et autres, a contribué à la naissance de différents langages de modélisation des processus d'affaires [Söderström, 2002].

Dans ce chapitre, nous allons présenter une étude sur différents langages de modélisation des processus d'affaires. Nous débuterons par les notions de base des processus d'affaires. Une classification des langages est proposée à la section 3. À la section 4, nous allons présenter les langages traditionnels de processus d'affaires. Ensuite, le langage de modélisation des workflows sera présenté à la section 5. À la section 6, il sera question des langages d'intégration des processus, puis la section 7 traitera des langages orientés objets. Une comparaison entre les langages sera exposée à la section 8. Enfin, une conclusion sera présentée à la dernière section de ce chapitre.

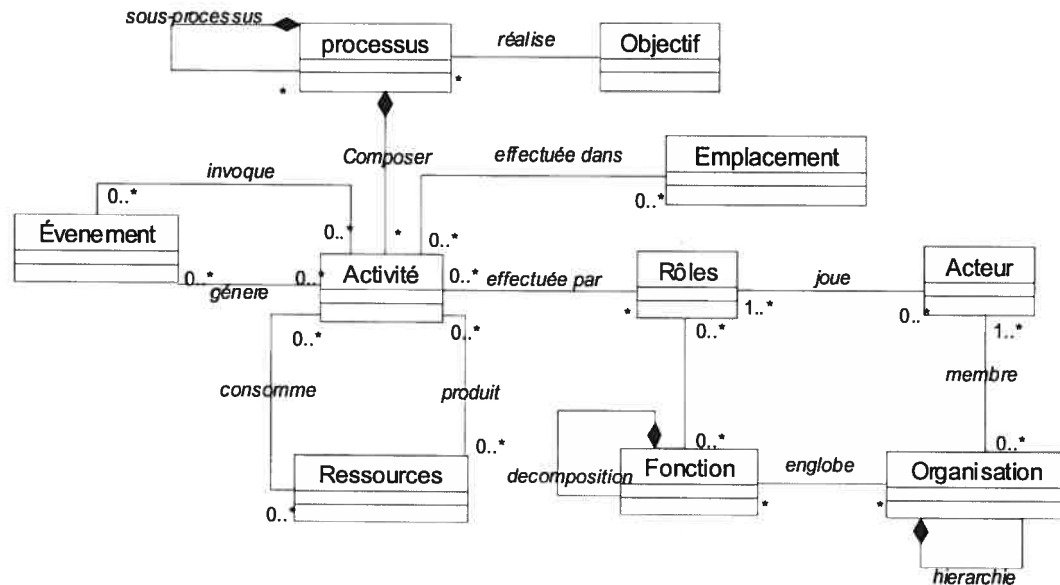
## **2.2 Les processus d'affaires**

Les entreprises ont une structure complexe formée d'une composition hiérarchique de départements et de leurs fonctions. La méthode traditionnelle pour représenter une entreprise est de construire ses fonctionnalités verticalement en fonction de son organigramme. Cet organigramme divise l'entreprise en un nombre de départements ou de sections tels que les ventes, la production, le marketing, etc. Chaque personne au sein de ces frontières organisationnelles a une tâche bien spécifique. Avec cette organisation fonctionnelle, le flux de travail est étape par étape, et les responsabilités sont des fonctions hiérarchiques. Cependant, une partie de ces fonctions ne se limite

pas à un seul département, certaines fonctions entrecoupent horizontalement plusieurs [Hans, 2000]. L'approche RPA (ré-ingénierie des processus d'affaires) a proposé une nouvelle vision des entreprises. Elle perçoit l'entreprise comme un grand ensemble de processus concurrents et communicants, réalisés par les ressources (ou acteurs) de l'entreprise qui exécutent les opérations nécessaires pour atteindre les objectifs fixés par celle-ci [Vernadat, 1996] [Meir, 2003]. Le but de cette approche est de permettre aux processus de franchir les frontières des départements et d'intégrer à la fois les activités et les informations permettant de réaliser les objectifs d'affaires visés par l'entreprise. Cette vue horizontale de l'entreprise, qui tend à ignorer les barrières organisationnelles, s'oppose aux vues classiques de décomposition par fonctions (ou vues verticales), elle représente le travail comme des flots et se prête mieux à l'analyse et à la réorganisation des processus opérationnels de l'entreprise, ainsi qu'à leur simulation ou au suivi de leur exécution [Vernadat, 1996] [Hammer & Champy, 1993].

### **2.2.1 Qu'est ce qu'un processus d'affaires**

Indépendamment du monde des affaires, un processus est un ensemble d'étapes partiellement ordonnées pour réaliser un objectif [Curtis, 1992]. Il n'existe pas de définition claire ou de définitions communes des processus d'affaires. Par exemple, pour Hammer et Champy, un processus d'affaires est « un groupe d'activités, qui à partir d'un ou plusieurs types d'entrées (input), crée une sortie (résultat) qui présente une valeur pour le client » [Hammer et Champy 1993], tandis que pour Davenport, un processus d'affaires est « un ordonnancement des activités de travail avec un début, une fin et des entrées/sorties clairement identifiés » [Davenport, 1990]. Pour Ould, les processus d'affaires sont « des activités comme des négociations aboutissant à une affaire. Généralement ils impliquent des personnes internes ou externes à l'entreprise » [Ould, 1995]. La Workflow Management Coalition [WfMC, 1999b] présente la définition suivante: « un processus d'affaires est un ensemble coordonné d'actions ou d'opérations qui sont reliées, en série ou en parallèle, dans le but d'atteindre un objectif d'affaires ou politique, normalement dans le contexte d'une structure organisationnelle, définissant des liens et des rôles fonctionnels ».



*Figure 2-1: Meta modèle primaire du processus d'affaires.*

Il existe des caractéristiques communes entre les différentes définitions d'un processus d'affaires. Pour simplifier, nous pouvons définir un processus comme étant un ensemble d'**activités** reliées. Ces activités sont accomplies par des **participants** qui remplissent des **rôles**. Ces activités consomment des **ressources** et en produisent d'autres. Elles peuvent être déclenchées par des **événements** comme elles peuvent en générer d'autres. Les activités peuvent être reliées par des **dépendances de ressources** et des **dépendances de contrôles**. Les participants peuvent être des acteurs agissant dans un contexte **organisationnel** qui traite des **fonctions** d'affaires spécifiques. Les rôles joués par les acteurs peuvent supporter les fonctions. Un processus permet de satisfaire un **objectif** et il est exécuté dépendamment des **règles d'affaires**. Un méta-modèle primaire qui démontre ces caractéristiques est présenté dans la Figure 2-1.

Il faut établir une distinction entre la définition de processus et les instances de processus. La définition de processus considère les types de données d'affaires ou les ressources. Une instance d'un processus tient compte des instances spécifiques des données. Par exemple, on peut définir le processus de traitement d'une commande client en terme de déclenchement des événements (un appel au département du service à la clientèle ou une soumission via le Net), en terme de types de ressources (les produits ne sont pas connus), et en termes de rôles (un représentant de service à la clientèle). Une instance d'un processus tient compte d'un client spécifique (« Monsieur X »), d'un contenu spécifique de la commande et d'un acteur spécifique (« Madame Y ») jouant le rôle du

représentant. Le méta-modèle de la Figure 2-1 schématise la définition du processus, où seule la classe acteur et ses associations contiennent la sémantique d'instance.

## **2.3 Langages de modélisation des processus d'affaires**

Dans cette section nous allons présenter dans un premier temps nos dimensions de classification de langages de modélisation de processus d'affaires. Dans un deuxième temps nous allons essayer une première catégorisation de ces langages.

### ***2.3.1 Dimensions de classification des langages de modélisation***

#### **2.3.1.1 Types des langages de modélisation**

Les langages de modélisation de processus d'affaires proviennent de différentes traditions scientifiques. Ils servent différents besoins et représentent différentes choses [Curtis et al. 1992]. Ould présente trois utilisations possibles d'un langage de modélisation des processus d'affaires [Ould, 1995]:

- Décrire un processus: Le modèle d'un processus permet de décrire ce dernier. Ainsi, le modèle doit être facile à comprendre pour les humains. Advenant le cas où les récepteurs seraient des machines, le langage doit aussi être formel [Curtis et al., 1992].
- Analyser le processus: l'analyse d'un processus consiste à évaluer ses propriétés. Cette analyse permet l'amélioration du processus suivant les changements économiques, politiques et technologiques. Si le processus est décrit formellement, nous pouvons vérifier facilement les propriétés structurelles telles que le couplage et la cohésion [Phalp et Shepperd, 2000] ou les propriétés dynamiques telles que l'absence d'inter-blocage et autres.
- Implémenter un processus: un processus peut être implémenté pour être simulé ou simplement exécuté. Selon le langage, nous pouvons étendre la modélisation pour vérifier le fonctionnement des activités du processus durant l'exécution et pour s'assurer que des contraintes spécifiques soient satisfaites [Curtis et al., 1992]. Naturellement, il n'y a que les langages formels qui permettent l'implémentation de processus.

Les concepteurs de langages peuvent mettre l'emphasis sur une de ces utilisations, au détriment des autres. De plus, un processus peut être modélisé à plusieurs niveaux d'abstraction pour différents objectifs. Par conséquent, l'utilisation des langages dépend des niveaux d'abstractions. Par exemple, si nous avons besoin de comprendre, d'évaluer ou possiblement de re-concevoir un processus d'affaires, alors le langage doit permettre de créer des modèles à un niveau d'abstraction

élevé. Par contre, si le processus doit être implémenté, le langage doit permettre de créer des modèles plus concrets pour englober les informations nécessaires à l'implémentation [Georgalopoulos et al., 1995] [Gale & Eldred, 1996].

Pour mieux gérer la complexité des modèles de processus d'affaires, plusieurs chercheurs ont identifié différentes vues de modélisation [Curtis et al., 1992]:

- *La vue fonctionnelle* : représente la dépendance fonctionnelle entre les éléments du processus. Elle représente le « **quoi** » (les activités à accomplir et les informations traitées dans ces activités) qui s'intéresse aux données et à leurs transformations. Cette vue utilise typiquement les diagrammes de flux de données (DFD; [Yourdon, 1989]).
- *La vue comportementale*: aussi appelée transactionnelle [Basu, 2000], représente le flux de contrôle entre les éléments d'un processus. Elle représente « **quand** » et « **comment** » les activités sont produites, et les contraintes qui sous-tendent ces activités. Elle s'intéresse à l'évolution du système dans le temps.
- *La vue informationnelle*: représente les entités informationnelles créées ou manipulées par un processus, leurs structures et les liens établis entre elles.
- *La vue Organisationnelle*: représente « **où** » (situé dans quelle unité organisationnelle, quels sont les lieux physiques utilisés pour stocker les entités), « **par qui** » (agents, rôles) les activités sont établies, et les mécanismes de communication physique utilisés pour transférer les entités.

La plupart des langages de modélisation prennent en compte ces vues mais en privilégient un sous-ensemble par rapport aux autres. Par exemple, la majorité des méthodes orientées objet prennent en compte les trois premières vues (ex. OMT [Rumbaugh et al. 1991], OOSE [Jacobson, 1992] et UML [Booch et al. 1999]). Ce qui est nouveau du point de vue ontologique, c'est la vue organisationnelle qui contient une description des participants dans un processus ainsi que l'emplacement physique et le contexte organisationnel du processus.

### 2.3.1.2 Types de modélisation

Les modèles de processus d'affaires possèdent une frontière plus large que le système logiciel considéré [Hans, 2000]. Ainsi, la modélisation d'un processus d'affaires consiste à représenter les activités du monde d'affaires à modéliser tandis que la modélisation d'un système d'information met l'emphase sur les parties à automatiser. Elle permet aussi de représenter les interactions des éléments du domaine (acteur, processus) avec d'autres systèmes existants. Dans ce cas, le système est représenté comme une « boîte noire ». Les modèles de processus d'affaires sont considérés comme des *modèles réels* par Isoda, tandis que les modèles de système d'information s'appellent

modèles *pseudo-réels* [Isoda, 2001]. Pour illustrer la distinction entre la modélisation du monde réel et la modélisation du monde pseudo-réel considérons l'exemple d'un système automatisé d'une bibliothèque (Figure 2-2). Le monde réel contient des livres, des utilisateurs, des bibliothécaires, des index physiques et des endroits physiques (sections, supports, étagères). Toutes ces entités interagissent avec ce système automatisé qui correspond au monde pseudo-réel. C'est ce système qui gère les tables d'index et qui trace les livres. La Figure 2-2-a présente le model statique du monde réel qui représente le fonctionnement de la bibliothèque avec un système automatisé. La Figure 2-2-b représente un modèle pseudo-réel du système automatisé lui-même.

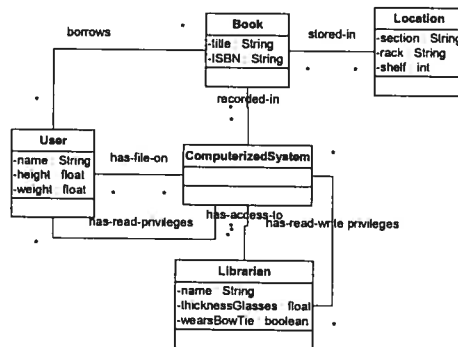


Figure 2-a. Real world model of library with computerized system

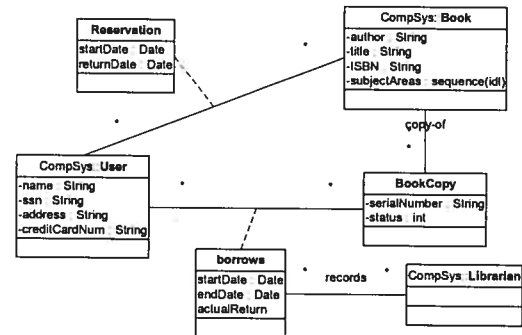


Figure 2-b. Pseudo-real world model of the computerized system

**Figure 2-2: Modélisation du monde réel et pseudo-réel (figure tirée de [Isoda, 2001])**

Avec cet exemple nous pouvons distinguer entre les deux types de modélisation. Avec la modélisation réelle on modélise les entités réelles du monde tandis qu'avec la modélisation pseudo réelle, on modélise les informations traitées par un système automatisé. Il est claire que la classe « User » ( ainsi que la classe « Librarian » et la classe « Book ») dans la Figure 2-2-a est différente de celle qui apparaît dans la Figure 2-2-b : La première représente l'utilisateur actuel tandis que la seconde représente la fiche utilisateur.

### 2.3.2 Classification des langages de modélisation de processus d'affaires

Dans le reste de ce chapitre, nous étudierons un certain nombre de langages qui ont été développés pour différents objectifs, mais qui sont utilisés pour décrire des processus d'affaires. Ces langages traitent différentes facettes de processus d'affaires (dynamique, fonctionnel, informationnel, organisationnel), et peuvent être plus ou moins formels selon l'utilisation. Il est difficile de catégoriser ces langages dans une seule dimension comme ils s'entrecoupent à travers les

différentes dimensions discutées plus tôt. Toutefois, les différents langages peuvent être regroupés dans quatre familles :

1. *Langages traditionnels de modélisation de processus*: Ces langages sont le résultat des travaux de l'ingénierie de l'information et de l'ingénierie des processus d'affaires. Cette catégorie contient par exemple, les réseaux de Petri, IDEF (ICAM Definition Method) [IDEF], EPC (Event Process Chain) [Keller et al., 1992], RAD (Role Activity Diagrams) [Ould, 1995], REA (Ressource-Event-Agent) [McCarthy, 1982] et récemment, BPML (Business Process Modeling Language) [BPMI, 2003].
2. *Langages de modélisation de workflow*: Ces langages sont orientés textes (« scripting ») pour décrire les workflows qui sont supportés par des systèmes de gestion de workflow (WfMS<sup>1</sup>). La plupart de ces langages sont formels et exécutables. Le Workflow Process Description Language (WPDL), qui sera présenté plus loin dans la section 2.5.1, est considéré comme un « interlingua » pour la description d'un workflow. Également, il existe d'autres formats d'échange, dont PIF [Lee et al., 1996] et PSL [NIST, 2002], qui sont établis pour soutenir l'échange de description de processus parmi différents outils.
3. *Langages d'intégration de processus*: Avec l'ouverture des entreprises sur le net, les entreprises se sont orientées de plus en plus vers le commerce électronique. Les exigences des commerces inter-entreprises sur le net sont différentes du domaine intra-entreprise [Dussart et al., 2002]. Ces exigences ont contribué à l'apparition de nouveaux langages de modélisation qui mettent l'emphase sur les mécanismes d'intégration en terme d'indépendance technologique, d'interfaces de programmation et de formats d'échange de données. Ces langages de modélisation ont une sémantique opérationnelle qui suppose l'existence d'un moteur de contrôle pour l'exécution des modèles. Trois initiatives seront présentées: RosettaNet [RosettaNet, 2003], ebXML [ebXML, 2001b] et BPEL4WS [Andrews et al., 2003].
4. *Langages orientés objet*: Les langages de modélisation orientés objets visent à représenter la solution logicielle du domaine plutôt que le problème du domaine d'affaires. Nous présenterons le langage de modélisation UML 1.x et ses primitives d'extensions avec le profil EDOC qui traite la modélisation d'entreprise [OMG, 2001]. De plus, nous présenterons UML 2.0, qui incorpore bon nombre de ces extensions [OMG, 2003].

---

<sup>1</sup> Selon la définition du glossaire WfMC (*Workflow Management Coalition*) [WfMC 1999b] un Workflow « est un outil informatique dédié à la gestion des processus. Cet outil définit, gère et exécute des processus en exécutant des programmes dont l'ordre d'exécution est pré-défini dans une représentation informatique de la logique des processus »

## 2.4 Langages traditionnels de modélisation de processus

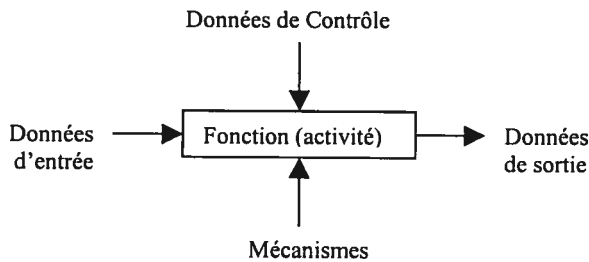
### 2.4.1 IDEF

IDEF (acronyme de **ICAM Definition Method**) (IDEF0 à IDEF14) est une suite de techniques conçues pour modéliser et analyser les entreprises [IDEF]. Elle fut créée dans le cadre du projet ICAM (Integrated Computer Aided Manufacturing) par l'U.S. Air Force. Ces diverses techniques sont le résultat d'un besoin industriel (la production) d'analyser les processus et d'être munis de techniques de communication décrivant le processus. Les langages les plus reconnus qui seront présentés par la suite sont: IDEF0 (Integration Definition for Function Modeling) pour la modélisation fonctionnelle d'un système, IDEF1,1X (Integration Definition for Information modeling), pour la modélisation des informations, et IDEF3 (Integration Definition for Process Descriptions Capture), qui représente l'aspect comportemental. La famille IDEF convient parfaitement à la modélisation des activités et du flux d'informations entre les activités. Il est aussi utilisé comme support pour la ré-ingénierie des processus d'affaires (RPA) [Bosilj, 2001].

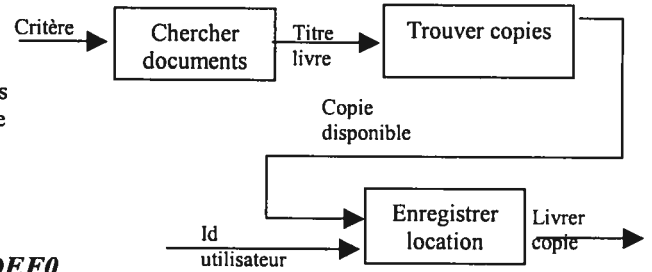
#### 2.4.1.1 SADT/IDEF0

IDEF0 est une méthode basée sur SADT (Structured Analysis Design Technique [Ross, 1977]). Son objectif est de modéliser les fonctions, les activités et les processus d'une entreprise, d'un système ou d'un sujet. Elle comporte non seulement un formalisme graphique mais aussi une méthodologie de développement des modèles. Un modèle IDEF0 décrit la fonctionnalité d'un système, ses entrées, ses sorties, les services et les autres fonctions dont il a besoin pour fonctionner. Le modèle IDEF0 est constitué d'une série hiérarchique de diagrammes. Chaque diagramme correspond à une vue fonctionnelle d'une partie du problème à un niveau d'abstraction donné. Une fonction à un niveau donné peut devenir un diagramme à un niveau plus bas. Le diagramme est constitué par un graphe de nœuds, chacun représentant une fonction (représentée par une boîte Figure 2-3). Les fonctions communiquent entre elles et avec leur environnement par l'intermédiaire de leurs interfaces (les flèches) qui représentent des flux de données (Figure 2-4). Les données qui passent entre les fonctions contrôlent implicitement le déroulement du processus.





**Figure 2-3 : Anatomie d'une fonction dans IDEF0**



**Figure 2-4: Diagramme fonctionnel avec IDEF0**

Avec la modélisation des processus d'affaires par IDEF0, le but principal est de montrer les éléments de transformation (données d'entrée) plutôt que les ressources qui font ce travail. IDEF0 couvre clairement la vue fonctionnelle d'un processus. Cependant, cette notation n'est pas aussi riche que celle d'un diagramme de flux de données (DFD) utilisée dans les techniques d'analyses et de conception structurées (ex. [Yourdon, 1989]).

### 2.4.1.2 IDEF1, 1X

La technique IDEF1 incorpore des principes de base permettant de produire des modèles informationnels nécessaires au déroulement des fonctions d'une entreprise. Ce modèle est composé de deux éléments fondamentaux: les diagrammes qui décrivent les relations entre les entités composant le système d'information modélisé (approche entité relation (E/R)) et les dictionnaires qui présentent la signification de chaque élément du modèle.

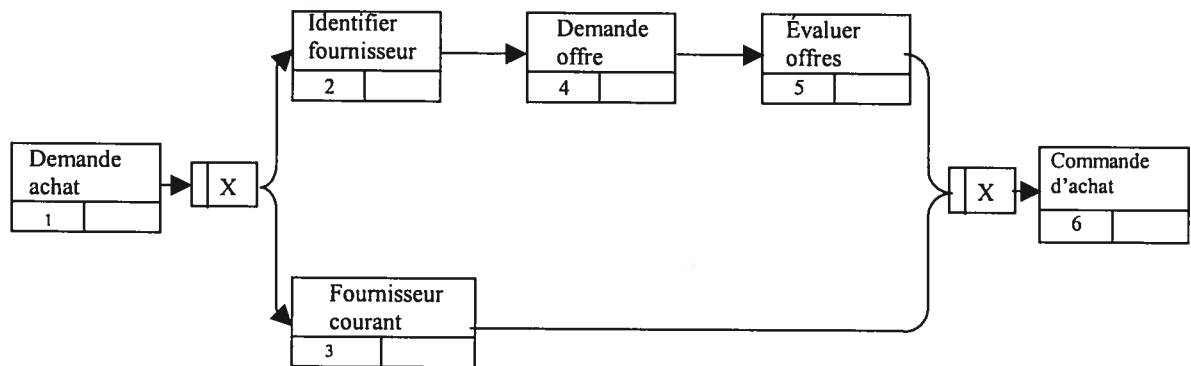
IDEF1X est une extension d'IDEF1, il introduit la notion du modèle de donnée sémantique qui contient une sémantique riche des relations et des concepts d'agrégation, de catégorisation et de spécialisation. De plus, IDEF1X supporte la notion de vues. Une vue est un modèle partiel et conditionné de données. Ce modèle est partiel étant donné que toutes les entités peuvent ne pas être représentées, et même pour une entité donnée, uniquement un sous-ensemble des attributs peut être inclus dans la vue. Le modèle est conditionné si le domaine d'un attribut dans une vue donnée est conditionné. En effet, IDEF1X permet de modéliser les données qui vont être manipulées par un logiciel. Ces données sont présentées dans des entités. Ces dernières représentent un sous-ensemble du monde réel qui doit être normalement modélisé dans une entreprise.

### 2.4.1.3 IDEF3

La méthode IDEF3 a été proposée pour remédier aux limites d'IDEF0 en matière de modélisation du comportement de l'entreprise [Mayer et al. 1995]. IDEF3 décrit les relations de succession et les

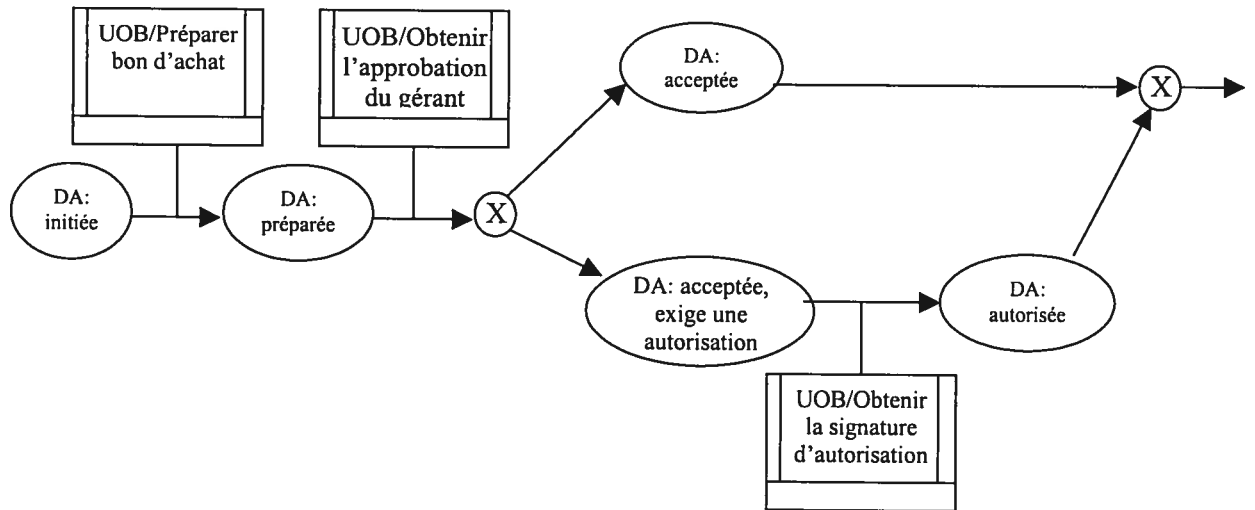
relations causales entre les activités, les objets et les événements d'un système, d'une organisation, avec la prise en compte de l'aspect temporel. IDEF3 permet de modéliser le monde réel des affaires dans le système automatisé, car elle est conçue pour supporter : a) des scénarios d'activités organisationnelles, b) des types de rôles des utilisateurs dans ces activités c) les cas d'utilisations. De plus, cette méthode supporte les classes utilisateurs et les interfaces objets, ce qui correspond à la modélisation logicielle (modélisation pseudo-réelle). IDEF3 représente un processus suivant deux stratégies qui peuvent référer l'une à l'autre:

1. *Stratégie centrée processus*: la description du processus est réalisée par des diagrammes de transitions qui présentent un scénario, complété par des documents d'information. Le diagramme de description des processus ressemble au diagramme d'activité de UML, mais il ne présente pas de façon explicite les ressources qui effectuent ces activités. Ce type de diagramme est composé de quatre types d'éléments (exemple à la Figure 2-5): Les unités de comportement (UOB) qui représentent les fonctionnalités, des événements ou des activités. Les liens pour relier les UOB entre elles, les connecteurs logiques ( ET, OU et OU exclusif représenté avec un X dans la figure 2-5) et les références permettant de faire un renvoi à une partie du modèle (ou à un autre modèle).



**Figure 2-5: Un diagramme de processus avec IDEF3. ( Exemple tiré de [Mayer et al. 1995])**

2. *Stratégie centrée objet* : La description des processus est réalisée par des diagrammes de transitions , complétés par des objets. Un objet est n'importe quelle chose physique ou conceptuelle identifiée et référée par des participants au domaine. Les modèles d'objets permettent à l'utilisateur de spécifier les règles qui gèrent la transition entre les états d'objets.



**Figure 2-6: Schéma de transition d'un objet (demande achat) avec IDEF3.**

Il existe trois types de schémas centrés objet : a) Schéma de transition: décrit le changement d'état d'un objet dans un seul scénario (voir exemple Figure 2-6). b) Schéma de transition étendu: Décrit le changement de plusieurs objets dans un scénario. c) Schéma d'objet: Décrit le changement d'état d'information à travers plusieurs scénarios.

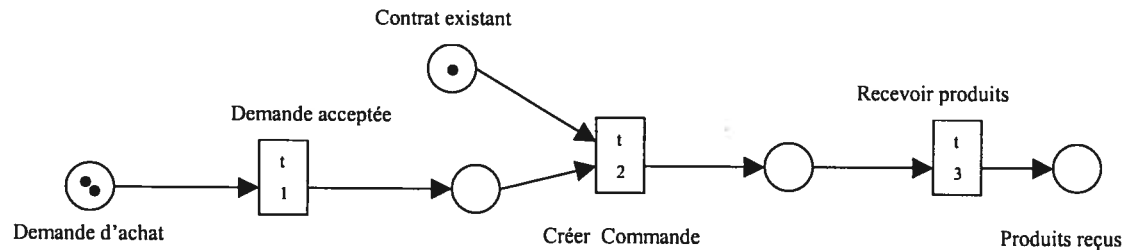
Finalement, IDEF est un standard de modélisation qui permet de spécifier formellement les aspects importants d'ingénierie des entreprises (la modélisation du mode réel). Cependant nous pouvons remarquer que ce qui manque à la famille IDEF ce sont les notions reliées aux acteurs et organisations. Peut être que IDEF12 (« Organization Modeling ») sera la méthode pour modéliser la vue organisationnelle, mais elle n'est pas activement développée.

### 2.4.2 Réseau de Petri

Le Réseau de Petri [Petri Nets] proposé par Carl Adam Petri en 1962 est devenu un formalisme important pour la modélisation, tant en informatique que dans de nombreuses autres disciplines. Les réseaux de Petri combinent une théorie mathématique bien fondée avec une représentation graphique du comportement dynamique des systèmes. Cette combinaison est la raison principale du grand succès des réseaux de Petri. Parmi les modèles de réseaux de Petri les plus simples (réseau de Petri classique), nous trouvons les réseaux places/transitions.

Un réseau place/transition est constitué d'un ensemble de places, d'un ensemble de transitions (disjoint de celui des places), et d'arcs reliant places et transitions: a) La place, représentée par un cercle et qui précède une transition est nommée « place d'entrée » et celle qui suit une transition est

nommée « place de sortie ». **b)** Un jeton est une ressource ou une condition . Chaque place contient un nombre entier non négatif de jetons. . L'ensemble de jetons dans une place décrit l'état du système à un instant donné. **c)** La transition est représentée par une boîte. Une transition est dite «franchissable» si chacune de ses places d'entrée contient un nombre de jetons supérieur ou égal à celui indiqué sur la flèche correspondante qui lie une place et à une transtion. En effet, une flèche peut être étiquetée d'un nombre naturel indiquant le nombre d'occurrence de cette flèche, c'est-à-dire le nombre de jetons nécessaires pour que la flèche soit «activée».



t(i) : transition I

**Figure 2-7 : Un processus modélisé avec le réseau de Petri classique**

Comme les réseaux de Petri ont été utilisés dans différents secteurs, différentes interprétations de ces concepts sont possibles. La table qui suit (tiré de [Murata, 1988]) en présente quelques unes

Place d'entrée	Transition	Place de sortie
Pré-conditions	Événement	Post-conditions
Données d'entrée	Étape de calcul	Données de sortie
Signaux d'entrée	Processeur	Signaux de sortie
Ressources entrées	Tâche	Ressources résultantes
Conditions	Clause	Conclusions
Mémoires tampon	Processeur	Mémoires tampon

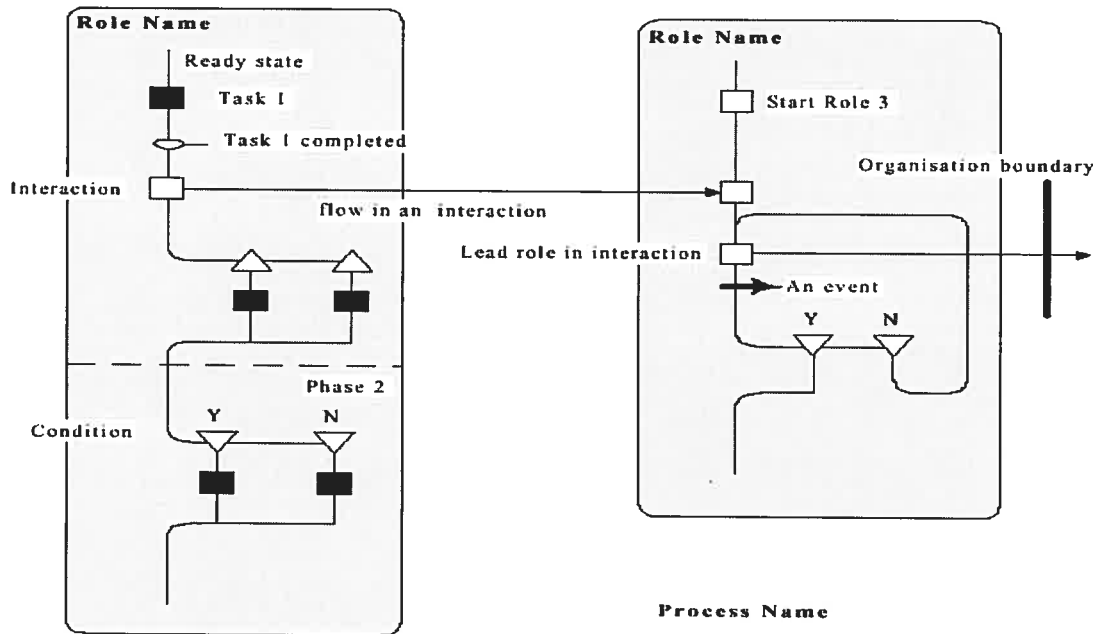
Le réseau de Petri classique permet de modéliser les états, les événements, les conditions, la synchronisation, l'itération, le choix, et le parallélisme. La modélisation d'un processus d'affaires simple est faisable avec un réseau classique (exemple à la Figure 2-7). Cependant, il ne prend pas en considération la modélisation des données, des acteurs, des systèmes complexes et du temps.

Des efforts pour résoudre ces problèmes sont identifiés dans la littérature notamment ceux de Aalst [Aalst, 1998].

Il est clair que les réseaux de Petri servent à modéliser l'aspect comportemental dynamique d'un processus. Les modèles sont difficiles à communiquer aux gens d'affaires. Cependant, différents chercheurs ont utilisé les réseaux de Petri pour garantir une sémantique formelle et pour vérifier la complétude (exactitude) de la définition des processus qu'ils modélisent (ex. [Phalp, 1998], [Glikas & Valiris, 1999], [Bosilj, 2001]).

### **2.4.3 Role activity Diagram (RAD)**

L'objectif de RAD est de « modéliser, et analyser les processus d'affaires » [Ould, 1995]. Les concepts de base de RAD ont été présentés par Holt et al. et plus tard enrichis par Ould [Ould, 1995]. La variante de RAD présentée par Ould s'appelle STRIM ('Systematic Technique for Role and Interaction Modelling'). En utilisant RAD, un processus est modélisé comme un nombre de rôles qui interagissent ensemble pour échanger des informations ou pour effectuer des activités. Un rôle peut être effectué par des acteurs tels qu'une machine, une personne ou un groupe de personnes. Dans chaque rôle, il y a un certain nombre d'activités qui sont ordonnées par leurs états. La Figure 2-8 montre les différents états d'un rôle et les transitions des activités avec les liens verticaux. Les événements externes sont représentés par des petites flèches, tandis qu'une activité est représentée par une boîte noire. Les interactions entre les rôles, représentées par des boîtes blanches peuvent être binaires ou multi-participants. Les données échangées à travers les interactions sont représentées par des entités. Une entité est une étiquette reliée aux activités et aux interactions entre les rôles. RAD utilise d'autres concepts et notations pour modéliser les conditions, l'itération, le choix et surtout le comportement dynamique (l'invocation d'un autre rôle tout au long du déroulement du rôle en cours). RAD présente un nombre de mécanismes d'abstractions, comme la composition et l'encapsulation.



*Figure 2-8: Notation de RAD [Murdoch, 1998].*

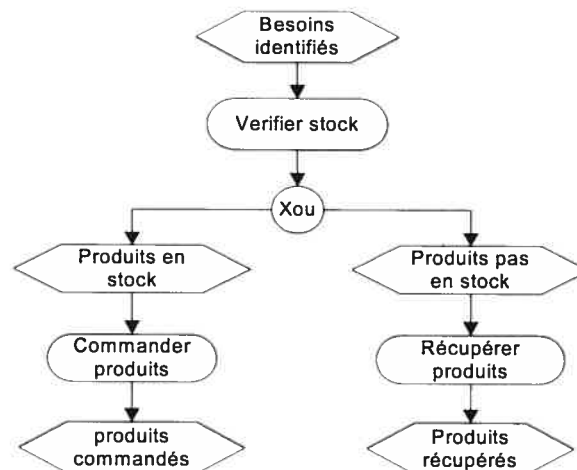
RAD recommande l'utilisation de ELH (« Entity Lifetime Histories ») pour représenter sous forme d'une hiérarchie les différents états qu'une entité ou une ressource prend durant le déroulement du processus. RAD ne permet pas de représenter les données: il suppose que les propriétés statiques des données et des ressources seront représentées en utilisant les diagrammes entités relations.

Ce langage est utile pour représenter le comportement des processus organisationnels, et connaître par la suite le fonctionnement d'une entreprise (ex. [Murdoch, 1998]). Selon Ould, STRIM (le successeur de RAD) est utile pour le développement des modèles communicatifs [Ould, 1995]. De plus, Ould propose une méthode d'analyse qui se base sur les modèles représentés avec STRIM en utilisant une combinaison de simulation, d'optimisation locale, et de détection des "antipatterns". La simulation se base sur l'utilisation de jetons qui se propagent dans tout le diagramme reflétant le progrès du processus (comme les réseaux de Petri). Les deux autres méthodes se basent sur des heuristiques qui soumettent un modèle STRIM à une série de validations ou de questions, destinées à identifier les redondances, les conflits, etc. Autres chercheurs ont essayé d'analyser formellement avec STRIM (ex. [Phalp & Shepperd, 2000]).

### 2.4.4 Event Process Chains (EPC)

EPC (Event driven Process Chain) est une méthode développée à l'institut des systèmes d'information (IWi) [IWI] en collaboration avec SAP [SAP, 2003] au début des 90 [Keller et al., 1992]. EPC est une technique de modélisation utilisée par de nombreuses sociétés pour modéliser leurs processus d'affaires. Elle est facile à comprendre et à employer par des gens d'affaires. Les composants principaux de EPC sont les fonctions, les événements et les connecteurs.

Une fonction correspond à une activité (étape de processus) à exécuter. Un événement décrit la situation avant ou après l'exécution d'une fonction. Cet élément peut jouer le rôle de post et pré-condition. Un événement a au plus un arc d'entrée et au plus un arc de sortie. Les connecteurs logiques sont utilisés pour spécifier le flux de contrôle. Ils ont la même sémantique que les connecteurs de IDEF3. Les connecteurs sont (ET, OU, OU exclusif). La Figure 2-9 présente un exemple de processus avec EPC.



**Figure 2-9: Un processus modélisé avec EPC**

Les progiciels intégrés (ERP) tel SAP et les outils RPA tel ARIS<sup>2</sup> qui utilisent les diagrammes EPC ont étendu cette notation pour traiter les ressources qui peuvent affecter le flux de contrôle. Avec les EPC étendus, nous pouvons présenter des objets de données et des unités organisationnelles. De même, ils ont ajouté d'autres modèles (diagrammes) pour différents points de vue (regarder [Curran 1998, 2000] pour SAP et [Scheer, 2002] pour ARIS):

**Le modèle fonctionnel:** ce modèle présente la hiérarchisation des fonctionnalités et leurs décompositions qui sont référencées dans le diagramme EPC. **Le modèle organisationnel :** Ce modèle permet de présenter la hiérarchisation des unités organisationnelles qui exécutent les

<sup>2</sup> ("Architecture of Integrated Information Systems")

fonctions. Ces unités sont des personnes, des équipes, des départements ou des divisions. La hiérarchisation peut être définie par un organigramme de l'entreprise.

**Le modèle de données :** Ce modèle traite les objets d'information les plus importants et il décrit les relations qui existent entre eux. Il est présenté comme un modèle entité-relation.

**Le modèle ELH (voir 2.4.3):** Ce modèle présente les différents états que les entités peuvent avoir durant le traitement du processus.

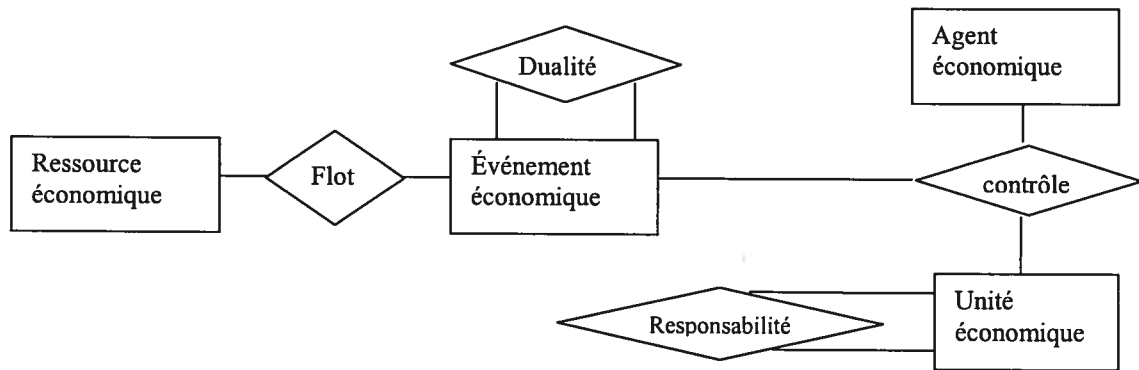
EPC est centré sur l'aspect comportemental d'un processus d'affaires. Avec son extension, nous pouvons identifier les autres points de vue d'un processus d'affaires. EPC peut être présenté hiérarchiquement par des niveaux d'abstraction. Ainsi, une fonction peut être plus détaillée à un niveau plus bas. Les fonctions élémentaires sont les fonctions qui ne peuvent plus être détaillées. EPC n'est pas formel, certains chercheurs ont établi une correspondance avec les réseaux de Petri pour pouvoir analyser formellement les processus décrits en EPC (ex.[Kindler, 2003], [Aalst,1999]).

### **2.4.5 Ressource Event Agent (REA)**

Le cadre REA a été créé par William E. McCarthy, principalement pour modéliser des systèmes d'information de comptabilité [McCarthy, 1982]. Ce cadre s'est avéré tellement utile pour une meilleure compréhension des processus d'affaires qu'il est devenu une des ontologies de modélisation des processus d'affaires pour des entreprises traditionnelles et des systèmes de commerce électronique [Geerts & McCarthy, 2002]. Le modèle de base de REA représente d'une façon expressive « ce qu'on a besoin de compter » dans une entreprise, comme les ressources et les événements économiques.

Les *ressources économiques* sont des objets rares et utiles qui sont sous le contrôle de l'entreprise [Ijiri, 1975]. Les *événements économiques* sont définis comme étant des phénomènes qui reflètent le changement de ressources suite à une production, à un échange, à une consommation ou à une distribution [Yu, 1976]. Le méta-modèle de base de REA est présenté à la Figure 2-10 avec la notation ER (entité-relation).





*Figure 2-10 : Le méta modèle de REA ( Tiré de [ McCarthy, 1982])*

Ce méta-modèle montre les événements économiques, les ressources qui sont incrémentées et décrementées suite à un événement et les agents qui participent à l'événement. La relation de dualité représente le lien entre les événements dans le contexte d'échange économique. Ce lien montre quelle ressource a été augmentée et quelle ressource a été diminuée en contrepartie. Le modèle différencie également les agents internes des agents externes pour un événement, et il établit des rapports récursifs parmi les unités économiques (agents internes).

La spécificité de REA est l'incorporation de la sémantique des objets économiques dans l'architecture d'information d'une société. REA n'est pas aussi riche que les langages de modélisation tels que les réseaux de Petri, RAD et EPC. Cela s'explique par le fait que l'objectif de base de sa création était de mémoriser les transactions effectuées durant l'exécution d'un processus d'affaires. En effet, REA est un modèle de données implémenté par une base de données à laquelle les utilisateurs et les logiciels peuvent accéder. La préoccupation de la modélisation des données et leur implémentation a deux conséquences : La première, négative, parce que la représentation du processus est limitée à ce que nous pouvons enregistrer et la deuxième, positive, parce que tout ce qui est modélisé avec REA est implémenté.

D'après les auteurs, les concepts de cette ontologie deviennent de plus en plus utilisés pour la représentation des processus publics, mettant l'emphase sur les transactions qui peuvent être effectuées entre les entreprises [Haugen & McCarthy 2000]. Le standard ebXML que nous aborderons plus loin, utilise les diagrammes et les concepts de REA.

#### **2.4.6 BPML**

Le consortium « Business Process Management Initiative » (BPMI), regroupement d'acteurs de l'« entreprise application integration » (EAI) et du « Business to Business » (B2B), a présenté un langage de description de processus d'affaires reconnu sous l'acronyme BPML [BPMI, 2004].

D'après BPML, BPML est un langage pour représenter des processus exécutables qui traitent tous les aspects des processus d'affaires d'entreprise, y compris les activités complexes, les transactions et leur compensation, la gestion des exceptions et la sémantique opérationnelle [BPML, 2003]. L'un des objectifs de BPML est d'intégrer les applications existantes de l'entreprise dans des processus de commerce électronique. BPML utilise le langage XML pour la description des processus d'affaires comme il utilise les schémas XML (XSD) pour spécifier les types de données manipulées par le processus à modéliser.

