



ASAP approche orientée : services pour un support agile et flexible des processus de conception de produit dans les systèmes PLM

Safa Hachani

► **To cite this version:**

Safa Hachani. ASAP approche orientée : services pour un support agile et flexible des processus de conception de produit dans les systèmes PLM. Autre. Université Grenoble Alpes, 2013. Français. <NNT : 2013GRENI094>. <tel-01204845>

HAL Id: tel-01204845

<https://tel.archives-ouvertes.fr/tel-01204845>

Submitted on 24 Sep 2015

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

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

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Génie Industriel**

Arrêté ministériel : 7 août 2006

Présentée par

Safa HACHANI

Thèse dirigée par **Lilia GZARA** et **Hervé VERJUS**

préparée au sein du **Laboratoire G-SCOP**
dans l'**École Doctorale I-MEP²**

**ASAP : Approche orientée
Services pour un support Agile
et flexible des Processus de
conception de produit dans les
systèmes PLM**

Thèse soutenue publiquement le **16 Avril 2013**,
devant le jury composé de :

Mme. Dominique RIEU

Professeur des Universités, Université Pierre Mendès France
(Présidente)

Mme. Corine CAUVET

Professeur des Universités, Université Paul Cézanne, Aix-Marseille III
(Rapporteur)

M. Hervé PANETTO

Professeur des Universités, Université de Lorraine (Rapporteur)

Mme. Mireille BLAY-FORNARINO

Professeur des Universités, Université de Nice (Examineur)

M. Aziz BOURAS

Professeur des Universités, Université Lyon 2 (Examineur)

Mme. Lilia GZARA

Maître de Conférences, Grenoble INP (Directeur de thèse)

M. Hervé VERJUS

Maître de Conférences, Université de Savoie (Co-directeur de thèse)



À mes parents qui m'ont toujours soutenue
À mon mari et ma fille à qui je souhaite une excellente vie

Remerciements

Il m'est agréable d'exprimer mes sincères remerciements à celles et ceux qui ont au fil des années, joué un rôle dans la réalisation de ce travail.

Je commencerais par évoquer ma gratitude à l'égard de mes directeurs de thèse M. Hervé Verjus et Mme. Lilia Gzara pour la confiance qu'ils ont placée en moi, en m'accompagnant dans mon parcours de "jeune chercheur". Ils m'ont soutenu jusqu'au bout, en m'apportant de précieux conseils dans toutes mes démarches. Leur encadrement et suivi ont beaucoup contribué à l'aboutissement de cette thèse. Leur coaching quotidien lors des derniers jours de la rédaction, m'ont été d'une grande aide. Puissent-ils trouver ici l'expression de ma gratitude et de ma reconnaissance.

Je voudrais exprimer ma reconnaissance envers les membres du jury. Pouvoir réunir pour cette occasion des chercheurs d'un tel niveau au sein de disciplines diverses est pour moi un très grand honneur et une marque d'encouragement pour la conduite de recherches interdisciplinaires. Je tiens à remercier Mme. Corine Cauvet et M. Hervé Panetto pour l'intérêt qu'ils ont porté à mes travaux en acceptant de les rapporter. Je remercie également Mme. Mireille Blay-Fornarino et M. Aziz Bouras qui ont accepté d'examiner mon travail de thèse. Je souhaite enfin remercier Mme. Dominique Rieu pour avoir accepté de présider le jury.

Je remercie la Région Rhône-Alpes pour avoir financé, par l'intermédiaire du projet ISPRI-PLM, ce travail de thèse.

Mes remerciements vont également à tous les membres du laboratoire G-SCOP, qui m'ont permis de passer des années agréables et enrichissantes. Je ne saurais oublier les colocataires successifs du bureau F215 : Sandra, Hamed, Valérie, Hélène... ainsi qu'à Laura, Hussein, Ahmed... que je remercie pour tous les bons moments passés ensemble.

Ces années de thèse resteront ancrées dans ma mémoire comme une période très enrichissante de ma vie. Cet aboutissement dans mes études ne pouvait se réaliser sans être associé au soutien des personnes les plus proches de moi : ma famille, une valeur à laquelle j'attache énormément d'importance, et particulièrement à mes parents attentifs, et affectueux, mes deux frères et ma soeur ouafa admirables. L'affection que vous m'avez apportée durant ces années loin de vous m'a aidée à garder le moral haut, à surmonter les difficultés et à achever ma thèse dans les meilleures conditions.

Enfin, mes remerciements les plus tendres vont à celui que cette thèse n'aurait pu aboutir sans lui. Mohamed, je tiens à te dire merci d'avoir été toujours là, pour la patience et l'affection que vous m'avez manifesté durant ces années et pour votre aide morale.