BPML est conçu comme un langage de programmation structuré. Il représente un processus d'affaires comme un ensemble d'activités exécutées dans un contexte. BPML définit 17 types d'activités catégorisés en des *activités simples* et des *activités complexes*. Parmi celles-là, on retrouve spécialement un ensemble d'activités de bas niveau qui correspondent aux structures de contrôle dans un langage de programmation (ex. « Call », « Until », « While », « Switch », etc.). La plupart des activités simples sont atomiques. Le contexte d'une activité est celui de son exécution. Il est caractérisé par un ensemble de propriétés spécifiques à l'activité (variables locales, exceptions, etc). En effet, ça correspond à la notion de portée dans les langages de programmation.

Le processus peut être vu comme une activité complexe qui exécute un ensemble de fonctionnalités (d'activités) dans un contexte spécifique. Il existe des processus visibles par tous les processus et d'autres qui sont locaux. Les processus peuvent être lancés par un appel explicite d'une autre activité, ou par un événement (signal). L'exemple suivant présente un processus simple qui calcule le temps que met une activité à finir (tiré de la spécification de BPML [BPML,2003]).

```
<property name="timeToPerform" type="xsd:duration">
  <value>PT24H</value>
</property>
<process name="calculateEndTime">
  <context>
    <property ref="tns:timeToPerform"/>
    <property name="endTime" type="xsd:dateTime"/>
  </context>
  <assign property="endTime"
    xpath="$tns:timeToPerform + func:currentDateTime()" />
</process>
```

Dans cet exemple, nous pouvons identifier les paramètres d'entrée/sortie du processus. Ces paramètres sont la lecture et l'écriture dans les propriétés (« ref » et « name ») du contexte. Ce sont des variables locales à un contexte. Également nous pouvons identifier les types, les valeurs par défaut et les expressions des différentes variables. Les types sont des schémas XML valides comme des types simples, complexes, dérivés et même anonymes définis par des éléments déclarés.

Avec BPML, il existe d'autres constructeurs utiles, notamment pour les notions des ordonnanceurs (« Schedules »), pour la technique de compensation de transactions et pour les propriétés de corrélations. Les céduteurs sont des collections des événements synchronisés qui déclenchent l'exécution d'un groupe d'activités/processus. Nous pouvons avoir plusieurs céduteurs « Schedules » dans un contexte. La technique de compensation permet d'effacer les effets des transactions de longue durée dans le cas d'échec. Les propriétés de corrélations vont relier les instances de processus. En effet, dans une organisation, nous avons plusieurs instances d'un même processus (ex. traiter commande client) qui s'exécutent en même temps. Nous pouvons corréler les activités (ou les sous-processus) en utilisant des variables d'instances comme des propriétés des processus. Par exemple, pour le processus « traiter commande client » d'habitude nous avons plusieurs instances. Pour relier des activités particulières à une même « commande client » nous utilisons la propriété « numéro commande ».

BPML n'adresse pas l'aspect organisationnel, de plus, la modélisation de l'aspect informationnel est faible. En effet, la modélisation des données (avec XSD) est limitée aux types des messages échangés.

Finalement, BPML est un langage permettant d'exécuter des processus mais n'est pas conçu pour la communication de la description d'un processus. En effet, les auteurs de ce langage précisent que son but est la validation et la simulation des processus. BPMI travaille présentement à concevoir une notation BPMN (Business Process Modeling Notation) afin de fournir aux entreprises une modélisation graphique de leurs processus. BPMN sera soutenu par un modèle interne pour générer BPML [BPMI, 2003].

## **2.5 Langage de modélisation de workflow (WPDL)**

La technologie Workflow vise à fournir un support informatique pour une gestion efficace des processus d'affaires. Cette gestion comporte non seulement la conception et l'ingénierie, mais également des activités qui vont plus loin dans le cycle de vie des processus d'affaires [Cardoso et al., 2002]. Les workflows les plus reconnus sont les processus de production et les processus administratifs. Le consortium international de gestion de Workflow (WfMC) formé d'un certain nombre d'organisations s'intéressant à la promotion de la technologie des workflows, a défini un certain nombre de standards. Parmi ces standards notons le modèle de référence des systèmes de

gestion de workflow<sup>3</sup> (WFMS) qui présente le méta-modèle de base pour représenter un processus d'affaires [WFMC, 2002a]. Ce méta-modèle (Figure 2-11) définit les éléments de base et les liens qui doivent exister pour supporter l'automatisation d'un processus. Ce méta-modèle vise à établir un format d'échange entre les différents modèles et outils de workflows. Il met en évidence la grammaire écrite, en EBNF (« Extended Backus-Naur Form »), d'un langage de description reconnu sous le nom de WPDL (Workflow Process Description Language).

WPDL est utilisé pour sérialiser les modèles des processus décrits suivant le méta-modèle à la Figure 2-11. Les entités principales de ce méta-modèle sont [WFMC, 1998]:

**Les activités** qui peuvent être automatisées ou qui nécessitent l'intervention d'un utilisateur. Les interactions humaines faisant intégralement partie du processus. **Les conditions de transitions** qui relient les activités par un flot de contrôle conditionné. **Les données pertinentes** sont les données créées ou utilisées avec chaque instance de processus durant l'exécution. Ces données ne recouvrent pas l'ensemble des données utilisées dans un processus. Elles représentent la partie des données utiles pour a) les conditions de transition, b) les règles d'affectation d'une tâche à une ressource ou c) l'invocation d'une application externe. **Les participants** qui peuvent être incarné par des personnes, des ressources, des rôles et des unités organisationnelles (non illustré dans la Figure 2-11).

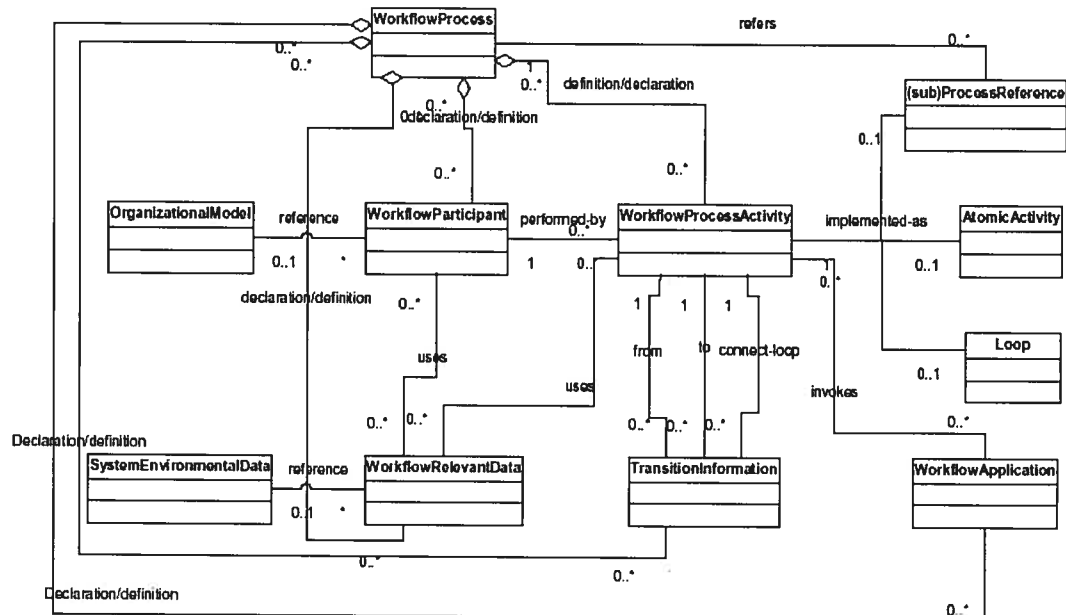


Figure 2-11: Méta- modèle de WfMS [WFMC, 1999a]

<sup>3</sup> Le système de gestion des workflows WFMS (*workflow management system*) est un système qui définit, implémente, intègre et gère l'exécution d'un ou de plusieurs workflows (systèmes d'information, unités organisationnelles, ressources physiques) à l'aide d'un environnement fonctionnant avec un ou plusieurs moteurs de workflow [WFMC, 1999]

Il faut noter que pour chacun des éléments du méta-modèle, la spécification contient un ensemble d'attributs de base et un ensemble d'attributs optionnels, comme elle permet aux utilisateurs d'ajouter leurs propres attributs. Ces attributs vont être compilés correctement avec les outils, mais leur sémantique d'exécution appartient aux utilisateurs et aux concepteurs d'outils.

Le WPDL se caractérise par la représentation des processus comme un graphe ordonné d'activités. WPDL propose également un aspect dédié à l'exécution en offrant la gestion des états des différents processus. Il permet de présenter l'état par lequel passe une activité. Cependant, il ne fournit pas les états d'un objet. WPDL met l'emphase spécialement sur la présentation des workflows. En d'autres termes, la préoccupation de base de ce langage est d'exécuter la description du processus. Les deux aspects informationnel et organisationnel sont faiblement traités avec ce langage. Également, ce langage ne contient pas de représentation graphique. La description d'un processus est fournie textuellement. À noter qu'une version XML de WfMC WPDL est en cours de développement dont le nom est XPDL (XML Process Definition Language) [WfMC, 2002b]. Des parties du standard WPDL sont suivies par de nouvelles technologies pour décrire la sémantique d'un processus (ex. BPEL4WS, BPML déjà présenté précédemment).

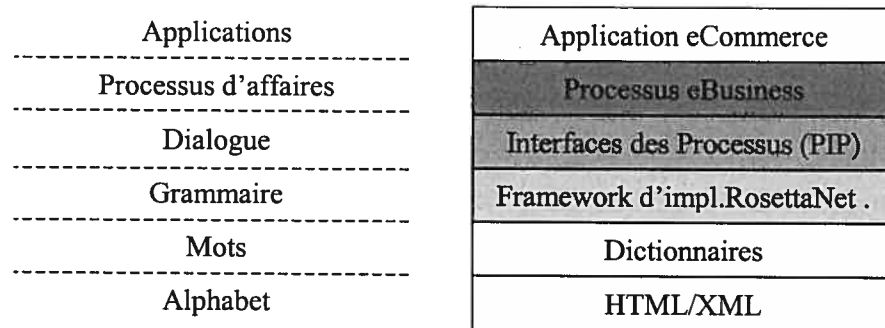
## 2.6 Langages d'intégration de processus

Dans la littérature on distingue deux types de processus: les processus d'affaires internes à l'entreprise (nommés aussi processus privés ou centrés) et les processus publics qui engendrent une interaction avec les partenaires externes (nommés aussi processus de collaboration). Les processus publics font partie des processus de l'entreprise, mais ils sont exposés publiquement dans le but d'une corrélation inter-entreprises. Ces processus permettent de spécifier le comportement des partenaires par rapport aux messages échangés, sans rendre public le comportement interne [Dayel, 2001]. Nous allons, présenter dans cette section des langages qui visent les processus publics.

### 2.6.1 RosettaNet

RosettaNet est un consortium qui regroupe des entreprises vouées aux technologies de l'information, aux productions des semi-conducteurs, aux télécommunications et aux composants électroniques. Ces entreprises collaborent pour créer et mettre en oeuvre des standards destinés au commerce inter-entreprises et à la gestion de la chaîne logistique [RosettaNet, 2003]. L'objectif est de présenter des normes à la portée des entreprises de taille moyenne afin de leur permettre de bénéficier de l'automatisation intégrale de la chaîne logistique. Ces normes traitent **la sémantique**

**des informations** échangées et les dialogues entre les entreprises. Les dialogues permettent aux applications de se comprendre et de réaliser des transactions par le biais d'échange de messages XML, dont le contenu (bon de commande, facture, etc.) est normalisé.



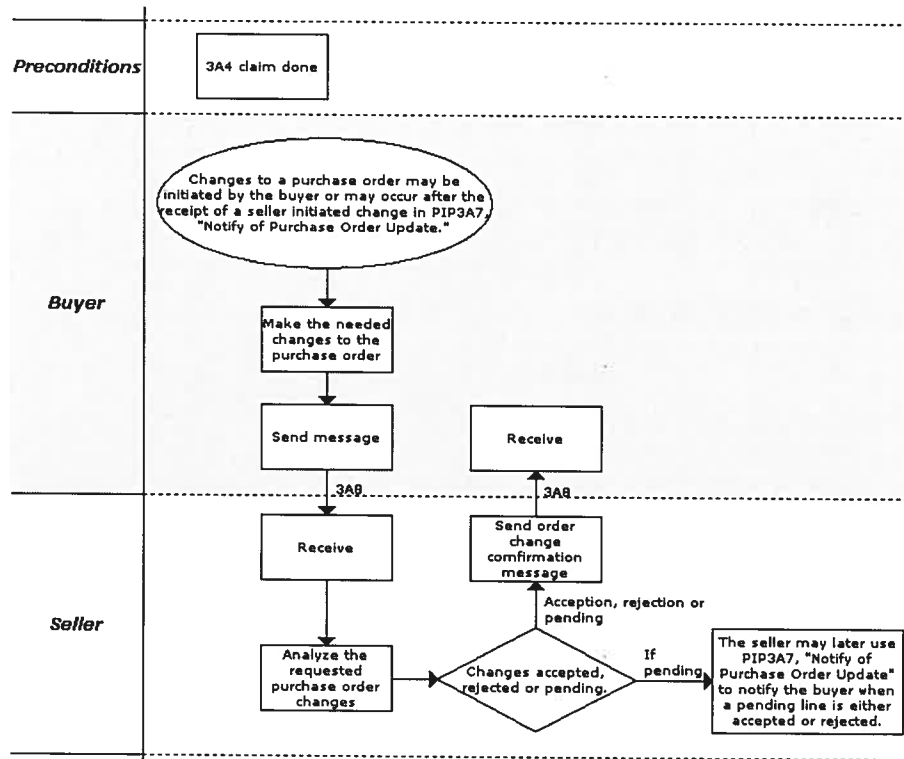
**Figure 2-12: Structure du standard RosettaNet**

La structure du standard de RosettaNet est formée de couches (Figure 2-12) pour permettre aux entreprises d'être compatibles à différents niveaux. Cette structure est formée de *Partner Interface Processes* (PIP), de dictionnaires et d'un framework d'implémentation (RNIF). Les dictionnaires définissent en quelque sorte les mots utilisés dans les transactions électroniques. Il existe deux types de dictionnaires: un pour les termes techniques et l'autre pour les termes métiers. Le dictionnaire des termes métiers contient la définition des termes d'affaires génériques indépendants de l'industrie (ex. compte, nomenclature, etc.). Le dictionnaire des termes techniques contient des vocabulaires communs pour décrire des produits et des services dans des industries spécifiques, surtout dans le domaine des composants électroniques et des technologies de l'information.

Le RNIF (RosettaNet Implementation Framework) agit comme une grammaire. Il couvre les formats des messages en XML (DTD), des directives d'utilisation des messages, des services de sécurité et des règles de paquetages et de dépaquetages des messages. Les PIPs sont des modèles qui précisent la séquence des messages échangés au cours du processus d'affaires (le dialogue), et contiennent des documents métiers. Ils ne sont pas en format exécutable et sont des processus publics qui passent à l'interface entre les partenaires.

La spécification PIP contient trois vues du processus : a) la vue opérationnelle d'affaire, pour représenter la sémantique d'affaire et spécifier les rôles des partenaires ainsi que leurs interactions pour accomplir un objectif métier. b) La vue du service fonctionnel décrit les composants de l'interaction leurs protocoles, et établit une correspondance entre les actions de PIP et les documents. c) La vue d'implémentation du framework qui spécifie les formats des messages.

En effet, un processus d'affaires typique commence avec une ou plusieurs activités internes à un partenaire. Ce processus se poursuit par l'envoi d'un message. Ce dernier sera suivi par un ensemble d'activités du partenaire répondant et possiblement par une réplique au premier partenaire. La Figure 2-13 présente un exemple de processus avec la notation diagramme d'activité de UML. Cet exemple illustre le PIP (3A8) pour une demande d'achat.



**Figure 2-13: PIP pour le changement dans une commande existante (tiré de [Rosettanet, 2003])**

Le standard RosettaNet utilise les notations UML et XML pour illustrer les processus. Son avantage majeur est qu'il met l'accent sur les processus et les fragments de processus qui passent entre les interfaces des partenaires, sur la sémantique d'interactions et les formats des messages. De plus, il identifie la nécessité d'avoir différentes vues pour représenter différents aspects d'un processus d'affaires.

## 2.6.2 Ebxml/BPSS

EbXML est un consortium international, créé principalement par OASIS et l'UN/CEFACT, qui visent à poser les nouvelles bases de normes d'échanges pour le commerce électronique B2B [ebXML, 2003]. Cette finalité est atteinte avec ebXML en se concentrant sur la notion de processus d'affaires externes (publics). «EbXML ne s'occupe pas des documents échangés dans les transactions commerciales mais bien de la formalisation du processus de transaction lui-même, le

processus métier » [Chauvet, 2002], ce qui n'était pas le cas avec RosettaNet, qui met l'emphase sur la sémantique des informations échangées et de leurs formats, comme nous l'avons déjà vu. EbXML met l'emphase sur l'intégration du processus et sur son infrastructure. En résumé, la famille des standards ebXML traite les secteurs suivants :

1. Un mécanisme standard et un langage pour décrire des processus d'affaires et leurs informations associées (le *Business Process Specification Schema* BPSS) [ebXML, 2001b].
2. Un standard pour permettre aux entreprises représenter leurs processus et leurs méthodologies de support, ex: Profils de Coopération des Entreprises (CPP, [ebXML, 2002]).
3. Un mécanisme pour rechercher et sélectionner des CPP à travers un registre public (ebXML *Registry Services* [ebXML, 2001c]).
4. Un mécanisme pour établir la correspondance entre deux CPPs pour produire un *accord de collaboration* (« Collaboration Protocol Agreements » ou CPA, [ebXML, 2002])
5. Une librairie qui contient un ensemble de processus d'affaires de base (« Core business processes ») et des documents qui couvrent les scénarios d'affaires fréquents [ebXML, 2001d]. Nous parlerons de cette librairie au chapitre 3.

Ce qui nous intéresse dans cette section est le mécanisme de description des processus d'affaires reconnu sous le nom de BPSS (Business process specification schema). Le BPSS est une spécification de schémas. Son méta-modèle sert à définir la collaboration entre des partenaires (B2B) dans un processus d'affaires public à travers les transactions et les échanges de documents. Le processus est défini comme une collaboration multi-participants (Multiparty Collaboration). Cette dernière implique un ensemble de partenaires (Business Partner Roles). Chaque partenaire joue un rôle autorisé dans une collaboration binaire et les collaborations multi-participants seront ramenées à un ensemble de collaborations binaire.

```
<BinaryCollaboration name="Product Fulfillment" timeToPerform="P5D">
  <InitiatingRole name="buyer"/>
  <RespondingRole name="seller"/>
  <BusinessTransactionActivity name="Create Order"
    businessTransaction="Create Order"
    fromAuthorizedRole="buyer"
    toAuthorizedRole="seller"/>
  <BusinessTransactionActivity name="Notify shipment"
    businessTransaction="Notify of advance shipment"
    fromAuthorizedRole="buyer"
    toAuthorizedRole="seller"/>
  <Start toBusinessState="Create Order"/>
  <Transition fromBusinessState="Create Order"
```



```

        toBusinessState="Notify shipment"/>
    <Success fromBusinessState="Notify shipment"
        conditionGuard="Success"/>
    <Failure fromBusinessState="Notify shipment"
        conditionGuard="BusinessFailure"/>
</BinaryCollaboration>

```

**Figure 2-14 : Une collaboration binaire simple avec BPSS de ebXML.**

Une collaboration binaire (ex. Figure 2-14) se déroule entre deux rôles (InitiatingRole et RespondingRole) et consiste en une ou plusieurs activités commerciales (Business Activity). Il existe 2 types d'activités : a) Activité transactionnelle (Business Transaction Activities) et b) Activité de collaboration (Collaboration Activities).

Une activité transactionnelle est atomique, elle décrit l'échange (Business transaction) entre 2 rôles. La transaction est décomposée en deux parties, la requête (Requesting Business Activity) et la réponse (Responding Business Activity). Il existe toujours une requête de demande (Request document flow), tandis que la réponse (Response document flow) est optionnelle. Une requête synchronisée est exigée pour établir un accord (agreement). Le passage d'un ordre d'achat ou la demande d'un catalogue constituent des exemples de Business Transaction Activity ; la négociation d'un contrat constitue un exemple de Collaboration Activity. Les activités commerciales sont séquencées dans une chorégraphie. La chorégraphie décrit l'ordre et les transitions entre Business Transactions ou sous-collaborations au sein d'une Binary Collaboration. La chorégraphie est décrite dans le BPSS ebXML à l'aide de concepts de diagramme d'activités tels qu'un état de début, un état de fin, des activités, des synchronisations, des transitions entre les activités ainsi que des gardes sur les transitions.

BPSS couvre adéquatement la vue comportementale. La vue informationnelle est partiellement traitée avec les structures des documents et les bibliothèques des composants (voir chapitre 3). Ce langage n'est pas aussi riche que BPEL4WS que nous allons présenter par la suite.

### **2.6.3 BPEL4WS**

Le « Business Process Executable Language For Web Service » (BPEL4WS, [Andrews et al., 2003]) est un langage permettant de modéliser les processus d'affaires exécutables dans le contexte des services Web. Ce langage correspond à la fusion des langages WSFL d'IBM et XLANG de Microsoft. BPEL4WS utilise WSDL pour spécifier les actions qui doivent s'effectuer au sein d'un processus opérationnel et pour décrire les services Web offerts par ce processus. L'idée de base de ce langage est qu'un processus peut être considéré comme une collaboration entre services ou tâches. BPEL4WS décrit l'ordre d'interaction (comme BPSS d'ebxml) entre les partenaires

d'affaires et plus précisément, entre les services Web. Il permet la distinction entre les 2 types de processus, ce qui n'est pas le cas avec BPML:

1. *Le processus exécutable*: permet de décrire les processus internes à l'entreprise. Ces processus sont spécifiés complètement et sont exécutables. Avec BPEL4WS, nous avons la possibilité d'avoir une représentation indépendante des interactions entre les partenaires, comme nous pouvons le faire également avec BPML.
2. *Le processus abstrait*: permet de décrire les processus de l'entreprise, exposés publiquement dans le but d'une corrélation inter-entreprise. Ces processus permettent de spécifier le comportement des partenaires par rapport aux messages échangés, sans rendre public le comportement interne.

La définition d'un processus avec BPEL4WS consiste en une partie déclarative qui introduit différents éléments. Cette partie est suivie par la description du processus. La partie déclarative contient les éléments suivants :

1. Une description des messages échangés entre les services. Le message est formé de parties. Chaque partie a un nom et un type. Le composant type est décrit en utilisant les types de XSD (types simples, types complexes, etc.) BPEL4WS permet l'utilisation d'autres types que ceux de XSD.

```
...
<message name="POMessage">
  <part name="customerInfo" type="sns:customerInfo"/>
  <part name="purchaseOrder" type="sns:purchaseOrder"/>
</message>
<message name="InvMessage">
  <part name="IVC" type="sns:Invoice"/>
</message>
...
```

2. Une description des services appelés : chaque service est formé d'un groupe d'opérations. Une opération a un nom, une entrée et une sortie. Une opération est invoquée par un message. Un service est défini en utilisant le concept « portType » de WSDL.

```
...
<portType name="purchaseOrderPT">
  <operation name="sendPurchaseOrder">
    <input message="pos:POMessage"/>
    <output message="pos:InvMessage"/>
    <fault name="cannotCompleteOrder"
      message="pos:orderFaultType"/>
  </operation>
</portType>
...
```

3. Une description des interactions (contrat) entre les partenaires de processus: chaque contrat (appelé service link type) définit des rôles et les associe à des interfaces/portTypes. Prenons

l'exemple d'une chaîne de valeurs. Dans cette chaîne nous avons les clients, les fournisseurs et l'entreprise. L'interaction entre un client et l'entreprise consiste à remplir une demande d'achat, tandis que l'interaction entre l'entreprise et le fournisseur consiste à commander des produits pour les stocker. Ces interactions sont gérées par deux types de contrat (service link types). Dans chaque type, on identifie les rôles joués par chaque interface de service.

```

...
<slnk:serviceLinkType name="invoiceLT">
  <slnk:role name="invoiceService">
    <slnk:portType name="pos:computePricePT"/>
  </slnk:role>
  <slnk:role name="invoiceRequester">
    <portType name="pos:invoiceCallbackPT"/>
  </slnk:role>
</slnk:serviceLinkType>
...

```

5. Une description des partenaires : considérant un contrat, il n'est pas clair quelle entité jouera quel rôle dans le contrat. Le BPEL4WS permet d'identifier les différents partenaires dans les différents contrats, les rôles qu'ils jouent dans le contrat (type de lien de service), et le rôle de l'entreprise dans ce contrat.

```

...
<partners>
  <partner name="customer"
    serviceLinkType="lns:purchaseLT"
    myRole="purchaseService"/>
  <partner name="invoiceProvider"
    serviceLinkType="lns:invoiceLT"
    myRole="invoiceRequester"
    partnerRole="invoiceService"/>
...
</partners>
...

```

6. Une déclaration des variables utilisées durant la collaboration d'affaires: ces variables sont définies dans la portée (contexte avec BPML) du processus et elles sont échangées entre les différentes étapes de processus comme messages d'entrée/sortie.

7. Une description du gestionnaire d'erreurs (Fault handler) : celui -ci indique la réponse désirée pour une exception particulière.

Le processus est décrit en utilisant un ensemble d'activités partiellement ordonnées. Ces activités correspondent aux invocations des opérations définies dans les différents services. Ces services seront assurés par les partenaires identifiés ci-dessus. Un processus peut contenir des activités séquentielles, des activités concurrentes, et il peut créer des liens de dépendances (pour le contrôle) entre ces étapes processus en utilisant le mécanisme de lien « link ». Pareillement à BPML, le BPEL4WS contient des descriptions sur le gestionnaire des exceptions, le mécanisme de

compensation et l'instance de corrélation. L'exemple qui suit présente une description en BPEL4WS du processus de demande d'achat qui se déclenche par l'arrivée d'une commande d'un client.

```

...
<sequence>
  <receive partner="customer"
           portType="lns:purchaseOrderPT"
           operation="sendPurchaseOrder"
           variable="PO">
  </receive>
  <flow>
    <links>
      <link name="ship-to-invoice"/>
      <link name="ship-to-scheduling"/>
    </links>
    <sequence>
      <assign>
        <copy>
          <from variable="PO" part="customerInfo"/>
          <to variable="shippingRequest"
              part="customerInfo"/>
        </copy>
      </assign>
      <invoke partner="shippingProvider"
             portType="lns:shippingPT"
             operation="requestShipping"
             inputVariable="shippingRequest"
             outputVariable="shippingInfo">
        <source linkName="ship-to-invoice"/>
      </invoke>
      ...
    </sequence>
    ...
  </flow>

```

BPEL4WS utilise des structures semblables à celles de BPML. Pourtant il est plus riche et plus clair. La distinction entre le processus exécutable et le processus abstrait permet la réutilisation des processus. BPEL4WS met l'accent sur l'aspect fonctionnel et comportemental d'un processus. Il présente les concepts organisationnels vis-à-vis des processus publics (rôles, partenaires), mais n'intègre pas les acteurs humains.

## 2.7 Langages de modélisation orientée objet

Des chercheurs dans la modélisation orientée objet ont argumenté que les modèles d'objet nous permettent de modéliser le monde réel d'une façon compréhensible pour tous les types de profils utilisateurs [Isoda, 2001]. Cependant, des experts de la modélisation d'affaires (ex. [Hans, 2000]) précisent que la notation d'UML n'est pas complète pour exprimer des processus d'affaires d'une

façon intuitive et naturelle. Des mécanismes d'extension de UML ont été utilisés pour représenter des processus d'affaires (voir par exemple l'EDOC [OMG, 2001]). Récemment, des concepts de modélisation des processus d'affaires ont été ajoutés à UML2 [OMG, 2003].

### **2.7.1 Enterprise Distributed Object Computing (EDOC)**

EDOC est un cadre de modélisation adopté par l'OMG en novembre 2001 comme norme pour modéliser des systèmes d'entreprise basés composants [OMG, 2001]. Son objectif est de permettre la conception orientée objet d'un système d'information d'entreprise et d'en dériver une implémentation à l'aide de composants métiers distribués, conformément au développement MDA et en utilisant UML. EDOC est défini comme un méta-modèle et comme un profil UML. Dans sa spécification, des correspondances sont établies entre les concepts du méta-modèle et les entités des différents profils qui le forment. Deux critères rendent ce cadre pertinent à nos objectifs : Une extension de UML pour la représentation explicite des processus d'affaires, et un mécanisme pour lier les composants d'affaires aux acteurs, et les activités aux processus d'affaires qui les initient. EDOC est formé de trois grandes parties:

1. L'ECA (entreprise collaboration architecture) qui contient des profils qui présentent différentes vues d'un système EDOC.
2. Le méta-modèle de patrons, qui permet de décrire des savoir-faire de ECA et qui permet de les appliquer dans des systèmes EDOC spécifiques.
3. Des méta-modèles techniques pour définir des plates-formes dépendantes des modèles. En particulier, la spécification décrit des méta-modèles pour Java, EJB, et d'autres.

Nous allons représenter et mettre l'emphase sur ECA qui contient cinq profils :

- a) Le profil *Component Collaboration Architecture* (CCA), représente un système comme un ensemble de composants collaborant pour jouer des rôles dans le but de réaliser une fonction.
- b) Le profil « *Entities* », présente des extensions UML pour représenter des concepts des domaines d'applications.
- c) Le profil « *Events* », présente des extensions UML pour modéliser des systèmes dirigés par l'événement.
- d) Le profil « *Business Process* » spécialise CCA et met l'emphase sur les composants qui présentent des activités, des tâches et des processus.
- e) Le profil « *Relationships* », présente des extensions UML pour traiter des associations qui sont plus riches sémantiquement.

Chacun des profils est articulé respectivement autour d'un concept de base: composant, objets métier, événements, processus et relations. Il faut noter que les concepts du profil CCA sont considérés comme le cœur architectural des autres profils à plusieurs niveaux de granularité. Ces différents profils sont utilisés pour représenter différentes vues des systèmes EDOC.

### 2.7.1.1 Spécification du profil CCA (component collaboration Architecture)

Le profil CCA présente un système comme étant un ensemble de composants qui interagissent entre eux via un ensemble de ports, suivant un ensemble de protocoles. Le concept de base est le composant connu sous le nom de composant de processus « process component ». Un composant peut être élaboré c.à.d composé d'autres sous-composants et peut contenir des propriétés pour définir des paramètres de configuration. Un composant élaboré est spécifié en utilisant deux constructeurs: une composition avec le diagramme de collaboration d'UML (Figure 2-15) et une chorégraphie avec le diagramme d'activité UML, pour montrer la coordination des activités des sous-composants pour accomplir la fonctionnalité d'un composant.

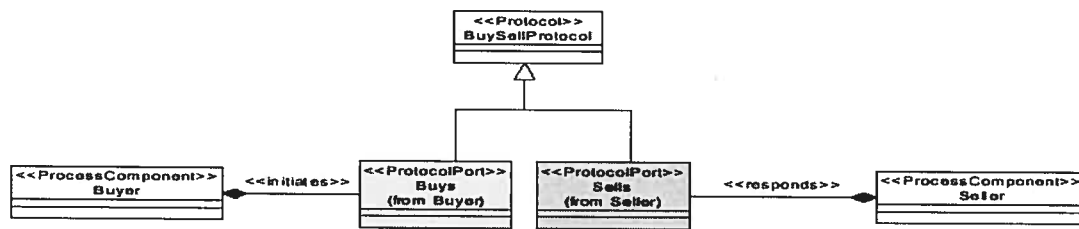


Figure 2-15: Une Composition

La Figure 2-16 présente la notation utilisée avec EDOC pour représenter un composant de processus. La notation utilise l'icône sous-système de UML et des notations pour les ports et les propriétés sont ajoutées à cette icône.

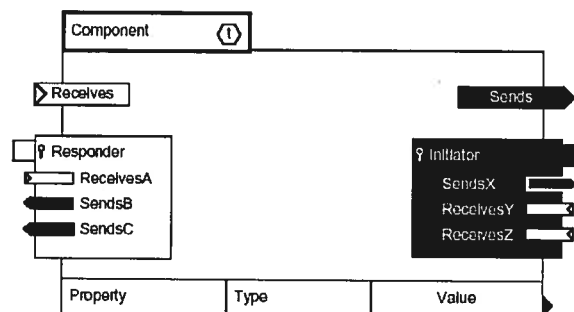


Figure 2-16 : Notation d'un composant avec EDOC

Le comportement d'un composant est spécifié en utilisant des protocoles. Un protocole est une séquence de messages "envoyer" et "recevoir" entre deux (ou plusieurs) composants de processus. Un port est une entité qui forme le point d'interaction (peut être bi-directionnel) du composant avec l'environnement (autres composants). Il existe différents types de ports. D'abord, le port « OperationPort » qui réalise des opérations requêtes/réponses ; dans la Figure 2.18 ces ports sont « receives » et « sends ». Pour les interactions complexes, il existe le port « ProtocolPort » ou « Multiport », qui est un groupe de ports dont les actions sont liées ensemble. Dans la figure 2.16, ces ports sont les ProtocolPorts « responder » et « initiator ». Le ProtocolPort « responder » se déclenche en recevant un message A d'un initiateur. Réciproquement, le ProtocolPort « initiator » permet au composant de jouer le rôle d'un initiateur. Ce protocole commence en envoyant un message X. Les étapes qui suivent impliquent des messages de « receives » et « sends ».

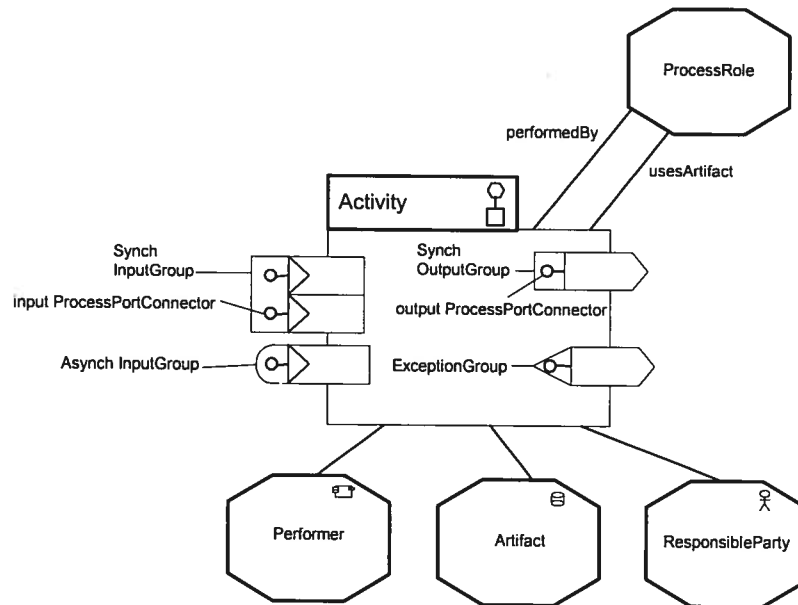
Les composants sont connectés en liant leurs ports par des connecteurs (ports connectors). Par exemple en liant un port de flux de sortie d'un composant A avec un port de flux d'entrée d'un autre composant B. Un composant élaboré montre la composition et les liens entre les différents sous-composants qui le constituent. Il faut noter qu'il existe un type spécial de composant, dont le nom est « communauté » (process community)). Ce composant est un ensemble de composants de processus qui n'interagit pas avec l'environnement externe de la communauté et qui n'a pas de ports externes.

### **2.7.1.2 Le profil du processus d'affaires (Business process profile)**

Ce profil est une spécialisation du profil CCA et réfère à d'autres profils pour exprimer le comportement du système dans le contexte d'affaires demandé. Un processus d'affaires est décrit comme une composition d'activités d'affaires (spécialisation de composant de processus). Ce profil permet de représenter :

- Les dépendances complexes entre les tâches d'affaires ainsi que leur concurrence sémantique (à noter que BPEL4WS fait un bon travail là-dessus).
- Les différentes façons de représenter le temps (la durée, la date limite).
- Le support pour la détection des exceptions.
- L'association entre les spécifications des tâches d'affaires et les rôles qui effectuent ces tâches ou bien qui sont nécessaires pour l'exécution des tâches.
- Les initiations des tâches spécifiques pour répondre à l'occurrence des événements d'affaires.
- Les actions qui agissent durant un processus d'affaires comme des événements.

Le méta-modèle de ce profil est composé de trois éléments de base: processus d'affaires, (« BusinessProcess »), tâche composée (« CompoundTask ») et activité (« Activity »). Le « BusinessProcess » tout comme le « CompoundTask » dérive du composant de processus « ProcessComponent » (CCA Profile). Ces deux types de composants sont constitués d'activités. La distinction entre les deux, est que le premier est utilisé comme une couche extérieure pour représenter un processus d'affaires, tandis que le second est utilisé pour décomposer récursivement un processus.



**Figure 2-17: Notation d'une activité avec EDOC**

Une activité peut soit invoquer une tâche composée « compoundTask », soit être réalisée par un acteur (performeur) si elle est élémentaire (Figure 2-17). De même, plusieurs activités d'une même tâche peuvent être accomplies par le même rôle. Une activité peut avoir des ressources (artéfacts) et un responsable, comme elle peut avoir une pré et une post-condition.

Un type spécial du composant processus d'affaires est le « Business Process Entity ». En effet, ce type de composant représente les instances des processus d'affaires, tandis que le composant « BusinessProcess » représente les définitions des processus d'affaires. Ce type est utilisé pour modéliser les processus de longue durée qui peuvent exiger une gestion ou une interaction de différentes instances durant leurs traitements (comme l'instance de corrélation avec BPML et BPEL4WS).

La spécificité d'EDOC est l'intégration des modèles de processus et les modèles d'information. A noter que EDOC vise à faciliter le développement de modèles d'entreprises, de systèmes ou



d'organisations (PIMs) en utilisant une approche de collaboration (d'intégration) récursive à différents niveaux de granularité et pour différents degrés de couplages. De plus, le profil du processus d'affaires, fait une séparation claire entre les différentes ressources qui sont impliquées dans un processus d'affaires, comme les acteurs et les artefacts (participants actifs et passifs).. Avec UML 2.0 (dont il sera question par la suite), nous retrouvons les critères de base d'EDOC.

### **2.7.2 Langage Unifié de Modélisation version 2 (UML 2.0)**

UML a été défini par l'OMG comme la notation normalisée des méthodes OO [Booch et al. 1999]. Différentes versions d'UML ont été publiées depuis. Nous tenterons de présenter la dernière version (2.0), qui est en cours de standardisation. Nous supposons que le lecteur possède une connaissance sommaire d'UML. UML 2.0 met l'emphase sur quatre secteurs [UML-RFP, 2000a], [UML-RFP, 2000b], [UML-RFP, 2000c]: a) les bases de UML; en essayant d'aligner la sémantique du méta-modèle d'UML et les standards MOF (Méta-Object Facility). b) les constructions au niveau utilisateur; pour établir une consolidation des constructions du méta-modèle existants, pour éliminer l'ambiguïté et l'ajout de nouvelles constructions et pour représenter des systèmes basés composants. c) Le langage des contraintes d' UML (OCL); un besoin d'alignement entre UML et OCL et d) un standard pour échanger les diagrammes en se basant sur la représentation graphique.

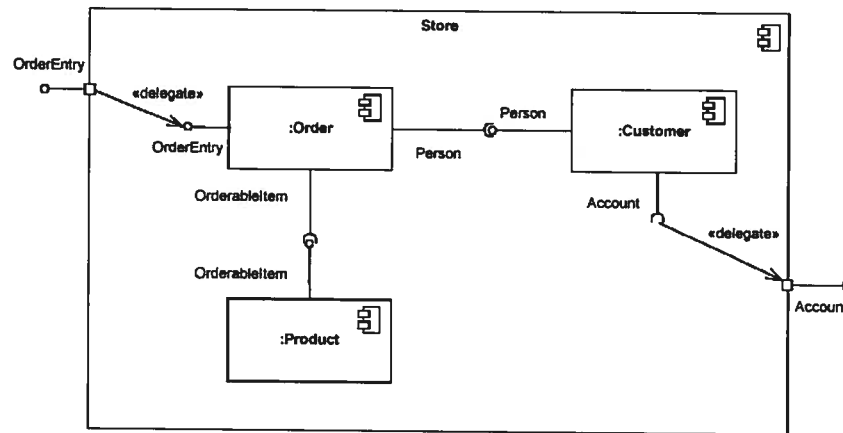
Ce qui nous intéresse est le secteur « constructions utilisateur » et les améliorations effectuées. UML2 introduit de nouvelles constructions dans le méta-modèle pour la représentation des aspects comportementaux et architecturaux:

1. Le changement dans la structure architecturale interne et le comportement est effectué en introduisant les notions de parties, de connecteurs, de ports et de généralisation.
2. La représentation des composants avec des ports et avec une structure interne constituée de parties connectées via des connecteurs. Ces composants offrent à leur environnement un accès parfaitement contrôlé et spécifié par l'intermédiaire d'une interface.
3. Les activités utilisent la sémantique des réseaux de Petri plutôt que les machines à états.
4. Les interactions sont améliorées par de nouveaux concepts, comme la composition, des références pour les diagrammes de séquences, les exceptions, les boucles et les alternatives.

UML2 semble en effet avoir intégré bon nombre d'idée d'EDOC qui font maintenant partie du méta modèle.

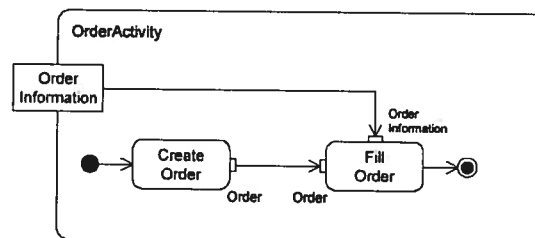
Concernant les concepts architecturaux, une notion de classes structurées est introduite et contribue à la présentation des modèles de données élaborés qui renforce l'encapsulation. La structuration

sera appliquée avec les différents « classifier ». Un nouveau type de classe est présenté avec le diagramme de classes. Il est structuré et associé à une collaboration permettant d'exprimer l'interaction entre les instances des parties de la classe composante. Cette collaboration vise à exprimer un aspect bien déterminé en identifiant les rôles que les instances jouent via les interfaces liées à la classe. UML 2.0 présente un type spécial de classes structurées (le concept de composant) et de ses éléments associés (Figure 2-18). Ce concept est utile pour deux cas: le développement par composant et la structuration du système basée sur les composants.



**Figure 2-18: Structure d'un composant élaboré avec UML2.0**

Au niveau des diagrammes d'activités, UML2 a intégré la sémantique des réseaux de Petri. De plus la nouvelle notation présente un support pour la détection des exceptions, la gestion d'erreurs, les contrôles, la présentation des activités composées de plusieurs actions (Figure 2-19), et la possibilité de modélisation des partitions (acteurs) multidimensionnelles. Ces partitions permettent de représenter les différentes classes (classifier) contribuant à la présentation des flux (de contrôles ou de données) de différents points de vues.



**Figure 2-19: Une activité composée des actions avec UML 2.0**

La version UML 2.0 est meilleure que celle de UML1.x, car elle est enrichie avec des notations de composition, d'agrégation comportementale et structurelle ainsi que des concepts de bas niveau. Il n'existe pas de support explicite pour la modélisation des processus d'affaires avec UML 2.0, pourtant cette version de UML possède tous les ingrédients nécessaires. UML 2 incarne la synthèse de plusieurs initiatives visant la modélisation de processus d'affaires.

## 2.8 Comparaison entre les langages de modélisation

La comparaison entre les différents langages est établie selon deux dimensions générales: a) la portée du langage (quelle vue du processus d'affaires est couverte) et b) l'intention d'utilisation du processus décrit par le langage.

Nous utiliserons pour ce faire la catégorisation proposée respectivement par Curtis [Curtis et al., 1992] et Ould [Ould, 1995] (revoir section 2.3.1). Nous récapitulons notre évaluation présentée dans ce chapitre dans le Tableau 2.1 où nous montrons les vues que chaque langage a englobées. Nous désignons par « Oui » les langages qui contiennent un ensemble de constructions cohérent pour représenter la vue en question, cependant, le niveau d'expression fourni par les langages peut varier de l'un à l'autre. Citons par exemple la notation REA qui est moins expressive que UML2 avec la vue dynamique. « Référence » indique que le langage en tant que tel ne maintient pas la vue en question, mais que les concepteurs du langage se sont efforcés de lier les modèles de cette vue à des modèles dans un autre langage. Par exemple, EPC et RAD ne contiennent pas de méta-modèle informationnel, mais les concepteurs de ces langages supposent que la description du processus est liée à un modèle de donnée exprimé avec la notation entité/relation. Idem pour BPML et BPEL4WS, qui supposent que les données sont exprimées avec des types de systèmes externes, comme le XML Schéma . Nous utilisons « en partie » si le langage présente un ensemble incomplet de constructions pour représenter la vue. Par exemple, dans le méta-modèle présenté à la Figure 2-1, nous avons utilisé trois entités pour représenter la vue organisationnelle (rôles, acteurs, et organisations). Plusieurs langages représentent la notion de *rôle*, mais pas nécessairement les notions d'*acteur* et d'*organisation*.

UML englobe largement et richement les vues dynamique, fonctionnelle, et informationnelle. Cependant, pour la vue organisationnelle, UML contient les ingrédients mais pas encore la portée (« focus or tone »). Avec son architecture à quatre niveaux, UML peut probablement prendre en charge n'importe quelle vue; il suffit de définir le méta-modèle correspondant en utilisant le MOF (Meta Object Facility). Toutefois, en terme de constructions de niveau utilisateur, UML2 se doit d'enrichir le vocabulaire. EDOC renferme un vocabulaire riche dont une partie a d'ailleurs été intégrée dans UML2. Pour la partie concernant le profil du processus d'affaires, nous ne savons pas

encore si elle sera introduite dans UML2 comme le CCA ou bien si elle sera construite au-dessus de UML2 .

Langage	Informationnelle	Fonctionnelle	Dynamique	Organisationnelle
IDEF0		Oui		
IDEF1	Oui			
IDEF3			Oui	
Petri nets			Oui	
RAD	Référence		Oui	Oui
EPC	Référence	Référence	Oui	Référence
REA	En partie	Oui	Oui	Oui
BPML	Référence		Oui	En partie
WPD	Référence		Oui	Référence
RosettaNet	Oui		Oui	
EbXML	Oui		Oui	En partie
BPEL4WS	Référence		Oui	Oui
EDOC	Oui	Oui	Oui	Oui
UML2	Oui	Oui	Oui	Ingrédients

**Tableau 2.1: Comparaison de langages de modélisation de processus en terme de portée.**

Le tableau 2.2, montre les utilisations possibles d'un modèle de processus d'affaires écrit dans différents langages. Nous avons utilisé la catégorisation de Ould (section 2.3.1). Comme nous pouvons constater, les langages qui communiquent la description de processus aux humains ne supportent pas souvent les tâches qui exigent une formalité et une précision concernant l'analyse quantitative et l'implantation. De plus, la lisibilité de l'aspect machine réfère plus à l'existence du format de sérialisation qu'au niveau de formalité.

Langage	Description		Analyse		Implantation	
	Humain	Machine	Qualitative	Quantitative	Simulation	Exécution
IDEF0	Oui		Oui			
IDEF1	Oui		Oui			
IDEF3	Oui		Oui			
Petri nets	Plus/moins	Oui	Plus/moins	Oui	Oui	Plus/moins
RAD	Oui		Oui	Plus/moins	Plus/moins	

EPC	Oui	Oui	Oui	Possiblement		
REA	Oui		Plus/moins	Oui		
BPML	Oui	Oui			Oui	
WPDL		Oui			Oui	Oui
RosettaNet	Oui		Oui		Plus/moins	
EbXML		Oui	Oui		Plus/moins	
BPEL4WS		Oui		Oui	Oui	Oui
EDOC	Oui	Oui	Oui	Possiblement	Oui	
UML2	Oui	Oui	Oui	Possiblement	Oui	

**Tableau 2.2: Utilisation potentielle des différents langages.**

Plusieurs langages se sont dotés de formats de sérialisation textuels qui supportent l'échange de description du processus. Citons EDOC et UML2 qui utilisent XMI (« XML Meta-data Interchange ») et BPEL4WS, ebXML, BPML ainsi que XPDL, qui utilisent XML. Par ailleurs, bien qu'il existe des langages qui permettent l'analyse quantitative, leur type d'analyse peut varier. Avec « peut-être » nous faisons référence au fait qu'un modèle peut être simplifié pour présenter quelques types d'analyses. Par exemple, des chercheurs ont souvent essayé d'extraire des réseaux de Petri des modèles dynamiques pour effectuer des analyses.

## 2.9 Conclusion

Dans ce chapitre, nous avons parcouru un ensemble de langages pour la représentation des processus d'affaires. Ces langages proviennent de différentes traditions scientifiques. Ils ont été divisés en quatre catégories: les langages de modélisation traditionnels, les langages de modélisation des workflows, les langages exécutables et les langages de modélisation orientée objet. Ces langages ont été définis en utilisant des techniques bien différentes. Nous avons remarqué que ces langages sont guidés par les besoins des utilisateurs; ainsi, ils sont toujours en évolution. Également, ces langages couvrent différentes facettes d'un processus d'affaires. Notre objectif est de choisir une notation qui a la capacité de représenter les différentes vues de processus, et possède les concepts nécessaires pour définir un processus d'affaires. Dans ce contexte, nous avons établi que la convergence des efforts vers le langage UML, en fait un langage de choix. En plus de ces concepts qui permettent de représenter un processus d'affaires, sa notation graphique facilite la lisibilité et la communication pour différents types d'utilisateurs, et son méta-modèle permet l'ajout de nouveaux concepts si nécessaire en utilisant des mécanismes d'extensions.

# **Chapitre 3. Initiatives de modélisation de processus d'affaires génériques**

## **3.1 Introduction**

Avec l'orientation des entreprises vers une vision par processus, le développement de systèmes d'information pour supporter les processus d'affaires est devenu de plus en plus basé sur l'hypothèse qu'il est possible de construire ces systèmes à partir d'éléments prédéfinis [Zachman, 1982]. Une telle approche doit permettre de réduire le temps d'élaboration de systèmes d'information, d'en améliorer la qualité et d'en faciliter la maintenance [Mili et al., 1995]. Dans ce chapitre nous étudions un certain nombre d'initiative de modélisation et de répertoire de processus d'affaires génériques ou des composants logiciels les supportant.

Dans la prochaine section, nous allons présenter nos critères de classification des différentes initiatives de modélisation en nous basant sur les objectifs visés par chaque initiative. Une distinction est établie entre les initiatives visant les processus internes et celles visant les processus d'intégration. Dans la troisième section, nous présenterons les initiatives qui répertorient les processus d'affaires internes. Les initiatives qui traitent des composants d'affaires sont décrites dans la quatrième section. À la cinquième section, nous aborderons les initiatives des cadres d'application. La sixième section, sera consacrée à des initiatives orientées processus externes. Finalement, une discussion suivie d'une conclusion seront présentées dans les deux dernières sections.

## **3.2 Cadre de classification des initiatives de modélisation**

De nombreuses recherches portant sur la réutilisation en ingénierie des systèmes d'information ont été menées récemment. Ces recherches portent aussi bien sur les composants logiciels qui interviennent durant la phase d'implantation des systèmes d'information, que sur les composants de conception, d'analyse, d'affaires, voire même d'ingénierie des besoins et d'interactions d'affaires. Plusieurs auteurs ont tenté de classer et d'évaluer ces travaux (ex: [Mili et al., 1995], [Fellner, 2000], [Medjahed, 2003]). Pour notre part, un objectif de notre recherche est de promouvoir la

réutilisation des modèles de processus d'affaires et des composants logiciels qui les supportent. Ainsi, nous avons porté notre attention sur un ensemble d'initiatives qui ont essayé de faciliter l'ingénierie des systèmes d'information en réutilisant des modèles complets ou partiels de processus d'affaires. Nous avons choisi de classer ces initiatives de modélisation en nous basant tout d'abord sur l'intention d'utilisation des modèles qu'elles fournissent et en essayant de les évaluer selon les critères suivants:

- *Les objectifs*: Ould indique que la raison de notre modélisation affecte nos modèles [Ould 1995 p.163]. L'objectif de chaque initiative dévoile le degré de correspondance visé entre le modèle à créer et la réalité. Mili et al. précisent que la description d'un point de vue vise les aspects essentiels à ce point de vue et élimine ceux qui ne lui sont pas pertinents [Mili et al., 2002].
- *L'organisation*: Les différentes approches exposent différentes méthodes pour classer et représenter les modèles qu'elles proposent. Connaître les stratégies et les méthodologies de présentation utilisées par chacune des initiatives aide à identifier les différents paramètres qui assurent la généricité des modèles. De plus, il faut savoir dans quelles circonstances ces modèles sont applicables pour identifier leur degré de réutilisation.
- *La notation de modélisation*, précédemment détaillée, permet d'identifier la qualité syntaxique des modèles.
- *Les mécanismes d'adaptation*: Étant donné que ces initiatives produisent des modèles génériques, des techniques d'adaptation et de spécialisation de ces modèles doivent être fournies. Parmi ces techniques, notons l'héritage, la substitution, la complétion, l'instantiation, et d'autres.
- *L'emballage* : il y a différentes considérations :
  - Le niveau de développement : Certaines initiatives produisent juste des modèles de processus d'affaires (exigences), d'autres des modèles d'analyse et d'autres des bibliothèques de classes.
  - Niveau de granularité : Certains composants sont de petite taille (à l'échelle d'une classe Java par exemple), d'autres sont des agrégats tels des patrons, des cadres d'application et des cadres d'entreprise.

Durant notre étude et dans l'évaluation des initiatives décrites dans les sections suivantes, nous avons identifié deux grandes familles: a) les initiatives de description des processus privés ou internes aux organisations, b) les initiatives orientées intégration ou coordination de processus.



**Figure 3-1: classification des initiatives de modélisation**

Dans chaque famille, nous avons identifié trois sous familles (Figure 3-1): a) les approches qui répertorient les processus (ex: ebXML), b) des approches qui répertorient les composants supportant ces processus et c) des approches qui concernent l'établissement de lien entre les approches précédentes. Par exemple, IBM San Francisco [Bohrer, 1998], Coad/Lefebvre [Coad et al., 1999] et RosettaNet [RosettaNet, 2003] se concentrent sur la description de composants logiciels. Accessoirement, ces initiatives présentent des fragments de processus supportés par les composants. Au sein de cette famille, on trouve des dépôts de modèles de niveau analyse (les composants de Coad & Lefebvre [Coad et al., 1999], patrons d'analyses [Fowler, 1997], etc.), ainsi que des approches incluant des composants au niveau conception et implémentation (code source, dont IBM San Francisco). À noter que RosettaNet est une initiative orientée vers l'intégration de composants inter-organisationnels. D'autres approches se concentrent sur le répertoriage de processus d'affaires, sans se préoccuper de leur support informatique, notamment l'exemple du manuel MIT («MIT Handbook») [MIT, 2004] et SCOR («Supply Chain Operations Reference-model») [SCOR, 2001]. Une autre famille est celle qui concerne l'établissement d'un lien entre les processus d'affaires et les composants d'affaires. Par exemple, le système de gestion de workflow (WfMS), qui vise à gérer et à ordonnancer les différentes activités et composants qui constituent un processus d'affaires [WfMC, 2002a] ( voir section 2.5).

Il faut noter que la distinction entre les catégories n'est pas nette. En effet, chaque initiative présentée va mettre l'accent sur un aspect plutôt qu'un autre. De plus, il existe également des approches qui englobent tous les aspects. Cependant, dans quelle mesure ces initiatives supportent-elles ces aspects? Prenons le cas du progiciel SAP [Curran, 1998,2000], qui, en plus de son domaine d'affaires, cherche à développer des systèmes d'information qui supportent directement des processus d'affaires. Malheureusement, ses mécanismes d'extension et d'instanciation ne sont pas publics. De même, les modèles et les composants de ce progiciel sont à usage interne, ils sont confidentiels et ils sont utilisés par les consultants de SAP.



### **3.3 Initiatives orientées description de processus d'affaires internes**

Les initiatives qui visent à décrire les processus d'affaires internes de l'entreprise ne se préoccupent pas de la description des interactions et des collaborations avec d'autres processus provenant de l'extérieur. Ce type d'initiative vise à présenter le contenu d'un processus et les connaissances qui peuvent le former. Dans les prochaines sections, nous allons aborder les initiatives qui traitent des modèles partiels, dont les patrons et d'autres qui traitent des modèles complets, dont les cadres d'application. En outre, il existe des initiatives qui visent spécialement à répertorier et classifier les processus d'affaires dont le projet du manuel MIT.

#### ***3.3.1 Initiative pour répertorier des processus d'affaires: MIT Process Handbook Project***

##### **Objectifs**

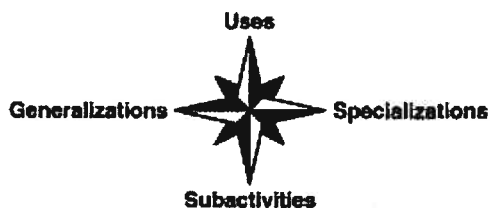
Ce manuel (« hand book ») est une bibliothèque en ligne, créée par le centre de coordination et le centre du commerce électronique du MIT (Massachusetts Institute of Technology). Ce manuel en service depuis 1991, sert à classifier et partager des connaissances concernant les domaines et les processus d'affaires [MIT, 2004]. Il est constitué de trois sections de base: a) des modèles génériques des processus d'affaires typiques qui s'appliquent à différents domaines d'affaires (ex: achat, production, ventes) b) des exemples spécifiques de modèles d'affaires provenant des organisations particulières c) un cadre de classification de ces modèles [Malone et al., 2003]. Ce manuel est une base de connaissances, qui permet aux différents types d'utilisateurs de mettre à jour leurs connaissances et de trouver de nouvelles idées pour gérer et créer les affaires. Il contient des alternatives et de bonnes pratiques pour concevoir un modèle d'affaires adéquat répondant à des besoins en appliquant la ré-ingénierie ou l'étalonnage (« benchmarking ») [Malone et al., 1999]. Les modèles génériques fournis peuvent être utilisés comme le point de départ pour la modélisation spécifique des entreprises particulières, des processus et des modules logiciels [Malone et al., 2003], rejoignant notre objectif de recherche. Cependant, ce manuel met l'accent sur la classification, et ce, sans tenir compte du support informatique. Ce n'est donc pas une technologie qui permet d'implémenter les processus.

## Méthodes et notations de représentation

Ce manuel ne fournit aucun modèle graphique ou normalisé des processus d'affaires. Une explication textuelle restreinte des processus est présentée. Toutefois, les concepteurs de ce manuel mentionnent l'intérêt que présente le formalisme « Process Interchange Format (PIF) », qui aide à transmettre la description des processus d'un système à un autre [Malone et al., 1999].

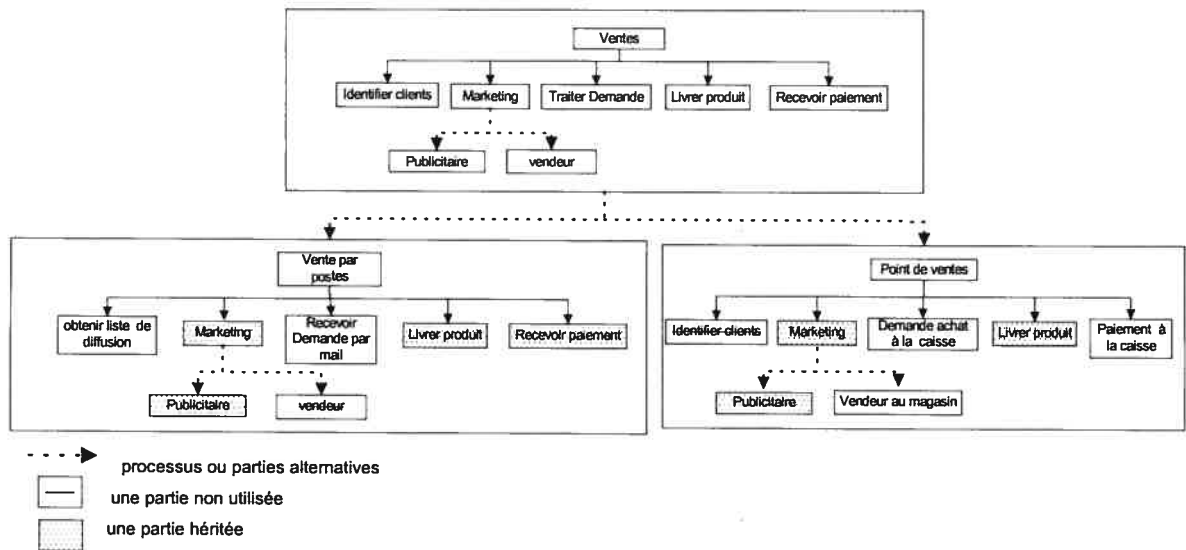
## Organisation

La classification et la représentation des processus d'affaires avec le manuel MIT utilise une approche de hiérarchisation qui se base sur les notions de spécialisation de processus basées sur le concept d'héritage de la programmation orientée objet et sur les concepts de dépendance que l'on retrouve dans la théorie de la coordination [Malone et al., 1999]. Les modèles du manuel vont du général au spécifique, en utilisant deux dimensions. La première basée sur l'agrégation de différentes parties d'un processus, et la seconde basée sur la spécialisation. Ces dimensions sont représentées graphiquement par une boussole de processus («process compass» Figure 3-2).



*Figure 3-2 :Boussole de processus (« Process Compass »)(figure extraite de [Malon et al. 1999])*

Par exemple, partant d'un processus générique de « vente d'un produit » qui est formé d'un ensemble de parties, nous pouvons avoir deux spécialisations « vente par la poste sur commande » et « ventes dans un point de vente » (Figure 3-3).



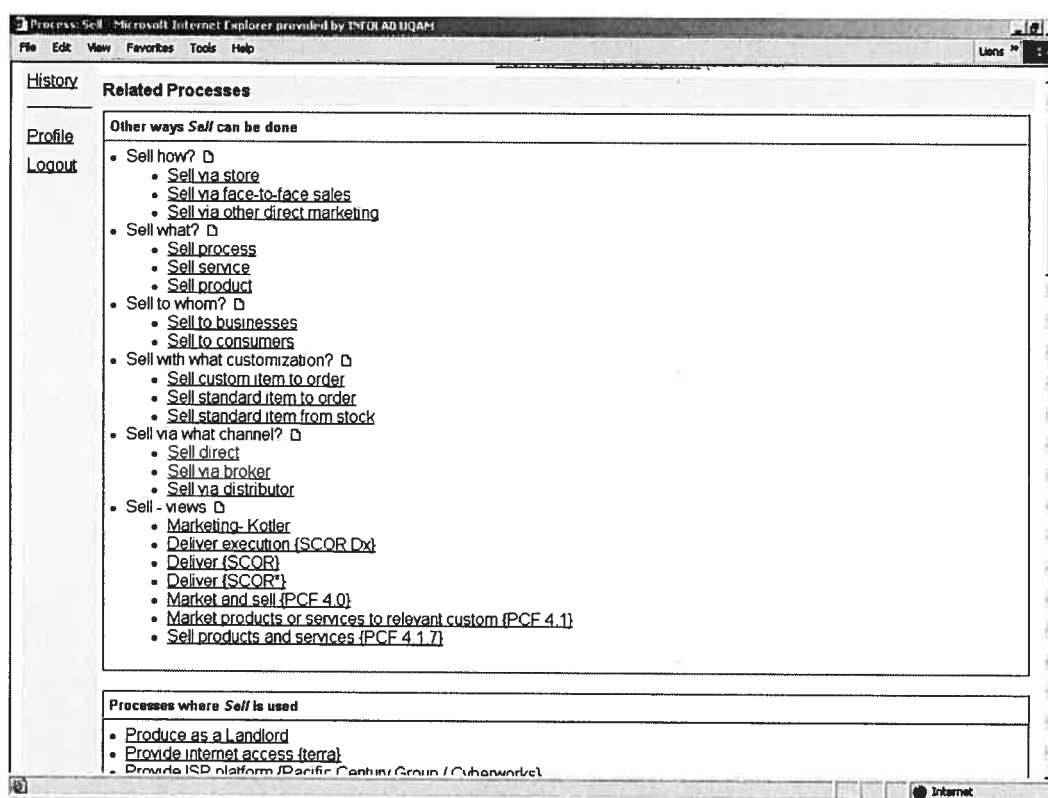
**Figure 3-3 :Processus de ventes alternatifs à différents niveaux de décomposition et de spécialisation [Malone, 1999]**

Un autre concept clef employé avec le manuel MIT est la *coordination*, considérée comme la gestion de dépendances entre les activités d'un processus. Chaque dépendance peut avoir un mécanisme de coordination associé. Le mécanisme de coordination peut être réutilisé, et il peut être un processus à part entière qui gère le flux de ressources et coordonne ainsi les activités connectées. Le manuel MIT présente trois types de dépendances: a) dépendance de flot: un processus produit une ressource consommée par une autre ressource ; b) dépendance partagée: deux processus consomment la même ressource ; c) dépendance remplie « fit »: deux processus produisent la même ressource. Ces dépendances sont elles mêmes des compositions ou des spécialisations d'autres dépendances plus élémentaires [Malone et al., 1999].

Le manuel MIT utilise aussi différentes vues pour organiser les modèles des processus d'affaires: vue par « fonctionnalités d'affaires » typiques qui passent dans différents domaines d'affaires (ex : gestion de chaîne logistique, marketing, ventes, etc.), vue par « types d'activités » (ex: détruire, créer, modifier, etc.) et vue par modèles d'affaires. Le manuel fournit plusieurs exemples avec chaque catégorie, comme il fournit un nombre de modèles développés par différentes organisations externes.

Pour la vue par « fonctionnalités d'affaires » ou vue par agrégation, le modèle de haut niveau est le modèle d'affaires production (« Produce as a business »). Le deuxième niveau contient les sous-processus : acheter, concevoir, fabriquer, vendre et gérer une affaire. L'idée est de placer toutes les

descriptions des processus dans une hiérarchie d'agrégation [MIT, 2004]. La plupart de ces sous-processus ont de nombreux types spécialisés reflétant des informations sur leurs variations selon les différentes entreprises. De même, ces sous-processus ont des dérivés et des alternatives qui dépendent de l'application et du domaine que l'on veut présenter. Si nous reprenons l'exemple du processus « ventes », la Figure 3-4 représente les dérivés et les alternatives de ce processus.

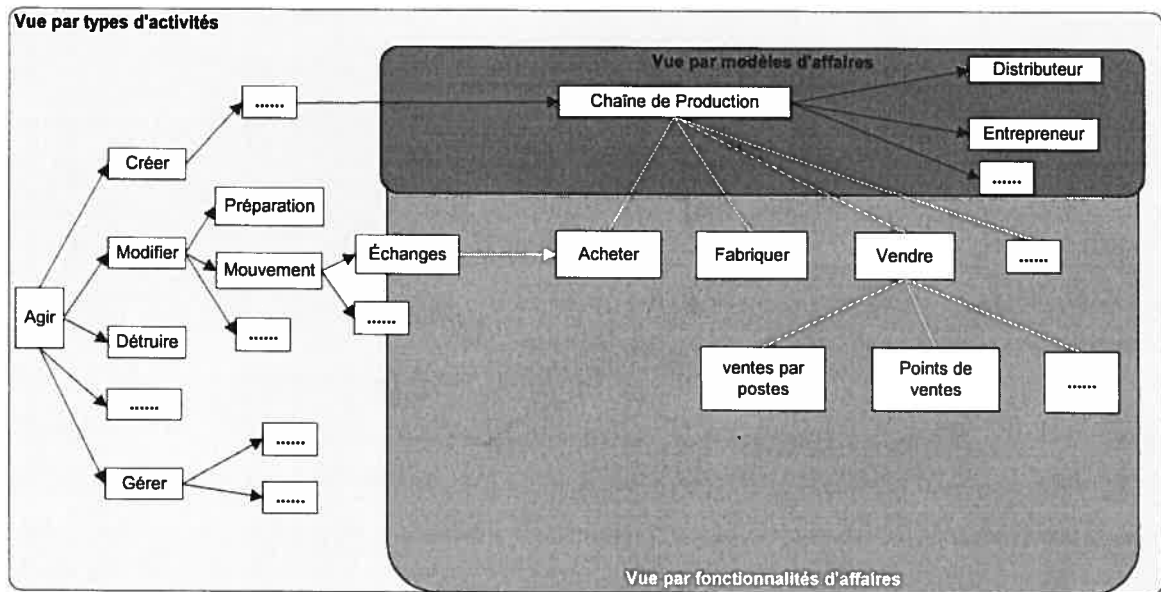


**Figure 3-4: les spécialisations du processus « ventes » (tiré de [MIT, 2004])**

Pour la vue par « type d'activité » ou vue de spécialisation, tous les modèles sont regroupés par familles ayant des caractéristiques communes. Par exemple, les processus d'affaires « ventes » et « achat » appartiennent à une catégorie d'activités plus générique, «échanges» qui est elle-même une spécialisation de la catégorie « mouvement ». Cette dernière appartient à la catégorie « modifier », qui est une spécialisation de la catégorie « agir ». « Agir » est la catégorie la plus générale : toutes les activités dans le manuel en sont une spécialisation directe ou indirecte (voir Figure 3-5).

Pour la vue par modèles d'affaires, le manuel MIT présente six archétypes. Ces modèles sont présentés suivant deux dimensions: a) l'objectif de l'entreprise et b) comment l'entreprise réalise des profits. Les six modèles sont: distributeur, producteur, intermédiaire, propriétaire, entrepreneur,

« attracteur » [Malone, 2003]. Malone et al. précisent qu'en pratique les entreprises utilisent une combinaison de ces archétypes, mais ils ont estimé que ces modèles peuvent être utilisés pour classifier les différentes combinaisons qui existent en réalité. Cependant, il faut noter que ces modèles sont introduits dans le manuel MIT comme une spécialisation du modèle de production « Produce as a business » dans un paquetage dont le titre est « produire avec quel modèle d'affaires ? ».



*Figure 3-5 : Plan simplifié du manuel MIT*

Pour récapituler, la Figure 3-5 représente un plan simplifié de la classification des processus utilisée dans le manuel MIT. Ce plan permet d'avoir un aperçu général de leur méthodologie. Nous identifions, que le processus « Ventes par poste » par exemple, peut être atteint par trois vues différentes.

### Mécanismes d'adaptation

Le manuel MIT utilise des techniques pour adapter ou spécialiser les processus génériques à un cas particulier :

- a) La spécialisation, en passant d'un niveau à un autre dans la hiérarchie fournie;
- b) La complétion, en supposant que l'utilisateur va suivre les directives fournies par les modèles pour construire son propre modèle dépendamment de son contexte d'affaires ;
- c) La substitution en remplaçant certains activités par d'autres spécifiques au domaine d'application (voir Figure 3-3).

Pour intégrer une nouvelle description au répertoire, deux approches peuvent être utilisées: «top down» et «bottom up». Le point de départ de l'approche «top down» est un processus similaire du répertoire à partir duquel on crée une spécialisation modifiable. Dans l'approche «bottom up», le processus est lié à des processus d'un niveau plus élevé du répertoire.

Le manuel utilise aussi l'approche par question pour organiser les processus. Cette approche dans le cadre du manuel MIT aide à regrouper les spécialisations des processus reliés dans des paquets (catégories). Ces paquets sont créés en se basant sur les questions habituellement applicables à une activité (comment ? quoi ? qui ? quand ? ou ? et pourquoi ?). Ces questions sont applicables à la plupart des activités et processus (voir Figure 3-4). Nous reviendrons là dessus dans le chapitre 4.

## **Emballage**

Ce manuel fournit des modèles des processus d'affaires. Cependant ces modèles sont des coquilles vides qui présentent seulement des descriptions textuelles des processus d'affaires, ils fournissent les fonctionnalités de haut niveau qu'un processus doit englober, mais ne fournissent pas de logique de comportement.

### ***3.3.2 Initiatives orientées répertoriage des composants d'affaires***

Dans la section précédente nous avons identifié l'initiative MIT qui vise à classifier des processus d'affaires sans se préoccuper du support informatique. Dans ce qui suit nous allons présenter deux initiatives qui visent à présenter les connaissances formant un processus d'affaires. Cependant, elles en répertorient des modèles partiels. La première initiative (les patrons d'analyse) concerne la modélisation du monde pseudo-réel des processus d'affaires tandis que la deuxième (les patrons d'affaires) traite de modélisation réelle (voir section 2.3).

#### **3.3.2.1 Les patrons d'analyse**

Un patron (pattern) est une solution à un problème récurrent qui pourra être réutilisée et adaptée à un contexte donné. Un patron d'analyse a pour but de guider les étapes d'analyse lors de l'ingénierie des systèmes d'information. Il permet d'identifier les problèmes récurrents dans l'expression des besoins des applications et de transformer ces expressions en des modèles réutilisables [Coad, 1992]. Une des premières contributions aux patrons d'analyses est celle de Coad (voir [Coad, 1997]). Les caractéristiques de ces patrons sont leur petite taille et leur généralité.

Martin Fowler a aussi proposé des patrons d'analyse pour quelques domaines d'affaires [Fowler, 1997]. D'autres travaux dans la littérature existent sur le sujet notons ceux de Hay [Hay, 1996]. Le site de la conférence PLOP « Pattern Languages of Programs Conference » contient aussi un ensemble d'articles sur des patrons d'analyses pour différents domaines d'affaire tels que la location, la gestion des ressources, etc... [PLOP].

## Objectifs

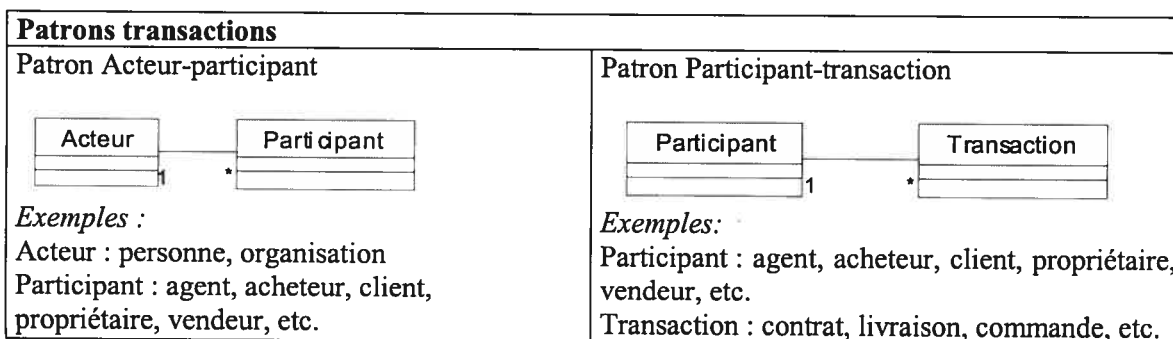
L'objectif des patrons, que ce soit d'analyse ou de domaine, est d'apprendre à l'utilisateur comment résoudre correctement les problèmes qu'il peut rencontrer durant l'analyse d'un domaine d'affaires dans le but de construire des systèmes logiciels. Cependant, les modèles fournis avec les patrons d'analyse ne couvrent pas les spécificités du domaine qu'ils présentent.

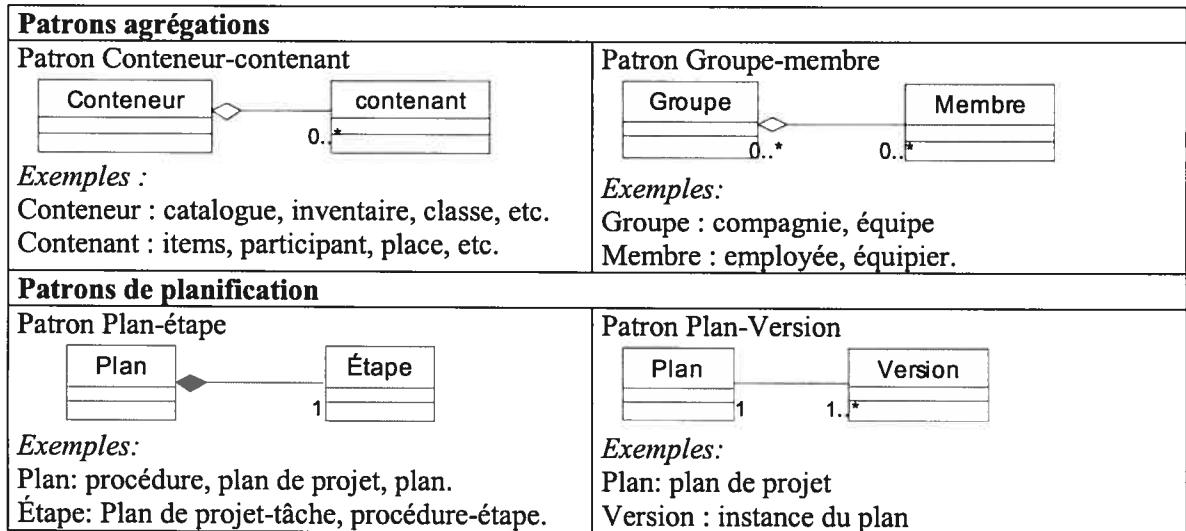
## Méthodes et notations de représentation

Il existe plusieurs formalismes permettant la description des patrons. Ces formalismes peuvent être structurés ou narratifs et peu structurés. En pratique, ces formalismes contiennent des rubriques particulières destinées à la compréhension et à la sélection des patrons, et qui sont souvent décrites en langue naturelle. Seules les rubriques exprimant les solutions contiennent des notations formelles (les diagrammes OMT avec les patrons de Coad, les diagrammes entité-relation avec les patrons de Hay, les diagrammes pré-UML avec les patrons de Fowler). Ces diagrammes présentent des modèles très détaillés du point de vue de la structure, complétés d'exemples bien choisis. Toutefois, avec la plupart des patrons, surtout ceux de Fowler, la description du comportement est abstraite et embrouillée par les explications textuelles.

## Organisation

Les patrons de Coad sont divisés en quatre catégories: transaction, agrégation, plan et interaction. La Figure 3-6 présente un nombre de ces patrons organisés par catégorie.





**Figure 3-6 : Échantillons des patrons fournis par Coad [Coad, 1997]**

Ces patrons sont de petite taille, formés au plus de trois entités et ne sont pas liés à un domaine particulier. Coad démontre leur applicabilité en ayant recours à quatre systèmes différents: points de ventes, inventaire, saisie commande, et temps réel. Ces patrons constituent une sorte de méta-modèle pour représenter un processus d'affaires (voir l'approche Coad et Lefebvre section 3.3.4.1 qui utilise ces patrons dans ce but [Coad et al., 1999]). La classification de ces patrons est claire, mais vu la généralité de ces derniers, il reste difficile d'identifier quand et comment chaque patron peut être utilisé dans différents domaines d'affaires.

Les patrons présentés par Fowler couvrent des domaines aussi variés que la comptabilité, les mesures, les observations, l'organisation, les inventaires, la planification, les transactions et les contrats. Il n'existe pas de classification claire de ces patrons et surtout on ne trouve pas une classification des patrons isolés. Les patrons sont présentés par domaine pour résoudre un problème trivial en fournissant plusieurs adaptations et généralisations du même problème. Pour identifier un patron, les utilisateurs doivent établir une correspondance entre le domaine qu'ils modélisent et les domaines présentés par Fowler pour choisir les patrons qui leurs sont utiles.

Il faut préciser, qu'il n'existe pas de structure de navigation précise pour tous les patrons d'analyses.

## Mécanismes d'adaptation

Le mécanisme d'adaptation fournis par Fowler se résume en une sélection suivie d'une adaptation libre en effectuant des substitutions ou des complétions quand c'est nécessaire. L'application des patrons d'analyse est étroitement liée aux problèmes du domaine identifié durant l'analyse.



L'utilisateur des patrons doit examiner le problème en détail et mettre au point une méthode pour employer/instantier le modèle dans le contexte spécifique où il les utilise [Fowler, 1997].

Coad présente des stratégies pour assister l'analyste durant la phase d'analyse en appliquant les patrons pour construire des modèles objets. Parmi ces stratégies, Coad utilise les questions pour identifier les attributs « **what I know** », les liens « **who I know** », et les services fournis « **what I do** » de chaque entité, ce qui constitue une technique qui permet de construire et d'étendre les modèles d'objets (cette technique sera détaillée au chapitre 4). Cependant, les stratégies de Coad ne sont pas formelles et sont difficiles à implémenter.

## **Emballage**

Les patrons sont présentés sous la forme de modèles d'analyses qui vont aider le concepteur à analyser et exprimer ses besoins. Ces modèles représentent des petits fragments des modèles de données des processus d'affaires et ils sont considérés comme des composants de modélisation.

### **3.3.2.2 Les patrons d'affaires**

Ces initiatives considèrent la modélisation du monde réel des domaines d'affaires plutôt que celui des systèmes logiciels (section 2.3.1.2). Rappelons que le modèle d'affaires permet à l'analyste de faire le point sur les concepts à implanter dans un logiciel ou à implanter d'une autre façon (ex: processus manuel). Nous avons identifié les travaux de Hans et al., qui présentent un catalogue des patrons d'affaires.

## **Objectifs**

Le catalogue est constitué de patrons permettant de construire des modèles structurels et comportementaux de domaines d'affaires [Hans, 2000]. Comme beaucoup d'autres chercheurs (voir chapitre 2), Hans et al. voient la nécessité d'avoir différentes vues pour gérer la complexité des domaines d'affaires: vue d'affaires, vue structurelle, vue comportementale, et vue de processus. Les vues ne forment pas de modèles séparés, ce sont plutôt des perspectives différentes sur les mêmes modèles d'affaires [Hans, 2000].

## Méthode et notations de représentation

Les différents patrons sont représentés par un formalisme formé de plusieurs rubriques. La plupart des rubriques sont textuelles. Mais elles sont également exprimées avec différents diagrammes de UML1.x. Cependant, des extensions ont été proposées au diagramme d'activités pour modéliser explicitement les processus d'affaires et les éléments qu'ils impliquent (Figure 3-7). La notation étendue ressemble beaucoup à celles de IDEF.

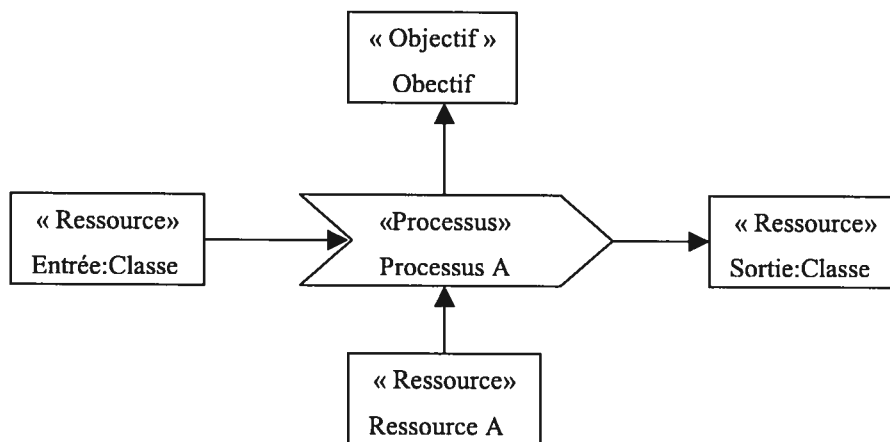


Figure 3-7: Diagramme d'activités étendue par Hans et al. [Hans, 2000].

## Organisation

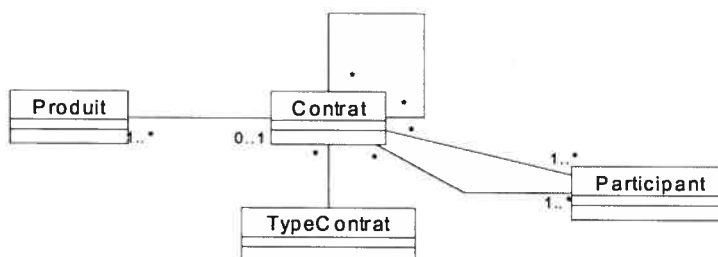
Le catalogue des patrons d'affaires fournis par Hans et al. aborde les mêmes domaines d'affaires que Fowler à l'exception du domaine d'analyse financière. Néanmoins, les patrons sont classifiés d'une façon indépendante du domaine d'affaires. Ils sont répertoriés en trois parties : patrons de processus, patrons d'objectifs et patrons de ressources.

- *Patrons de processus*: ce sont des patrons comportementaux et fonctionnels qui facilitent la modélisation des domaines d'affaires. Ces patrons sont divisés en trois sous-catégories : a) ceux qui décrivent la façon de représenter un processus (ex: patron pour présenter la structure de base d'un processus (Figure 3-7)), b) ceux qui représentent la différence entre un processus décrit et son instance (ex : patron instance processus-processus) et c) ceux qui servent de support pour décrire des problèmes communs et leurs solutions dans le contexte d'un processus en exécution (ex: patron d'utilisation de ressource, patron de l'état d'instance d'un processus).
- *Patrons d'objectifs* : ces patrons décrivent des résolutions typiques pour analyser et modéliser les objectifs d'affaires. Ces patrons sont: le patron d'attribution des objectifs à un processus d'affaires, le patron de décomposition des objectifs et le patron

d'identification des objectifs et leurs problèmes. En effet, la modélisation des objectifs d'affaires influence le processus de modélisation du domaine d'affaires, comme elle représente les cibles d'affaires désirés par le domaine.

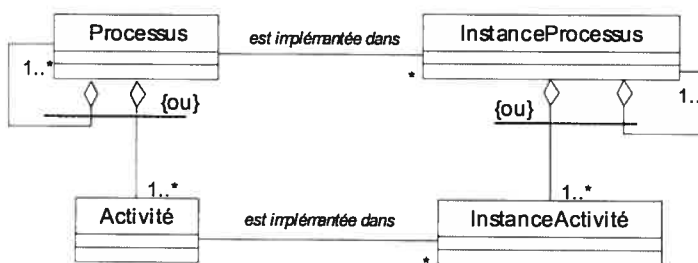
- *Patrons des ressources et des règles*: Ce sont des patrons pour modéliser les ressources actives ou passives qui participent à un processus d'affaires (ex: patron acteur-rôle, patron contrat, patron document).

La plupart des patrons de ressources et quelques patrons de processus sont des modèles d'affaires. Par contre, la plupart des patrons de processus et les patrons des objectifs sont des patrons méthodologiques pour résoudre des problèmes de modélisation. En d'autres termes, ils présentent des heuristiques de modélisation. Par exemple, le patron de structure « contrat » fournit les directives pour modéliser les concepts majeurs qui forment un contrat (Figure 3-8).



**Figure 3-8 : Patron « Contrat » [Hans, 2000]**

Par ailleurs, si nous prenons par exemple le patron de processus « Instance processus-processus » (Process-Process instance pattern), nous constatons que ce patron sert à clarifier la distinction entre un processus et son instance (Figure 3-9).



**Figure 3-9 : Patron « Instance processus-processus » [Hans, 2000]**

## Mécanismes d'adaptation

Etant donné que ces patrons sont des patrons d'affaires, il n'existe pas de mécanisme d'extension spécifique. C'est à l'utilisateur de préciser ses besoins et de choisir le patron correspondant.

En d'autres termes, la technique utilisée pour appliquer un patron est la même que celle des patrons d'analyse; la sélection.

## **Emballage**

Ces modèles sont des patrons d'affaires. Ils sont de deux types: a) des patrons de solutions et b) des patrons d'heuristiques de résolution. Ces modèles aident à préciser et modéliser les exigences d'affaires.

### **3.3.3 Initiatives orientées cadre d'applications**

Avec les patrons d'analyse ou d'affaires, les domaines d'affaires traités sont esquissés plutôt qu'entièrement définis. Les patrons d'analyse, tels que ceux de Fowler, mettent l'emphase sur la présentation des solutions aux problèmes que nous trouvons durant l'analyse d'un système d'information. Hans et al. vont en amont et présentent des patrons pour modéliser les domaines d'affaires selon différents points de vue. Néanmoins, les différents patrons sont utiles d'une façon ou d'une autre pour modéliser les systèmes d'informations. Ces patrons sont des fragments de modèles, et nous pouvons même les considérer comme des composants logiques réutilisables qu'ils soient logiciels ou non. Ils sont individuellement parfaitement spécifiés mais assemblés ils peuvent causer des problèmes sémantiques.

Différents chercheurs ont essayé d'établir des patrons d'analyses génériques réutilisables qui visent à systématiser des bibliothèques de patrons d'analyses en établissant un lien entre eux notons les travaux de Johonnesson [Johonnesson, 1999]. D'autres chercheurs ont identifié le besoin d'avoir des liens entre différents patrons pour présenter des modèles complexes. Le résultat était une agrégation sémantique de plusieurs patrons simples pour représenter un modèle de processus d'affaires. Notons les travaux de Fernandez [Fernandez, 2000], de Paludo [Paludo et al., 2000] et d'Arsanjanni [Arsanjanni, 1999]. Ces différentes recherches illustrent le besoin d'avoir un cadre global de modélisation pour présenter un processus d'affaires récurrent, au lieu de se limiter à des fragments de processus de petite taille. Nous décrivons par la suite des cadres d'application qui proposent leurs propres approches de conception et d'implémentation du modèle des processus et des composants logiciels que nous en avons déduit: les modèles de composants d'entreprise de Coad et Lefebvre [Coad et al., 1999], le projet San Francisco [IBM, 2003] et les modèles de référence du progiciel SAP [Curran et al., 2000].

### 3.3.3.1 Les processus et les composants d'entreprise [Coad et al., 1999]

#### Objectifs

Coad et al. fournissent un ensemble de composants d'affaires qui présentent des modèles de processus d'affaires sous le nom de composants génériques au domaine. Ces composants aident à la construction du modèle d'une application ou d'une famille d'applications [Coad, 1999], comme ils peuvent aider dans la ré-ingénierie des processus d'affaires. Coad et al. ont introduit un composant indépendant du domaine « Domain-Neutral Component » (DNC) qui est construit à partir des concepts qui représentent un processus d'affaires (voir chapitre 2) [Coad et al., 1999] (Figure 3-10). En effet le DNC est une constellation de patrons de Coad [Coad, 1992] qui utilisent la notion d'archétypes<sup>4</sup> coloriés [Coad, 1997]. Il peut être appliqué et spécialisé dans les domaines d'affaires pour modéliser les composants d'affaires. Il faut noter que les composants génériques représentés par Coad et al. sont des modèles de processus d'affaires qui ne sont pas assez détaillés. Ils doivent être étendus ou modifiés selon le contexte du domaine à modéliser.

#### Méthodes/notations de représentation

La notation utilisée avec cette approche est le langage de modélisation UML, notamment les diagrammes de séquences, les diagrammes de classes et les stéréotypes. Ces derniers sont utilisés pour représenter les différents archétypes en leur appliquant différentes couleurs. La notation par couleur vise à faciliter la lisibilité et la compréhension. Elle offre une technique pour analyser rapidement un problème du domaine et construire un modèle objet [Coad et al., 1999].

Les différents archétypes sont (Figure 3-10):

- L'archétype moment/intervalle (MI) (Rose): représente les objets qui dépendent d'un moment ou d'un laps de temps. Ces objets retracent les événements ou les activités d'affaires. Ils lient le modèle du composant et contiennent les méthodes importantes pour calculer les résultats et pour les évaluer.
- L'archétype rôle (Jaune) : représente un rôle joué par quelqu'un ou quelque chose. Il génère un événement ou accomplit une activité. Il est associé au moment/intervalle et contient des méthodes pour évaluer la performance des acteurs.

---

<sup>4</sup> L'archétype est une forme à partir de laquelle on groupe une catégorie. Il n'est pas fixe comme les stéréotypes

- L'archétype description (Bleu) : représente une entrée semblable au catalogue, un jeu des valeurs qui s'appliquent répétitivement.
- L'archétype participant/place/chose (Vert) : Représente un joueur de rôle, c'est une entité uniquement identifiable.