Table des matières

INTRODUCTION GÉNÉRALE.....	1
Chapitre I : Gestion des processus de conception : Contexte et Problématique.....	5
1. Introduction	6
2. Processus de conception de produits manufacturés.....	6
2.1. Notion de processus.....	6
2.2. Spécificités du processus de conception de produit manufacturés	7
2.3. Exigences vis-à-vis de la gestion des processus de conception.....	8
3. Systèmes de gestion des processus de conception : Etat de l'art.....	9
3.1.Modélisation des processus de conception de produits manufacturés dans l'industrie ...	10
3.2. Outils support à l'exécution des processus de conception de produits manufacturés dans l'industrie	13
4. Approches pour la mise en œuvre de processus flexibles	16
4.1. Cadre de référence pour la flexibilité des processus	16
4.2. Approches de flexibilisation des processus	19
4.3. Analyse des approches de flexibilisation des processus.....	22
4.4. Synthèse.....	24
5. Conclusion.....	25
Chapitre II : Etat de l'art et Démarche pour la Gestion flexible des Processus de Conception de produits manufacturés	27
1. Introduction	28
2. État de l'art sur l'orientation services.....	28
2.1. L'approche orientée services	28
2.2. Les concepts principaux de l'orientation services	30
2.3. Synthèse.....	33
3. Orchestration de services.....	35

3.1.	Présentation des approches et langages d'orchestration.....	35
3.2.	Comparaison des approches et langages d'orchestration	38
3.3.	Synthèse sur les approches et langages d'orchestration	39
4.	Notre démarche pour la gestion flexible des processus de conception de produits manufacturés dans le PLM.....	40
4.1.	Objectif de la thèse	40
4.2.	Notre approche	41
4.3.	Architecture conceptuelle de notre approche	43
4.4.	Pertinence de notre approche vis-à-vis de nos objectifs.....	46
5.	Conclusion.....	47
Chapitre III :_Définition des Catalogues de Services du niveau métier, fonctionnel et technique		
		49
1.	Introduction	50
2.	Méthodes d'identification de services : État de l'art	50
2.1.	Méthodes d'identification des services existantes.....	51
2.2.	Évaluation des méthodes d'identification.....	55
3.	Démarche proposée pour l'identification de différentes catégories de services.....	57
3.1.	Identification des services du niveau métier – services de conception de produits manufacturés (SCP).....	58
3.2.	Identification des services fonctionnels PLM	70
3.3.	Catalogue de services logiciels PLM.....	77
4.	Conclusion.....	79
Chapitre IV :Usage des services: définition du processus par composition de services et alignement entre les orchestrations métier, fonctionnelle et technique.....		
		81
1.	Introduction	82
2.	Cadre pour la définition et la mise en œuvre des processus de conception de produits manufacturés	82
3.	Définition du processus de conception par composition de services	85

3.1.	Caractéristiques de notre approche de composition de services.....	86
3.2.	Éléments de modélisation du processus de conception	88
4.	Méta-modèles d’orchestration spécifiques au niveau métier, fonctionnel et technique	92
4.1.	Méta-modèle d’orchestration métier	93
4.2.	Méta-modèle d’orchestration de services fonctionnels PLM	95
4.3.	Méta-modèle d’orchestration de services logiciels PLM	97
5.	De la définition vers la mise en oeuvre du processus de conception de produits manufacturés	99
5.1.	Modèle de déploiement générique.....	99
5.2.	Déploiement métier des opérations de SCPs.....	102
5.3.	Déploiement fonctionnel des opérations de services fonctionnels PLM.....	103
5.4.	Alignement entre les différents modèles d’orchestration de notre approche.....	105
6.	Synthèse.....	113
7.	Conclusion.....	114
Chapitre V : Illustration de la démarche sur un cas d’application.....		117
1.	Introduction	118
2.	Présentation de l’étude de cas.....	118
2.1.	Définition et analyse du problème.....	119
2.2.	Résolution du problème.....	120
2.3.	Définition de la solution	121
3.	Application de la démarche d’usage de services pour la définition et la mise en oeuvre d’un processus – Itération 1	122
3.1.	Phase de modélisation du processus de conception.....	122
3.2.	Phase de déploiement métier	128
3.3.	Phase de déploiement fonctionnel	133
4.	Application de la démarche d’usage de services lors du changement d’un processus déjà défini et mis en oeuvre – Itération 2.....	133
5.	Conclusion.....	138

Conclusion Générale	141
Références Bibliographiques.....	145
Annexe A : Exemple de fiches descriptives des services de conception de produit et des services fonctionnels PLM – Version en anglais des deux catalogues de services	155
Annexe B : Fiches de correspondance Métier/Fonctionnel et les orchestrations fonctionnelles résultat de la transformation des opérations métier – cas d’application	163

Liste des figures

Figure 1 : Typologie des processus d'entreprise selon [Débauche 04].....	7
Figure 2 : Environnement du domaine de la conception.....	8
Figure 3 : Taxonomie de la flexibilité selon [Regev 06].....	18
Figure 4: Cadre d'interaction selon SOA.....	29
Figure 5 : Chorégraphie de services.....	32
Figure 6 : Orchestration de services.....	33
Figure 7 : Les différentes couches du langage PXL [Verjus 11].....	37
Figure 8: Changement de la composition dans un processus de conception de produits manufacturés.....	41
Figure 9: Processus de conception de produits manufacturés à base de composition de services.....	42
Figure 10: Réutilisation d'un processus de conception de produits manufacturés à base de service dans plusieurs systèmes PLM.....	42
Figure 11: Architecture conceptuelle de l'approche proposée.....	44
Figure 12 : Processus de conception de produits manufacturés sous forme d'orchestration de SCP.....	45
Figure 13: Alignement entre les différents niveaux de notre approche.....	46
Figure 14: Processus d'identification des services selon [Erl 05].....	53
Figure 15 : Méthode d'identification des services de haut et bas niveau [Kohlborn 09b].....	54
Figure 16 : Démarche d'identification des services du niveau métier et du niveau fonctionnel PLM.....	58
Figure 17: Croisement entre actions métier et objets métier de la conception de produits manufacturés.....	64
Figure 18 : Matrice de dépendance entre les objets métier du domaine de la conception.....	68
Figure 19 : Raffinement du groupement des opérations métier par décomposition en corps de métier.....	69
Figure 20 : Catalogue des services de conception de produits manufacturés (SCP).....	70
Figure 21 : Catégories de données PLM.....	72
Figure 22 : Les attributs d'un objet PLM.....	73
Figure 23: Structuration d'objets PLM.....	74
Figure 24: Extrait de la matrice DF.....	76
Figure 25 : Extrait de la matrice CF.....	77
Figure 26 : Catalogue de services fonctionnels PLM.....	77
Figure 27: Interrogation de système PLM en utilisant PLM services.....	78
Figure 28 : Système réel, modèle et méta-modèle.....	82
Figure 29: Aperçu de l'approche proposée.....	84
Figure 30 : Phases du cadre méthodologique pour la définition et mise en œuvre des processus de conception de produits manufacturés.....	84
Figure 31 : Processus de conception : Enchaînement entre les opérations des SCP.....	85
Figure 32 : Fréquence d'occurrence de concepts BPMN [Zur Muehlen 08a].....	89
Figure 33 : Méta-modèle d'orchestration.....	90

Figure 34 : Méta-modèle d'orchestration métier	93
Figure 35 : Formalisme de représentation des concepts du méta-modèle d'orchestration métier .	95
Figure 36 : Exemple de processus de conception défini selon le formalisme proposé	95
Figure 37 : Méta-modèle d'orchestration fonctionnelle.....	96
Figure 38 : Méta-modèle d'orchestration technique	97
Figure 39 : Alignement entre une orchestration métier et une orchestration fonctionnelle	99
Figure 40 : Alignement entre une orchestration fonctionnelle et une orchestration technique ...	100
Figure 41 : Méta-modèle générique de déploiement	101
Figure 42 : Méta-modèle de correspondance Métier/Fonctionnel.....	102
Figure 43 : Méta-modèle de correspondance Fonctionnel/Technique	104
Figure 44 : Alignement : Génération d'une seule version exécutable Vs. Génération de toutes les versions exécutables du processus.....	107
Figure 45 : Alignement : transformations entre les modèles d'orchestrations de services métier, fonctionnels et logiciels.....	109
Figure 46 : Méthode de traitement des modifications [Riviere 04].....	119
Figure 47 : Fragment d'orchestration métier «Définition Solution».....	123
Figure 48 : Fragment d'orchestration métier «Définition Périmètre et objectif du Problème» ...	123
Figure 49 : Fragment d'orchestration «Définition Devis».....	123
Figure 50 : Fragment d'orchestration «Définition Spécification Technique»	124
Figure 51 : Fragment d'orchestration «Ordonnancement du Processus De conception»	124
Figure 52 : Fragment d'orchestration «Analyse Fonctionnelle».....	124
Figure 53 : Fragment d'orchestration «Analyse Dysfonctionnelle»	124
Figure 54 : Fragment d'orchestration «Analyse Test».....	124
Figure 55 : Fragment d'orchestration «Évaluation d'une solution».....	125
Figure 56 : Fragment d'orchestration «Validation d'une Solution».....	125
Figure 57 : Fragment d'orchestration «Conception d'une Solution»	126
Figure 58 : Processus de traitement des modifications.....	127
Figure 59 : Fiche de correspondance métier/fonctionnel de l'opération ElaborerDemandeDeModification	129
Figure 60 : Fiche de correspondance métier/fonctionnel de l'opération DemanderEvaluationDemandeDeModification	130
Figure 61 : Orchestration fonctionnelle : déploiement métier du processus de traitement des modifications	131
Figure 62 : Orchestration Fonctionnelle DefinitionSolution_OF	132
Figure 63 : Changement dans le processus de traitement des modifications	134
Figure 64 : Fragment d'orchestration « Choix de composants standards ».....	134
Figure 65 : Fiche de correspondance métier/fonctionnel de l'opération ConsulterCS	135
Figure 66 : Fiche correspondance métier/fonctionnel de l'opération choisir composants standards	135
Figure 67 : Fiche correspondance métier/Fonctionnel de l'opération diffuser choix composants standards.....	136
Figure 68 : Orchestration fonctionnelle ChoixdeComposantsStandard_OF	136
Figure 69 : Changement dans le déploiement métier du processus de traitement des modifications	137