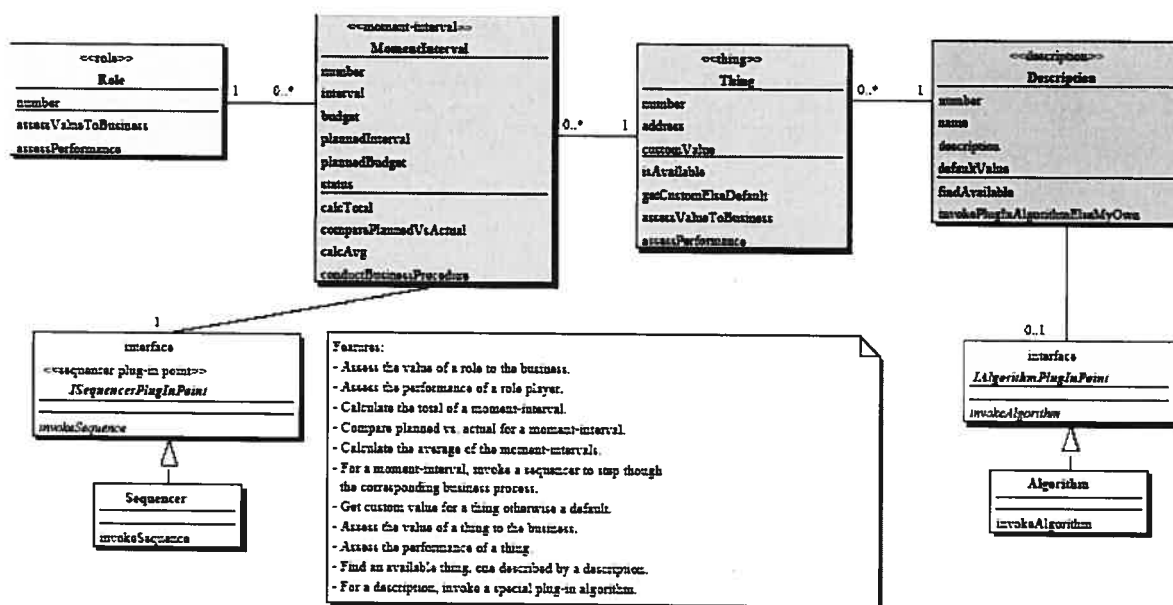


Figure 3-10 : Composant indépendant du domaine [Lefebvre et al., 1999]

## Organisation

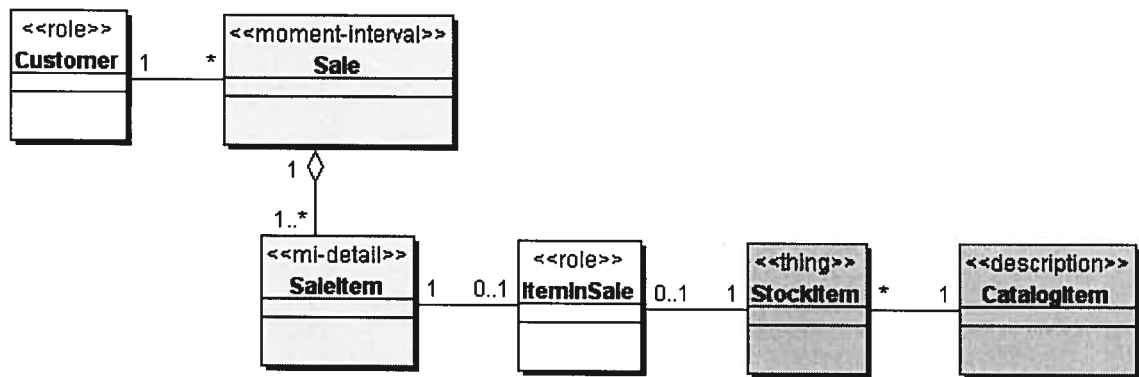
Coad et Lefebvre utilisent deux niveaux de classification: Le premier niveau distingue les processus d'affaires horizontaux reconnus: la gestion, le support, la vente, l'achat et la production. Cette classification rencontre celle du modèle de référence de la chaîne de valeurs de Porter [Porter, 1985] (voir la section SAP pour comprendre ce modèle). Le deuxième niveau présente les sous-processus de base qui constituent les processus horizontaux. Chacun des sous-processus est une fonctionnalité d'affaires (ex : gestion d'inventaire, gestion de ressources humaines, etc.) présentée sous forme d'un composant élaboré.

Il faut noter qu'il existe un composant particulier sous le nom de la gestion du lien «relationship management» qui contient une présentation des ressources pouvant participer à un processus d'affaires, ainsi que leur rôle. Ce composant permet de mettre en évidence la nécessité d'avoir une vue organisationnelle à part (nous aborderons ce point au chapitre 4).

## Mécanismes d'adaptation

Les auteurs de cette approche suggèrent deux techniques de réutilisation: l'application du DNC au domaine à modéliser ou la réutilisation des composants déjà présentés dans le catalogue.

Pour appliquer le DNC à un domaine particulier il faut identifier les différents éléments de base du domaine (les acteurs, les événements et les ressources) pour une activité d'un processus. Ensuite, il faut attribuer ces éléments aux différents éléments du DNC en suivant les différentes directives présentées par les auteurs. Par exemple : Si nous prenons une activité simple de ventes, les concepts identifiés sont clients, ventes, produits et catalogue. Pour chaque concept nous attribuons l'archétype correspondant du DNC et nous obtenons le modèle suivant (Figure 3-11):



*Figure 3-11: Le composant processus de ventes (figure extraite de [Coad et al., 1999])*

En ce qui concerne les composants du catalogue, ils peuvent être utilisés tels quels. Ces composants peuvent également être étendus en ajoutant de nouvelles fonctionnalités à travers les points d'extension (plug-in point) identifiés dans chaque modèle. Les points d'extension sont des interfaces qui vont être implémentées par l'extension (Figure 3-10). Nous pouvons aussi modifier les modèles en éliminant les classes non désirées, en ajoutant des classes ou en modifiant les classes existantes et leurs annotations.

## Emballage

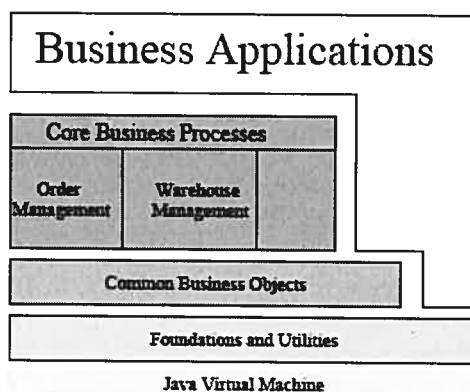
Les composants réutilisables de Coad et al. sont des modèles d'analyse qui facilitent l'analyse et la spécification des processus d'affaires. Ces modèles sont fournis sous la forme de fichiers enregistrés dans un outil de modélisation, qui permet de générer en langage Java ou C++ le code correspondant.

### 3.3.3.2 Le cadre d'application IBM San Fransisco

IBM San Fransisco est un cadre d'application qui fournit des composants (considérés comme des minis-cadres) qui aident à la création des applications d'affaires. Un cadre d'application est un ensemble typé de classes destiné à la fabrication d'un type donné d'applications. Récemment, les travaux avec ce cadre se sont intégrés dans « IBM Business components for WebSphere » pour traiter des composants qui touchent aux affaires électroniques [IBM, 2003]. Dans cette section nous allons parler de ce cadre et de sa stratégie de réutilisation.

#### Objectifs

L'objectif du cadre d'application « IBM San Fransisco » est de permettre le développement d'applications de gestion en s'appuyant sur un noyau de base commun et en réutilisant des composants d'affaires. Ainsi, l'apport de ce cadre doit représenter environ 40% des efforts de développement d'une solution, sachant que les 60% restants sont de la responsabilité de l'utilisateur de ce cadre [IBM, 1998]. Ces développements complémentaires incluent les spécificités nationales et métiers d'une organisation, les interfaces et autres formes de valeur ajoutée. Ces spécificités permettent de présenter l'importance de la personnalisation d'un processus d'affaires dépendamment de son contexte. Le guide utilisateur fournit des modèles de processus sous-jacents à différents sous-ensembles du cadre. Ces modèles permettent de mieux comprendre les composants logiciels correspondants.



*Figure 3-12 : Architecture Générale de l'approche IBM San Fransisco.*

Le cadre San Fransisco est formé de trois couches (Figure 3-12) [Bohrer, 1998]. La couche « fondation » forme l'infrastructure de base et permet de construire les niveaux supérieurs. La couche objets d'affaires communs, (« common business objects » ou CBO) regroupe différents



objets de gestion et de services communs à plusieurs domaines applicatifs. La dernière couche celle des processus d'affaires de base (« Core Business Processes » ou CBP), fournit les structures et les comportements utiles à un domaine d'application donné.

Les auteurs du cadre prétendent que les modèles spécifiques à une industrie sont faciles à construire à partir des composants et des utilitaires fournis par l'infrastructure du cadre d'applications. Une étude de cas visant la création d'un processus de gestion de ressources humaines a permis de valider cette hypothèse [Jauf, 2000].

### **Méthodes/notations de représentation**

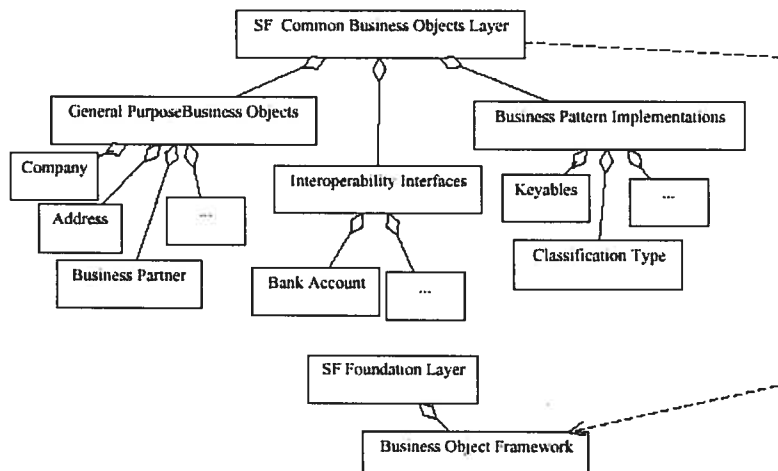
Ce cadre d'application est implémenté en Java. Toutefois, les fragments des modèles des processus identifiés avec les guides d'utilisation sont représentés avec le langage UML et plus spécifiquement, les diagrammes de classe et de collaboration.

### **Organisation**

Le cadre « IBM San Fransisco » fournit des modèles de processus à deux niveaux d'abstractions: les processus d'affaires de base « Core Business Processes » et les objets d'affaires communs « Common business objects »

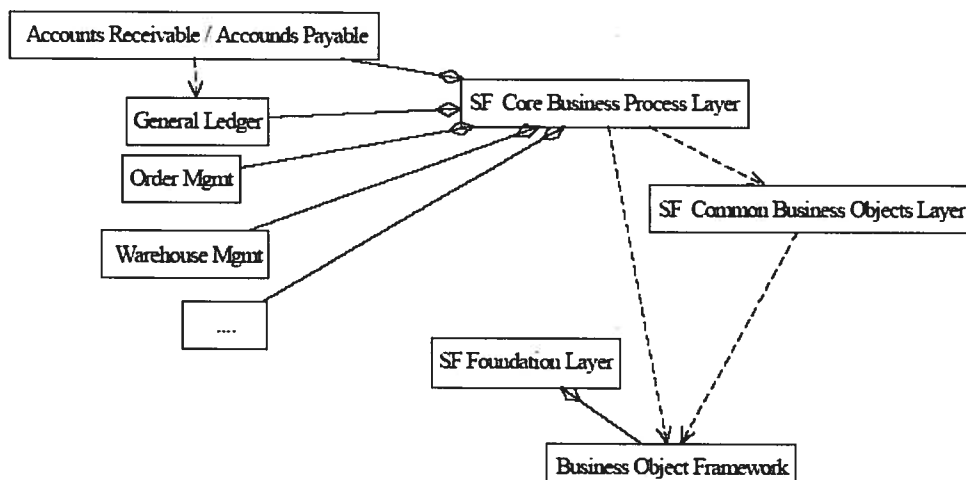
- 1) Les objets d'affaires communs (CBO) représentent des services applicatifs généraux. Ils sont groupés en trois catégories (Figure 3-13):
  1. Les objets d'affaires généralement utilisés dans plusieurs domaines d'application, par exemple l'adresse, le partenaire d'affaires, la monnaie, les méthodes de paiement, etc.
  2. Les objets d'affaires pour interfacier entre les composants d'affaires et assurer l'interopérabilité entre les applications. Par exemple les interfaces aux banques, et les interfaces aux comptes.
  3. Les objets qui présentent les mécanismes généraux de conception pour des applications d'affaires en se basant sur les patrons de conception fournis par IBM (voir [Carey, 2000]) (ex « Extensible Item » pour permettre l'ajout ou la suppression du comportement d'un objet durant l'exécution).

Les objets qui forment les CBO ne sont pas utilisés indépendamment comme des bibliothèques de classes, mais ils sont utilisés à travers le CBO en un seul bloc. De plus, certains CBO dépendent les uns des autres.



**Figure 3-13 : Classification des CBO (figure extraite de [Henn, 1998])**

2) Les processus d'affaires de base (CBP) touchent trois grands domaines : a) la finance (« General ledger, Accounts payable, Accounts receivable »), b) la gestion d'inventaire, et c) la gestion de la commande. Actuellement, ces modules portent les noms des composants d'affaires communs avec la terminologie IBM. Ces composants sont considérés comme des sous-modèles utilisés avec des applications d'affaires plus grandes, comme la gestion de la chaîne logistique [IBM, 2003]. De plus, il existe une dépendance entre les processus et les objets communs d'affaires [Bohrer, 1998] (Figure 3-14).



**Figure 3-14: Les composants de IBM San Francisco (figure extraite de [Henn, 1998])**

## Mécanismes d'adaptation

Le cadre IBM propose différentes façons d'adapter un composant existant, dépendant de l'appariement entre le besoin et le composant [Henn, 1998]:

- Le processus d'affaires de références fournit toute les fonctionnalités nécessaires pour l'application visée. Dans un tel cas, les modèles sont utilisés tel quels. Un modèle peut contenir une partie à instancier et une autre à configurer.
- Le processus d'affaires fournit la fonctionnalité, mais il y a une différence dans certains algorithmes de l'application: dans ce cas-ci, les points d'extension fournis avec les modèles sont identifiés et étendus. Les extensions peuvent être conformes aux patrons de conception fournis par IBM ou bien utilisent les approches d'extensions de classe présentées ci-après.
- Le processus d'affaires n'a pas la fonction équivalente à l'application visée. Dans ce cas, l'application est créée en utilisant les classes d'objets d'affaires communs (CBO) ou bien en créant de nouvelles classes à partir de la couche fondation.

Les approches utilisées pour établir une extension des classes fournies avec ce cadre sont les suivantes [IBM, 1998]:

- Sous-classification pour ajouter une nouvelle classe.
- Sous-classification pour remplacer une classe existante dans le but de changer le comportement (par exemple: Abstract factory pattern). À noter que IBM interdit le remplacement des CBO, puisqu'ils sont partagés par plusieurs processus.
- Extension d'une classe par agrégation d'autres classes.
- Extension d'une classe par utilisation de propriétés : si une classe a besoin d'informations supplémentaires (un attribut ou une relation), les attributs sont ajoutés en créant une sous-classe. Si la classe hérite de « PropertyContainer », elle peut alors être étendue sans modifier la classe ou créer une nouvelle sous-classe. Un «PropertyContainer » peut contenir un nombre arbitraire d'objets correspondant à des propriétés, chacune avec un nom qui lui est associé.

## Emballage

Les modèles fournis par ce cadre sont présentés sous forme de composants et d'objets d'affaires qui constituent une librairie de classes. Cette librairie peut être utilisée au niveau conception et implantation d'une application d'affaires. Il existe deux niveaux de granularité dans ce cadre les CBO qui forment une solution par partie comme les patrons et les CBP qui forment une solution complète pour des fonctionnalités d'affaires.

### **3.3.3.3 Modèles de référence de l'ERP SAP**

Un système ERP (Enterprise Resource Planning) est un système d'information « préfabriqué » intégrant les principales fonctions nécessaires à la gestion des flux et des procédures de l'entreprise [Davenport, 1998]. En pratique, un système ERP fournit des modèles de référence pour bien refléter les domaines d'applications où il est intégré. Dans cette section nous allons présenter les modèles de référence du système ERP de SAP (« systems, Applications, and Products in Data processing »).

#### **Objectifs**

Un système ERP doit supporter les processus d'affaires des entreprises. Le but des modèles de référence de SAP (« blueprints ») est de documenter les processus standards et de faciliter le développement en se basant sur les meilleures pratiques. Ainsi, la construction d'un modèle d'affaires ne part pas de zéro («from scratch») et ne nécessite pas de tout réinventer à chaque fois. Selon la procédure d'utilisation des modèles de SAP, il est suggéré que l'entreprise cliente compare sa modélisation avec les différents modèles proposés et qu'elle fasse coïncider sa structure avec les parties identiques des modèles génériques. De cette façon, elle visualise les endroits décisifs de sa compagnie.

Du point de vue technique, SAP est un cadre d'application pour les entreprises. Il implémente une architecture client/serveur à 3 niveaux: le niveau interface client (présentation), le niveau logique (serveur d'application) et le niveau physique (un serveur de base de données). Les nouvelles technologies proposées par SAP prennent en compte l'Internet et fournissent un quatrième niveau: la couche Web, qui rend possible les interactions inter-organisationnelles via Internet [Curran, 2000].

#### **Méthodes/notations pour la représentation**

Les modèles de SAP sont présentés par EPC (décrit dans le chapitre 2) et une documentation textuelle. Récemment, un cadre de modélisation des processus d'affaires qui se base sur EPC, ARIS («Architecture of Integrated information Systems» ) a été conseillé par SAP. Les outils de ARIS sont intégrés dans SAP [Scheer, 2000].

#### **Organisation**

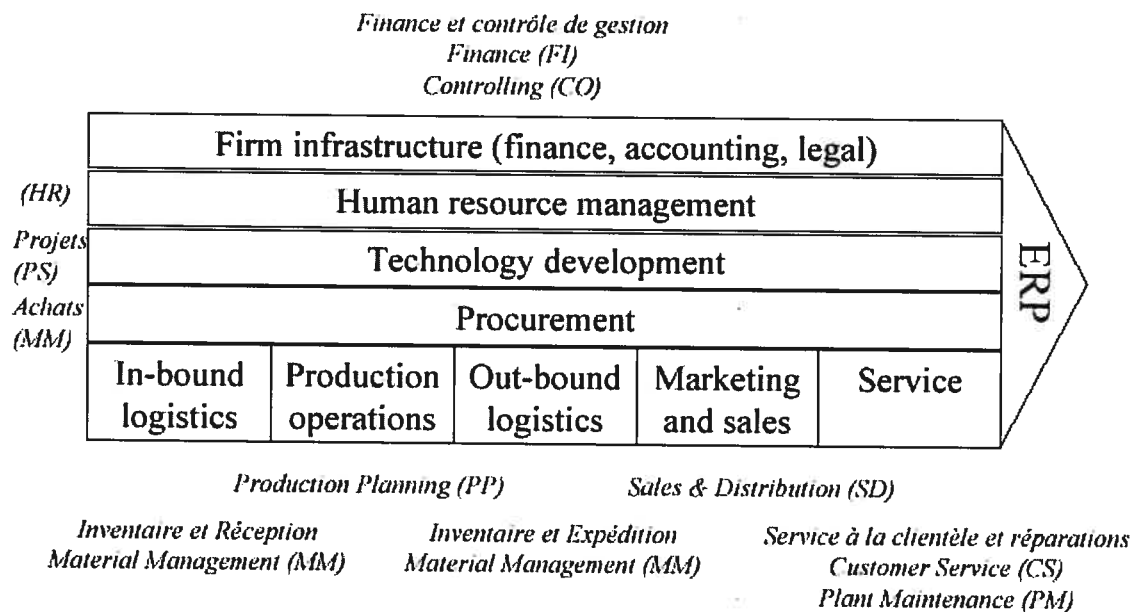
Les processus d'affaires peuvent être organisés en deux types: a) les processus sectoriels qui confèrent à une entreprise les utilisant une identité unique b) les processus génériques

(horizontaux) qui sont des processus identiques pour toutes les entreprises, mais qui changent de forme [Meir, 2003]. En effet, SAP a organisé ces modules selon ces deux types [SAP, 2003] :

- Les modules horizontaux nécessaires à la gestion d'une entreprise et qui couvrent les grands domaines fonctionnels : la finance, la logistique (achat, vente, production, planification) et la gestion. Pour chacun d'entre eux, SAP offre des fonctionnalités complètes. Les modèles de référence qui présentent ces modules proposent différents scénarios, dépendant du contexte des exigences de l'entreprise [Curran, 1998].
- Les modules sectoriels sont propres à un métier spécifique: par exemple la pharmaceutique, l'assurance ou les médias. Ces processus utilisent des fonctionnalités qui sont définies dans les modules horizontaux.

Il existe des modules pour gérer les affaires via le web. SAP fournit alors des interfaces qui permettent l'intégration des applications d'autres fournisseurs.

La classification des modules horizontaux se base sur la chaîne de valeur de « Porter » (Figure 3-15) [Curran,1998,2000].



**Figure 3-15: chaîne de valeur de Porter et les modèles de SAP**

Porter regroupe les activités (processus) en deux catégories: les processus primaires et les processus de support. Les activités primaires sont essentiellement de la création à la vente du produit, incluant son transport au client et le service après vente. Les activités de support viennent appuyer les activités primaires, en assurant l'approvisionnement, la gestion des ressources humaines, les

activités administratives et la recherche et le développement des technologies [Porter, 1985]. Dans la Figure 3-15, nous pouvons remarquer comment la classification de Porter est reprise par SAP pour classer leurs modèles (en italique).

Les modèles de référence comprennent trois agrégations, les scénarios, les processus et les transactions [Gulledge 2002] [Curran 1998, 2002]:

*Scénarios*: un scénario est une composition de fonctionnalités de haut niveau qui forment un processus d'affaires dans un contexte donné. Par exemple, pour le processus approvisionnement, nous pouvons avoir différents scénarios comme: approvisionnement par sous-traitance, approvisionnement par consignation, etc. A un niveau élevé les scénarios sont des chaînes de valeurs (end-to-end) pour présenter l'enchaînement d'un ensemble de processus.

*Processus*: le processus est une fonctionnalité qui décrit la séquence procédurale des transactions d'affaires. Un processus est illustré avec EPC.

*Transaction des processus d'affaires*: c'est la description de la tâche élémentaire d'un processus. Elle met l'accent sur les unités organisationnelles qui effectuent cette tâche et les objets/documents/données d'affaires nécessaires à cette tâche.

Il faut noter qu'avec les modèles de références repérés dans la littérature spécialisée, seulement quelques diagrammes EPC pour les scénarios de base de chaque catégorie ont été identifiés (regarder [Curran 2000, 1998]). Toutefois, une explication textuelle est fournie pour présenter les différents scénarios possibles. Pour accéder à tous ces diagrammes, il faut se référer au produit SAP qui est malheureusement à usage interne.

## **Mécanismes d'adaptation**

L'approche utilisée pour personnaliser un système SAP est descendante (« top-Down<sup>5</sup> »). Une fois que l'application (qui contient la logique d'affaires) visée est identifiée, les paramètres fournis par l'application seront instantiés selon les besoins spécifiques de l'organisation. SAP offre des modèles comme des solutions descriptives (pas normatives). Ces modèles peuvent être utilisés tels quels ou bien étendus par l'entreprise avec une gamme d'outils fournis par SAP. L'application acquise après l'instantiation crée un processus d'affaires qui remplace le processus d'affaires à automatiser. Il arrive souvent que le processus déployé ne corresponde pas exactement au processus

---

<sup>5</sup> SAP nomme cette approche «configure to order»

remplacé; par conséquent, un effort énorme est nécessaire pour contrôler le changement dans l'organisation.

Il faut noter que, les modèles de référence fournis mettent l'emphase sur la description d'un processus exécutable et sur la structuration des données, mais il ne montre pas les personnalisations alternatives existantes. En effet, un modèle de SAP montre les fonctionnalités entières du système en supposant que le système au complet est utilisé, alors qu'en réalité, un sous-ensemble est typiquement utilisé. Ainsi, les modèles de référence ne sont pas conçus pour la personnalisation, mais pour représenter les processus d'affaires au complet. Par conséquent, les effets de la personnalisation d'un processus ou d'une structure de données sur d'autres processus ne sont pas clairs avec ces modèles [Rosemann, 2003].

## **Emballage**

Les modèles de référence de SAP sont des processus d'affaires complets. Ces processus sont aussi sous-jacents aux divers outils de configuration d'un système ERP. Ils sont utiles pour la modélisation des exigences et des besoins d'un domaine d'affaires.

## **3.4 Initiatives orientées vers l'intégration des processus inter-entreprises**

Le commerce électronique inter-entreprises permet à des entreprises de faire des affaires par le biais d'une collaboration entre leurs systèmes d'information et d'internet. Cet avènement a poussé un certain nombre d'organismes à standardiser ces collaborations et les processus d'affaires qui les gèrent. Dans cette section nous allons décrire de tels standards: RosettaNet et EbXML. A noter que ces standards ont déjà été présentés au chapitre 2.

### **3.4.1 L'initiative RosettaNet**

#### **Objectifs**

Rappelons que RosettaNet est une initiative de standardisation des documents d'affaires et des processus d'interactions entre partenaires dans le contexte du commerce électronique inter-entreprises (voir section 2.6.1). Ce standard édicté s'intéresse à simplifier la mise en place et la

gestion des processus d'affaires publics pour un ensemble de domaine d'application. Pour ce faire, RosettaNet fournit un catalogue des modèles génériques de processus sous forme de PIP (Partner Interface Processes). Un PIP présente la partie des processus d'affaires des partenaires qui implique une interaction entre eux

## Méthodes/notations pour la représentation

Les langages utilisés avec ce standard sont XML et UML (voir chapitre 2).

## Organisation

Les PIP sont catégorisés par grande fonction commerciale (les clusters) puis par des processus d'affaires impliquant plusieurs partenaires commerciaux (les segments). À l'intérieur d'un PIP, on trouve une nouvelle classification en un certain nombre « d'activités », elles-mêmes subdivisées en « actions » [Chauvet, 2002].

*Exemple:*

Cluster 3: Order Management

- Segment A: Quote and Order Entry
  - PIP 3A4: Manage Purchase Order
    - Activity: Create Purchase Order
      - Action: Purchase Order Request
      - Action: Purchase Order Acceptance
- Segment B: Transportation and Distribution
- Segment C: Returns and Finance

Segment D: Product Configuration

Les catégories de processus couvrent l'administration, la recherche de partenaires, la recherche d'informations de produits, la gestion de commandes, la gestion d'inventaire, la production, la gestion d'informations marketing et le service après-vente.

Chaque PIP est identifié par sa catégorie, son segment et son numéro d'ordre dans le segment considéré. Les PIPs ne fournissent pas toutes les étapes des processus de chacun des partenaires. Toutefois, certains PIPs suggèrent l'utilisation d'autres PIPs complémentaires dans certains dialogues, sans oublier leurs pré-conditions d'utilisation. Un PIP contient la spécification des rôles des partenaires, des activités impliquées entre ces rôles, du contenu, du type et de la séquence des messages échangés durant l'interaction entre ces rôles. Les messages échangés entre deux partenaires sont de deux types, des actions d'affaires (« business action ») et des signaux d'affaires (« business signal »). Les premiers types sont des messages avec un contenu (ex : ordre d'achat), tandis que ceux du second sont des messages de réception (positifs ou négatifs). À noter que ces



derniers sont spécifiés et qu'ils font parties du cadre d'implémentation RNIF (« RosettaNet Implementation Framework »). Rappelons que ce dernier définit les choix techniques pour le format et l'échange des messages RosettaNet.

### **Mécanismes d'adaptation**

Ses spécifications doivent être utilisées telles quelles. En effet, pour que les processus publics soient interopérables dans la communauté RosettaNet, le format des informations et les séquences des échanges doivent être conformes aux spécifications RosettaNet. Si une entreprise utilisant RosettaNet désire effectuer un changement, elle doit soumettre une demande à l'organisation RosettaNet [RIG, 2003]. Par la suite, les organisations doivent implémenter de nouveaux processus publics ou modifier les processus existants pour supporter les processus publics fournis par RosettaNet.

### **Emballage**

Les modèles de ce standard sont des modèles d'analyses. Ces modèles décrivent des collaborations binaires entre deux partenaires. Ils sont présentés comme des composants d'affaires publics réutilisables, sans oublier que chaque composant contient des DTD et des directives nécessaires pour les formats des messages échangés entre les partenaires.

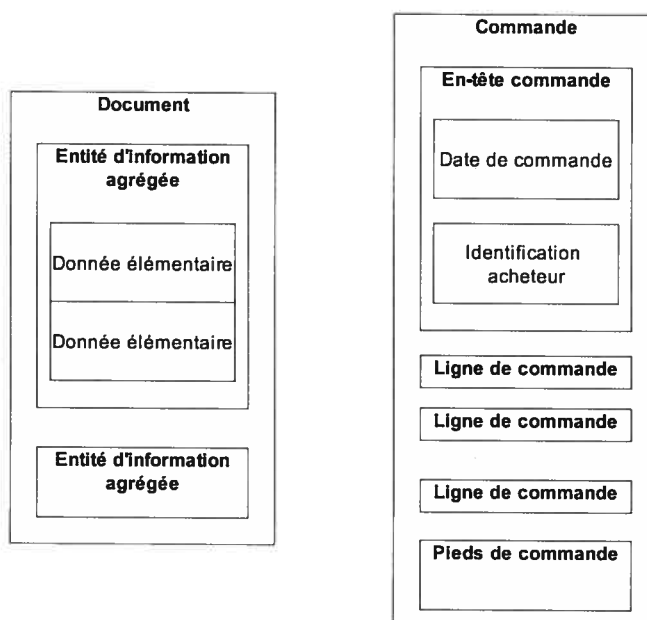
## **3.4.2 Initiative EBXML**

### **Objectifs**

Rappelons que ebXML est une technologie de commerce électronique basée sur les échanges de messages XML. Le standard ebXML couvre trois aspects du commerce électronique (voir chapitre 2): 1) le contenu des documents d'affaires échangés, 2) le format des documents et 3) un registre dans lequel les entreprises peuvent publier ou rechercher des services. De plus, une des préoccupations de la technologie ebXML est de fournir une librairie qui contient un ensemble de processus d'affaires de base et de composants des documents qui couvrent la plupart des scénarios d'affaires [ebXML, 2001d]. L'objectif de cette librairie est d'inciter l'interopérabilité en standardisant non seulement la notation pour représenter un processus, mais les processus eux-même. Pour les processus d'affaires communs, ebXML propose d'avoir un catalogue des processus

qui sont indépendants d'une industrie spécifique [ebXML, 2001d]. Ces processus sont regroupés selon une classification que nous présenterons dans la prochaine section. De plus, un des objectifs de ce catalogue est de fournir une correspondance entre les processus et les documents d'affaires.

Les documents d'affaires ebXML sont définis en utilisant une spécification externe de document ou en assemblant selon des règles précises des structures d'information de plus bas niveau appelées composants communs [ebXML, 2001a]. Ces composants communs sont eux aussi présentés dans des bibliothèques. Par exemple (Figure 3-16): Le document commande ne constitue pas un composant commun réutilisable, mais il est assemblé à partir de composants communs.



*Figure 3-16 : Document commande (ebXML)*

Il faut noter que les travaux sur ebXML en général et sur les bibliothèques en particulier sont toujours en cours de standardisation. Alors nous pouvons identifier, d'après les spécifications qui sont fournies, la méthode d'organisation et le mécanisme d'adaptation prévus. Cependant nous n'avons pas vu d'exemples concrets pour pouvoir illustrer leur méthodologie.

### **Méthodes/notations pour la représentation**

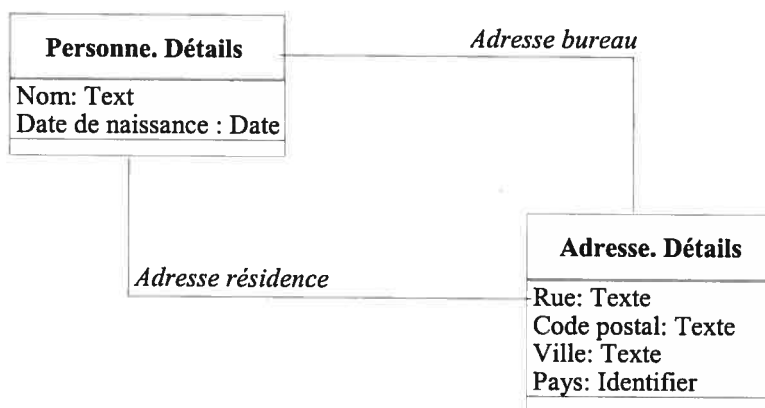
Les modèles de processus d'affaires sont formellement décrits avec le «Business Process Specification Schema» ou BPSS (un schéma XML et une DTD) (présenté à la section 2.6.2). EbXML recommande une méthodologie UMM (UN/CEFACT Modeling Methodology) et fournit des fiches descriptives pour créer les modèles de processus d'affaires qui vont être décrits par la suite avec BPSS [ebXML, 2001b]. Ces fiches peuvent être utilisées pour modéliser toutes sortes de processus d'affaires. Toutefois, la concentration de l'effort d'ebXML est sur l'intégration des

systèmes des partenaires. La méthodologie UMM utilise les diagrammes d'UML, mais avec des caractéristiques spécifiques. Cette méthodologie est en cours de standardisation avec le groupe UN/CEFACT<sup>6</sup>.

## Organisation

Comme nous l'avons déjà mentionné, ebXML utilise deux types de catalogues dans son cadre sémantique: Les composants communs et les processus d'affaires.

1. **Les composants communs d'affaires («core components»)** [ebXML, 2001a]: Ce catalogue contient des structures de données élémentaires (téléphone, adresse, date, etc...) que les activités d'affaires utilisent le plus fréquemment. Elles sont définies de manière à ce qu'elles soient réutilisables en leur attribuant des noms neutres et des identifiants uniques et ceci en respectant une convention de nomination. Il existe quatre catégories de composants communs : Les Composants Communs Élémentaires, les Composants Communs Associatifs, les Composants Communs Types et les Composants Communs Agrégés.



**Figure 3-17 : Exemple d'un composant commun d'EbXML**

Prenons l'exemple de la Figure 3-17, nous avons deux composants communs agrégés, « Personne.Détails » et « Adresse.Détails ». Chaque composant a un certain nombre de propriétés (des caractéristiques d'affaires). La plupart de ces propriétés sont elles mêmes des composants communs élémentaires. Ces derniers représentent une caractéristique d'affaires particulière et leurs valeurs autorisées sont définies par un type de donnée. Les propriétés « adresse résidence » et « adresse bureau » sont deux Composants Communs Associatifs et leurs structures sont décrites par « Adresse.Détails ».

<sup>6</sup> [http://webster.disa.org/cefact-groups/tmg/doc\\_bpwg.html](http://webster.disa.org/cefact-groups/tmg/doc_bpwg.html) lien révisé le 2 octobre 2003.

Ces composants se transforment en des entités d'information d'affaires (Business Information Entities) par migration avec les contextes de domaines d'affaires. Le résultat sera un ensemble de composants spécifiques au contexte (ou composants transversaux) qui vont être utilisés avec l'assemblage d'un document d'affaire. Si nous reprenons l'exemple précédent, la combinaison du contexte avec les composants communs aboutit à la constitution d'un ensemble d'entités d'informations comme :

US\_Personne.Details (Aggregate Business Information Entity)

US\_Personne.US\_Residence.US\_Adresse (Association Business Information Entity)

US\_Personne.Nom.Text (Basic Business Information Entity).

2. **Les processus d'affaires communs [ebXML, 2001d]**: La classification de ce catalogue se base sur la chaîne de valeur de Porter. Chaque processus commun du catalogue appartient généralement à une certaine catégorie d'activité normative de cette chaîne. Les catégories normatives sont décomposées en sous-catégories qui représentent des fonctionnalités génériques qui englobent des collaborations. Le catalogue inclut des références croisées avec d'autres standards d'échanges électroniques dont RosettaNet, X12, EDIFACT, et d'autres (Figure 3-18).

Common Processes	Business Normative Category	Normative Category	Sub-EDIFACT including sub-sets	X12 including sub-sets	xCBL 3.0	RosettaNet Partner Interface Process
Request Quote Response	Procurement Management	Pre-Order and Quote	REQOTE, QUOTES	840, 843	Quote, RequestForQuotation	PIP3A1
Query Price And Availability	Procurement Management	Pre-Order and Quote	PROINQ, AVLREQ, TIQREQ	879		PIP3A2

**Figure 3-18 : Échantillon du catalogue de processus d'affaires commun d'ebxml [ebXML, 2001d]**

D'autres méthodes pour cataloguer les processus d'affaires sont présentées ; notons le catalogue qui utilise la méthodologie REA (Ressource Event Agent) et la vue transactionnelle.

EbXML propose également une structure d'annuaires répartis. Ces annuaires recouvrent une partie des fonctionnalités de la spécification UDDI, des services d'annuaire (Registry Services) et un modèle d'information associé (Registry Information Model), similaires à UDDI. Pareillement à UDDI, les annuaires ebXML enregistrent les processus métier et les composants à travers une API

de publication. Ils offrent également une API de requête permettant à une application de récupérer ces différentes informations. Ces annuaires sont physiquement répartis et des mécanismes de réplication et de propagation sont prévus pour maintenir l'intégrité de la vue logique [Chauvet 2002].

## **Mécanismes d'adaptation**

Nous avons deux aspects à considérer: l'utilisation des composants communs et leurs dérivés pour créer des documents commerciaux, et l'utilisation des processus d'affaires génériques.

Un composant commun est sélectionné et spécialisé dépendamment de la sémantique du domaine d'affaires où il est appliqué. Cette sémantique est représentée par un ensemble de catégories contextuelles dont la localisation géopolitique, le secteur d'activité, etc. Des règles de contextes représentés sous formes d'une DTD XML sont utilisées, pour spécialiser un composant. Une règle indique quelles actions sont effectuées sur le composant si un appariement est apparu avec une valeur donnée. Les actions peuvent être d'ajouter, de demander ou d'éliminer de l'information. Par exemple, dans un contexte d'affaires spécifique où le contexte localisation géopolitique prend la valeur EU (États Unis), le composant commun « Adresse », est spécialisé en y ajoutant des informations concernant les États. L'entité d'information résultante de cette spécialisation sera enregistrée sous un nom unique formé du nom du contexte et du nom du composant stéréotypé dans la librairie propre aux composants, pour une utilisation ultérieure [ebXML, 2002a].

Avec les processus d'affaires, une spécialisation s'applique sur les spécifications qui sont faiblement couplées aux contraintes technologiques et aux détails de relations d'affaires (exemple: les paramètres chronologiques) [ebXML, 2001b]. Pour effectuer cette spécialisation, un élément nommé groupe de substitution « SubstitutionSet » est placé dans la spécification du processus le plus générique afin de le rendre plus spécifique en redéfinissant selon les besoins les documents échangés et les valeurs des attributs contenus dans le processus d'affaires. Un autre élément « Include » se présente dans la spécification et permet d'inclure d'autres instances de processus d'affaires dans celle que l'on est en train de décrire [ebXML, 2001b].

## **Emballage**

Les modèles fournis avec ebXML permettent d'assurer l'implémentation d'un processus d'affaires public. Nous avons les composants communs qui sont des éléments de données individualisés pour la création des documents d'affaires. Nous avons aussi des spécifications de processus d'affaires

communs (exprimés en UML et XML) employés par différentes industries qui peuvent être utilisées pour élaborer des processus personnalisés en fonction d'un contexte d'affaires spécifique.

### 3.5 Discussion

L'objectif de notre recherche est de réutiliser des composants d'affaires génériques pour construire des applications d'affaires. Pour ce faire, nous devons établir une association entre les processus et les composants d'affaires génériques. Idéalement, nous devons trouver une façon pour organiser les modèles des composants génériques en se basant sur les processus d'affaires. De même, nous devons trouver une méthode pour spécialiser ces composants. L'étude dans les sections précédentes a révélé que toutes les initiatives en général encouragent la réutilisation; cependant, leurs portées et leurs utilités diffèrent.

En effet, quelle que soit la granularité des modèles des composants réutilisables, nous sommes souvent amenés à en sélectionner puis à en utiliser plusieurs lors du développement des applications. Certains travaux présentés mettent l'accent sur des modèles de processus d'affaires partiels parmi lesquels les patrons d'analyses et les patrons d'affaires. Ces patrons sont des micro-architectures qui doivent être utilisées dans un contexte plus large et leurs classifications et leurs catégorisations ne sont pas claires. Avec les cadres d'application, des modèles de processus plus complets sont traités et une catégorisation plus claire est utilisée.

Certains cadres d'application fournissent le processus et les produits de développement nécessaires pour représenter, concevoir, implémenter et déployer un concept d'affaires dans un système logiciel. En d'autres termes, ces cadres fournissent les composants logiciels nécessaires pour supporter les processus d'affaires. Cependant, la granularité des ontologies des cadres d'application est très variable. Ces ontologies dépendent de la complexité du processus représenté. Elles sont parfois spécifiques à des conceptions architecturales (IBM San Francisco).

Pour les différents modèles de processus fournis par les cadres d'application, en particulier les modèles de références du SAP et les composants de Coad et Lefebvre, nous remarquons que malgré la représentation à un niveau générique du contenu des processus, leurs variabilités ne sont pas explicites. Il faut alors pouvoir identifier et comprendre les différentes variétés et les options fournies par ces modèles pour pouvoir les personnaliser et les utiliser selon nos besoins, ce qui n'est pas évident.

D'autres part, la classification des processus d'affaires dans la plupart des initiatives se basait sur le modèle de référence de Porter [Porter, 1985]. Avec le manuel MIT, la classification est différente. Elle se base sur la hiérarchisation et facilite la recherche et la sélection des processus. Malheureusement, le manuel MIT ne représente pas le processus d'affaires du point de vue structurel, tout comme il ne contient pas de modèles décrits avec une notation.

D'autres initiatives dont ebXML et RosettaNet ont essayé d'établir une correspondance entre les processus et les composants génériques, dans le cadre des relations inter-entreprises. La spécificité d'ebXML se résume par l'amélioration de l'interopérabilité sémantique grâce à la conception de composants communs et au mécanisme contextuel. Ces composants permettent également la réutilisation, mais l'objectif principal est toujours l'interopérabilité. Malgré ces efforts, ebXML manque toujours d'un modèle formel sur lequel la représentation sémantique se fonde. RosettaNet n'occasionne pas ce type de problème étant donné que le contexte est lié au composant. Cependant, ces composants sont des interfaces et n'impliquent pas tous les processus d'affaires, comme ils ne permettent pas d'aboutir à un niveau suffisant d'interopérabilité ou de souplesse.

Finalement, ces initiatives de catalogage de composants d'affaires et logiciels réutilisables que nous avons étudiées se servent de fragments de modèles de processus principalement pour documenter les composants, mais n'établissent *pas* de liens entre processus et composants logiciels de façon *systématique*.

### **3.6 Conclusion**

Dans ce chapitre, nous avons passé en revue certains des travaux qui constituent des cheminements assez différents permettant la modélisation et la réutilisation des processus d'affaires. Nous avons classifié les différentes initiatives selon leurs utilités d'application. Cependant, selon notre objectif, nous constatons qu'aucune de ces initiatives n'est complète. Toutefois, il existe des caractéristiques tirées des différentes initiatives que nous allons élaborer dans notre approche au chapitre suivant notons surtout la méthodologie de hiérarchisation du manuel MIT.

# **Chapitre 4. Approche proposée**

## **4.1 Introduction**

Les processus d'affaires sont à la base des systèmes d'informations d'une industrie donnée. Certains processus d'affaires sont réutilisables dans différentes industries. Pour une bonne réutilisation de ces modèles, il faut trouver une méthode pour les classifier et les spécialiser dépendamment des domaines d'affaires. Dans les chapitres précédents, nous avons passé en revue différentes initiatives de modélisation et leurs méthodologies de classification et de spécialisation. Nous avons établi que la classification proposée par le manuel MIT pour présenter les processus d'affaires et leurs composants est la plus adéquate. Rappelons que l'une des techniques de recherche utilisées par ce manuel se base sur une stratégie par question. Cette technique est facile à assimiler par des utilisateurs ayant des profils différents. Cependant, les questions sont très spécifiques à chaque processus et l'approche ne peut être utilisée de façon systématique sur un reposoir de processus génériques. Nous proposons donc de généraliser cette approche. Pour pouvoir peupler notre reposoir, nous utiliserons les différents modèles génériques fournis dans la littérature avec les différentes initiatives étudiées (chapitre précédent). La représentation de processus générique mettra l'accent sur le concept de « variabilité » et elle utilisera une notation étendue d'UML.

Dans ce chapitre, nous allons commencer dans la première section par une explication sur notre méthodologie pour adapter des processus génériques. Dans la deuxième section, nous présenterons la classification descriptive des processus génériques. Dans la troisième section, nous expliquerons comment représenter un processus générique. La quatrième section traitera la procédure suggérée pour spécialiser les processus génériques. Un exemple sera présenté à la cinquième section. Nous concluons dans la sixième section.

## **4.2 Une méthodologie d'adaptation de processus génériques**

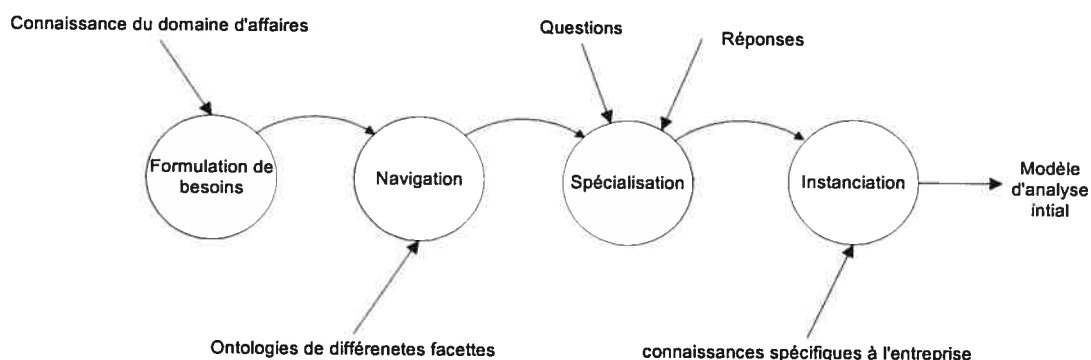
En général, pour peupler notre reposoir avec de modèles génériques, nous devons procéder par une approche ascendante (« bottom up »), qui pousse à généraliser des modèles de processus d'affaires à partir des modèles spécifiques, pour pouvoir les rendre abstraits et génériques. En effet, la plupart



des initiatives qui ont présenté de modèles génériques (chapitre 3), ont procédé de cette façon, et ceci est prouvé par les différentes alternatives et les fonctionnalités optionnelles que nous trouvons dans les différents modèles génériques de différentes initiatives. Ceci met en évidence le fait que l'origine de ces modèles est en effet spécifique aux domaines. Dans notre cas, la meilleure façon pour identifier différents modèles et pour les représenter d'une façon générique est d'examiner les différentes initiatives énumérées au chapitre 3.