Liste des tableaux

Tableau 1 : Comparaison des langages de modélisation des processus de conception	12
Tableau 2: Comparaison des outils supports à l'exécution des processus de conception	15
Tableau 3: Critères de comparaison des approches de mise en œuvre de la flexibilité.....	19
Tableau 4: Synthèse des approches de flexibilisation vis-à-vis de la problématique.....	22
Tableau 5: Comparaison des approches d'orchestration.....	38
Tableau 6: Comparaison des méthodes d'identification de services.....	56
Tableau 7 : Objets métier du domaine de la conception de produits manufacturés	60
Tableau 8: Catégories d'activités du processus de conception.....	62
Tableau 9 : Liste d'action métier du domaine de la conception de produits manufacturés	63
Tableau 10 : Liste révisée des d'opérations métier de type Production.....	65
Tableau 11 : Liste révisée des opérations métier de type Demande.....	66
Tableau 12: Entrée / sortie génériques des opérations métier du domaine de la conception	67
Tableau 13 : Objets PLM par catégorie.....	72
Tableau 14: Extrait des entrées et des sorties des opérations fonctionnels PLM	75
Tableau 15 : Extrait de l'identification des cas d'utilisation des opérations fonctionnelles PLM .	76
Tableau 16 : Extrait de l'identification des objectifs des opérations fonctionnelles PLM	76
Tableau 17: Attributs communs à tous les éléments du méta-modèle d'orchestration	91
Tableau 18: Attributs spécifiques de l'élément OrchestrationMétier	94
Tableau 19 : Attributs spécifiques de l'élément FlotdeContrôle.....	94
Tableau 20: Attributs spécifiques de l'élément OrchestrationFonctionnelle	96
Tableau 21 : Attributs spécifiques de l'élément FlotdeContrôle.....	97
Tableau 22: Attributs spécifiques de l'élément OrchestrationTechnique	98
Tableau 23 : Attributs spécifiques de l'élément FlotdeContrôle.....	98
Tableau 24: Concepts de déploiement métier d'une opération métier.....	103
Tableau 25: Concepts de déploiement fonctionnel d'une opération fonctionnelle PLM.....	105
Tableau 26 : Avantages et inconvénients de chacune des deux démarches d'alignement.....	107
Tableau 27: Règles de transformation du modèle métier vers le modèle fonctionnel.....	110
Tableau 28: Règles de transformation du modèle fonctionnel vers le modèle technique	112
Tableau 29: Fragments d'orchestration métier composant le processus de traitement de modification.....	123
Tableau 30: Opérations de services de conception de produits manufacturés composant le processus de traitement de modification	126

Liste des Abréviations

ISPRI-PLM	Intégration par les Services des PR ocessus Industriels - Application au contexte du PLM
PLM	Product Lyfecycle Management (gestion de cycle de vie de produit en français)
GOSPI	Gestion et Organisation des Systèmes de Production et de l'Innovation
BPM	Business Process Management (ingénierie des processus métier en français)
WFM	Workflow Management (gestion de workflow en français)
OMG	Object Management Group
UML	Unified Modeling Langage
BPMI	Business Process Management Initiative
BPMN	Business Process Modeling Notation
XPDL	XML Process Definition Language
WfMC	Workflow Management Coalition
XML	Extensible Markup Language
WFMS	WorkFlow Management Systems (système de gestion de workflow en français)
GED	Gestion Electronique de Documents
SGBD	Systèmes de Gestion de Base de Donnée
SGDT	Système de Gestion de Données Techniques
CAO	Conception Assistée par Ordinateur
ERP	Enterprise Ressource Planning
PGI	Progiciels de Gestion Intégrée
YAWL	Yet Another Workflow Language
BPMS	Business Process Management Systems
SOA	Service Orienteed Architecture
AOS	Architecture Orientée Service

WSDL	Web Service Description Language
WSFL	Web Service Flow Language
WS-BPEL	Web Services Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
BPML	Business Process Management Language
PXL	Process eXtensible Language
IDM	Ingénierie Dirigée par les Modèles
MDE	Model Driven Engineering
CIM	Computer Independent Model
OM	Orchestration Métier
SCP	Service de Conception Produit
PDS	Product Design Service
PIM	Platform Independent Model
OF	Orchestration Fonctionnelle
PSM	Platform Specific Model
OT	Orchestration Technique
SM	Service Métier
SL	Service Logiciel
E/S	Entrée/Sortie
DF	Dépendance Fonctionnelle
CF	Contrainte Fonctionnelle
PLCS	Product Life Cycle Support
STEP	STandard for Exchange of Product data model
M _{PC}	Modèle de Processus de Conception
MM _{PC}	Méta-Modèle de Processus de Conception
M _F	Modèle Fonctionnel
MM _F	Méta-Modèle Fonctionnel
UD.OF	User Defined. Orchestration Fonctionnelle (orchestration fonctionnelle définie par l'utilisateur en français)
Pr.OF	Predefined. Orchestration Fonctionnelle (orchestration fonctionnelle prédéfinie en français)