Une fois que le reposoir est peuplé par les modèles, l'approche pour utiliser ces modèles est descendante (« top down »). Cette dernière permet de passer du modèle générique à d'autres plus spécifiques. Le fonctionnement de notre approche à ce stade est formé de quatre étapes (Figure 4-1):

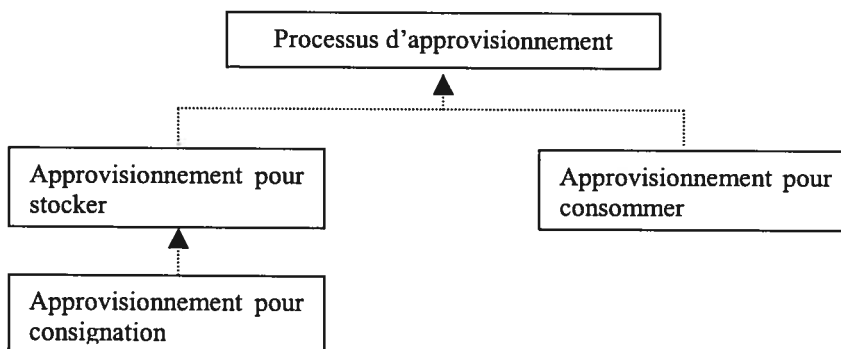
- a) Formulation des besoins pour préciser l'objectif du processus à modéliser.
- b) Navigation dans un reposoir des modèles génériques pour se placer sur le processus d'affaires le plus proche, en se basant sur une classification descriptive.
- c) Spécialisation du modèle générique en utilisant une technique par question.
- d) Instanciation du modèle spécialisé dépendamment des connaissances spécifiques à une entreprise pour obtenir un modèle d'analyse.



**Figure 4-1 : Les étapes de l'approche**

Pour pouvoir formuler nos besoins nous devons cerner les besoins fonctionnels de notre domaine d'affaires. Pour ce faire, une description informelle du processus d'affaires est nécessaire. Cette description permet de spécifier implicitement les objectifs de notre application. À partir de cette description le concepteur n'a pas toujours d'idées précises sur le processus générique qu'il cherche. Pour résoudre ce problème nous avons besoin d'une classification descriptive du reposoir, pour faciliter la navigation (section 4.3). Cette classification va permettre de retrouver les modèles dépendamment de nos objectifs. Après avoir identifié le modèle générique dans le reposoir, nous

devons déterminer s'il existe des processus génériques variantes, qui en sont des spécialisations alternatives. Ces spécialisations peuvent être adéquates à nos besoins ou non. Exemple :



La recherche à ce niveau peut être effectuée avec la spécialisation du modèle générique selon nos besoins. La spécialisation du modèle du processus d'affaires générique se fait par rapport à certaines dimensions de variabilité que nous allons présenter dans la section 4.4. Une fois le processus générique spécialisé, nous devons l'adapter (instancier) à une entreprise spécifique pour avoir un modèle d'analyse.

Dans les sections qui suivent nous allons essayer de répondre aux questions: a) Quels types de classification descriptive utiliser ? b) Comment représenter les processus ? c) Comment spécialiser le processus ?. Pour ce qui est de l'instanciation, nous supposons que cette étape peut être effectuée manuellement par substitution ou par ajout d'autres liens ou concepts. Une étude plus approfondie fera l'objet de recherches futures.

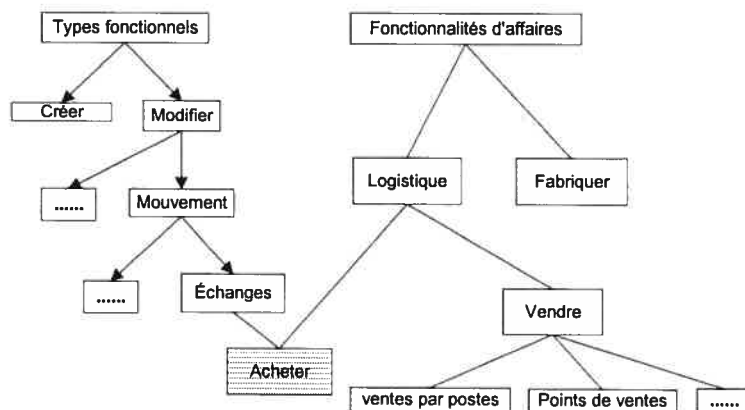
### 4.3 Classification descriptive

Cette classification facilite la recherche des modèles génériques car elle fournit des facettes ontologiques claires qui permettent de localiser facilement les modèles. En effet, en partant d'une classification qui se base sur des attributs simples appartenant à différentes ontologies reconnues dans les domaines d'affaires, il devient simple de se placer sur les modèles des processus les plus proches de ce que nous voulons modéliser. Nous avons vu dans le chapitre précédent que la classification de Porter [Porter, 1985] a été adoptée par plusieurs initiatives comme l'approche de Coad et Lefebvre [Coad et al., 1999], les modèles de références de SAP [Curran, 1998] et ebXML [ebXML, 2001d]. Cette classification reste limitée aux modèles génériques des processus d'une chaîne de valeur. Elle permet néanmoins la distinction entre deux types de processus génériques : primaires et de support.. Avec le manuel MIT, en plus de la classification de Porter, une autre

classification plus générique et vaste est présentée. Cette classification regroupe les processus génériques dans des catégories qui ont des parents communs. Par exemple, le processus d'affaires « Ventes » appartient à une catégorie plus générique «Échanges» et ce dernier appartient à la catégorie « Mouvement » et ainsi de suite. « Agir » est la catégorie la plus générale : toutes les activités dans le manuel en sont une spécialisation directe ou indirecte. De plus, le manuel MIT permet d'avoir différentes variantes d'un même processus. Pour notre reposoir, nous allons nous baser sur les différentes classifications identifiées dans le chapitre précédent pour représenter une classification qui traite les facettes suivantes:

1. La fonctionnalité d'affaires: cette facette décrit la fonctionnalité du processus à un niveau d'abstraction élevé. Elle présente l'objectif de ce processus. Quatre fonctionnalités sont identifiées: la production, la logistique, le support et la planification. Toutefois, il est intéressant de faire des distinctions entre les processus de support et les processus primaires [Lefebvre, 1996] [Porter, 1985].
2. Le type fonctionnel: cette facette présente les catégories génériques auxquelles les processus appartiennent (créer, modifier, détruire, etc.) [MIT, 2003].
3. Le type de processus: on a deux types: les types sectoriels sont spécifiques à un domaine d'affaires (exemple: dans le domaine pharmaceutique, on a des processus de tests propres aux cliniques ou des processus de recherche pharmacologique), et les types génériques qui sont identiques dans toutes les entreprises, mais qui changent de forme (exemple : processus d'approvisionnement) [Meir, 2003] [SAP, 2003].
4. La dépendance: cette facette (voir IBM San Fransisco [IBM, 1998]) identifie des composants et des processus d'affaires qui sont parfois dépendants les uns des autres (par exemple, la gestion de commandes et la gestion d'inventaire). Ainsi, un processus particulier X dépend d'un autre Y, si X utilise un ou plusieurs modules de Y. D'autres part, les concepts de dépendance de MIT poussent à construire un lien entre les processus qui s'intègrent pour aboutir à un objectif qui dépasse les frontières de chaque processus. Par exemple, avec la chaîne de valeur, chaque processus complète le précédent pour pouvoir servir un client.

La Figure 4-2 est un petit exemple pour montrer comment fonctionne la navigation avec deux facettes.



<i>Processus</i>	<i>Fonctionnalité</i>	<i>Type fonctionnel</i>	<i>Type processus</i>	<i>Dépendance</i>
Acheter	Logistique	Échanges	Génériques	Gestion d'inventaire, Gestion de finance,

**Figure 4-2 : Exemple de navigation**

Ajoutons le fait que nous allons adopter dans notre reposoir l'approche de hiérarchisation du manuel MIT, qui se base sur la décomposition et la spécialisation, puisqu'elle permet de présenter le processus à plusieurs niveaux. Cette hiérarchisation a deux avantages [Johannesson, 2000]: a) elle permet une représentation concise puisqu'un nouveau processus peut hériter une grande partie de sa description d'un processus de niveau plus abstrait; uniquement les changements seront introduits explicitement; b) la navigation peut être faite vers le haut pour identifier des versions plus générales d'un processus, ou bien vers le bas pour trouver des versions plus spécifiques des processus.

Rappelons que cette classification est descriptive et non structurelle. Elle peut servir comme étape initiale pour trouver le bon processus générique dans lequel seront utilisées des dimensions structurelles pour le spécialiser.

#### **4.4 Représentation des processus génériques**

Nous allons utiliser l'exemple d'un processus d'approvisionnement pour décrire notre approche tout au long de ce chapitre. Une description informelle est fournie ici, mais dans les sections qui suivent, différentes vues de ce processus vont être présentées avec plus de détails. Pour une description

complète, il faut se référer à l'annexe 1. D'une façon informelle, un processus d'approvisionnement est une fonction qui consiste à fournir en temps voulu à l'entreprise, les ressources et les services nécessaires auprès de sources d'approvisionnement. Toutefois, ce processus dépend des types de ressources, d'agents et d'événements (traitements) visés. Différentes variantes peuvent exister. Ces variantes vont dépendre de l'objectif visé par le processus. Par exemple, les variantes identifiées surtout par le progiciel SAP sont: approvisionnement pour stocker, pour consommer, par consignation, par sous-traitance, par transfert, pour un client, etc.

#### **4.4.1 Critères de représentation du modèle générique**

Pour avoir des modèles de processus personnalisables et qui supportent la spécialisation, il faut prendre en considération le concept de « variabilité <sup>7</sup> » souvent utilisé dans le domaine du développement des lignes de produits (« product family ») (ex: [Halmans 2003]). La variabilité se traduit au niveau des modèles génériques par des « points de variation ». Massen a identifié trois types de variabilités au niveau des caractéristiques d'un système [Massen 2002]:

- *Des caractéristiques optionnelles:* Ces caractéristiques peuvent être choisies ou non.
- *Des caractéristiques alternatives:* Une seule caractéristique doit être choisie parmi plusieurs.
- *Alternatives optionnelles:* une combinaison entre des caractéristiques optionnelles et alternatives doit être considérée. En d'autres termes, nous pouvons avoir des alternatives à un point de variation, mais nous pouvons décider de ne choisir aucun d'eux.

Ajoutons le fait qu'il peut y avoir des dépendances entre les points de variations. En effet, les variabilités ne sont pas indépendantes les une des autres, une variabilité A peut exiger d'autres variabilités, ou une variabilité C est visible si, et seulement si, des caractéristiques précises sont choisies de la variabilité B.

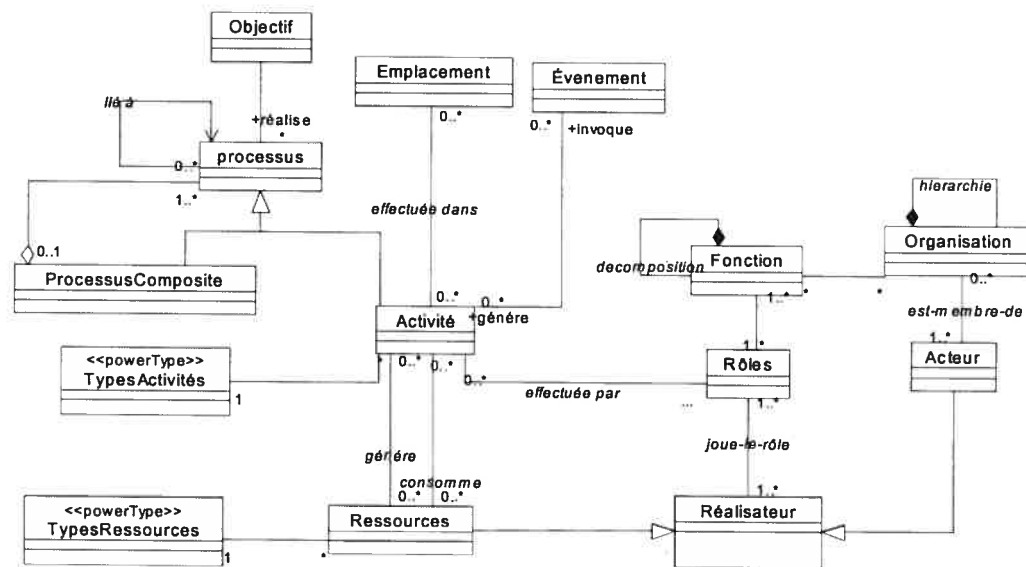
#### **4.4.2 Méta-modèle élaboré du processus d'affaires**

Une première vue du méta-modèle tiré de différentes définitions d'un processus d'affaires a été présentée au chapitre 2. Ce méta-modèle permet de comprendre les concepts de base qui constituent un processus ainsi que les liens entre eux. Ces concepts ont été traités et élaborés par différents langages de modélisation dépendamment des besoins et des objectifs, comme nous en avons déjà

---

<sup>7</sup> La *variabilité* se définit comme le fait qu'un système s'adapte, se spécialise, se configure en fonction du contexte [Van Gorp , 2000].

discuté dans le chapitre 2. Cependant, pouvoir construire un méta-modèle qui satisfait toutes les exigences potentielles des domaines d'applications est difficile. Néanmoins, nous avons essayé de présenter un méta-modèle plus détaillé (Figure 4-3) à partir de l'analyse de différents langages et des différentes ontologies vues au chapitre 2.



**Figure 4-3 : Méta-modèle élaboré d'un processus d'affaires.**

Ce méta-modèle ne présente pas les concepts qui illustrent les aspects exécutables d'un processus d'affaires, comme les liens de dépendance et de contrôle et la gestion des exceptions et des compensations transactionnelles. Certes, ces types de concepts sont nécessaires pour la modélisation d'un processus d'affaires, mais avec notre approche, nous allons nous contenter de représenter les processus à un niveau qui nous permet d'avoir des modèles d'analyse génériques.

Compte tenu de la variété de concepts et de leurs relations sémantiques, il convient de structurer davantage les modèles de processus et de regrouper ces concepts en dimensions (vues). Rappelons les différentes vues d'un processus d'affaires présentées par Curtis. Ces différentes vues, représentées ci-dessous, sont reliées pour caractériser et représenter un modèle de processus multidimensionnel (Figure 4-4). Il faut noter que la vue organisationnelle représente l'environnement du processus d'affaires. En effet, elle est liée aux différentes vues car elle vise à préciser « qui fait quoi et pourquoi ».

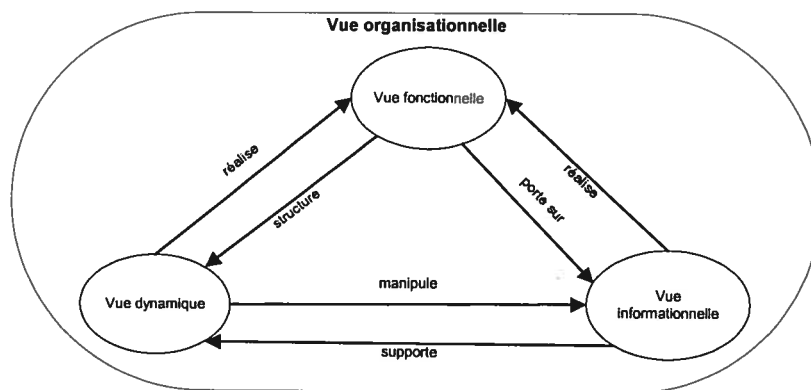


Figure 4-4: Modélisation d'un processus réutilisable à l'aide de quatre vues

### 4.4.3 Vues de représentation du modèle générique.

#### 4.4.3.1 Vue fonctionnelle

Cette vue présente la dépendance fonctionnelle entre les éléments du processus (activités, sous-processus). Avec cette vue notre préoccupation sera de connaître les objectifs du processus à traiter et sa structure fonctionnelle.

#### Concepts de modélisation

Comme nous voyons dans le méta-modèle (Figure 4-3), un processus X peut être composé d'autres processus (nommés sous-processus) ou d'activités. Un sous-processus peut former un module (composant) réutilisable par plusieurs processus. Une activité est une fonction atomique (élémentaire), effectuée par un seul rôle à un moment donné, et qui n'est plus décomposable. Les notions de fonction ou d'activité peuvent être utilisées comme des synonymes. Nous considérons qu'un processus est de haut niveau s'il ne fait partie d'aucun autre processus. Dans cette vue, le but primordial est d'identifier les processus de haut niveau et les sous-processus constituants. Les liens entre les sous-processus et les activités peuvent être conditionnés ou obligatoires. Conditionnés si la présence d'un sous-processus (resp. activité) peut exiger la présence d'autres sous-processus (resp. activités). Obligatoires si le sous-processus (resp. activité) engendre la présence d'autres processus (resp. activités).

#### Notation

Pour représenter cette vue, nous allons utiliser la notation des diagrammes d'activité d'UML2.0 de haut niveau. Pour représenter les variabilités nous allons utiliser les « stéréotypes » :

« Optionnel » pour représenter la fonctionnalité qui n'est pas obligatoire. Ce stéréotype est utilisé avec l'activité.

« Xou » utilisé avec le connecteur de décision pour représenter les fonctionnalités alternatives. Ce type de connecteur est associé à des notes qui contiennent les règles d'affaires dans le cas où le choix du cheminement du flux est lié à la présence d'autres processus [Roseman, 2003].

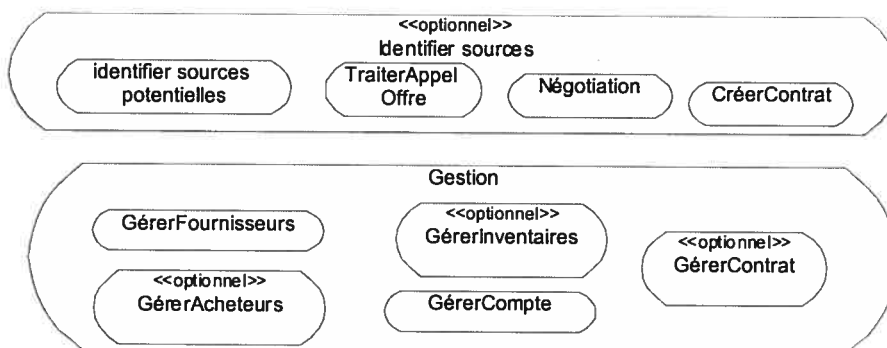
« Extension » pour représenter une variante qui peut dériver d'une activité.

« Inclue » une activité variante qui n'existe que si elle vient d'une autre activité.

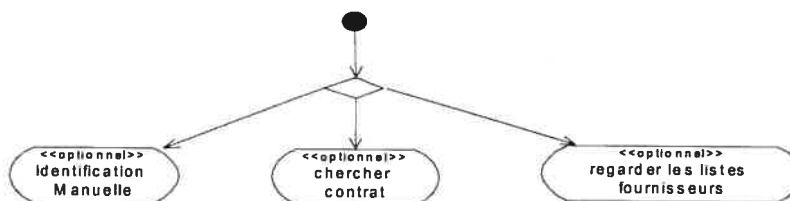
En retournant à notre exemple, le processus d'approvisionnement, le point de vue fonctionnel est formé des sous-processus suivants:



Les trois dernières étapes du cycle de vie du processus d'approvisionnement (traiter commande, recevoir ressources et paiement) sont des sous-processus obligatoires. Le sous-processus « identifier sources » peut lui-même être décomposé en des activités élémentaires.



L'activité « identifier sources potentielles » peut consister en plusieurs alternatives, comme illustré ci-bas.





### 4.4.3.2 Vue informationnelle

Cette vue décrit les informations traitées par le processus. Elle représente l'aspect statique et structurel du processus du point de vue objets, entités et informations nécessaires pour le déroulement du processus.

#### Concepts de modélisation

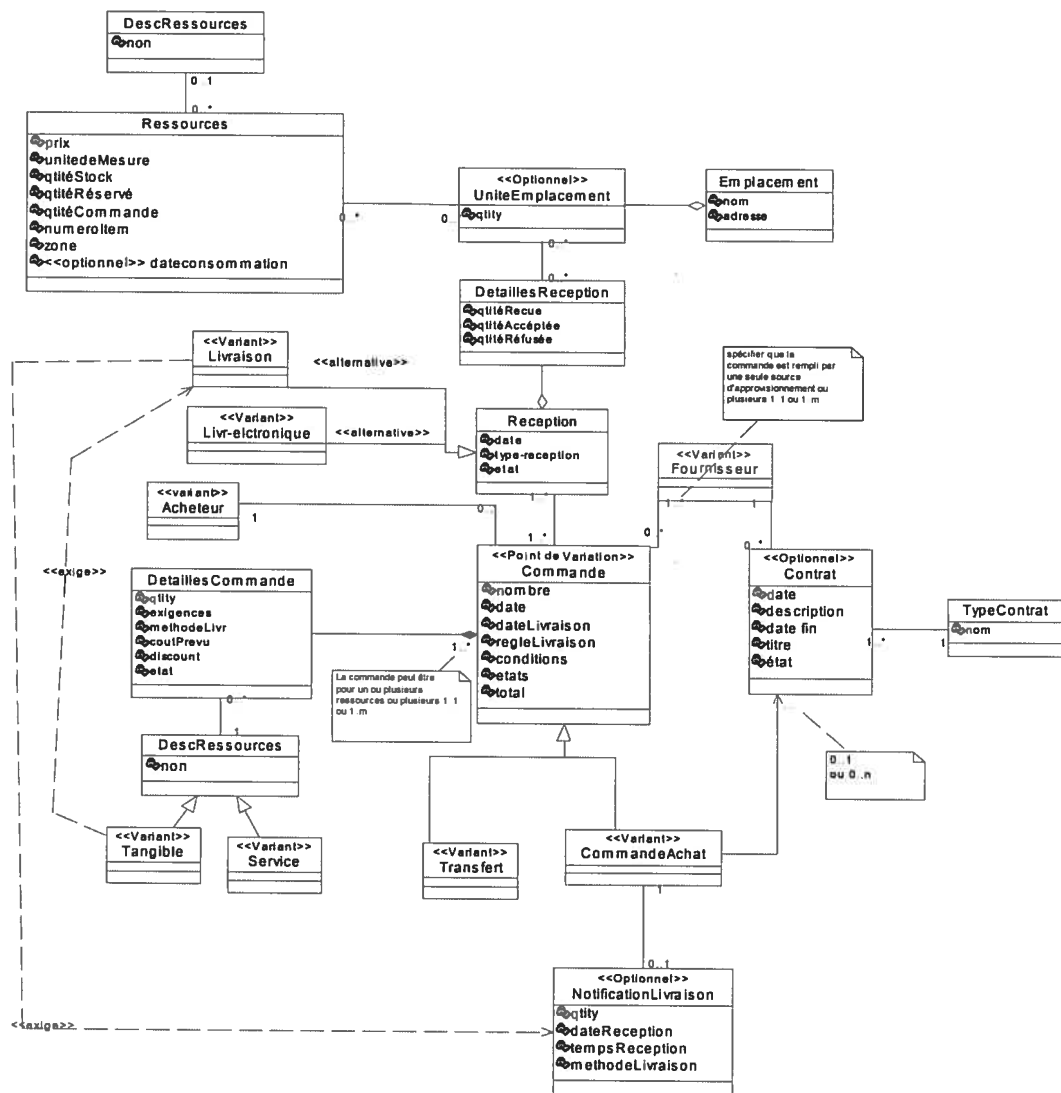
Avec cette vue, il faut pouvoir présenter tous les concepts et les données présentés dans le méta-modèle du point de vue informations et données, à l'exception des concepts qui touchent à la modélisation organisationnelle. Ces derniers vont être représentés séparément.

#### Notation

Le diagramme de classe de UML est la notation que nous allons utiliser pour représenter la vue informationnelle. Cependant, il faut avoir une notation pour: a) les entités optionnelles et les entités qui exigent au moins une instantiation ou une spécification, b) les liens optionnels ou obligatoires avec des valeurs par défaut et c) les cardinalités variables qui nous permettent de spécifier la cardinalité dépendamment de notre modèle (ex: 1,x remplacer x par 1 ou m) [Rosemann, 2001]. Pour prendre en considération ces variabilités nous augmentons les diagrammes de classes avec des notations présentés par Clauss [Clauss, 2001]:

- a. Chaque point de variation sera décrit avec une classe et un stéréotype « point de variation ». Chaque variante est décrite par une classe avec un stéréotype « variant » et une valeur marquée pour les conditions. Les conditions spécifient quand la variante peut être utilisée.
- b. Les liens entre un point de variation et ces variantes sont modélisés par des liens d'héritage.
- c. Pour présenter des liens optionnels, nous utilisons les cardinalités traditionnelles.
- d. Dans le cas où deux entités seraient alternatives, nous allons utiliser le stéréotype « alternative » avec les liens.
- e. Des interactions entre les entités utilisent des dépendances comme « mutex » et « exige ».
- f. Pour traiter la variabilité des cardinalités nous allons utiliser des notes.
- g. Pour représenter les entités optionnelles qui n'influencent pas le cœur du processus, nous allons utiliser les stéréotypes « optionnelles ».

Si nous prenons les sous-processus commande d'achat et réception du processus générique, le modèle informationnel (incomplet) qui les illustre est le suivant. À noter que ce modèle est inspiré des modèles Coad & Lefebvre et du cadre IBM San Fransisco.



#### 4.4.3.3 Vue organisationnelle

Cette dimension présente les participants du processus et regroupe les entités responsables de l'exécution de ses activités.

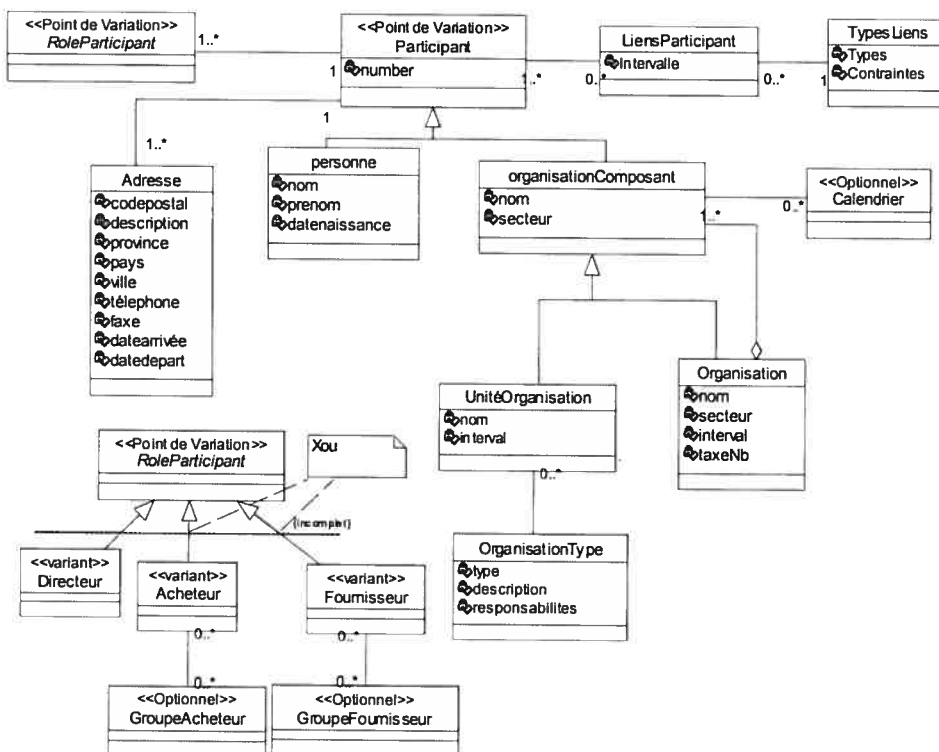
#### Concepts de modélisation

Avec cette vue, il faut représenter la structure organisationnelle et les règles qui permettent d'assigner les entités organisationnelles à des rôles [Patron rôle de Hans 2000]. Les entités organisationnelles représentent un ensemble d'acteurs comme les départements, les groupes et les projets. Le type de lien qui peut exister entre les acteurs et les rôles (respectivement les entités

organisationnelles) est par exemple « joue-le-rôle » (respectivement est « est-membre »). Un autre type de lien peut exister entre les rôles pour présenter les règles d'assignation. Hans suggère l'utilisation des notions logiques OU et Xou pour ce type de lien.

## Notation

Le diagramme de classe de UML est utilisé pour représenter un modèle organisationnel avec les mêmes extensions que celles du modèle informationnel. Pour représenter cette vue dans le cadre d'un processus d'approvisionnement, nous avons utilisé les modèles génériques a) du composant « relation management » de Coad/Lefebvre b) et du patron « party » de Fowler qui permet de présenter la vue organisationnelle comme un modèle objet.



### 4.4.3.4 Vue comportementale

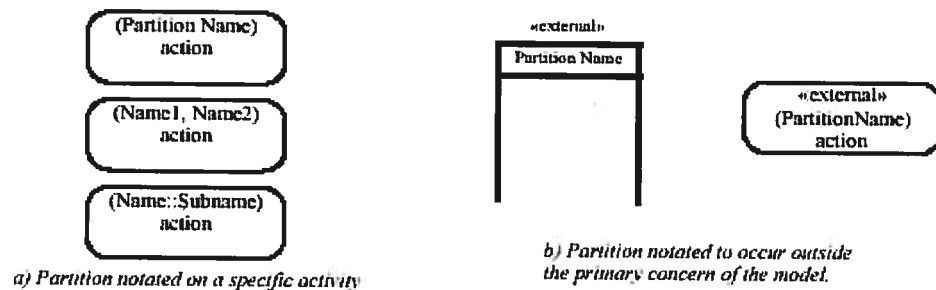
Cette vue représente l'aspect dynamique ou comportemental d'un processus. Typiquement on y retrouve deux types d'informations : a) Les diagrammes d'états montrant l'évolution des entités impliquées dans le processus (acteurs, ressources). b) Le fonctionnement dynamique de tout le processus qui forme une continuation de la modélisation fonctionnelle, en mettant l'accent sur la séquence d'interaction et en raisonnant sur des activités de granularité plus fine.

## Concepts de modélisation

Le concept fondamental de modélisation dans cette vue est l'activité qui forme l'élément élémentaire d'un processus. Au niveau global, ce qui nous intéresse est le déroulement des différentes activités, les flux de contrôle (surtout les conditions), les flux de données et les différents acteurs qui effectuent les activités. Une activité élémentaire peut être formée d'un ensemble d'actions. Ces actions pouvant être de différents types (opérations, invoquer un comportement, actions de communication comme les signaux, etc.). Ces actions ne peuvent pas être décomposées en d'autres actions dans l'activité qui l'englobe.

## Notation

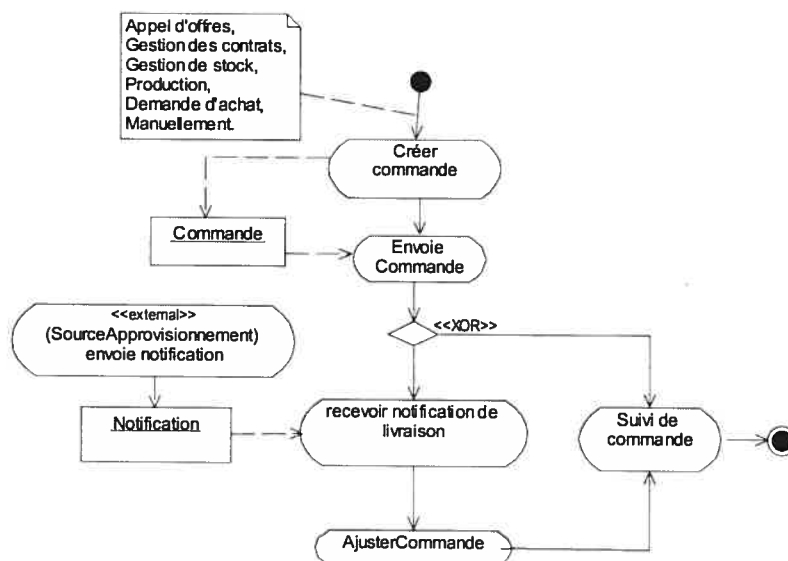
Le fonctionnement dynamique va être représenté avec le diagramme d'activité de UML 2.0 où les rôles joués par les acteurs sont présentés par des partitions (« swimlanes ») (Figure 4-5) et ceux joués par des ressources sont présentés par les objets.



**Figure 4-5 : Spécification UML2.0**

Nous allons mettre l'emphase sur les connecteurs de contrôle et surtout sur le stéréotype « XOUC », pour représenter les alternatives des entrées et sorties qu'une activité peut utiliser.

Prenons l'exemple du sous-processus traiter commande du processus d'approvisionnement. Le modèle comportemental est:



Les activités dans ce modèle sont typiquement représentées sous forme de méthodes ou d'opérations. Elles peuvent aussi être représentée comme des entités dans le diagramme informationnel. Dans ce cas, elles seront, elle-mêmes, manipulées par un système d'information: elles seront planifiées, cédulées, tracées, suivies, arrêtées, reprises, etc.

Il faut noter que la notation des machines à états finis et dans une moindre mesure les diagrammes d'activités ne se prêtent pas très bien à la spécialisation ou à l'extension, comme par exemple le cas des diagrammes de classes que l'on peut modifier d'une façon incrémentale. Par conséquent, il sera difficile de visualiser le rapport entre un modèle comportemental et ses spécialisations.

## 4.5 Procédure de spécialisation de processus génériques

Pour passer d'un modèle générique à un autre spécifique nous allons proposer une approche qui se base sur une stratégie par question. En effet, cette stratégie est utilisée par le manuel MIT. Elle permet de naviguer dans le manuel et de passer d'un niveau dans la hiérarchie des modèles de processus à d'autres niveaux plus spécifiques. Elle permet aussi de présenter les alternatives possibles aux activités ou aux processus. Toutefois, l'ensemble de questions identifiées n'est pas le même pour tous les processus. Les questions dépendent du type de processus. Les questions les plus utilisées sont: Comment? Quoi? Qui? Quand? Où? Pourquoi? (revoir section 3.3.1). Différents travaux ont utilisé la stratégie par question pour construire des modèles d'applications. Notons

l'approche de BIAIT (Business Information Analysis and Integration Technique) [Carlson, 1980] et l'approche de Wohed [Wohed, 2000].

### 4.5.1 Approches par questions

#### 4.5.1.1 BIAIT

La technique par questions « Business Information Analysis and Integration Technique » (BIAIT), proposée dans les années 80 par Carlson, a visé la classification des modèles d'entreprises, l'identification des objectifs et les informations qui leur sont reliées [Carlson, 1980]. BIAIT considère que chaque entreprise fournit un bien ou un service à un client. Ce bien ou service est produit suite à une «commande». Cette dernière peut s'appliquer à des choses, à des compétences ou à des espaces. BIAIT est formée de sept questions présentées à la Figure 4-6. Ces questions portent sur le concept de « commande ». Les réponses binaires (oui/non) vont permettre de reconnaître la catégorie de modèles qui caractérise l'entreprise.

1. *Le fournisseur envoie-t-il une facture au client ou celui-ci paie-t-il sa transaction comptant ou l'équivalent ?*
2. *Le fournisseur livre-t-il dans le futur ce qui est commandé par le client ou ce dernier l'emporte-t-il immédiatement?*
3. *Le fournisseur doit-il conserver un dossier sur le client ?*
4. *Les prix sont-ils négociés i.e. différents d'un client à l'autre ou sont-ils fixes peu importe le client ?*
5. *Le produit ou service est-il loué par le fournisseur qui en conserve les droits de propriété ou est-il acheté par le client qui en acquiert ainsi les droits?*
6. *Le fournisseur fait-il un suivi du produit ou service qui a été vendu i.e. sait-il exactement à quel client le produit ou service spécifique a été vendu, lui permettant de faire des rappels, d'offrir des mises à jour ou de répondre aux plaintes?*
7. *Le produit ou service est-il développé sur mesure selon les spécifications du client ou est-il livré à partir de l'inventaire?*

**Figure 4-6: Questions BIAIT [Carlson, 1980]**

Cette proposition de BIAIT a eu un impact conceptuel certain, puisqu'elle a joué le rôle de précurseur dans l'étude des modèles génériques, dont les travaux de [Sowa, 1984] [Shaw et Gaines, 1983], [Fertuck, 1992 ] et [Lefebvre 1996].

Les questions de BIAIT ont pour objectif d'assister la modélisation des entreprises, puisqu'elles facilitent la détermination d'un ensemble d'activités qui vont modifier l'aspect fonctionnel dépendamment de nos besoins (nos réponses) [Fertuck, 1992]. Exemple (tiré de la thèse de

Lefebvre) [Lefebvre, 1996]: Pour la question «est ce que le fournisseur facture le client?» Si le client paie comptant, par chèque ou carte de crédit, il n'y a pas d'activités correspondantes. Si le fournisseur doit envoyer une facture plus tard, alors le fournisseur doit : approuver le produit, et tenir le compte-client.

Notons qu'il est difficile d'interpréter ces questions dans certains domaines d'application. Pour résoudre ce problème, Lefebvre établit une analogie dans ses études de cas pour construire des modèles génériques. Il a précisé qu'il faut interpréter les termes utilisés dans le modèle générique, pour les adapter au modèle de l'entreprise [Lefebvre 1999]. Nous allons utiliser le même principe d'interprétation.

#### 4.5.1.2 Approche Wohed

Wohed propose un nombre prédéfini de questions pour modéliser le domaine [Wohed 2000] [Wohed 2000b]. Ses questions sont liées à des patrons pour suggérer une solution à sélectionner. Dans son premier travail, Wohed présente les questions liées au domaine de réservation. Ces questions ont été généralisées dans son deuxième travail en les liant aux patrons « transaction » de COAD [Coad 1997]. D'après Wohed, avec la généralisation des questions, la sémantique de quelques-unes est perdue, ce qui l'a poussé à trouver des substitutions adéquates de questions pour chaque domaine [Wohed 2000b]. Les questions génériques proposées de Wohed sont:

- *Est-ce que la transaction consiste en un ou plusieurs objets?*
- *Est-ce que la transaction concerne un objet spécifique ou les caractéristiques d'un objet?*
- *Est-ce que toutes les transactions ont le même caractère ou sont-elles divisées en plusieurs catégories?*
- *Est-ce qu'il est nécessaire de conserver les informations sur la transaction qui devance la transaction en cours? Si oui, est ce que la transaction qui précède concerne un(des) objet(s) spécifique(s) ou la spécification de(s) objet(s).*
- *Est-ce que la transaction doit être associée à d'autres éléments?*
- *Est-ce que la transaction est faite par l'intermédiaire d'une partie tierce? Si oui, est-ce qu'il est important de mémoriser les informations sur cette partie?*

Ce qui nous intéresse dans cette approche est le fait que d'une part elle met l'accent sur la modélisation de la vue informationnelle, et que d'autre part elle suggère quatre catégories de questions qui peuvent être liées à des points de variabilités:

- Questions pour identifier les super-types.
- Questions pour la cardinalité.

- Questions pour la généralisation/spécialisation.
- Questions pour identifier les catégories des objets et les liens

### 4.5.2 Principes de notre approche

Les questions des différentes approches sont liées à des processus spécifiques. Ces questions ne seront pas utilisées d'une façon systématique sur un ensemble de processus génériques dans un référentiel. Nous proposons de généraliser les questions et de les relier à des éléments du méta-modèle (Figure 4-3). Ces questions génériques vont être applicables à différents modèles génériques. En d'autres termes, pour chaque processus générique nous pouvons spécialiser ces questions génériques dépendamment du domaine du processus. Les réponses aux questions spécialisées vont permettre de spécialiser les modèles représentant le processus dépendamment de l'entreprise.

Pour mieux comprendre nous allons présenter un exemple simple. Supposons que nous avons un fragment d'un modèle informationnel spécifique qui représente la commande et la livraison des produits d'une compagnie (Figure 4-7).

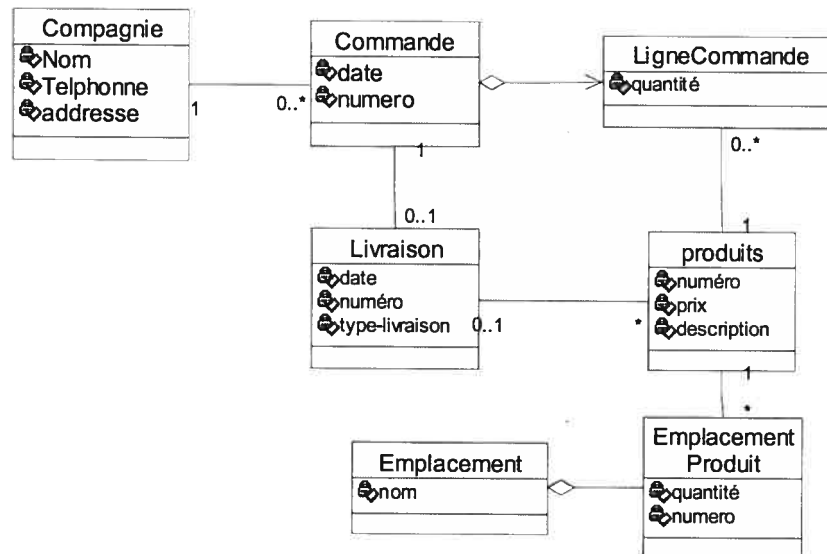


Figure 4-7 : Vue informationnelle spécifique à un domaine.

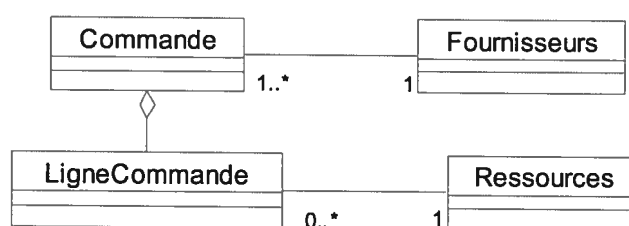
A partir de ce modèle nous pouvons nous poser la question « *Est-ce que les produits commandés sont toujours tangibles?* ». Cette question nous indique que nous pouvons avoir deux types de produits : ceux liés à des emplacements et ceux qui sont consommés sans être stockés.

En effectuant une abstraction de cette question toute en la liant aux éléments du méta-modèle du processus d'affaires (Figure 4-3), la question devient : *Est ce que les <<ressources>> sont*



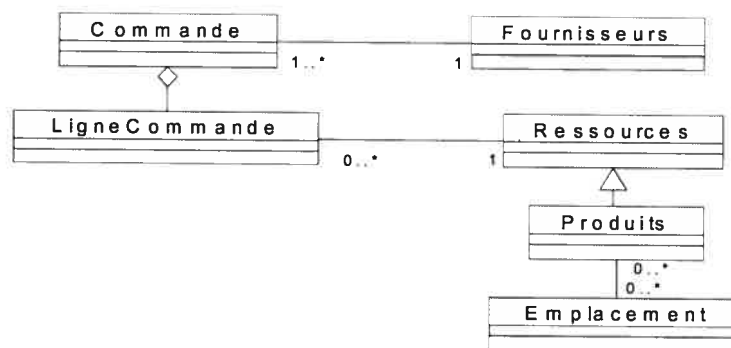
*tangibles ou non?*. Si nous prenons cette question générique et que nous essayons de l'appliquer à des modèles de processus génériques nous aurons d'autres questions spécialisées dépendantes du domaine. Par exemple, reprenons les sous-processus « commande » et « livraison ». Les questions spécialisées seront: *Est-ce que les produits commandés sont tangibles ou non? Est ce que les produits livrés sont tangibles ou non?* Les réponses à ces questions vont influencer la spécialisation de ces modèles génériques.

La Figure 4-8 est une représentation générique d'une « commande ». Si nous spécialisons la question générique dépendamment de ce modèle nous aurons la question suivante: *Est ce que les ressources sont tangibles ou non ?*:

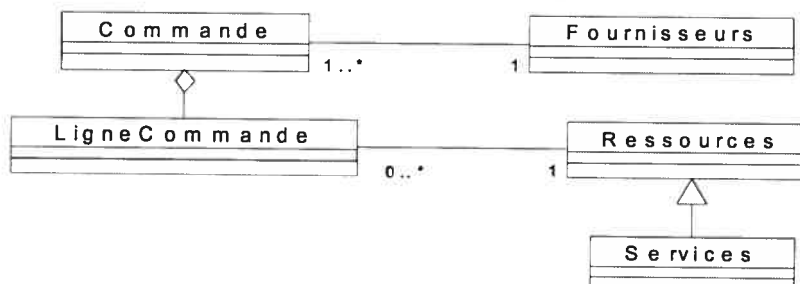


**Figure 4-8 : Représentation générique du modèle informationnel d'une commande**

- La réponse positive à cette question génère le modèle suivant à partir du reposoir:



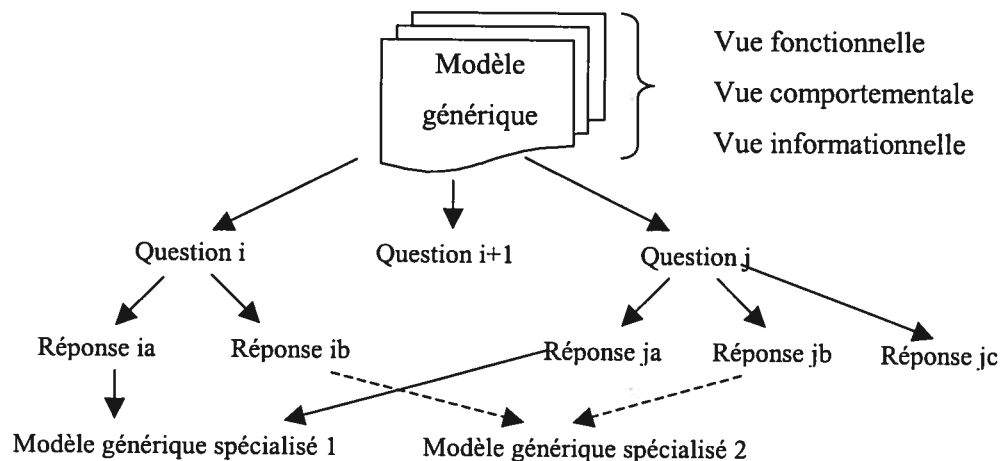
- La réponse négative génère un autre modèle:



Comme nous pouvons le remarquer nous avons deux niveaux de spécialisation : a) la spécialisation des questions dépendamment du domaine d'affaires du modèle générique que nous voulons représenter. b) La spécialisation des modèles génériques dépendamment des réponses aux questions spécialisées.