INTRODUCTION GÉNÉRALE

L'intensification de la concurrence et l'évolution des technologies ont abouti à une complexification des processus de lancement de nouveaux produits. Afin de demeurer concurrentielle et être en mesure de répondre aux défis du marché actuel, les organisations se doivent d'être agiles. L'agilité de l'entreprise est définie comme *'la capacité d'une organisation à détecter les changements qui se produisent dans son environnement et à réagir efficacement à ce changement'* [Overby 05]. L'agilité implique que les organisations soient en constante recherche d'équilibre face aux contraintes de plus en plus fortes de l'environnement concurrentiel. Que ce soit d'un point de vue inter ou intra-organisationnel, leurs systèmes, supports aux activités de l'entreprises, doivent pouvoir intervenir rapidement, en adoptant notamment une approche processus et en ayant une démarche de gestion des processus métiers (Business Process Management – BPM). Van der Aalst et al. définissent le BPM comme : *'la gestion des processus métiers en utilisant des méthodes, des techniques et des logiciels pour modéliser, exécuter, contrôler et analyser les processus organisationnels en s'appuyant sur des acteurs qui peuvent être : des êtres humains, des organisations, des applications, des documents et autres sources d'information'* [Van Der Aalst 03]. L'automatisation, complète ou partielle, d'un processus métier comporte deux aspects. D'abord, la modélisation du processus (sa représentation, le plus souvent sous forme graphique). Ensuite, l'implémentation de ce modèle dans un environnement d'exécution.

Dans ce travail de thèse, nous nous intéressons au métier de la conception de produits manufacturiers ; on parle alors de processus de conception. Les processus de conception constituent un moyen pour organiser les interactions entre les acteurs d'une organisation tout au long de la phase d'ingénierie du produit. Ces processus sont marqués par leur évolutivité et leur incomplétude. Impliquant plusieurs points de vue, la conception ne peut être réalisée efficacement que par la collaboration de plusieurs acteurs (clients, fournisseurs, donneurs d'ordre, représentants marketing, Bureau d'étude, Bureau de méthode, production, achats ...) ; des acteurs dont les multiples points de vue doivent être intégrés. Chaque point de vue détient diverses exigences et contraintes (qualitatives, temporelles, financières) [Pahl et Beitz 96], qui évoluent au fil du projet de conception et qui sont souvent mal connues au début du projet.

De toute évidence, les entreprises doivent s'adapter aux changements. En contrepartie, les modèles et les paradigmes de modélisation proposés dans la littérature pour supporter les processus sont simplistes, voire incohérents avec cette nouvelle réalité opérationnelle. En conséquence, les organisations sont confrontées à plusieurs obstacles, y compris la mise en œuvre limitée de nouvelles méthodes de travail car souvent les méta-modèles sur lesquels ces processus sont fondés ont tendance à être rigides et ne sont pas en mesure de mettre en œuvre facilement les changements prévus ou imprévus. Au niveau technologique, les systèmes PLM (Product Lifecycle Management) [saaksvuori 05] fréquemment utilisés dans les entreprises industrielles pour supporter la conception, sont également peu agiles et ne permettent pas d'adapter rapidement les implémentations pour répondre aux changements survenant dans la structure du processus ; changements imposés par des contraintes internes et externes à l'entreprise.

Suite à ces deux constats, d'une part l'instabilité des processus de conception et d'autre part la rigidité des modèles de processus et de leur système support, nous identifions un besoin de mise en œuvre d'une nouvelle solution permettant l'élaboration et l'exécution de modèles de processus souples et le support flexible des changements survenant dans les processus de conception. Malgré l'importance des processus de conception dans l'entreprise (en termes de coût engagés et d'incidences sur le reste des processus industriels), le problème de rigidité des solutions support au processus de conception demeure. C'est sur ces constats que se base notre problématique de recherche. Ainsi ce travail de thèse a pour objectif de proposer une approche de gestion flexible des processus de conception dans le PLM. Cette question de flexibilité impacte nécessairement le système d'information, considéré auparavant comme un outil d'intendance au service de l'entreprise, et qui est devenu aujourd'hui au cœur de son fonctionnement ; sa flexibilité conditionne les performances de l'entreprise. En effet, les systèmes d'information actuels freinent les évolutions nécessaires pour répondre aux nouvelles orientations métier et sont perçus comme résistants aux changements. Ceci est dû au manque de flexibilité au niveau du système informatique, support au métier de l'entreprise. À l'issue des différents constats et analyses que nous avons avancés, nous pouvons conclure que les entreprises doivent avoir une réflexion approfondie non seulement sur la manière de représenter leur processus métier mais aussi sur le système informatique support à ces processus, en l'occurrence le PLM, afin d'assurer une flexibilité efficace. Nous abordons donc notre problématique selon deux aspects :

- La mise en place d'une démarche permettant d'accroître l'agilité de l'entreprise en favorisant la réutilisation et en réduisant des solutions existantes (en termes de fonctions métier, fonctions logiciels et processus métier) ;
- L'accélération du changement du point de vue technique. L'objectif est de répondre rapidement aux demandes de changement en proposant des mécanismes génériques permettant la projection du changement sur le système support et son automatisation d'une manière flexible. Ce qui permettra de réduire le temps de définition et mise en œuvre des processus.