Les questions spécialisées vont permettre d'effectuer des choix entre les différentes variantes possibles que le processus générique présente. Les réponses à ces questions décrivent le cadre dans lequel la réutilisation de chaque modèle d'un même problème est adaptée. En principe, la combinaison des différentes réponses à ces questions doit fournir différents modèles (Figure 4-9).

Ces questions touchent aux différentes vues qui représentent un processus d'affaires. Elles influencent certaines vues plus que d'autres (voir Tableau 4.1). Il existera des réponses qui vont spécialiser différentes parties du même modèle qui représente une vue. Également, il y aura des réponses qui vont spécialiser différentes parties de modèles de différentes vues.



**Figure 4-9 : Combinaisons des réponses**

Toutes les combinaisons ne sont peut-être pas possibles, et toutes les questions n'influencent pas en particulier le processus en question, comme elles ne sont pas toujours pertinentes.

Dans la section suivante, nous allons proposer un ensemble de questions génériques. Elles sont généralisées à partir des stratégies des questions identifiées et des initiatives de modélisation générique déjà présentées au chapitre 3. Cependant, ces questions ne sont pas définitives. La mise en œuvre de notre méthodologie dans différents domaines d'application pourra nous amener à modifier cet ensemble par des ajouts, retraites ou modifications.

### 4.5.3 Les questions proposées

Pour faciliter la catégorisation des questions, nous allons diviser le méta-modèle (Figure 4-10) en trois zones : a) La zone O (organisationnelle) regroupe les questions qui touchent aux concepts mettant l'emphase sur l'aspect organisationnel d'un processus d'affaires. b) La zone C (comportementale) s'intéresse aux concepts qui symbolisent des préoccupations fonctionnelles et comportementales. c) La zone I (informationnelle) englobe les concepts qui traitent les informations manipulées par un processus.

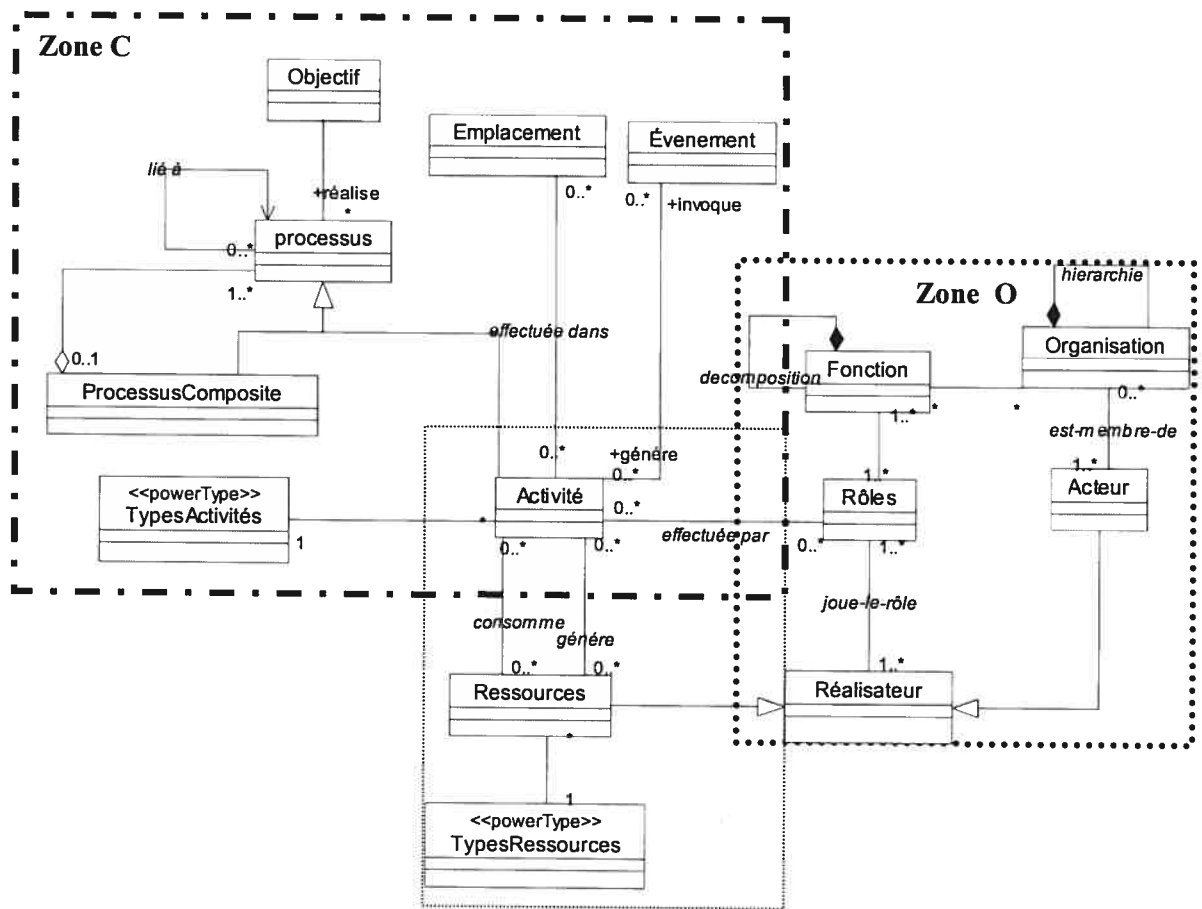


Figure 4-10: Zones des questions dans le Méta-modèle

#### Zone O:

1. *Est-ce que deux ou plusieurs <<rôles>> sont joués par le même <<réalisateur>> en même temps?* Le réalisateur peut effectuer plusieurs rôles en même temps, ce qui influence le déroulement du processus étant donné que les activités effectuées avec ces rôles peuvent se combiner ou être simplifiées. Par exemple, on peut avoir des activités effectuées pour le

compte d'un tiers acteur, prenons le cas d'une commande de produits d'un fournisseur effectuée par une entreprise pour le compte d'un de ses clients.

2. *Est-ce que <<l'acteur>> est une personne ou une <<organisation>>? Des contraintes institutionnelles (légales, réglementaires) influencent le traitement et les activités selon le type d'un acteur.*
3. *Est-ce que les <<réalisateurs>> qui jouent un <<rôle>> donné sont différenciés? Ou est-ce que les <<types d'activités>> sont influencés par les <<types de réalisateurs>>? Un groupe d'acteurs peut avoir des privilèges spécifiques, ce qui influence le processus. Par exemple, un client peut faire partie d'un groupe qui bénéficie d'un escompte spécifique pour un type donné de produit. La réponse à cette question influence la vue fonctionnelle, comme elle définit le type d'activité et la fonctionnalité nécessaire pour réaliser le processus.*
4. *Est-ce que le <<réalisateur>> est interne ou externe? Le type de réalisateur influence le déroulement du processus. Par exemple, dans le cas d'un processus d'approvisionnement, si la source d'approvisionnement (l'acteur) est interne, alors le processus va se baser sur le transfert entre départements.*
5. *Est-ce que toutes les informations concernant les <<réalisateurs>> qui jouent un <<rôle>> sont mémorisées? Il faut prendre en considération les fonctionnalités de gestion des acteurs comme la maintenance des historiques, l'enregistrement des informations, la planification des comptes, et d'autres. Notons le cas d'un supermarché, où il n'est pas intéressant de mémoriser tous les clients qui achètent des produits.*

#### **Zone C:**

6. *Est-ce que l'<<emplacement>> est indispensable pour <<l'activité>>? Ou est-ce que l'<<emplacement>> pour effectuer l'<<activité>> est virtuel ou non? Une ressource peut dépendre de son emplacement, par exemple un cours est offert à l'université dans un campus donné et non pas à distance. De plus, une activité peut être effectuée juste dans un emplacement spécifique, par exemple livrer les produits à une adresse donnée, ou une commande doit être effectuée sur place et non pas virtuellement via le net.*

7. *Le déroulement d'un <<processus>> est-il régi par des accords?* Dans ce cas, il y a des vérifications à faire par rapport à un plan prédéfini. Par exemple, l'achat d'un produit se base sur un contrat d'affaires que les partenaires doivent respecter.
8. *Étant donné le <<processus>> initial, peut-il y avoir des contraintes additionnelles de pré-réquisition, d'accessibilité et d'utilisation?* Un processus peut avoir des conditions additionnelles d'applicabilité. De plus, un processus peut n'avoir lieu que dans le siège (ex: certaines transactions bancaires ne peuvent avoir lieu qu'au siège).
9. *Est-ce qu'il existe une durée déterminée pour le traitement d'une <<activité>>?* Une activité est parfois liée à un intervalle de temps. En cas de dépassement de cet intervalle, le comportement change. Par exemple, un processus dans un système à temps réel est fortement lié à des contraintes de temps réel.

**Zone I :**

10. *Est-ce que l'<<activité>> manipule une <<ressource>> concrète ou une représentation de la <<ressource>>?* Une activité peut interagir avec une représentation de l'objet (type d'objet) et non pas l'objet concret. Par exemple : nous pouvons réserver un ticket de voyage avec le vol Lb200 le jeudi 7 septembre 2004, classe économique, dans un avion. Par la suite la réservation sera avec le type et l'achat avec l'objet.
11. *Est-ce que les informations concernant les <<activités>> effectuées par un <<réalisateur>> sont mémorisées?* La mémorisation des activités pousse à avoir des fonctionnalités pour gérer les données et pour pouvoir effectuer des recherches sur ces données. Toutefois, il existe parfois des activités manuelles qui ne nécessitent pas une mémorisation, il faut seulement mémoriser le résultat.
12. *Est-ce que les <<ressources>> sont tangibles ou non ?* Le traitement et la représentation des produits tangibles sont différents de ceux des produits non tangibles. Par exemple: si les ressources sont tangibles, il faut ajouter des fonctionnalités pour gérer l'inventaire. Sinon, il faut ajouter les fonctionnalités concernant la gestion des coûts et finances.
13. *Est-ce que les <<ressources>> sont en quantité limitée ou illimitée?* Il existe des ressources qui sont à toutes fins pratiques en quantité illimitée. Ce type de ressources

nécessite parfois une représentation et une gestion spéciale, comme c'est le cas pour l'électricité.

**14. Est-ce que les <<ressources>> sont périssables ou non?** Il existe des ressources de nature périssable. Par exemple, certains services livrés à un client ne peuvent pas être stockés en avance et livrés par la suite. De plus, cette question est liée aux deux précédentes. On peut avoir des ressources qui sont tangibles et épuisables en même temps, par exemple: les aliments sont tangibles, mais ils peuvent être périssables et par la suite épuisables.

**15. Est-ce que la consommation réduit la valeur (qualité ou quantité) de la <<ressource>> ou non?** Nous devons tenir compte de deux considérations avec la consommation; la diminution de la valeur quantitative ou qualitative. Prenons le cas de la consommation de la nourriture (quantitative) et le visionnement d'un film sur support informatique (qualitative). Parfois, l'intention d'utilisation d'une ressource influence le traitement du processus qui utilise la ressource. Par exemple: l'achat d'un livre pour le mettre en stock et le vendre par la suite (utilisation matérielle, quantitative), ou bien pour consommer l'information dans ce livre avec la location (utilisation informationnelle, qualitative). Il faut noter que le type de ressources influence parfois l'intention d'utilisation, et vice versa. Par exemple: un livre est tangible, mais son utilisation peut être non matérielle.

Ces questions vont influencer les différentes vues qui représentent un processus d'affaires. Le tableau qui suit montre les liens entre les questions et les vues. Avec «Oui», nous insistons sur le fait que la question touche principalement la vue désignée, alors qu'avec «en partie», nous montrons que la question influence partiellement la vue. Étant donné que la vue dynamique (comportementale) dépend de la vue fonctionnelle (la dynamique est le « comment » du « quoi »), tout ce qui influence le fonctionnel influence aussi indirectement la dynamique. Cependant, certains facteurs toucheront principalement le déroulement du processus (dynamique), mais à moindre mesure le traitement fonctionnel.

Questions	Information.	Fonction.	Dynami.	Organisa.
1. <i>Est-ce que deux ou plusieurs rôles sont joués par le même réalisateur?</i>		Oui	Oui	Oui
2. <i>Est-ce que l'acteur est une personne ou une organisation?</i>			Oui	Oui
3. <i>Est-ce que les réalisateurs qui jouent un rôle donné sont différenciés? Ou</i>		En partie	Oui	Oui

<i>est-ce que les types d'activités sont influencés par les types de réalisateur?</i>				
<i>4. Est-ce que le réalisateur est interne ou externe?</i>	Oui	Oui	En partie	
<i>5. Est-ce que toutes les informations concernant les réalisateurs qui jouent un rôle sont mémorisées?</i>	En partie	Oui		Oui
<i>6. Est-ce que l'emplacement est indispensable pour l'activité?</i>	Oui	En partie	En partie	
<i>7. Le déroulement d'un processus est-il régi par des accords (ex: contrat)?</i>	Oui	Oui	Oui	
<i>8. Étant donné le processus initial, peut-il y avoir des contraintes additionnelles de pré-réquisition, d'accessibilité et d'utilisation?</i>		En partie	Oui	
<i>9. Est-ce qu'il existe une durée déterminée pour le traitement d'une activité?</i>		En partie	Oui	
<i>10. Est-ce que l'activité manipule un objet concret ou une représentation de l'objet?</i>	Oui	En partie	En partie	
<i>11. Est-ce que les informations concernant les activités effectuées par un acteur sont mémorisées?</i>	Oui	Oui	En partie	
<i>12. Est-ce que les ressources sont tangibles ou non ?</i>	Oui	Oui	Oui	
<i>13. Est-ce que les ressources sont en quantité limitée ou illimitée?</i>	Oui	Oui	Oui	
<i>14. Est-ce que les ressources sont périssables ou non?</i>	Oui	Oui	Oui	
<i>15. Est-ce que la consommation réduit la valeur (qualité ou quantité) de la ressource ou non?</i>	En partie	Oui	En partie	

**Tableau 4.1: Tableau des questions et des vues**

## 4.6 Exemple de spécialisation

Dans cette section, nous allons spécialiser les questions pour les adapter au domaine de processus d'approvisionnement. Rappelons qu'un processus d'approvisionnement est une fonction qui consiste à fournir en temps voulu à l'entreprise, les ressources et les services nécessaires auprès de sources d'approvisionnement (annexe 1). Nous allons présenter les différentes combinaisons de réponses de ces questions par vues et les modèles correspondants (annexe 2). Rappelons que les questions identifiées auparavant ne sont pas toutes pertinentes pour notre domaine.

## 4.6.1 Les questions appliquées au processus générique d'approvisionnement

### A) Modèle Organisationnel

Rappelons le modèle organisationnel générique de la section 4.4.3.3:

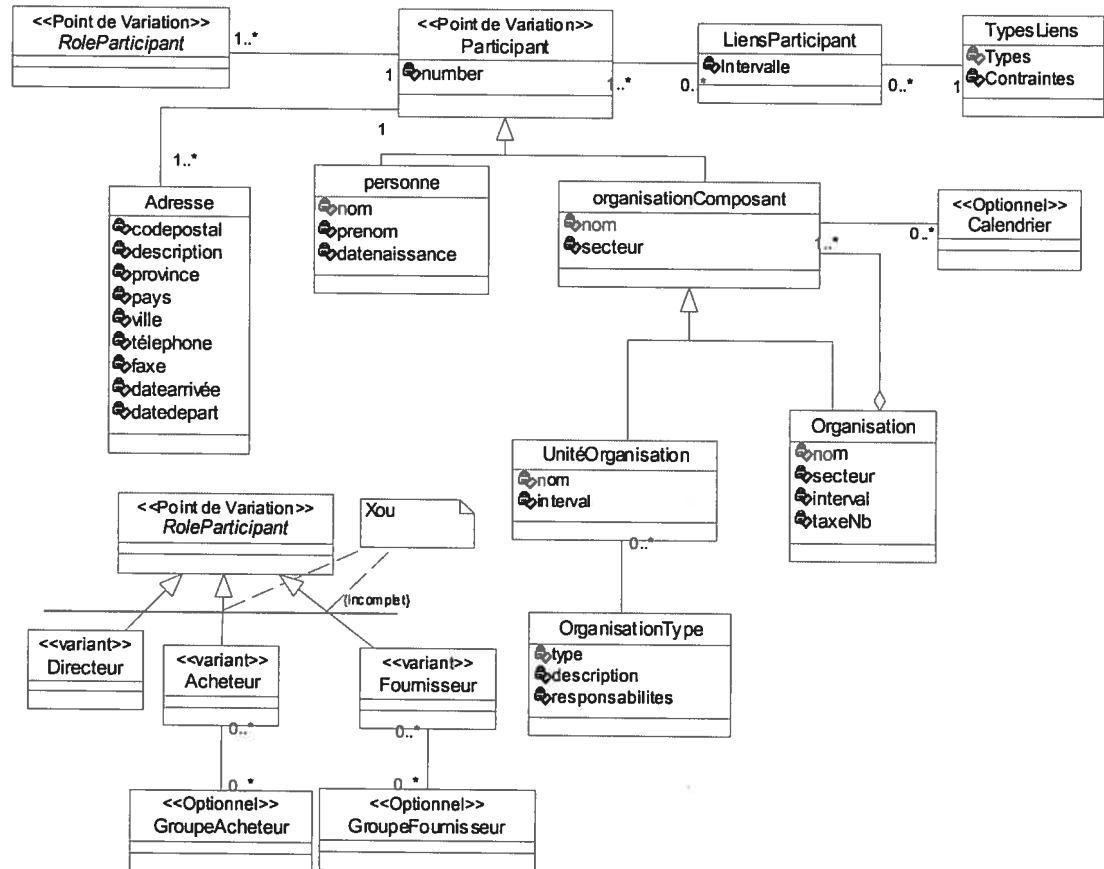


Figure 4-11 : Modèle organisationnel générique

Les questions spécialisées à partir des questions génériques pour ce modèle sont:

1. Est-ce que les rôles de l'acheteur (<<rôle>>) et du directeur (<<rôle>>) peuvent être joués par le même participant (<<acteur>>)?
2. Est-ce que le participant (<<acteur>>) qui joue le rôle de fournisseur (<<rôle>>) est une organisation ?
3. Est-ce que le participant (<<acteur>>) qui joue le rôle d'acheteur (<<rôle>>) est une organisation?

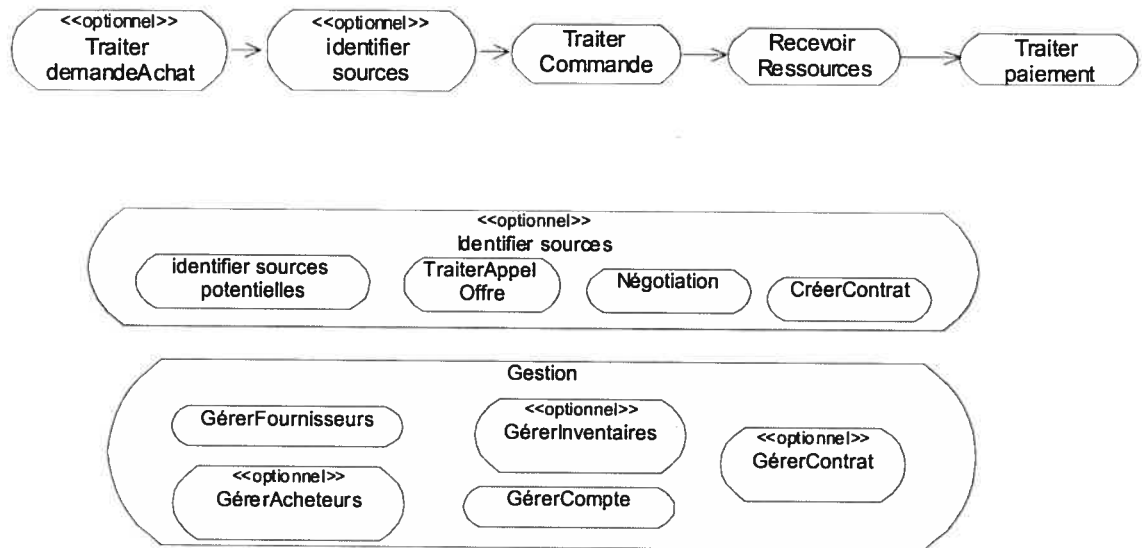


4. Est-ce que le fournisseur (<<rôle>>) fait partie d'un groupe de fournisseurs différenciés?
5. Est-ce que l'acheteur (<<rôle>>) fait partie d'un groupe de demandeur différenciés?

Si toutes les réponses sont positives, le modèle est identique au modèle générique. Pour différentes représentations du modèle organisationnel, il faut se référer à l'annexe2. Dans cette annexe, nous pouvons identifier les différentes combinaisons des réponses et leurs modèles correspondants.

## B) Modèle fonctionnel

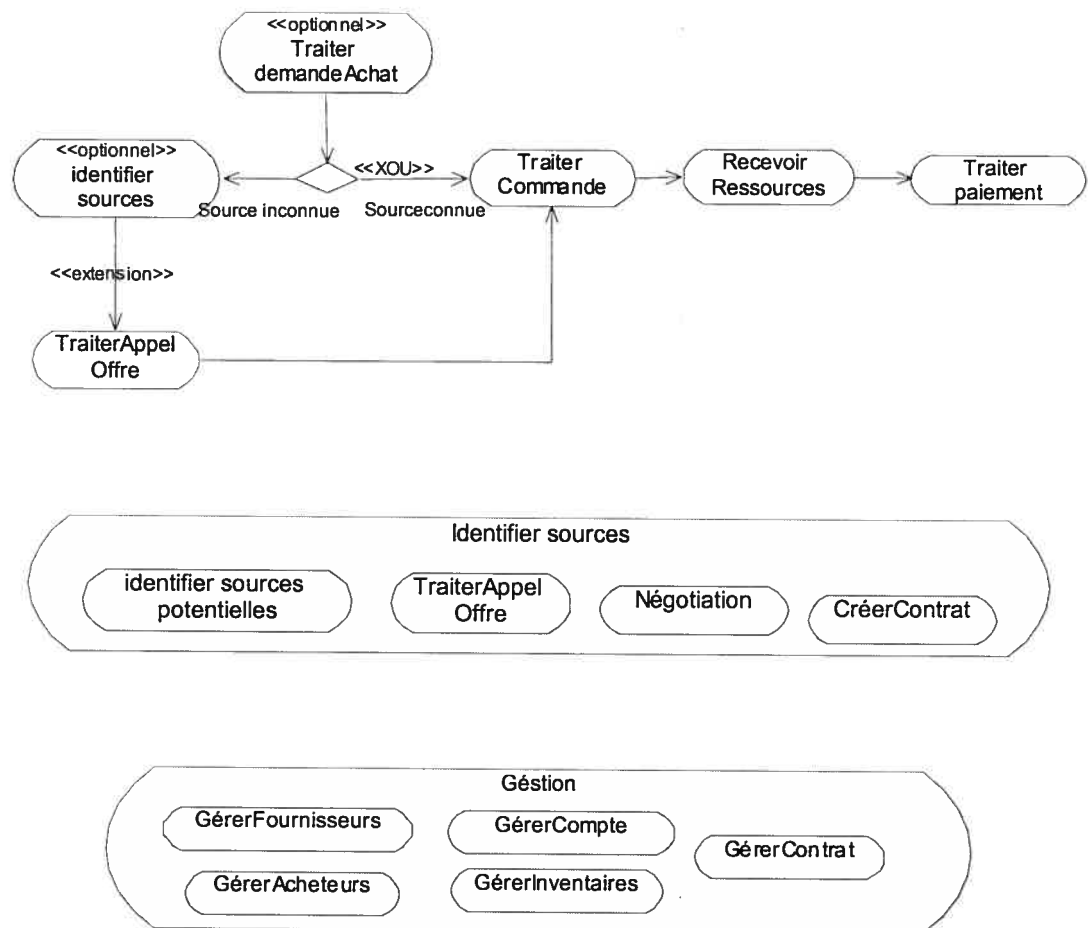
Le modèle fonctionnel générique de processus d'approvisionnement est le suivant :



Les questions spécialisées pour cette vue sont :

1. Est-ce que les rôles de l'acheteur (<<rôle>>) et de directeur (<<rôle>>) peuvent être joués par le même participant (<<acteur>>)?
2. Est-ce que le participant (<<acteur>>) qui joue le rôle de fournisseur (<<rôle>>) est externe?
3. Est-ce que le participant (<<acteur>>) qui joue le rôle d'acheteur (<<rôle>>) est externe?
4. Est-ce que le processus d'approvisionnement (<< processus>>) se base sur un contrat (<<ressource>> ) ?
5. Est-ce que les informations concernant les commandes (<< activité>>) des fournisseurs (<<rôle>>) sont mémorisées?
6. Est-ce que les informations concernant les réceptions (<< activité>>) sont mémorisées?
7. Est-ce que les informations concernant les appels d'offres (<< activité>>) sont mémorisées?

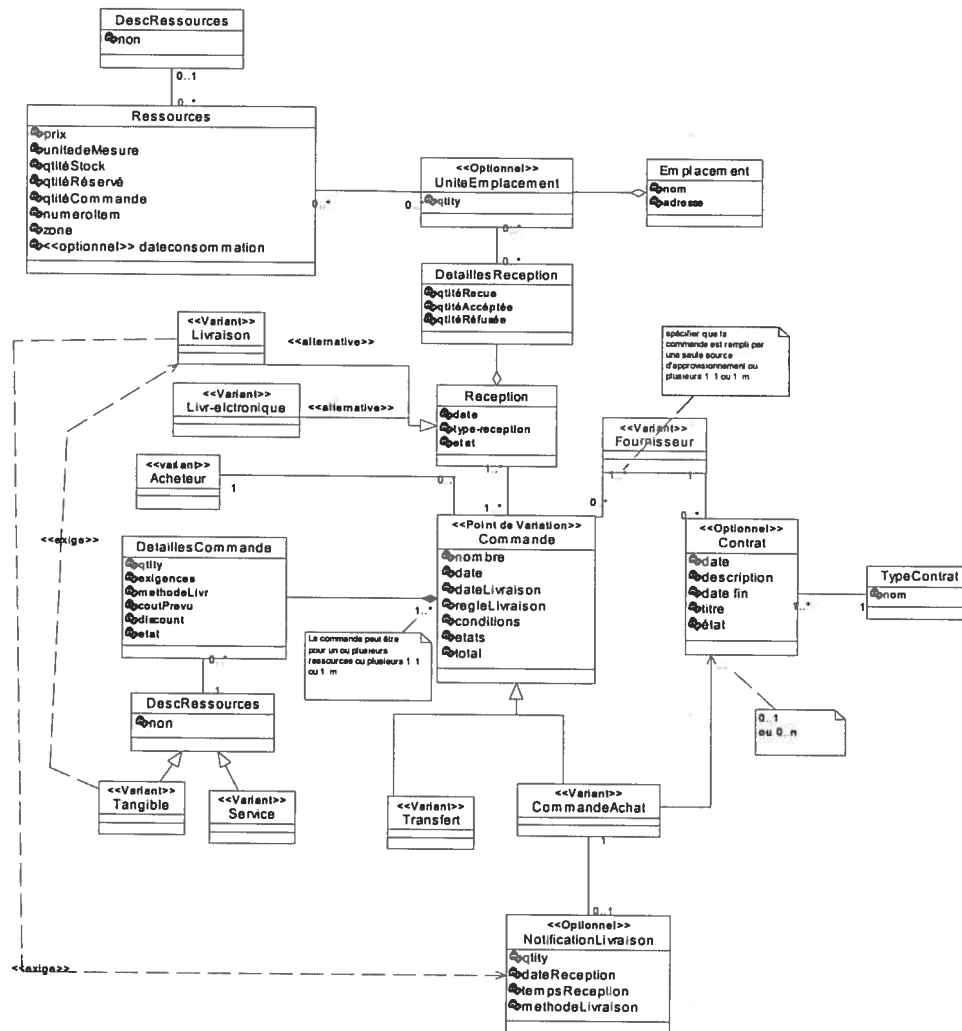
8. Est-ce que les produits (<<ressource>>) sont tangibles ?
  9. Est-ce que les produits (<<ressource>>) sont en quantité limitée?
  10. Est-ce que les produits (<<ressources>>) sont périssables?
  11. Est-ce que la consommation (<< activité>>) du produit (<<ressources>>) réduit la quantité?
- Les combinaisons des réponses à ces questions fournissent différents modèles. La Figure 4-12 représente le modèle qui suppose que les réponses à toutes les questions sont positives. Pour voir différents modèles qui représentent différentes combinaisons de réponses, référez-vous à l'annexe2.



**Figure 4-12 : Modèle fonctionnel d'un processus d'approvisionnement**

## C) Modèle informationnel

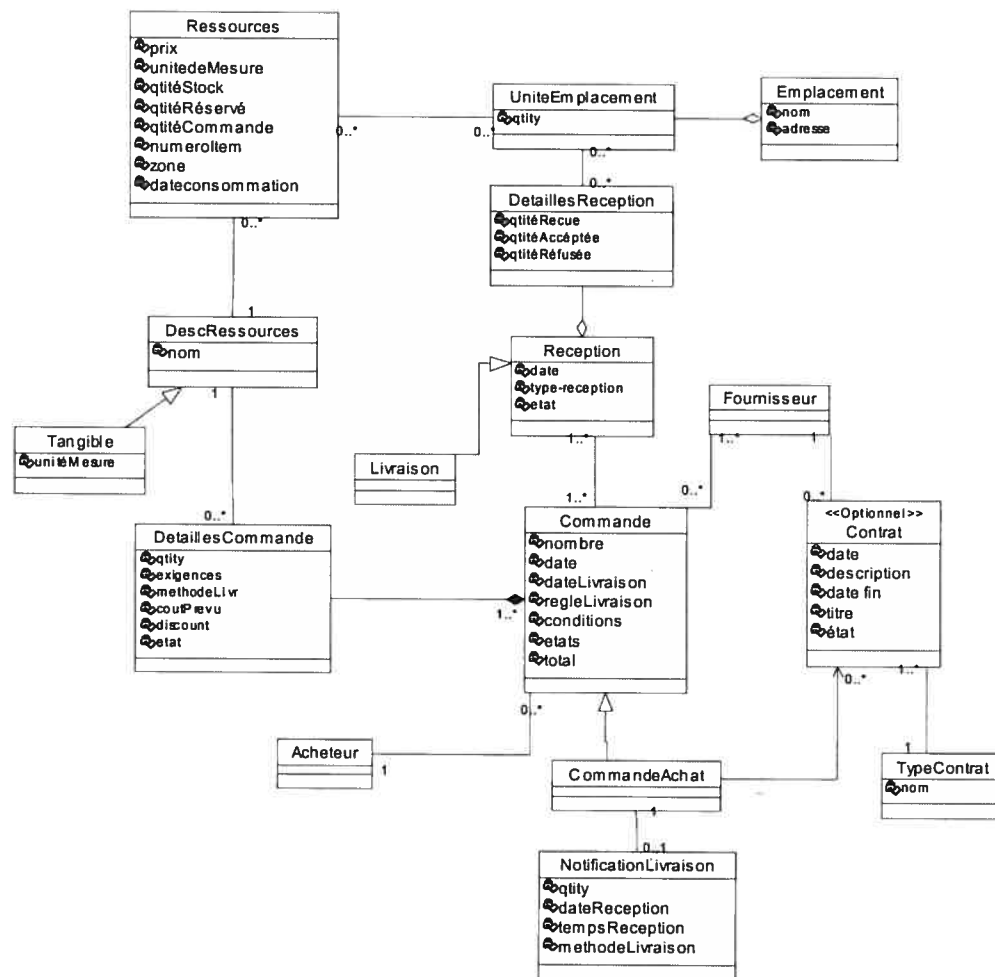
Le modèle informationnel générique représenté déjà à la section 4.4.3.2 est :



1. Est-ce que le participant (<<acteur>>) qui joue le rôle de fournisseur (<<rôle>>) est externe?
2. Est-ce que le participant (<<acteur>>) qui joue le rôle d'acheteur (<<rôle>>) est externe?
3. Est-ce que la livraison (<<sous-processus>>) est liée à un emplacement (<<emplacement>>)?
4. Est-ce que la commande (<<activité>>) manipule une ressource concrète (<<ressource>>)?
5. Est-ce que la livraison (<<sous-processus>>) manipule une ressource concrète (<<ressource>>)?
6. Est-ce que le processus d'approvisionnement (<<activité>>) se base sur un contrat (<<ressource>>)?
7. Est-ce que les produits (<<ressources>>) sont tangibles ou non?

8. Est-ce que les produits (<<ressources >>) sont en quantité limitée ou illimitée?
9. Est-ce que les informations concernant les commandes (<<activité>>) des fournisseurs (<<rôle>>) sont mémorisées?
10. Est-ce que les informations concernant les livraisons (<<sous-processus>>) sont mémorisées?
11. Est-ce que les informations concernant les appels d'offres (<<sous-processus >>) sont mémorisées?
12. Est-ce que la livraison (<<activité>>) concerne un produit (<<ressource>>) tangible ou non ?
13. Est-ce que les produits (<<ressource>>) sont périssables ou non?

Les réponses à ces questions permettent de contextualiser le modèle informationnel en facilitant le choix entre les entités, les attributs et les liens nécessaires au processus d'affaires. Si nous reprenons le modèle informationnel pour représenter les sous-processus commande d'achat et la réception, le modèle qui représente des réponses positives à ces questions est:



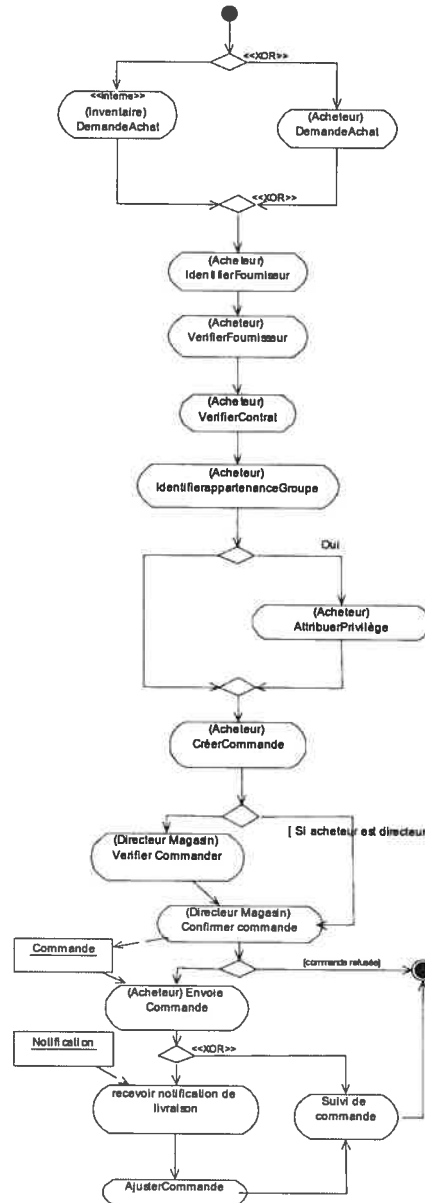
## D) Modèle comportemental

1. Est-ce que les rôles de l'acheteur (<<rôle>>) et de directeur (<<rôle>>) peuvent être joués par le même participant?
2. Est-ce que le fournisseur (<<rôle>>) fait partie d'un groupe de fournisseurs différenciés?
3. Est-ce que l'acheteur (<<rôle>>) fait partie d'un groupe d'acheteurs différenciés?
4. Est-ce que le processus d'approvisionnement (<<processus>>) se base sur un contrat (<<ressources>>) ?
5. Est-ce que les produits (<<ressource>>) achetées sont tangibles ou non?
6. Est-ce que les produits (<<ressource>>) sont en quantité limitée ou illimitée?
7. Quelles sont les contraintes de pré-réquisition (temps voulu), d'accessibilité (emplacement voulu) et d'utilisation nécessaire (objet voulu) de (<<sous-processus>>) « traiter commande »?
8. Quelles sont les contraintes de pré-réquisition (temps voulu), d'accessibilité (emplacement voulu) et d'utilisation nécessaire (objet voulu) de (<<sous-processus>>) «reception commande »?
9. Quelles sont les contraintes de pré-réquisition (temps voulu), d'accessibilité (emplacement voulu) et d'utilisation nécessaire (objet voulu) de (<<sous-processus>>) «paiement »?
10. Est-ce que la commande (<<activité>>) nécessite une durée déterminée?

Les questions 7-8-9-10 sont liées étroitement au domaine d'application spécialisé, nous ne pouvons pas les utiliser avec les combinaisons, mais les réponses à ces questions sont efficaces pour l'élaboration des modèles dynamiques pour un domaine d'application spécifique. Les questions 5 et 6 vont influencer le comportement de l'inventaire qui est un sous-processus à part entière.

Rappelons que le comportement d'un processus dépend de l'industrie et du domaine d'application. Nous pouvons peut-être prévoir un comportement général, mais les détails du déroulement sont vraiment internes et propres à l'entreprise et à ses politiques. Par conséquent, la spécialisation de processus est plus difficile, surtout si nous mélangeons les acteurs et par la suite nous enlevons des partitions « swimlanes ». Également, il n'est pas évident d'enlever des opérations internes, de changer les états, de supprimer des échanges de validations et d'autres.

Pour les réponses positives, le modèle comportemental est présenté à la Figure 4-13. Si les réponses ne sont pas positives, il faut se référer à l'annexe 2:



**Figure 4-13: Modèle comportemental pour traiter une commande**

#### 4.6.2 Exemple de situation

Supposons qu'un magasin de disques ait besoin de modéliser son processus d'approvisionnement. Le processus doit permettre d'acheter des ressources tangibles (des disques) pour pouvoir les

stocker et les vendre par la suite aux clients. L'achat se fait toujours suivant un contrat avec un fournisseur reconnu. Cependant, l'identification des fournisseurs potentiels se fait manuellement, pour créer par la suite des contrats. N'importe quel employé (représentant ou directeur) du magasin peut commander des disques, mais aucune commande n'est envoyée au fournisseur sans l'approbation du directeur du magasin. Une notification de réception de la commande doit parvenir du fournisseur. La livraison de disques se fait au magasin selon la date fournie avec la notification.

Après avoir parcouru le reposoir et identifié le modèle générique du processus, le concepteur doit répondre à un ensemble de question et doit les interpréter dépendamment du contexte de son domaine (le magasin de disques). Dans le Tableau 4.2 qui suit, un ensemble de questions interprétées est présenté. La combinaison de réponses à ces questions va permettre de générer le modèle de processus correspondant à notre magasin à partir de notre reposoir.

<i>Numéro</i>	<i>Question spécifique au processus générique d'approvisionnement</i>	<i>Formulation des questions pour l'entreprise de disques</i>	<i>Réponses</i>
<i>1</i>	Est-ce que les rôles de l'«acheteur» et de «directeur» peuvent être joués par le même «participant» ?	<i>Est-ce que les rôles du représentant et du directeur peuvent être joués par le même employé ?</i>	<i>Oui</i>
<i>2</i>	Est-ce que l'«acheteur» est une organisation ?	<i>Est-ce que l'acheteur est une organisation ?</i>	<i>Non</i>
<i>3</i>	Est-ce que le «fournisseur» est une organisation ?	<i>Est-ce que le fournisseur du disque est une organisation ?</i>	<i>Oui</i>
<i>4</i>	Est-ce que le «fournisseur» fait partie d'un groupe de fournisseurs différencié ?	<i>Est-ce que le fournisseur du disque fait partie d'un groupe de fournisseurs différenciés ?</i>	<i>Oui</i>
<i>5</i>	Est-ce que l'«acheteur» fait partie d'un groupe d'acheteurs différencié ?	<i>Est-ce que l'acheteur fait partie d'un groupe d'employés différenciés ?</i>	<i>Non</i>
<i>6</i>	Est-ce que le «fournisseur» est externe ou interne ?	<i>Est-ce que le fournisseur est externe ?</i>	<i>Oui</i>
<i>7</i>	Est-ce que l'«acheteur» est externe ?	<i>Est-ce que le rôle d'acheteur peut être effectué pour un acteur externe ?</i>	<i>Oui</i>
<i>8</i>	Est-ce que les informations concernant les «commandes» des «fournisseurs» sont mémorisées ?	<i>Est-ce que toutes les informations concernant les commandes sont mémorisées ?</i>	<i>Oui</i>
<i>9</i>	Est-ce que les informations concernant les «livraisons» sont mémorisées ?	<i>Est-ce que les informations concernant les livraisons sont mémorisées ?</i>	<i>Oui</i>
<i>10</i>	Est-ce que les informations concernant les «appels d'offres» sont mémorisées ?	<i>Est-ce que les informations concernant les d'appels d'offres sont mémorisées ?</i>	<i>Non</i>

11	Est-ce que le <<processus d'approvisionnement>> se base sur un <<contrat>> ?	<i>Est-ce que le processus d'approvisionnement se base sur un contrat ?</i>	<b>Oui</b>
12	Est-ce que l'activité de <<commande>> manipule une <<ressource concrète>> ?	<i>Est-ce que la commande manipule une représentation du disque ?</i>	<b>Oui</b>
13	Est-ce que les <<ressources >> sont tangibles ou non ?	<i>Est-ce que les disques sont tangibles ?</i>	<b>Oui</b>
14	Est-ce que les <<ressources >> sont en quantité limitée ?	<i>Est-ce que les disques sont en quantité limitée ?</i>	<b>Oui</b>
15	Est-ce que les <<ressources>> sont périssables ou non ?	<i>Est-ce que les disques sont périssables ?</i>	<b>Non</b>
16	Est-ce que l'activité de <<commande>> nécessite une durée déterminée ?	<i>Est-ce que la commande des disques nécessite une durée déterminée ?</i>	<b>Non</b>

**Tableau 4.2 : Tableau d'interprétation de questions spécialisées**

Rappelons que les réponses aux questions vont influencer les différentes vues qui représentent un processus. La spécialisation du modèle organisationnel se fait en combinant les réponses dont les numéro sont 1, 2, 3, 4, 5. Le modèle généré par cette combinaison est à la Figure 4-14. La spécialisation du modèle fonctionnel se fait en combinant les réponses 6,7,8,9,10,11,13,14,15. Le modèle généré est à la Figure 4-15. La spécialisation du modèle comportemental se base sur les réponses des questions 1,2,3,4,5,11,13,14,15,16. Nous n'allons pas appliquer la spécialisation de tout le modèle comportemental pour toutes les réponses. Nous allons traiter les réponses 1 et 11 pour montrer comment le modèle comportemental réagit (Figure 4-16). Pour le modèle informationnel, il faut rappeler qu'il est influencée par différentes questions qui proviennent des autres vues surtout les questions 6,7,8,9,10,11,12,13,14,15, étant donné qu'il traite tous les concepts du processus d'une façon statique (Figure 4-17).