Ce travail de thèse a été mené et s'est inscrit dans le cadre du projet régional ISPRIT-PLM¹: Intégration par les services des PProcessus Industriels - Application au contexte du PLM; projet du Cluster de recherche GOSPI² : Gestion et Organisation des Systèmes de Production et de l'Innovation. Ce projet s'intéresse aux apports des méthodologies et approches orientées services au domaine du PLM. L'intérêt pour ces approches part du fait que :

- L'alignement des infrastructures technologiques aux changements rapides et à la dynamique actuelle du business reste l'un des facteurs de succès pour les entreprises. Il garantit une certaine flexibilité et une facilité d'échange et de partage de l'information, en créant et exécutant des services qui peuvent être partagés et réutilisés le long de l'entreprise et avec ses partenaires ;
- Le PLM, domaine en plein essor, dispose de certaines spécificités techniques (traitement des données et connaissances, gestion des processus et cycles de vie, support des standards métiers, etc.) qui nécessitent une prise en compte de l'hétérogénéité des outils utilisés par les différentes filières d'ingénierie et une véritable mise en place d'une structure d'ingénierie des systèmes.

Nos travaux s'inscrivent dans ce cadre et ont pour objectif de contribuer à l'étude et à la mise en œuvre de méthodologies d'intégration et de déploiement de solutions de systèmes d'information pour le PLM plus flexibles et agiles que les architectures traditionnelles de systèmes

¹ <http://cluster-gospi.fr/spip.php?page=cadre&skell=fiche&ztable=projet&zid=32>

² <http://www.cluster-gospi.fr/>

d'informations monolithiques et rigides. Nous étudierons les différents aspects et spécificités des processus industriels, particulièrement les processus de conception de produit.

Nous pouvons décliner l'objectif de nos travaux en plusieurs sous-objectifs résumés par les points suivants :

- En quoi les nouvelles propositions améliorent-elles l'agilité de l'entreprise ?
- Où on est-on du côté standardisation (OMG, OASIS...) ?
- Comment décliner les processus métiers en services ?
- Comment décliner ces services en solutions informatiques (SOA) ?

Le premier sous-objectif propose une réflexion sur les améliorations possibles et les impacts sur l'organisation. Le deuxième sous-objectif vise à proposer un état de l'art préalable sur la dimension service et standardisation au niveau PLM afin de mieux cerner les conditions instrumentales nécessaires pour la mise en place d'une solution au problème d'agilité des processus de conception dans le PLM. Les deux derniers sous-objectifs proposent une réflexion autour des méthodologies permettant de définir les processus métier de l'entreprise sous forme de services et comment les traduire en solution informatique.

Ce manuscrit est constitué de cinq chapitres.

Le **chapitre I** définit le contexte dans lequel se situe notre travail. Il trace l'espace dans lequel évolue notre travail de recherche (domaine de conception de produit) et les systèmes de gestion des processus utilisés dans ce domaine (PLM, Workflow, Groupware, etc.). Ceci conduit à cerner la problématique liée aux processus dans le domaine de conception de produit et les limites des systèmes et approches de gestion de ces processus. La deuxième partie de ce chapitre examine les approches de flexibilisation des processus en général et conduit à dégager les moyens qui permettent d'approcher la problématique, en l'occurrence les approches orientées services.

Le **chapitre II** a comme objectif de développer les concepts liés aux approches orientées services (*i.e* service et composition de services) et de décliner leurs usage vis-à-vis de notre problématique. Ceci nous permettra de présenter l'architecture globale de notre approche de gestion flexible de processus de conception. La réalisation de cette approche nécessite trois étapes : d'abord une analyse du domaine pour l'identification des services nécessaires pour la définition et la mise en œuvre des processus de conception de produit. Ensuite, la définition des processus sous forme d'orchestration de services. Enfin, la déclinaison du processus en solution automatisable.

Le **chapitre III** décrit les catalogues de services nécessaires à la mise en place de notre approche et leurs méthodes d'identification.

Le **chapitre IV** décrit la démarche d'usage des services identifiés. Il présente les méta-modèles permettant la représentation du processus sous forme d'orchestration de services (au niveau de la définition et de l'exécution). Il détaille par la suite l'approche d'alignement proposée pour automatiser et généraliser le passage de la définition du processus vers sa mise en œuvre.

Le **chapitre V** illustre l'expérimentation de l'approche sur un cas d'application et montre comment nos propositions répondent à notre problématique.

La conclusion et les perspectives clôturent ce manuscrit de thèse en présentant un bilan du travail effectué et un ensemble de perspectives liées à la poursuite de ce travail.

Chapitre I :

Gestion des processus de conception : Contexte et Problématique

1. Introduction

Ce chapitre a pour objectif de situer le contexte de nos travaux. Après avoir présenté les processus de conception, leurs spécificités, et les exigences vis-à-vis de leur système support (§2), la suite de ce chapitre donne un aperçu de la façon avec laquelle ces processus sont gérés actuellement (§3). Les exigences et périmètres du système de gestion nécessaires au regard des spécificités des processus de conception seront d'abord définis (§3.1). Nous étudions ensuite l'état actuel des systèmes de gestion ; d'abord en matière de modélisation de processus (§3.2) et ensuite en matière d'exécution (§3.3). Cette étude nous permet d'analyser le potentiel et les manques des systèmes existants vis-à-vis des exigences attendues en matière de gestion des processus de conception de produits manufacturés.

La dernière partie de ce chapitre est consacrée aux approches offrant un support à la flexibilité pour les processus (§4). Un parcours de la littérature sur la taxonomie de la flexibilité permettra d'abord de préciser le besoin en termes de flexibilité pour les processus de conception de produits manufacturés (§4.1). Ensuite, nous examinerons les approches existantes permettant d'apporter de la flexibilité au processus (§4.2). Nous ferons une analyse de ces approches par rapport aux critères de flexibilité données en section (§4.1) et des exigences des processus de conception données en section (§2.3). Cette analyse permettra de dégager leurs limites et de pouvoir trouver une alternative à la problématique d'agilité et de flexibilité liée aux processus de conception de produits manufacturés (§4.3).

2. Processus de conception de produits manufacturés

Dans ce qui suit nous rappelons ce qu'est un processus et nous présentons les différents types de processus d'entreprise afin de situer les processus de conception dans cette typologie. Nous faisons par la suite un parcours du métier de la conception de produits manufacturés afin de présenter les spécificités de ces processus.

2.1. Notion de processus

Comme il sera fait largement appel au concept de processus dans ce manuscrit, nous rappelons dès à présent cette notion. Différentes définitions de processus sont proposées dans la littérature. Parmi les définitions existantes, nous citons celle de [Thérout 02]: *un processus est défini comme un enchaînement partiellement ordonné d'exécution d'activités qui, à l'aide de moyens techniques et humains, transforme des éléments d'entrée en éléments de sortie en vue de réaliser un objectif dans le cadre d'une stratégie donnée*. Les définitions proposées par [Vernadat 07] et [Morley 07] vont dans le même sens en rajoutant la notion d'événement qui conditionne l'exécution du processus. Selon [Vernadat 07] *A process is a partially ordered sequence of steps executed to perform some enterprise goals. Process execution is triggered by one or more event occurrences*. La définition de [Morley 07] *defini un processus d'entreprise comme un ensemble d'activités, entreprises dans un objectif déterminé. La responsabilité d'exécution de tout ou partie des activités par un acteur correspond à un rôle. Le déroulement du processus utilise des ressources et peut être conditionné par des événements d'origine interne ou externe. L'agencement des activités correspond à la structure du processus*.

Dans la littérature, on trouve différentes classifications des processus ; les processus de pilotage, les processus opérationnels et les processus de support [Débauche 04] (cf. Figure 1). Selon cette classification, les processus de conception, sujet de notre étude, font partie des processus opérationnels. Tels qu'ils sont définis, les processus opérationnels ont pour fonction d'accomplir une mission dans un domaine donné et utilisent plusieurs fonctions de l'entreprise afin de créer de la valeur pour le client. La norme AFNOR X50-127 [FDX50-127 02] définit le processus de conception de la façon suivante : partant des besoins exprimés, le processus de conception définit pas à pas le

produit qui doit répondre aux besoins et aux attentes par des choix successifs portant sur des points de plus en plus détaillés. Nous définissons un processus de conception de produits manufacturés de la manière suivante : *un processus de conception de produits manufacturés représente les activités du métier de la conception de produits manufacturés. Il est ainsi considéré comme une structuration des activités de ce domaine, organisées dans le temps et dans l'espace.* Le terme structuration signifie qu'un processus peut être modélisé. Dans ce qui suit nous faisons un parcours du domaine de la conception de produits manufacturés pour mieux comprendre les spécificités de ces processus et les exigences vis-à-vis de leur gestion.

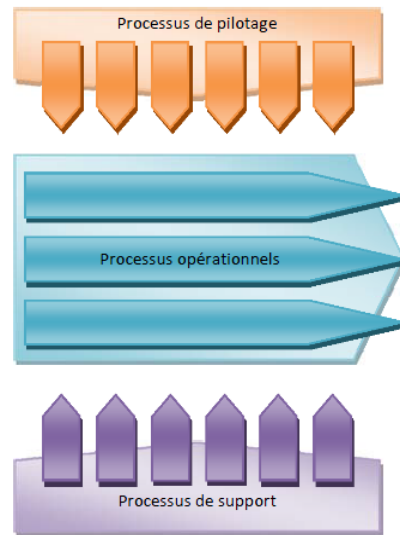


Figure 1 : Typologie des processus d'entreprise selon [Débauche 04]

2.2. Spécificités du processus de conception de produit manufacturés

Le terme de métier est couramment utilisé dans le domaine industriel. Le métier d'une entreprise représente un ensemble d'activités d'un domaine donné nécessitant des compétences et savoirs-faire des acteurs de l'entreprise [Ulmer 11]. Une deuxième définition proposée par [Martin 04] affirme que la réalisation des activités collectives dans un domaine particulier, impliquant différents acteurs partageant un but commun, est la définition même du métier d'une entreprise. Dans la suite de ce mémoire, nous considérons la notion métier, restreinte au domaine de la conception, de la manière suivante : *le métier de la conception regroupe l'ensemble des savoirs et savoirs-faire mobilisés lors des activités liées à la conception des produits manufacturés.*