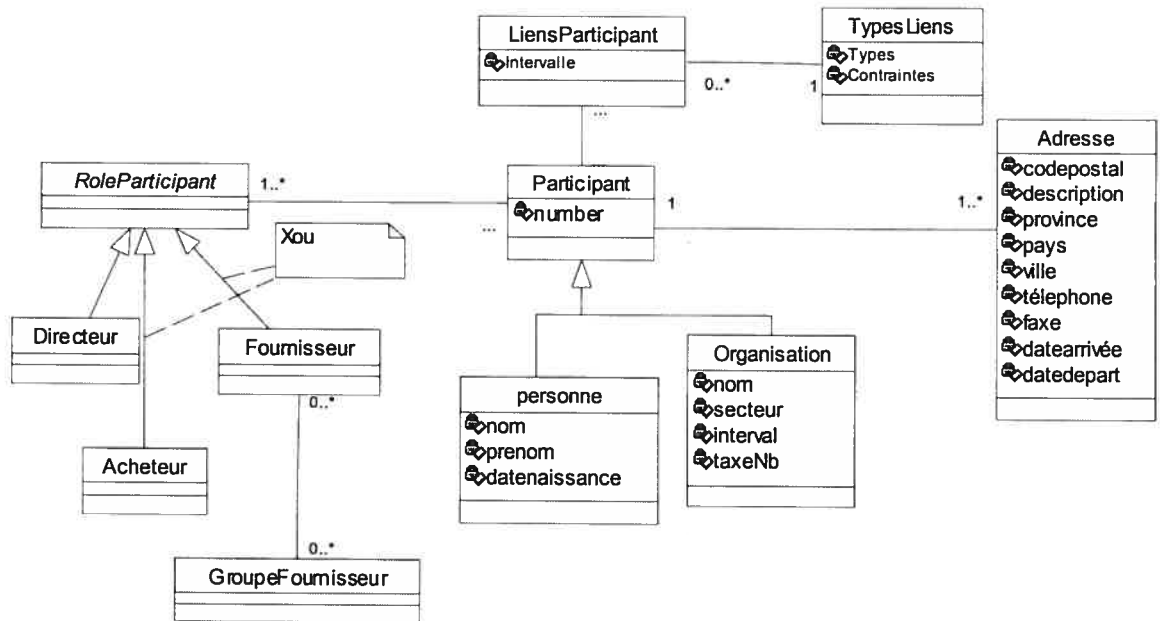


Figure 4-14 : Modèle organisationnel du magasin de disques

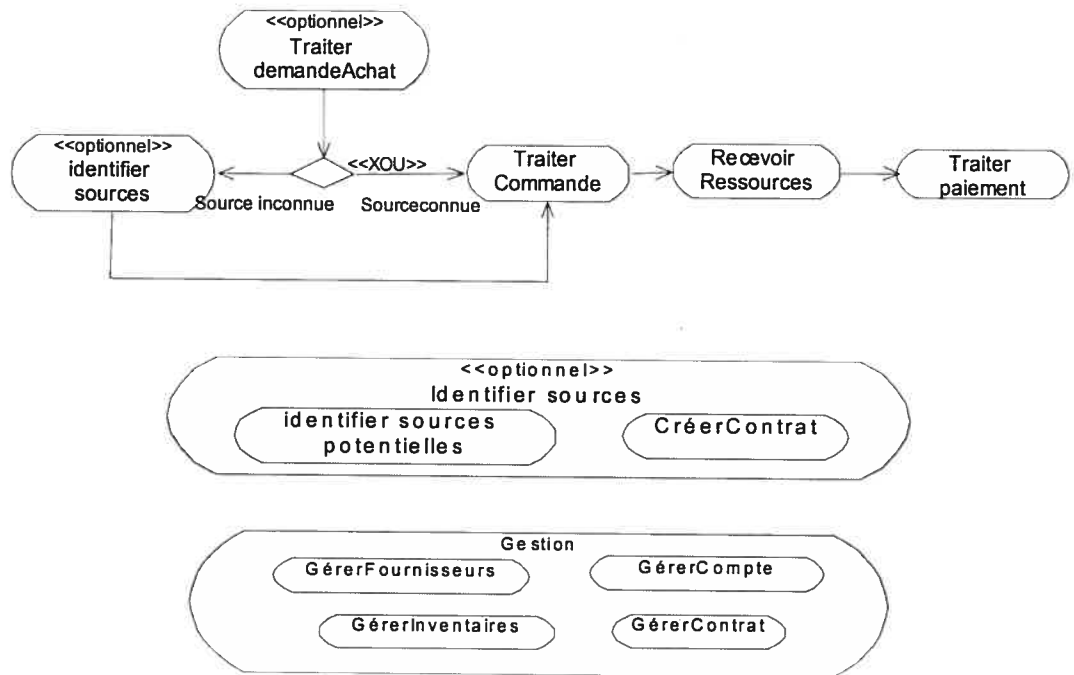


Figure 4-15 : Modèle fonctionnel du processus d'approvisionnement d'un magasin de disques

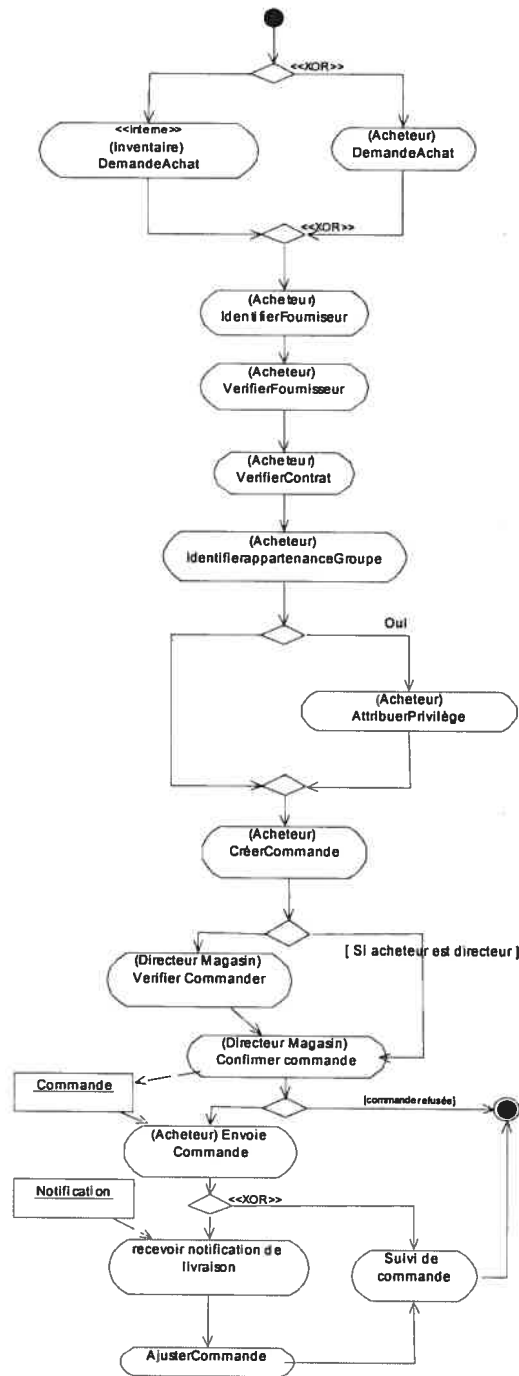
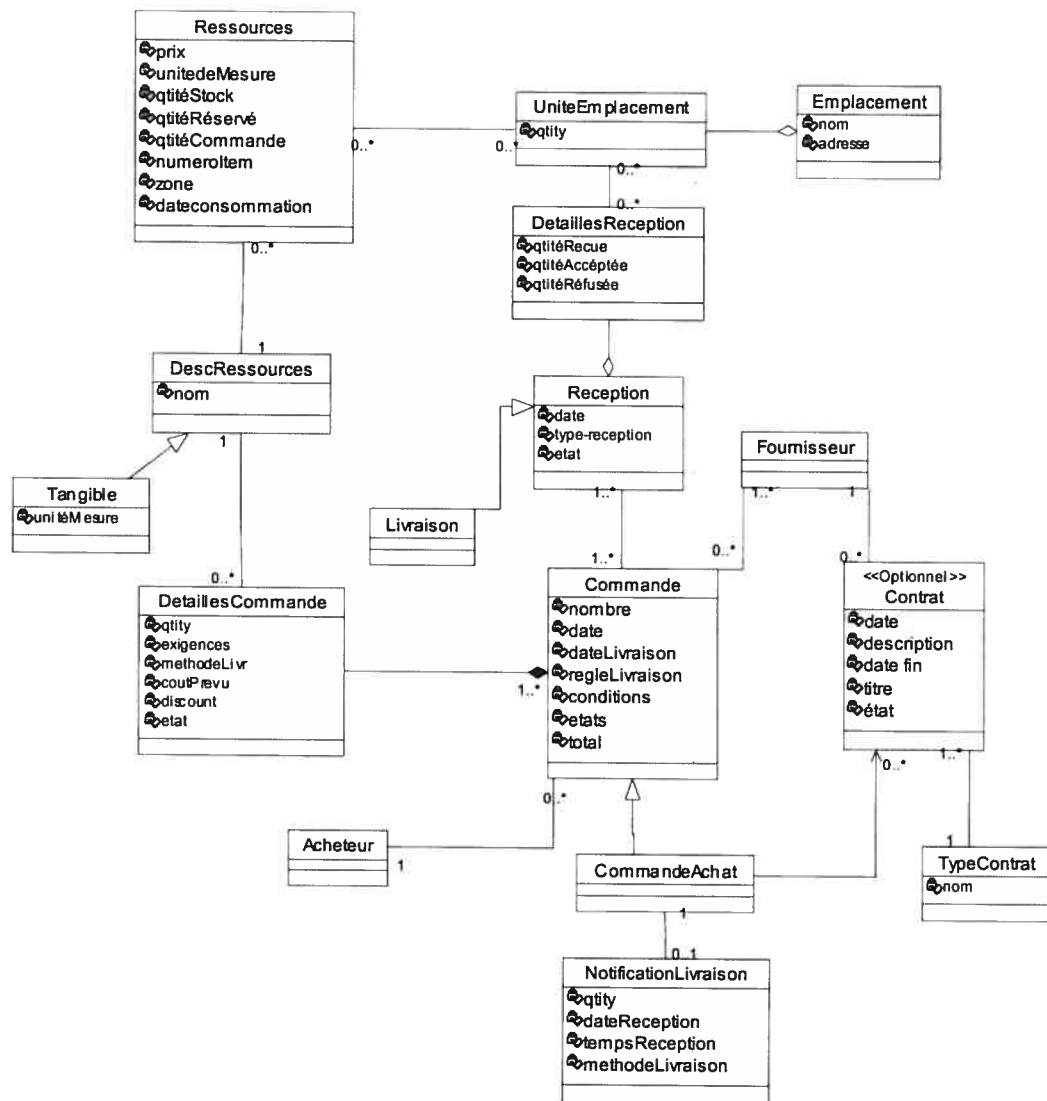


Figure 4-16 : modèle comportemental du processus d'approvisionnement d'un magasin de disques



*Figure 4-17 : modèle informationnel du processus d'approvisionnement d'un magasin de disques*

### 4.6.3 Instanciation

Le modèle obtenu des étapes précédentes est un modèle initial d'analyse. Le concepteur va par la suite l'instancier, le raffiner et l'enrichir dépendamment des préoccupations de l'entreprise pour avoir un modèle d'analyse adéquat à ses besoins. Comme nous l'avons déjà indiqué, cette étape n'est pas couverte dans notre travail. Cependant, nous suggérons un ensemble de mécanismes qui peuvent être utilisés dont:

- a) *La spécialisation*: Ce mécanisme permet de créer des sous-types qui spécialisent des types abstraits. La spécialisation peut être effectuée en modifiant les définitions existantes.
- b) *L'extension*: Ce mécanisme permet d'ajouter au modèle des concepts qui sont pertinents pour sa complétude.
- c) *La substitution*: Ce mécanisme permet de remplacer un élément par un autre. Nous pouvons remplacer un paramètre par une constante, ou renommer les éléments par des définitions correspondantes.
- d) *L'omission et la sélection*: Ces mécanismes permettent d'éliminer les éléments non pertinents.

Exemple: remplacer le produit par ( disque) et unité emplacement par (étage, section, etc.)

## 4.7 Conclusion

L'utilité des modèles génériques est de permettre d'une part leur réutilisations et leur spécialisation dépendamment d'un domaine d'affaire, et d'autre part de fournir un outil de comparaison et d'amélioration pour les entreprises qui utilisent déjà des modèles spécifiques. Avec les modèles génériques, nous avons identifié le besoin d'une classification descriptive, d'une représentation et d'un mécanisme de spécialisation de ces modèles. Pour la classification descriptive nous avons repris des attributs de différentes ontologies identifiées au chapitre 3 surtout celles de l'initiative MIT. Nous avons représenté le processus d'affaires générique d'une façon structurelle avec quatre vues. Pour chaque vue, une notation a été utilisée pour illustrer la variabilité qui peut exister dans un modèle générique. Cette notation n'est pas formelle, elle a été utilisée pour montrer le besoin d'avoir des concepts qui représentent la variation. Pour la tâche de spécialisation, nous avons proposé une approche par question. Cette approche permet la génération de modèles spécialisés dépendamment des domaines d'affaires. Un ensemble de questions génériques liées aux concepts du méta-modèle du processus d'affaires a été suggéré. Finalement, le modèle spécifique résultant de cette méthodologie constitue une base du modèle d'analyse du domaine de l'entreprise que nous modélisons.

# Chapitre 5. Conclusion

## 5.1 Résumé et conclusions

Notre étude visait la génération des modèles d'analyses spécifiques au domaine de systèmes d'informations, à partir des modèles de processus d'affaires génériques. Pour aboutir à cet objectif, idéalement, nous devions disposer d'une notation pour représenter les processus d'affaires. Nous devions avoir un réservoir de modèles génériques des processus et des composants logiciels qui les supportent. Ainsi, une classification adéquate pour faciliter la recherche et la sélection était nécessaire. De plus, pour réussir le passage d'un modèle de processus générique à des modèles spécifiques, nous avons besoin de mécanismes d'extension et d'instantiation qui permettent la personnalisation d'une façon transparente de ces modèles génériques.

Un ensemble de langages qui permet de représenter des processus d'affaires provenant de différentes traditions scientifiques a été étudié. Ces langages couvrent différentes facettes d'un processus d'affaires, comme ils peuvent se compléter pour décrire un processus. Vu la convergence des efforts de modélisation vers le langage UML, nous avons établi, qu'il est le plus adéquat pour décrire un processus d'affaires, grâce à sa notation reconnue et à ses mécanismes d'extensions.

Afin d'identifier des modèles génériques de processus d'affaires candidats aux réutilisations et les composants logiciels qui les supportent, de choisir une classification de ces modèles et de trouver un mécanisme qui permet le passage d'un composant générique à un autre spécifique, nous avons passé en revue un ensemble d'initiatives. Ces initiatives sont utiles pour la mise en oeuvre d'une approche systématique d'ingénierie des systèmes d'information par réutilisation. Ces différentes initiatives ont été évaluées principalement selon leur façon de classer les modèles partiels ou totaux des processus d'affaires. De même l'évaluation est faite au niveau des mécanismes d'adaptations suggérés par ces initiatives. Selon nos objectifs, nous avons constaté qu'aucune de ces initiatives n'est complète.

Nous avons proposé une nouvelle approche qui se base sur la classification fournie par le manuel MIT mais en effectuant certains amendements. Nous avons représenté le modèle générique des

processus d'affaires en quatre vues: comportementale, organisationnelle, fonctionnelle et informationnelle. Nous avons suggéré une approche par questions comme mécanisme de spécialisation et de navigation dans le reposoir des modèles. Cette approche permet d'associer aux concepts du méta-modèle qui représentent un processus d'affaires un ensemble de questions génériques. Ces questions doivent être spécialisées suivant le domaine d'affaires pour un modèle générique d'affaires. Dépendamment des réponses fournies par les utilisateurs aux différentes questions spécifiées, un modèle spécifique au domaine sera sélectionné ou généré. Le modèle spécialisé constitue un premier modèle d'analyse qui va être spécifié par le concepteur pour former le modèle d'analyse de l'application d'affaires visée. Une étude de cas, qui représente un processus d'approvisionnement générique et ses questions spécialisées, a été présentée en partie, pour prouver notre approche. Il faut noter que la spécialisation des modèles génériques est plus ou moins facile dans les différentes vues, surtout qu'il existe des questions qui peuvent influencer différentes vues à la fois et que les vues peuvent être dépendantes l'une de l'autre. L'application de notre approche reste à prouver sur d'autres domaines d'affaires et la recherche exploratoire est à poursuivre

Finalement, les initiatives de catalogage de composants logiciels réutilisables étudiées dans notre recherche n'établissent *pas* de liens entre processus et composants logiciels de façon *systématique*. Nous croyons cependant qu'il est possible de générer totalement ou partiellement des modèles d'analyse à partir des modèles de processus. Cet objectif soulève plusieurs problèmes de recherche que nous avons esquissés, mais qui n'ont pas été abordés dans ce travail

## 5.2 Directions futures

L'approche présentée constitue un pas de plus vers l'élaboration des systèmes d'information par réutilisation des composants d'affaires génériques. Cependant, certains points restent soit à étudier soit à compléter:

### 1. *Problèmes méthodologiques :*

- Identifier les mécanismes qui permettent de passer du modèle spécifique d'un domaine d'application à un modèle d'analyse. En effet, le modèle informationnel est assez proche du modèle objet, il reste à regarder les modèles fonctionnels et comportementaux.
- Il faut avoir une approche standardisée pour modéliser un processus réutilisable. En effet, dans notre recherche, nous avons identifié des modèles candidats à la réutilisation en étudiant des initiatives existantes; cependant, les modèles identifiés ne couvrent pas tous les domaines. Alors, il faut pouvoir modéliser et analyser des domaines d'affaires par une

méthodologie qui permet de peupler le reposoir. Prenons, comme exemple, l'approche d'ebXML qui utilise une méthode standardisée UMM pour identifier les processus d'affaires publics.

**2. Validation d'autres domaines et cas :**

- Essayer d'appliquer notre méthodologie dans différents domaines pour pouvoir vérifier si nos questions couvrent les besoins.
- Le développement d'un outil qui supporte notre méthodologie.

**3. Documentation et explication :**

- En utilisant des modèles génériques de notre reposoir, et en les adaptant, il sera essentiel d'associer aux différents modèles spécifiques des informations qui expliquent ce qui distingue un modèle d'un autre. Pour distinguer les différentes solutions d'un même problème, nous proposons de contextualiser chaque solution en décrivant la situation pour laquelle elle a été développée. Ainsi, le contexte trace l'ensemble des décisions de conception (sélection de règles de gestion ou de stratégies) ayant conduit à cette solution particulière. Pour ce faire une méthode pour représenter le contexte est nécessaire [Garner, 1999]
- Établir des directives de réutilisation pour chaque modèle générique. Ces directives vont permettre d'indiquer la finalité d'un modèle ainsi que les situations dans lesquelles sa réutilisation est pertinente. Ces directives peuvent prendre des formes diverses, allant de simples indications à de véritables spécifications d'opérations de manipulation particulières.
- Des dictionnaires, permettant d'indiquer la sémantique des termes d'affaires aux différents utilisateurs (prenons le cas de RosettaNet, et ebXML) seront utiles dans le futur. Dans ces dictionnaires, il faut utiliser une nomenclature pour assurer la cohérence.

## Bibliographie

- [Aalst, 1998] van der Aalst W.M.P., "The Application of Petri Nets to Workflow Management", Journal of Circuits, Systems and Computers, vol. 8(1), 21-66, 1998.
- [Aalst, 1999] van der Aalst W.M.P., "Formalization and Verification of Event-driven Process Chains," Information and Software technology, vol. 41 no. 10, 1999, pp. 639-650.
- [Andrews et al., 2003] Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, Aivana Trickovic, Sanjiva Weerawarana, Business Process Execution Language for Web Services (BPEL4WS) version 1.1 , 5 May 2003.
- [Arsanjani, 1999] Arsanjani A., Service provider: A domain pattern and its business framework implementation. In Proceedings of Pattern Languages of Programs (PLoP'99), 1999.
- [Basu, 2000] Basu, A. and Blanning, R., A Formal Approach to Workflow Analysis. Information Systems Research, 11(1):17-36, March 2000.
- [Bézivin, 2002] J.Bézivin, X.Blanc, Promesses et Interrogations de l'Approche MDA, Développeur Référence, septembre 2002. <<<http://www.devreference.net>>>.
- [BPMI, 2003] <http://www.bpmi.org/>, révisé le 10 Juin 2004.
- [BPML, 2003] Business Process Modeling Language, Business Process Management Institute, January 24, 2003.
- [Brassard, 2002] Michel Brassard and Mario Cardinal, Addressing Problems with Model Driven Architecture, Developer.com, , May 2002.
- [Bohrer, 1998] K. Bohrer , Architecture of the San Francisco frameworks, IBM System Journal Volume 37, Number 2, 1998 << <http://www.research.ibm.com/journal/sj/372/bohrer.html>>>.
- [Bosilj, 2001] Vesna Bosilj-Vuksic, Vlatka Hlupic: Petri nets and IDEF diagrams: applicability and efficacy for business process modelling. Informatica (Slovenia) 25(1): 2001.
- [Booch et al. 1999] G. Booch, J. Rumbaugh, I. Jacobson: "The Unified Modeling Language User Guide", Addison-Wesley, 1999, ISBN 0-201-57168-4.
- [Cardoso et al. 2002] Cardoso, J. and A. Sheth, Implementing QoS Management for Workflow Systems. Technical Report. LSDIS Lab, Department of Computer Science, University of Georgia, Athens, GA. 2002.
- [Carey 2000] James Carey, Brent Carlson and Tim Graser, SanFrancisco™ Design Patterns: Blueprints for Business Software, 2000, Addison-Wesley.
- [Carlson, 1979] W.M. Carlson, "Business Information Analysis and Integration Technique (BIAIT) - the new horizon." Data Base, 10 (4), p. 3-9, Spring 1979.
- [Garner, 1999] Garner B. et Raban R. « Context management in modeling information systems », Information and Software Technology, Vol. 41, Numero14, p.957-961, 1999.
- [Chauvet, 2002] Jean-Marie Chauvet, Services Web avec SOAP, WSDL, UDDI, ebXML., 2001, EYROLLES.



- [Clauss, 2001] M. Clauss. Modeling variability with UML. In Proceedings of the Young Researchers Workshop GCSE'01, Third International Symposium on Generative and Component-Based Software Engineering, Erfurt, Germany, September 2001.
- [Coad, 1992] P. Coad, Object Oriented Patterns, Communications of the ACM, vol. 35 n°9, September 1992.
- [Coad, 1997] Object Models: Strategies, Patterns & Applications, 2nd ed., Coad, Mayfield & North, Yourdon Press, Prentice Hall, 1997.
- [Coad et al., 1999] Coad P., Lefebvre E., De Luca J., Java Modeling in Color with UML: Enterprise Components and process, Yourdon Press, Prentice Hall, 1999.
- [Curran,1998] [2000] Curran, T., Keller, G., SAP R/3, Business Blueprint, Prentice Hall PTR, 1998, 2000 2° edition.
- [Curtis et al., 1992] Curtis, B., Kellner, M.I. and Over, J.: "Process modeling", Communications ACM, 35, (9), pp. 75- 90, 1992.
- [CWM] Data Warehousing, CWM and MOF Resource Page, <http://www.omg.org/cwm/>.
- [Davenport 1998] Davenport, T.H. Putting the enterprise into enterprise systems. *Harvard Business Review* (July–Aug. 1998), 121–131.
- [Davenport & Short, 1990] Davenport, T.H. and J. Short, The new industrial engineering: information technology and Business Process Redesign, *Sloan Management Review*, summer 1990.
- [Dayal et al., 2001] U. Dayal, M. Hsu, and R. Ladin, "Business Process Coordination: State of the Art, Trends and Open Issues," in Proceedings of the 27th Very Large Databases Conference, VLDB 2001, Roma, Italy, pp. 3-13.
- [Dussart et al., 2002] Aymeric Dussart, Benoit Aubert & Michel Patry, "An Evaluation of Inter-Organizational Workflow Modelling Formalisms," technical report 2002s-64, CIRANO (<http://www.cirano.qc.ca>) , july 2002.
- [ebXML,2003] ebXML - Enabling A Global Electronic Market Page << <http://www.ebxml.org>>>
- [ebXML, 2001] Business Process and Business Information Analysis Overview v1.0, 2001, <<<http://www.ebxml.org/specs/bpOVER.pdf>>>.
- [ebXML, 2001a] ebXML Core Components Dictionary, version 1.04, 2001.
- [ebXML, 2001b] ebXML, Business Process Specification Schema v1.01, 2001. <<[http://www.ebxml.org/specs/ebBPSS\\_print.pdf](http://www.ebxml.org/specs/ebBPSS_print.pdf)>>.
- [ebXML, 2001c] ebXML, Registry Services Specification v 2.0, 2001, <<<http://www.ebxml.org/specs/ebrs2.pdf>>>.
- [ebXML, 2001d] ebXML, Catalog of Common Business Processes v1.0, 2001, <<<http://www.ebxml.org/specs/bpPROC.pdf>>>.
- [ebXML, 2002] ebXML, Collaboration-Protocol Profile and Agreement Specification v2.0, 2002. <<<http://www.ebxml.org/specs/ebcpp-2.0.pdf>>>.

[ebXML, 2002a] ebXML, Context and Re-Usability of Core Components v1.04, 2001,  
<< <http://www.ebxml.org/specs/ebCNTXT.pdf> >>

[Ellis et al., 1984] Ellis Horowitz and John B. Munson, "An Expansive View of Reusable Software," IEEE Transactions on Software Engineering, vol. 10(5), pp. 477-487, 1984.

[Fernandez, 2000] Eduardo B. Fernández, Xiaohong Yuan: Semantic Analysis Patterns. 183-195 Conceptual Modeling - ER 2000, 19th International Conference on Conceptual Modeling, Salt Lake City, Utah, USA, October 9-12,2000, Proceedings <http://www.informatik.uni-trier.de/~ley/db/conf/er/er2000.html>.

[Fellner, 2000] Klement J. Fellner , Klaus Turowski, Classification Framework for Business Components, Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8, p.8047, January 04-07, 2000.

[Fertuck, 1992 ] Fertuck, L., "Systems Analysis and Design with CASE Tools", WCB, 1992.

[Fowler, 1997] M. Fowler, Analysis Patterns – Reusable Object Models, Addison-Wesley, 1997.

[Gale & Eldred, 1996] Thornton Gale and James Eldred, Getting Results with the Object Oriented Enterprise Model, SIGS Books, ISBN 1-884842-16-X, January 1996.

[Geerts & McCarthy, 2002] Guido Geerts and William E. McCarthy ,“An Ontological Analysis of the Primitives of the Extended-REA Enterprise Information Architecture”. The International Journal of Accounting Information Systems (2002, Vol. 3), pp. 1-16

[Georgakopoulos, 1995] Georgakopoulos, D., M. Hornick, and A. Sheth, An Overview of Workflow Management: From Process Modeling to Infrastructure for Automation. Distributed and Parallel Databases, An International Journal, 1995. 3(2):119-153.

[Glikas & Valiris, 1999] Michalis Glykas and George Valiris, “Formal methods in object-oriented business modelling,” The Journal of Systems and Software, vol 48 (1999), pp. 27-41.

[Gruhn & Wellen, 2001] Volker Gruhn and Ursula Wellen, “Analyzing a process landscape by simulation,” The Journal of Systems and Software, vol 59 (2001), pp. 333-342.

[Gulla, 2000] Jon Atle Gulla , Terje Brasethvik, On the Challenges of Business Modeling in Large-Scale Reengineering Projects, Proceedings of the 4th International Conference on Requirements Engineering (ICRE'00), p.17, June 19-23, 2000.

[Gulledge, 2002] Thomas R. Gulledge. Georg Simon. Rainer A. Sommer Using ARIS to Manage SAP Interoperability, Book Chapter, In Business Process Excellence - ARIS in Practice, Scheer, Jost, Abolhassan and Kirchmer (Editors), Springer Verlag, 2002.

[Halmans, 2003] Günter Halmans and Klaus Pohl. Communicating the variability of a software-product family to customers s. In Proceedings of the Software and Systems Modeling, volume 2, pages 15–36. Springer, February 27,2003.

[Hammer & Champy, 1993] Hammer M., and Champy, J., Reengineering the Corporation, Harper Business, 1993, New York.

[Hans, 2000] Eriksson Hans, and Penker, M.; Business Modeling with UML. John Wiley & Sons, Inc., New York, 2000.

[Haughen & McCarthy,2000] Haugen, R & McCarthy W.E., “REA, a semantic model for Internet supply chain collaboration”, OOPSLA Workshop on Business Object Components: Enterprise Application Integration, OOPSLA 2000, Minneapolis, Minnesota.

- [Hay, 1996] D.C. Hay, Data Model Patterns: Conventions of Thought, Dorset House Publishing, 1996.
- [Henn, 1998] Juergen Henn, IBM San Francisco - Object-oriented infrastructure and reusable business components for distributed, multi-platform business applications - implemented entirely in Java, Springer Verlag 1998.
- [IBM, 2003] Business Components for WebSphere Application Server - Features and benefits - IBM Software <<<http://www-306.ibm.com/software/awdtools/businesscomponents/features/index.html>>>.
- [IBM, 1998] SanFrancisco concept and facilities, SanFrancisco documentation, IBM Corporation, 1998.
- [IDEF] I-CAM Definition (IDEF) <<<http://www.idef.com/>>>
- [Ijiri, 1975] Ijiri, Y., The Theory of Accounting Measurement, American Accounting Association, Sarasota, Florida, 1975.
- [Isoda, 2001] Sadahiro Isoda, "Object-Oriented real-world modeling revisited," Journal of Systems and Software, vol. 59 (2001), pp. 153-162.
- [IWI] IWi - Institut für Wirtschaftsinformatik <<<http://www.iwi.uni-sb.de>>> consulté le 2003-12-10.
- [Jacobson,1992] Ivar Jacobson, Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Wokingham, UK, 1992.
- [Jacobsan et al.,1997] Ivar Jacobson, Martin Griss, and Patrik Jonsson. Software Reuse. Architecture, Process and Organization for Business Success. Addison-Wesley, 1997.
- [Johannesson, 1999] Paul Johannesson, Petia Wohed, The deontic pattern - a framework for domain analysis in information systems design; Data & Knowledge Engineering 31 (1999) 135±153.
- [Johannesson 2000] P. Johannesson , B. Wangler, and P. Jayaweera, Application and Process Integration - Concepts, Issues, and Research Directions, Fossil Conference, Springer, 2000.
- [Jauf, 2000] Jaufmann E., and Logan D. , The use of IBM SanFrancisco core business processes in human resources scheduling, IBM System Journal, Volume 39, Number 2, 2000.  
<< <http://www.research.ibm.com/journal/sj/392/jaufmann.html>>>
- [Keller et al., 1992] Keller, G; Nüttgens, M, Scheer, A.-W, "Semantische Prozeßmodellierung auf der Grundlage, publication of the Institut für Wirtschaftsinformatik, paper 89, Saarbrücken, 1992.
- [Kindler, 2003] Kindler, E., "On the semantics of EPCs: A framework for resolving the vicious circle" technical report, Reihe Informatik, University of Paderborn, Paderborn, Germany, August 2003.
- [Lee et al., 1996] J.Lee, M.Gruninger,Y.Jin, T.Malone, A.Tate, G. Yost, and other members of the PIF Working Group, The PIF Process Interchange Format and Framework (May 24,1996),1996, <http://ccs.mit.edu/pif8.html>.
- [Lefebvre, 1996] Lefebvre, É., 1996, Améliorer les méthodes de planification informatique: une approche pluraliste, Thèse de doctorat, Université de Grenoble II.
- [Lefebvre et al., 1999] Lefebvre, E.; Lévesque, G. , Enterprise-Component Models , in (GL98) 11èmes Journées internationales: Le Génie Logiciel & ses Applications , Paris, France , 1998 .
- [Levy, 1986] Leon S. Levy, "A Metaprogramming Method and its Economic Justification," IEEE Transactions on Software Engineering, vol. SE-12, no. 2, pp. 272-277, February 1986.

- [Malone et al.,1999] T.W. Malone, K. Crowston, J. Lee, B. Pentland, C. Dellarocas, G. Wyner, J. Quimby, C.S. Osborn, A. Bernstein, G. Herman, M. Klein, and E.O'Donnell, "Tools for inventing organizations: Toward a handbook of organizational processes," *Management Science* 45(3) pp 425-443, March, 1999.
- [Malone et al., 2003] Thomas W. Malone, Kevin Crowston, and George A. Herman, *Organizing Business Knowledge: The MIT Process Handbook*, Cambridge, MA: MIT Press, 2003.
- [Martin, 1985] James Martin, in *Fourth Generation Languages. Volume I: Principles.*, Prentice-Hall, 1985.
- [Moriconi et al., 1995] Moriconi, M., Qian, X. et Riemenschneider, R.A (1995). Correct architecture refinement. *IEEE Transactions on Software Engineering* , 21(4):356-372.
- [Massen, 2002] Thomas van der Massen and Horst Lichter. *Modeling Variability by UML Use Case Diagrams*. In *Proceedings of the International Workshop on requirements Engineering for Product Lines (REPL'02)*, pages 19–25. AVAYA labs, September 2002.
- [Mayer et al., 1995] Mayer, R.J., C. P. Menzel, M.K. Painter, P.S. deWitte, T. Blinn, and B. Parakath, *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*, Knowledge Based Systems Inc., September 1995.
- [McCarthy, 1982] William E. McCarthy. "The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment," *The Accounting Review* (July 1982) pp. 554-78.
- [McIlory, 1969] M. McIlory, "Mass-Produced Software Components", *Software Engineering Concepts and Techniques: NATO Conference on Software Engineering*, J.M. Buxton & al (eds), New York 1969.
- [MDA, 2003] MDA Guide Version 1.0.1, OMG/2003-06-01, <<<http://www.omg.org/docs/omg/03-06-01.pdf>>>, June 2003.
- [Meir, 2003 ] Meir H. Levi, *The Business Process (Quiet) Revolution; Transformation to Process Organization*, Handbook on Enterprise Architecture, 2003 ISBN 3-540-00343-6.
- [Medjahed, 2003] B. Medjahed, B. Benatallah, A. Bouguettaya, A. H. H. Ngu, and A. Elmagarmid. *Business-to-Business Interactions: Issues and Enabling Technologies*. *The VLDB Journal* , Volume 12, Number 1, May 2003.
- [MIT, 2004] <<<http://ccs.mit.edu/ph/>>>, consulté le 2004-03-10.
- [Mili et al., 1995] Hafedh Mili, Fatma Mili, Ali Mili: *Reusing Software: Issues and Research Directions*. *IEEE, Transactions on Software Engineering*, vol. 21, n°6, June 1995.
- [Mili et al., 2002] Hafedh Mili, Ali Mili, Sherif Yacoub, and Edward Addy, *Reuse-Based Software Engineering: Techniques, Organization, and Controls*, John Wiley & Sons, 2002, ISBN 0-471-39819-5.
- [MOF] Meta-Object Facility (MOF), version 1.4, <http://www.omg.org/technology/documents/formal/mof.htm>
- [Murata, 1988] Murata, T. Petri nets: Properties, analysis and applications. *Proc. IEEE* 77, 44 (Apr.), 541-580.1988.
- [Murdoch, 1998] Murdoch J, McDermid J, Astley K, Wilkinson P, "Modelling Engineering Design Processes with Role Activity Diagrams", *Integrated Design and Process Technology Conference*, Berlin, 6-9 July 1998.
- [NIST, 2002] *The Process Specification Language (PSL 2.0)*, September 2002, NIST, <http://ats.nist.gov/psl/>.

[UML-RFP, 2000a] OMG, A UML 2.0 RFP that is primarily concerned with the refinement and extension of UML 1.x semantics and notation. Document ad/00-09-02: UML 2.0 Superstructure RFP.

[UML-RFP, 2000b] OMG, A UML 2.0 RFP that is primarily concerned with specifying an OCL metamodel. Document ad/00-09-03: UML 2.0 OCL RFP.

[UML-RFP, 2000c] OMG, A UML 2.0 RFP that is primarily concerned with architectural alignment, restructuring and extension mechanisms. Document ad/00-09-01: UML 2.0 Infrastructure RFP.

[OMG, 2001] OMG, EDOC: UML Profile for Enterprise Distributed Object Computing, Document ptc/2001-12-04, December 2001.

[OMG, 2003] OMG, UML2: UML 2.0 Superstructure Final Adopted specification, Document ptc/03-08-02, September 2003.

[Ould, 1995] M. Ould, Business Processes: Modelling and Analysis for Re-engineering and Improvement, 1995, John Wiley & Sons.

[Paludo, 2000] Paludo M, Burnett R. et Jamhour E., Patterns Leveraging Analysis Reuse of Business Processes, Proceedings of the 6th International Conference on Software Reuse: Advances in Software Reusability, Lecture Notes In Computer Science, Springer-Verlag, 2000.

[Petri Nets] Petri Nets World: Online Services for the International Petri Nets Community. On-line à <<http://www.daimi.au.dk/PetriNets>>.

[Phalp, 1998] Keith Thomas Phalp, "The CAP framework for business process modeling," Information and Software Technology, vol 40 (1998) , pp. 731-744.

[Phalp & Shepperd, 2000] Keith Phalp and Martin Shepperd, "Quantitative analysis of static models of processes," The Journal of Systems and Software, vol. 52, pp. 105-112, 2000.

[Plop] Pattern Languages of Programs conference <<<http://www.hillside.net/conferences/plop.htm> >> consulté le 2003-11-25.

[Poole, 2001] John D. Poole. "Model-Driven Architecture : Vision, Standards And Emerging Technologies". ECOOP 2001, Workshop on Metamodeling and Adaptive Object Models, April 2001.

[Porter, 1985] Porter, Michael E., "Competitive Advantage". 1985, Ch. 1, pp 11-15. The Free Press. New York.

[RIG, 2003] RosettaNet Implementation Guide ( RIG) Fundamentals v 1.1, 22 January 2003.

[RosettaNet, 2003] RosettaNet , eBusiness Standards for the Global Supply Chain. < <<http://www.rosettanet.org>>>.

[Ross, 1977] Ross, D., and K. Schoman, "Structured Analysis for Requirements Definition," IEEE Trans. Software Engineering, vol. 3, no. 1, January 1977.

[Rosemann, 2001] Rosemann, M. and Shanks, G. Extension and Configuration of Reference Models for Enterprise Resource Planning Systems. Proceedings of the 12th Australasian Conference on Information Systems (ACIS 2001), eds.: G. Finnie, D. Cecez-Kecmanovic, B. Lo. Coffs Harbour, 4-7 December, 2001.

[Rosemann, 2003] M. Rosemann, W.M.P. van der Aalst, A Configurable Reference Modelling Language, Technical report, FIT-TR-2003-05, Queensland University of Technology, Brisbane, 2003.

[Rumbaugh, 1991] J. Rumbaugh, M.R. Blaha, W. Lorensen, F. Eddy, W. Premerlani, 'Object-Oriented Modeling and Design' Prentice Hall, 1991, ISBN 0-13-629841-9.

[SAP, 2003] SAP home page <<<http://www.sap.com>>>.

[SAP, 2003b] SAP Help Portal <<<http://help.sap.com>>>.

[Scheer, 2000] August-Wilhelm Scheer, Markus Nuttgens, ARIS Architecture and Reference Models for Business Process Management. Business process management 2000: 376-389.

[SCOR, 2001] Version 5.0a August, 2001 Supply Chain Council & Supply Chain Operations Reference (SCOR) Model Overview. Pittsburgh.

<http://www.supply-chain.org/slides/SCOR5.0OverviewBooklet.pdf> consulté le 14 octobre 2003.

[Shlaer et Mellor 1993 ] Shlaer, S. et Mellor, S.J. (1993). A Deeper Look at the Transition from Analysis to Design. Journal of Object-Oriented Programming, 5(9):16-21.

[Shaw et Gaines, 1983] Shaw, M.L.G. & Gaines, B.R. A computer aid to knowledge engineering. Proceedings of British Computer Society Conference on Expert Systems, 263-271 (December) 1983. International Journal of Man-Machine Studies 26(4), 435-452.

[Siegel, 2001] Developing in OMG's Model-Driven Architecture Jon Siegel and the OMG Staff Strategy Group. Revision 2.6 OMG, Novembre 2001.

[Sodhi, 1998] J. Sodhi, P. Sodhi, Software Reuse, Domain Analysis and Design process, collection Software Development, Mc Graw Hill, 1998, ISBN 0-0705-7923-7.

[Söderström 2002] Eva Söderström, Birger Andersson, Paul Johannesson, Erik Perjons, and Benkt Wangler, Towards a Framework for Comparing Process Modelling Languages, Advanced Information Systems Engineering, 14th International Conference, CAiSE 2002, Toronto, Canada, May 27-31, 2002, Proceedings. Lecture Notes in Computer Science 2348 Springer 2002, ISBN 3-540-43738-X.

[Sommerville, 1996] Ian Sommerville "Software engineering" fifth edition Addison-Wesley 1996.

[Sowa, 1984] Sowa, J.F. Conceptual Structures: Information Processing in Mind and Machine. Reading, Massachusetts, Addison-Wesley, 1984.

[Vernadat 1996] F.B. Vernadat, Enterprise Modeling and Integration: Principles and Applications, Chapman & Hall, 1996.

[WfMC,1998] Workflow Management Coalition, The Workflow Management Application Programming Interface Specification, WfMC-TC-1009 - V 2.0, July 1998.

[WfMC,1999a] Workflow Management Coalition, Interface 1: Process Definition Interchange Process Model. WfMC-TC-1016-P, Version 1.1, October 1999.

[WfMC,1999b] Workflow Management Coalition, Terminology and Glossary, Document number WfMC-TC-1011, Feb. 1999.

[WfMC, 2002a] ed. Layna Fischer, An introduction to workflow, in The Workflow Handbook 2002, the Workflow Management Coalition (2002), ISBN 0-9703509-2-9.

[WfMC,2002b] The Workflow Management Coalition, Workflow Process Definition Interface -- XML Process Definition Language, October 2002, document number TC-1025.

[Wohed, 2000] Wohed Petia, Conceptual Patterns for Reuse in Information Systems analysis, Advanced Information Systems Engineering: 12th International Conference, CAiSE 2000, Stockholm, Sweden, June 2000. Proceedings.

[Wohed, 2000b] P. Wohed, M. Bouzeghoub, Z. Kedad and E. Métais (Eds.), Conceptual Patterns- A Consolidation of Coad's and Wohed's Approaches , Proc. from the 5th International Conference on Applications of Natural Language to Information Systems NLDB'2000, LNCS1959, Springer, pp 340-351, 2000.

[Yacc ] S.C. Johnson, "Yacc: Yet Another Compiler Compiler", UNIX Programmer's Manual-Supplementary Documents, 7 Edition, AT&T Bell Laboratories, Indianapolis, USA.

[Yourdon,1989] Yourdon, E. N., Modern Structured Analysis, Prentice-Hall, 1990.

[Yu, 1976] Yu, S. C., The Structure of Accounting Theory, The University Presses of Florida, 1976.

[Zachman, 1982] John A. Zachman, "Business Systems Planning and Business Information Control Study: A Comparison". IBM Systems Journal, vol. 21, no.1, 1982 IBM Publication G321-5160.

# Annexe1: Description d'un processus d'approvisionnement

Pour élaborer les sous-processus qui constituent le scénario de base d'un système d'approvisionnement on va présenter le scénario du cycle général d'activités d'achats de SAP [SAP help]:

1. *Demandes achats*: Les besoins des ressources sont identifiés dans les services (directement ou par un autre processus (planification des besoins<sup>8</sup>, processus de fabrication<sup>9</sup>) indirectement. La demande d'achat est une demande formelle (interne) qui définit le besoin relatif à un article ou une prestation de service. Cette demande peut spécifier la source d'approvisionnement. Dans ce cas probablement un contrat d'achat avec cette source, ou bien une histoire d'achat pour cette source existent ( groupe de fournisseurs). Comme, la source peut être interne (un autre département). Si la source d'approvisionnement est identifiée nous passons directement à l'étape traitement de commande. *Les ressources peuvent être de différents types ce qui influence le processus d'approvisionnement.*
2. *Identifier les sources d'approvisionnement* : Déterminer les sources potentielles directement (manuellement ou d'une façon interactive avec un système automatisé) en fonction des commandes antérieures ou bien en utilisant d'autres techniques comme la publicité ou une partie tierce. Éventuellement, ce processus se réfère au processus gestion des fournisseurs pour ajouter ou sélectionner des fournisseurs ce processus. D'autant plus, les ressources exigées pour une demande peuvent être liées à des contrats d'achats de longs termes. Il faut noter que ce processus peut être effectué avant qu'une demande d'achat est présentée ( ex: le chef de département juge utile d'acheter un produit présenté par un fournisseur) ou bien il est fait en réponse à une demande d'achat. Habituellement, on peut avoir de fournisseurs et de contrats qui représentent les sources externes d'approvisionnement, ainsi que les unités départementales qui représentent les sources internes d'approvisionnement.

---

<sup>8</sup> Le processus planification des besoins propose des ressources à commander en fonction de la consommation, des stocks existants, des résultats des recherches appliqués pour des nouveaux produits.

<sup>9</sup> Le processus de fabrication crée automatiquement une demande d'achat s'il exige l'utilisation des pièces non gérées en stock ou s'il contient une opération de traitement externe ( par exemple sous-traitance).



3. *Traiter les appels d'offres*: Des appels d'offres peuvent être éventuellement envoyés aux fournisseurs (probablement par voie électronique). Ces appels d'offres peuvent être liés à une réquisition ou bien peuvent être entrés manuellement. La compagnie spécifie la date limite aux fournisseurs pour soumettre leurs offres. Les offres des fournisseurs conformes aux exigences de l'appel d'offre seront retenues. Une comparaison entre ces offres (prix, durée, qualité) est établie. Des lettres de refus peuvent être envoyées (automatiquement). Le fournisseur conforme aux exigences sera sélectionné. Une propriété nécessaire doit être fournie avec les appels d'offres "les principales échéances (le délai de la soumission d'offre)", "les prix" et "des conditions nécessaires" (ex: la livraison)
4. *Traitement des commandes*: Une commande d'achat peut être créée en se basant sur les informations de la demande d'achat et de l'offre (automatiquement ou manuellement). La commande d'achat contiendra des informations comme les règles de livraison, de paiement et une description détaillée des sources. La commande peut être effectuée à l'une des divisions de l'entreprise et non à un fournisseur. Une lettre de réception de commande est obtenue de la source d'approvisionnement ou il fournit la date de livraison (conforme ou ajustée). Un sous-processus "suivi commande" pour la vérification sur les jours de relance spécifiés et lancés, si nécessaire.
5. *Réception ressources*: Ce processus va s'assurer que les ressources reçues sont conformes aux exigences spécifiées dans les commandes et que n'importe quelles transactions financières ou d'inventaires sont initiés. Ce processus dépend des types de ressources. Toutefois, le personnel de réception confirme la réception des ressources.
6. *Gérer les fournisseurs et le paiement*: ce processus permet de gérer les fournisseurs et leurs factures ainsi que les différents paiements effectués pour eux.

**Des variantes possibles d'un processus d'approvisionnement sont :**

**Consignment** : Lors de l'achat de marchandises en consignation, les marchandises commandées appartiennent toujours au fournisseur, même après la livraison. Elles deviennent la propriété de l'entreprise approvisionnée uniquement lorsqu'elles sont consommées. La quantité consommée/prélevée est payée au fournisseur périodiquement, par exemple, tous les mois.

**Sous-traitance** : Dans le cas de la sous-traitance, le fournisseur (sous-traitant) reçoit les composants qui lui permettent de fabriquer un produit. Le produit est ensuite commandé par votre

entreprise par le biais d'une commande d'achat. Les composants nécessaires au fournisseur pour la fabrication du produit commandé sont indiqués dans la commande et fournis au sous-traitant. Ces composants peuvent être déterminés par le système via une nomenclature.

**Transfert physique à l'aide d'une commande de transfert :** Dans le cadre de la gestion des transferts physiques, les marchandises sont approvisionnées et fournies en interne. Une division commande les marchandises à une autre division.

**Gestion des livraisons directes :** Dans le cadre de la gestion des livraisons directes, l'entreprise passe une commande client à un fournisseur externe, lequel envoie les marchandises directement au client. La commande client n'est pas gérée par l'entreprise, mais par le fournisseur. Les postes de livraison directe peuvent être saisis dans les demandes d'achat, dans les commandes d'achat et dans les commandes client.

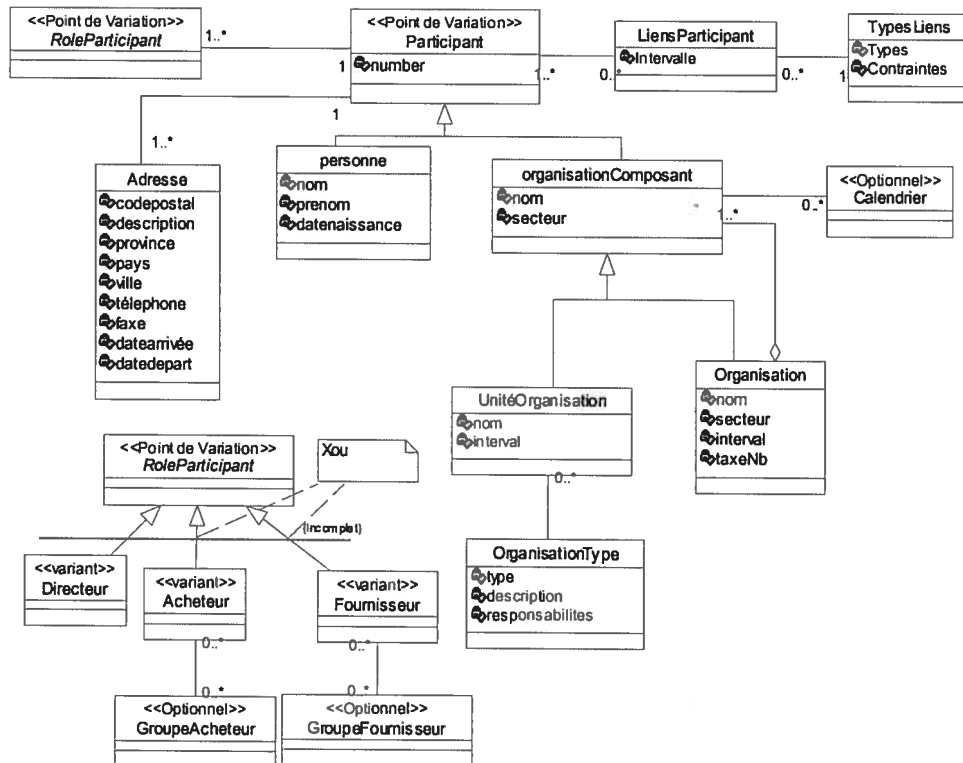
**Emballages consignés :** L'entreprise commande des marchandises à un fournisseur. Ces dernières sont livrées avec des emballages consignés (palettes, conteneurs) qui appartiennent au fournisseur et sont stockés dans les locaux du client jusqu'à leur retour chez le fournisseur.

**Gestion des articles pipeline :** Dans la gestion des articles pipeline, l'entreprise n'a pas besoin de commander ni de stocker l'article concerné. Il est immédiatement disponible à la demande via un pipeline (par exemple, l'huile ou l'eau) ou tout autre type de câble (par exemple, l'électricité). La consommation de l'article est régulièrement payée au fournisseur.

# Annexe2 : Modèles de processus d'approvisionnement

## A) Modèles organisationnels

Le modèle générique organisationnel est le suivant :



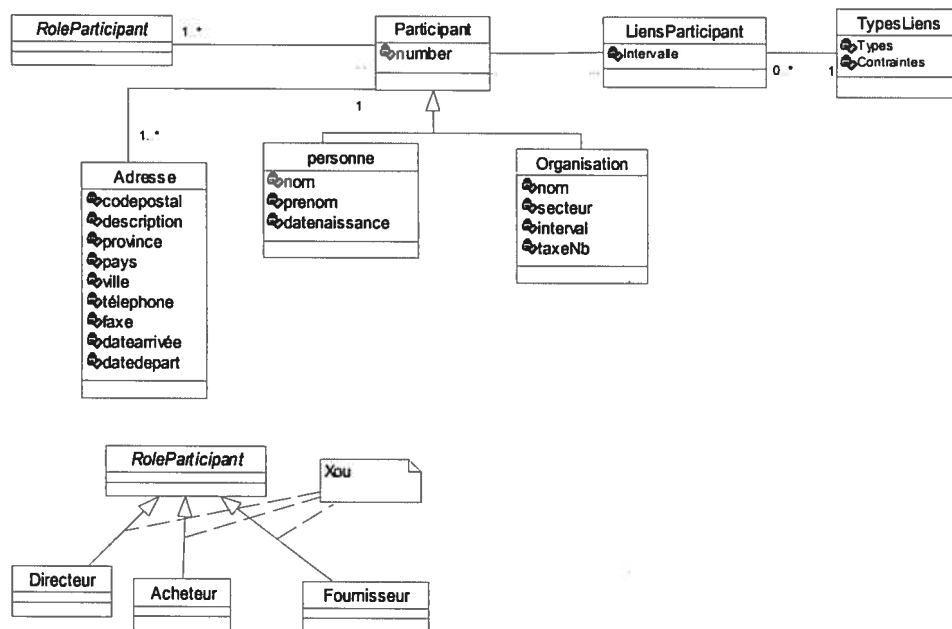
Les questions spécialisées pour un processus d'approvisionnement de point de vue organisationnel sont :

1. Est-ce que les rôles de l'acheteur (<<rôle>>) et de directeur (<<rôle>>) peuvent être joués par le même participant (<<acteur>>)?
  - a. Oui
  - b. Non
2. Est-ce que la participant (<<acteur>>) qui joue le rôle d'acheteur (<<rôle>>) est organisation?
  - a. Oui
  - b. Non

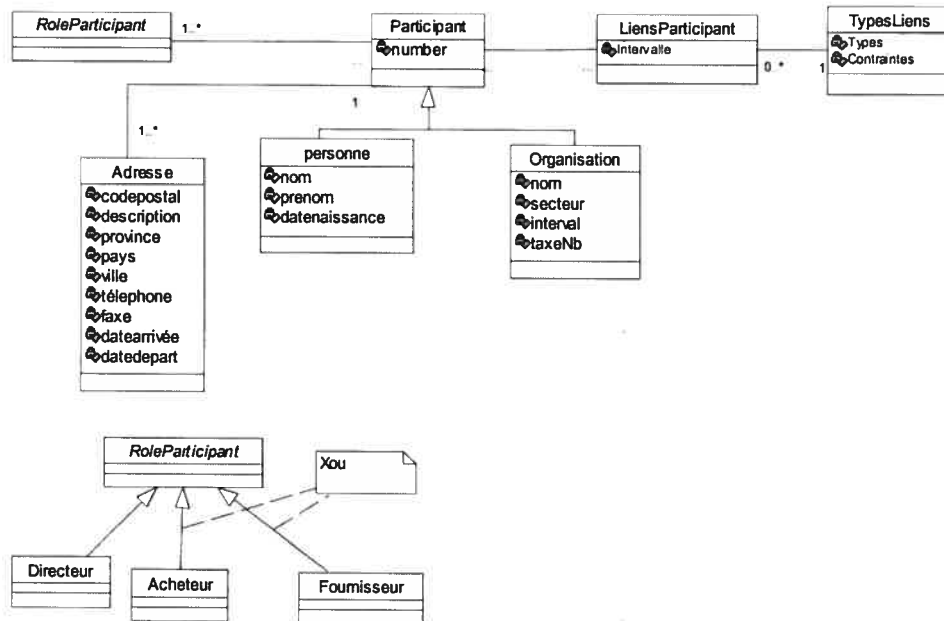
3. Est-ce que la participant (<<acteur>>) qui joue le rôle de fournisseur (<<rôle>>) est une organisation ?
  - a. Oui
  - b. Non
4. Est-ce que le fournisseur (<<rôle>>) fait partie d'un groupe de fournisseurs différenciés?
  - a. Oui
  - b. Non
5. Est-ce que l'acheteur (<<rôle>>) fait partie d'un groupe de demandeurs différenciés?
  - a. Oui
  - b. Non.

Par la suite nous trouvons quelques modèles suivant différentes combinaisons de réponses.

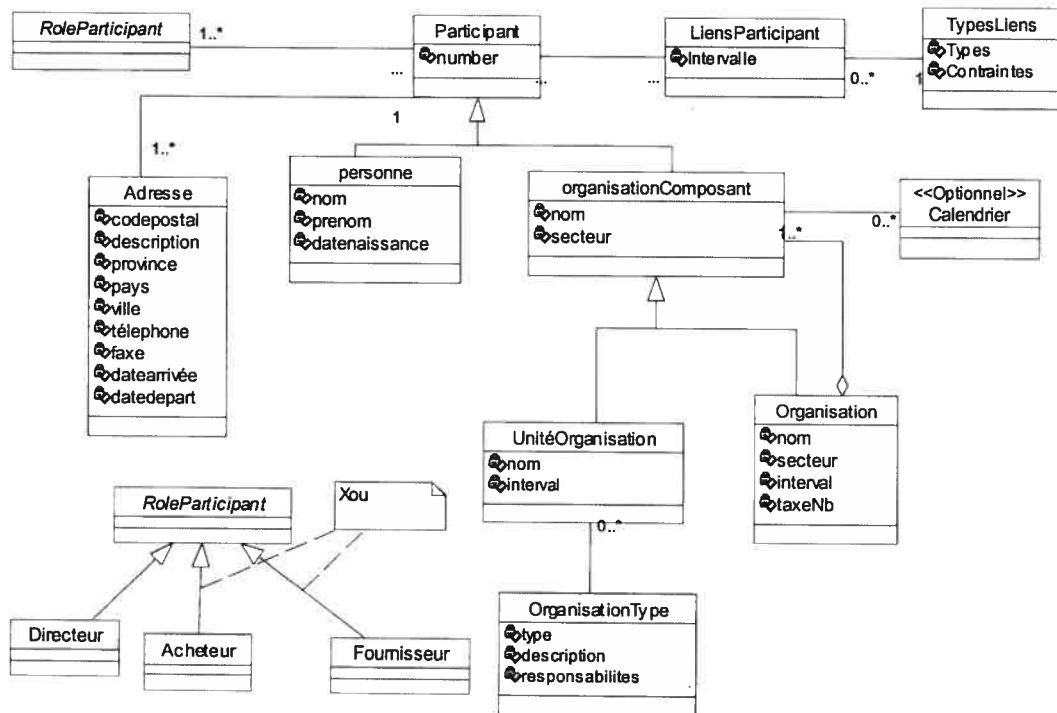
Modèle A (1b, 2b,3b, 4b, 5b)



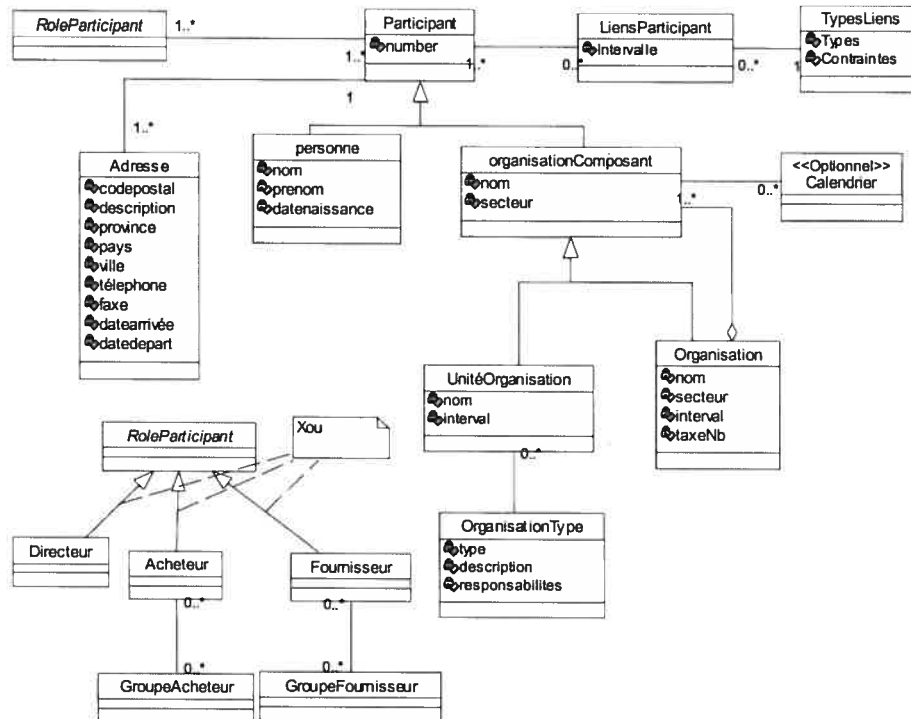
## Modèle B (1a, 2b,3b, 4b, 5b)



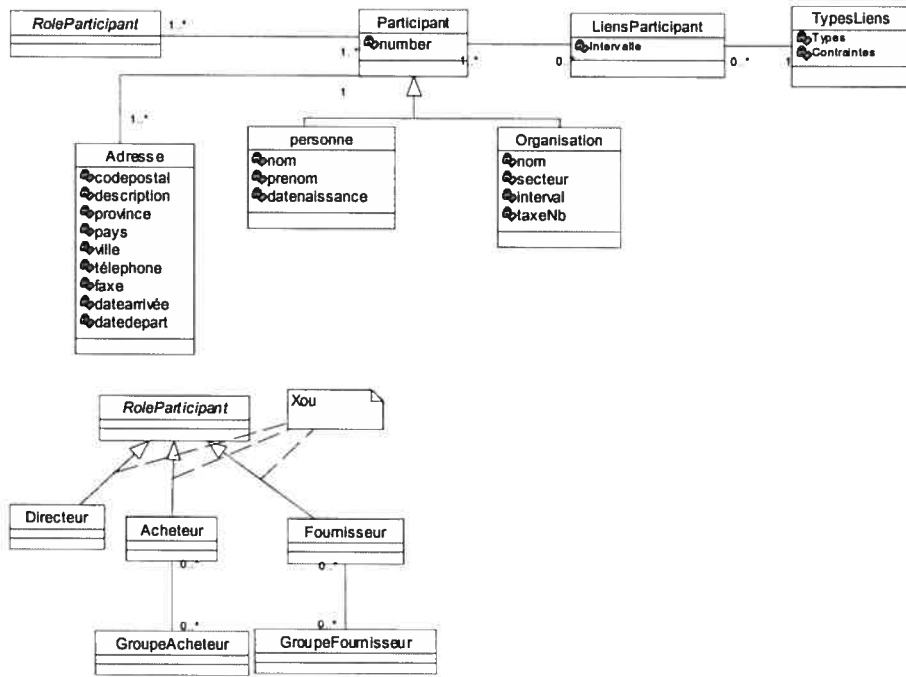
## Modèle C(1a, 2a,3b/3a, 4b, 5b)



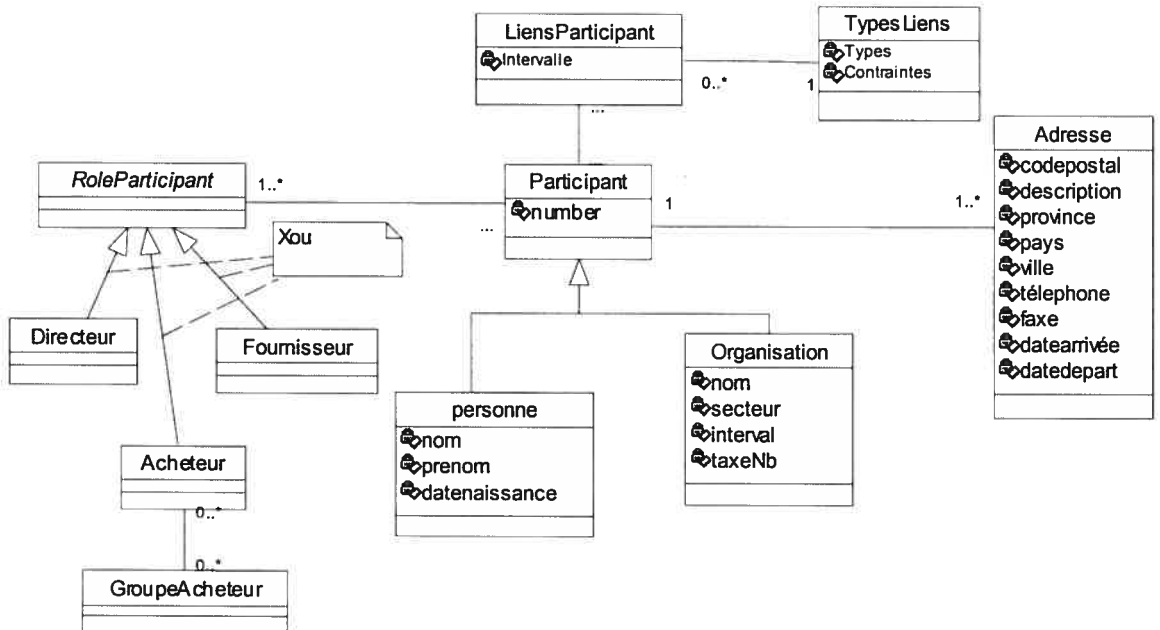
## Modèle D (1b, 2a, 3a, 4a, 5a)



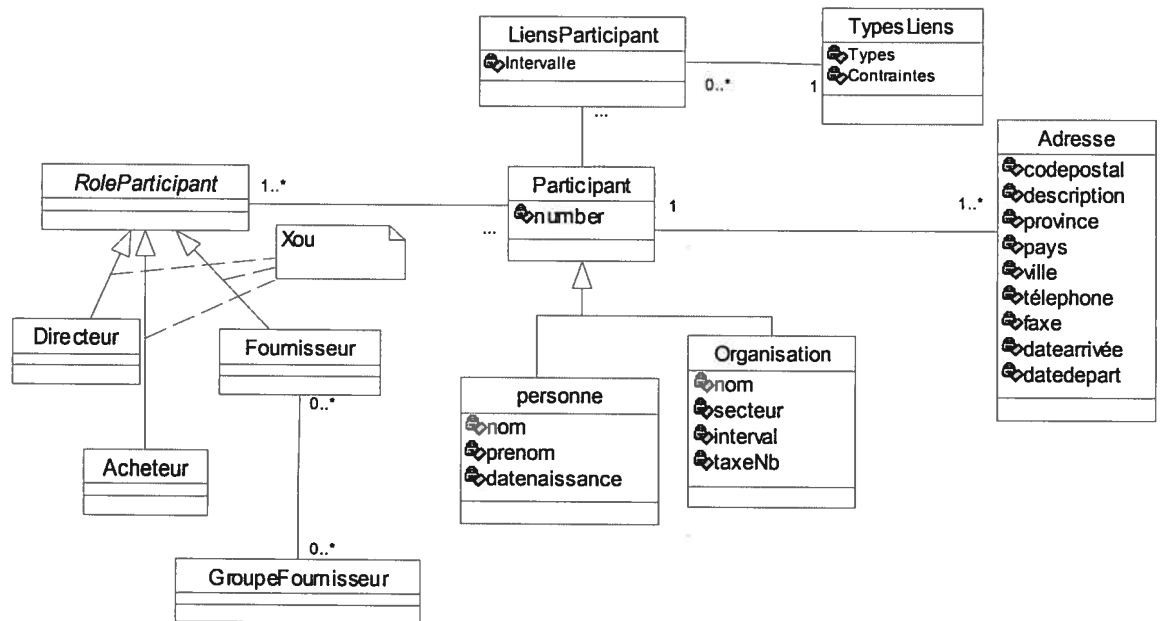
Modèle E (1b, 2b, 3a/3b, 4a, 5a)



Modèle F (1b, 2b,3b, 4b, 5a)



## Modèle G (1b, 2b,3b, 4a, 5b)

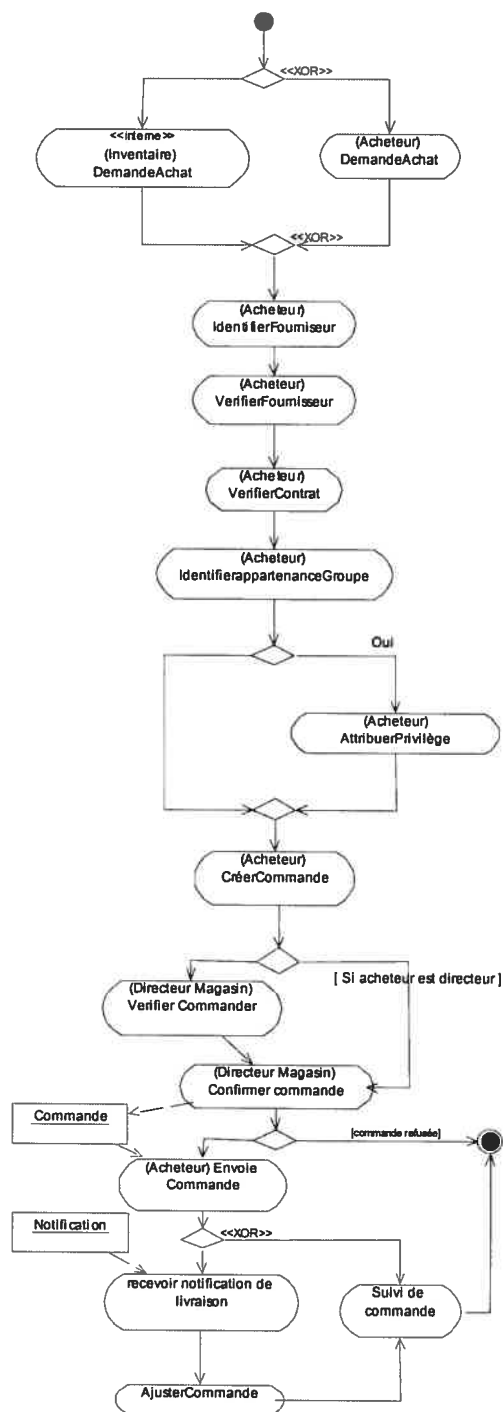
**B) Modèles comportementaux**

Les questions spécialisées pour les sous-processus traiter commande sont :

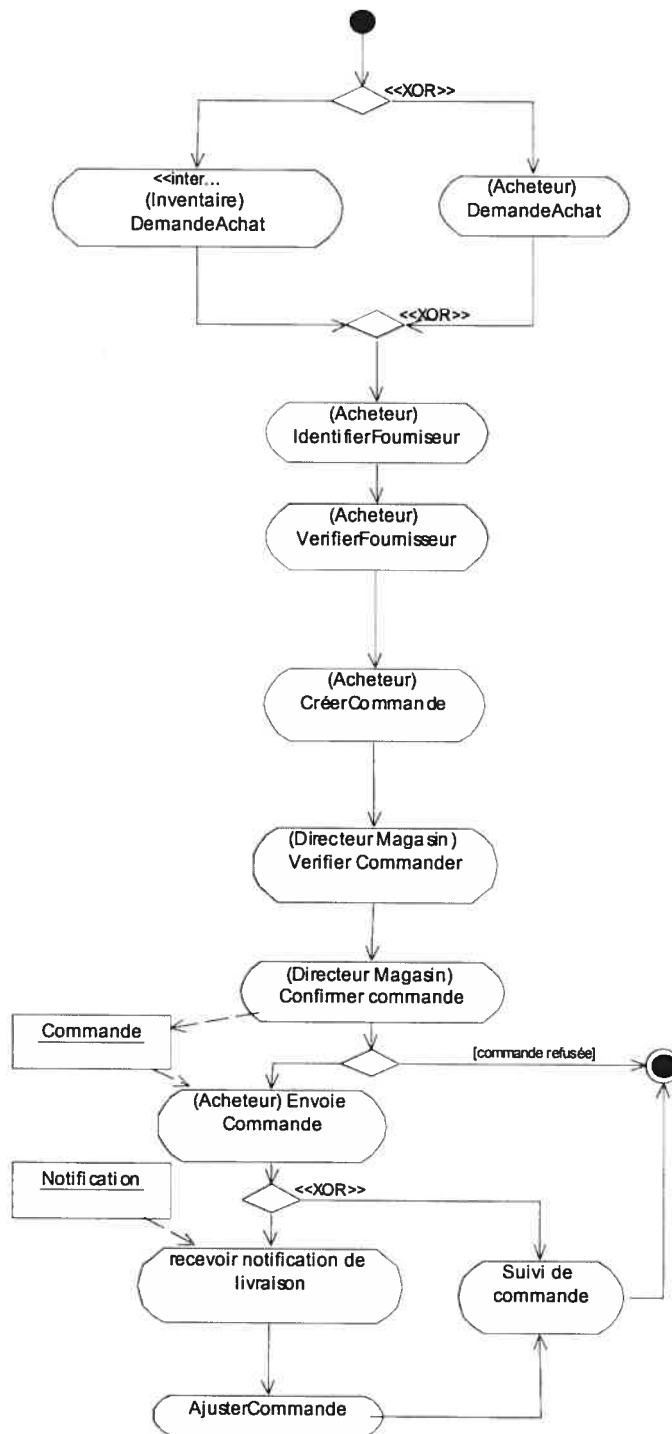
1. Est-ce que les rôles de l'acheteur (<<rôle>>) et de directeur (<<rôle>>) peuvent être joués par le même participant?
  - a. Oui
  - b. Non
2. Est-ce que le fournisseur (<<rôle>>) fait partie d'un groupe de fournisseurs différenciés?
  - a. Oui
  - b. Non
3. Est-ce que l'acheteur (<<rôle>>) fait partie d'un groupe d'acheteurs différenciés?
  - a. Oui
  - b. Non
4. Est-ce que le processus d'approvisionnement (<<processus>>) se base sur un contrat (<<ressources>>) ?
  - a. Oui
  - b. Non



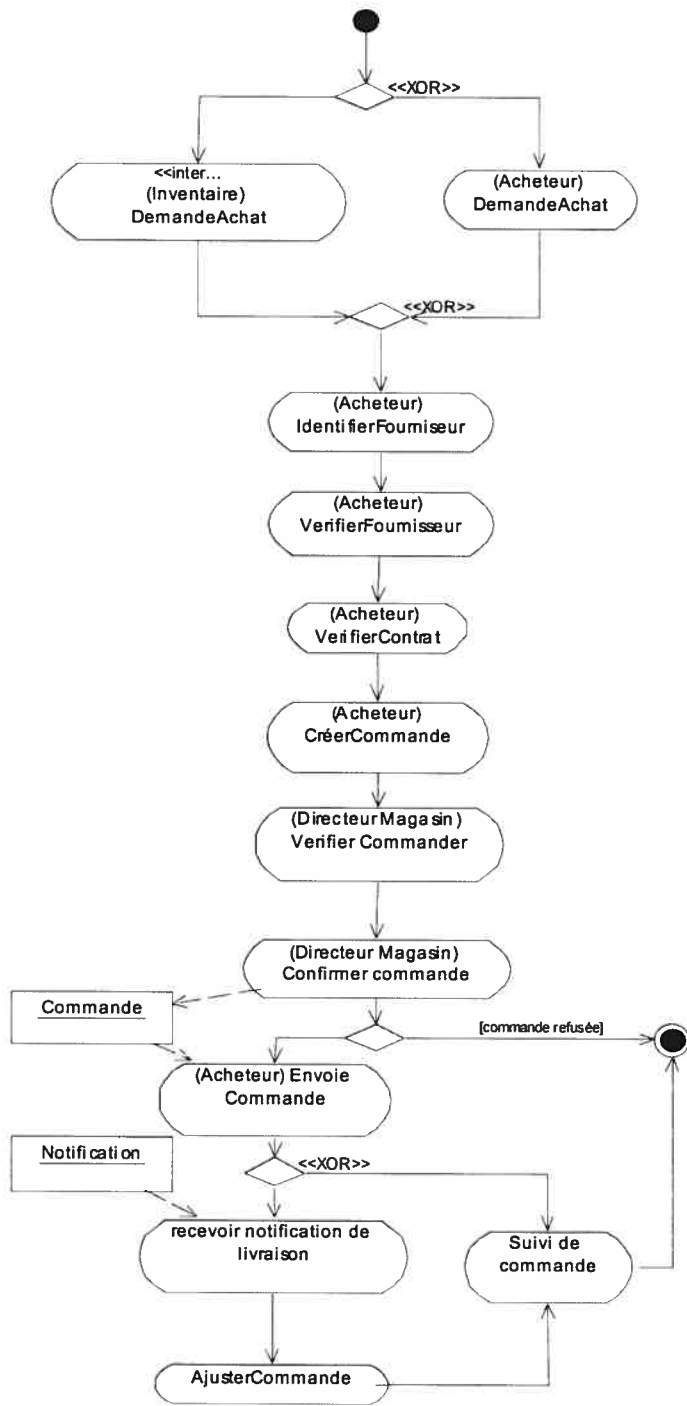
Modèle A ( 1a, 2a,3a,4a)



## Modèle B ( 1b, 2b,3b,4b)

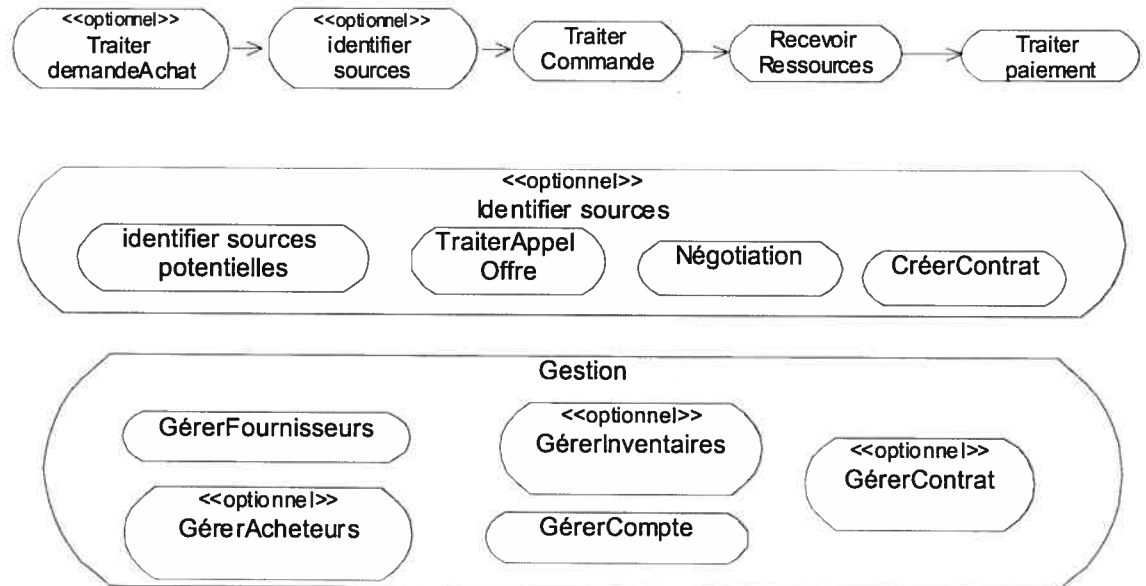


Modèle C ( 1b, 2b,3b,4a)



## C) Modèles fonctionnels

Le modèle fonctionnel générique du processus d'approvisionnement est:



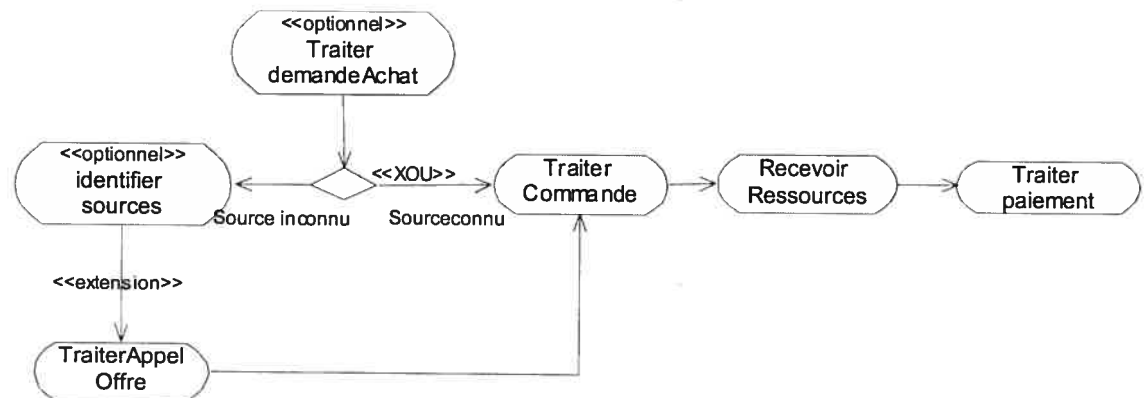
Les questions spécialisées sont :

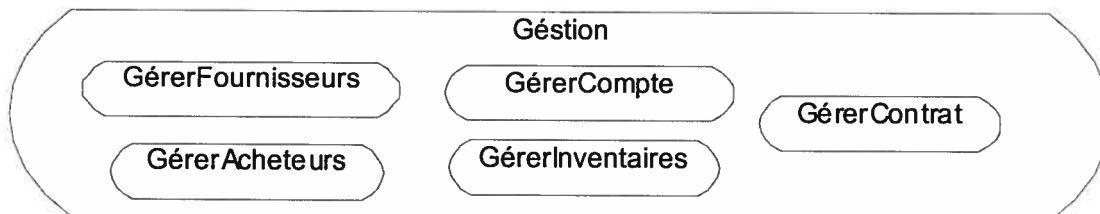
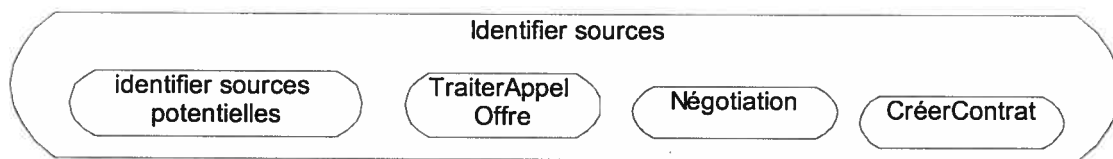
1. Est-ce que les rôles de l'acheteur (<<rôle>>) et de directeur (<<rôle>>) peuvent être joués par le même participant (<<acteur>>)?
  - a. Oui
  - b. Non
2. Est-ce que la participant (<<acteur>>) qui joue le rôle de fournisseur (<<rôle>>) est externe?
  - a. Oui
  - b. Non (interne)
3. Est-ce que la participant (<<acteur>>) qui joue le rôle d'acheteur (<<rôle>>) est externe?
  - a. Oui
  - b. Non (interne)
4. Est-ce que le processus d'approvisionnement (<< processus>>) se base sur un contrat (<<ressource>> )?
  - a. Oui
  - b. Non
5. Est-ce que les informations concernant les commandes (<< activité>>) des fournisseurs (<<rôle>>) sont mémorisées?
  - a. Oui

- b. Non
6. Est-ce que les informations concernant les réceptions (<< activité >>) sont mémorisées?
- a. Oui
- b. Non
7. Est-ce que les produits (<<ressource >>) sont tangibles ?
- a. Oui
- b. Non
8. Est-ce que les produits (<<ressource >>) sont en quantité limitée?
- a. Oui
- b. Non ( quantité illimitée)
9. Est-ce que les produits (<<ressources >>) sont périssables?
- a. Oui
- b. Non
10. Est-ce que la consommation (<< activité >>) du produit (<<ressources >>) réduit la quantité?
- a. Oui
- b. Non ( réduit la qualité)

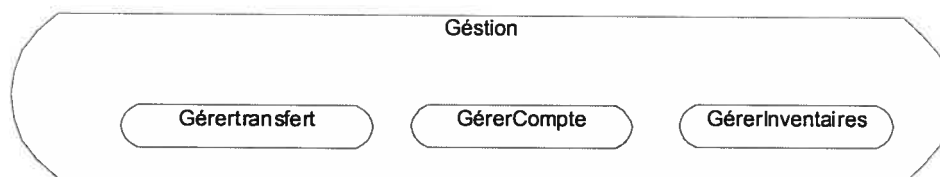
Nous allons présenter quelques combinaisons des réponses qui fournissent différents modèles. Il faut noter que parfois la négation d'une question entraîne la négation d'autres ( comme les questions 2-4). De plus, parfois la réponse positive à une question peut englober l'existence optionnelle d'une fonctionnalité ( question 4).

Modèle A (1a, 2a, 3a, 4a, 5a, 6a, 7a, 8a, 9a, 10a)

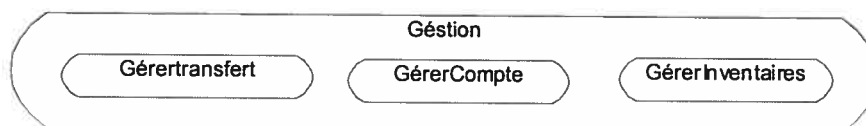




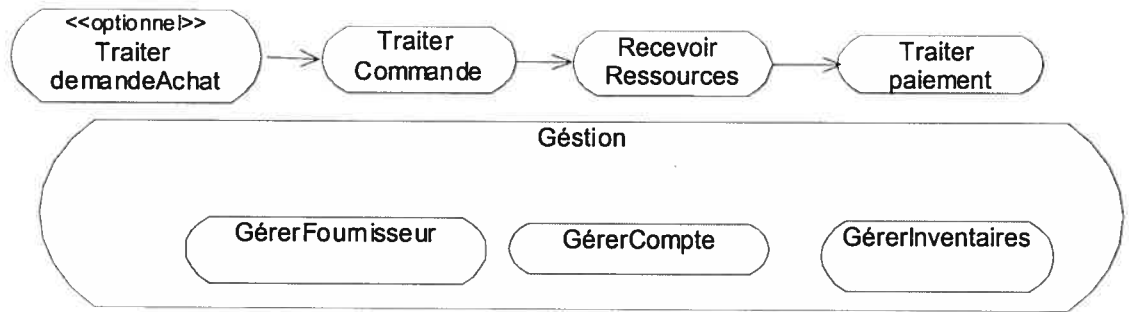
Modèle B (1b, 2b,3b, 4b, 5a, 6a, 7a, 8a, 9a, 10a)



Modèle C (1a, 2b,3b, 4b, 5a, 6a, 7a, 8a, 9a, 10a)

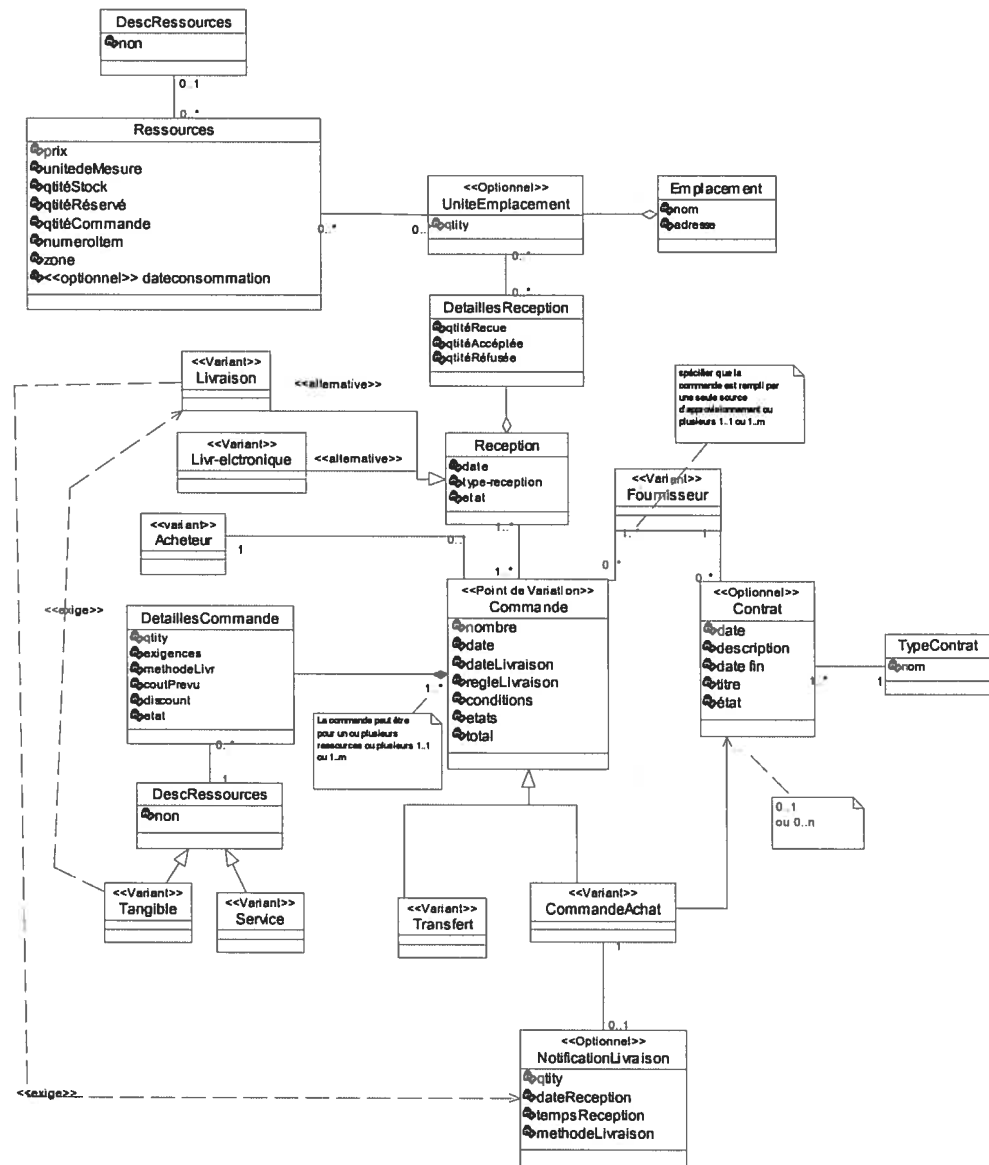


Modèle D (1a, 2a,3b, 4b, 5a, 6a, 7a, 8a, 9a, 10a)



## D) Modèles informationnels

Le modèle générique des sous-processus traiter commande et livraison est le suivant :



Les questions spécialisées pour ces sous processus sont :

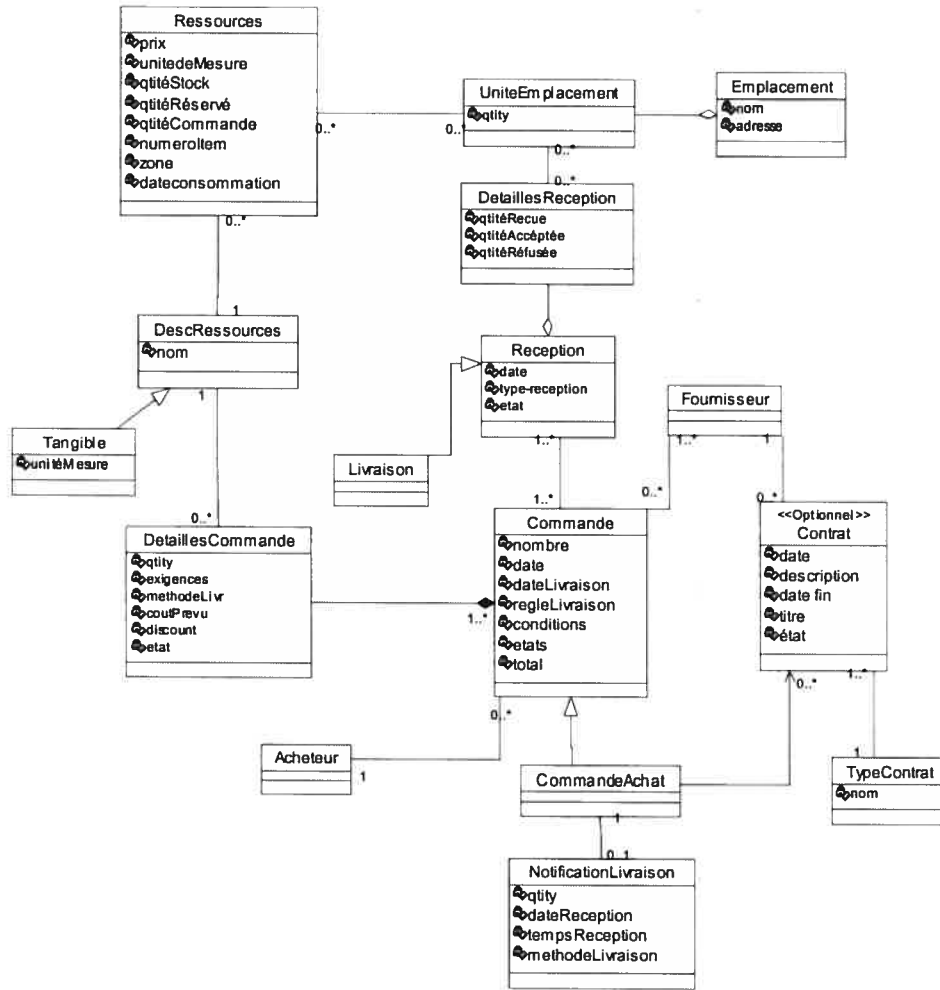
1. Est-ce que la participant (<<acteur>>) qui joue le rôle de fournisseur (<<rôle>>) est externe?
  - a. Oui
  - b. Non
2. Est-ce que la livraison (<<sous-processus>>) est liée à un emplacement (<<emplacement>>)?



- a. Oui
  - b. Non
3. Est-ce que la commande (<< activité>>) manipule une ressource concrète (<<ressource>>)?
    - a. Oui
    - b. Non
  4. Est-ce que la livraison (<<sous-processus>>) manipule une ressource concrète (<<ressource>>)?
    - a. Oui
    - b. Non
  5. Est-ce que le processus d'approvisionnement (<<activité>>) se base sur un contrat (<<ressource>>)?
    - c. Oui
    - d. Non
  6. Est-ce que les produits (<<ressources >>) sont tangibles ou non?
    - a. Oui
    - b. Non
  7. Est-ce que les produits (<<ressources >>) sont en quantité limitée ou illimitée?
    - a. Oui
    - b. Non
  8. Est-ce que les informations concernant les commandes (<<activité>>) des fournisseurs (<<rôle>>) sont mémorisées?
    - a. Oui
    - b. Non
  9. Est-ce que les informations concernant les livraisons (<<sous-processus>>) sont mémorisées?
    - a. Oui
    - b. Non
  10. Est-ce que les produits (<<ressource>>) sont périssables ou non?
    - a. Oui
    - b. Non

Quelques modèles génériques spécialisés comme résultats aux questions sont :

Modèle A (1a, 2a, 3a, 4a, 5a, 6a, 7a, 8a, 9a, 10a)



## Modèle b (1b, 2a, 3a, 4a, 5a, 6a, 7a, 8a, 9a, 10a)