Ce domaine de conception est devenu très complexe [Larsson 05]. En effet, la transformation de l'environnement économique au cours de ces dernières années a poussé le secteur industriel vers l'accroissement de la compétitivité. Les évolutions technologiques, les contraintes économiques et la concurrence mondiale poussent les entreprises à développer de nouveaux produits innovants, avec une meilleure qualité, avec des coûts de production maîtrisés et des délais réduits. Cette concurrence touche aussi l'organisation du travail dans une volonté de réduire les coûts bien que les produits deviennent de plus en plus complexes. L'entreprise étend donc son réseau de fournisseurs pour avoir un plus grand panel de prix. En même temps, le produit doit répondre au besoin de qualité exigée par le client [Ulrich 00]. La vision traditionnelle de celle-ci, était considérée comme une pratique individuelle, est révolue. La conception est ainsi devenue une pratique complexe, impliquant plusieurs points de vue et ne peut être réalisée efficacement que par la collaboration de plusieurs acteurs. En effet, les acteurs de la conception proviennent de plusieurs disciplines et possèdent des cultures et

expertises différentes. Le processus de conception nécessite l'intégration de leur multiples points de vue (cognitif, technique, social, économique, organisationnel, etc.). Chaque point de vue détient un ensemble d'exigences et de contraintes [Pahl et Beitz 96], qui peuvent évoluer et qui sont souvent mal connues au début du projet. De plus, grâce à la thèse avancée par [Simon 89], toute personne qui crée une nouvelle chose (idée de produit, etc.) ou un objet est considérée comme un acteur de la conception. Simon affirme à travers l'approche de traitement d'information symbolique [Simon 73], que la résolution d'un problème part du fait que la définition du problème est incomplète. Ainsi, il faut bien structurer le besoin avec le client, considéré aussi comme acteur de la conception en se basant sur sa vision, afin de pouvoir le satisfaire. Quand le besoin est bien défini, grâce aux échanges avec le client, les spécifications sont transmises aux concepteurs et la conception commence dès que possible. Toutefois, même si le BE et le client échangent dès le début afin de bien définir le besoin, les exigences du client peuvent évoluer au cours du temps ce qui se passe pratiquement dans la plupart des projets. De plus, le projet de conception d'un produit complexe (tel un véhicule) est lui-même complexe car son organisation se construit graduellement en fonction des informations et des contraintes très variées, évolutives et incertaines qui sont traitées progressivement par les acteurs du projet. Le domaine de la conception regroupe donc plusieurs types d'acteurs et obéit à diverses contraintes. La Figure 2 montre l'environnement dans lequel se déroule le processus de conception de produits manufacturés.

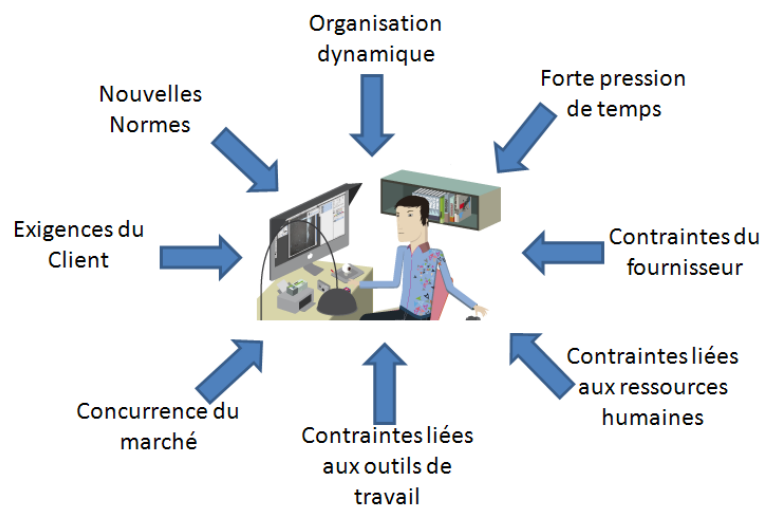


Figure 2 : Environnement du domaine de la conception

2.3. Exigences vis-à-vis de la gestion des processus de conception

Du fait de la complexité du domaine de la conception et de la diversité des acteurs impliqués, la gestion des processus de conception est soumise à plusieurs exigences :

- les processus de conception possèdent une structure complexe. Il faut pouvoir les représenter entièrement. De plus, ils sont réalisés par des acteurs métier. Ils doivent donc être bien définis et clairement compris par ces acteurs métier. Cela nécessite d'avoir des *formalismes de représentations simples et expressifs* ;
- ils sont collaboratifs et nécessitent des acteurs de métiers différents. Ces acteurs peuvent avoir des contraintes et donc ne pas être disponibles au bon moment. Un *outil de communication et de synchronisation* est essentiel pour ces processus ;
- ils sont tendus, la concurrence du marché induit une forte pression pour pouvoir mettre des produits sur le marché dans les plus brefs délais et avec des coûts réduits. *La capacité de*

réutilisation et d'automatisation des tâches répétitives et de l'enchaînement des activités est une caractéristique essentielle des outils de coordination de ces processus (permet de réduire les délais et de décharger la ressource responsable de la coordination) ;

- ils sont évolutifs, l'ordre d'exécution des activités qui les composent peut être initialement inconnu ou partiellement connu. Ils impliquent la nécessité de prendre en compte les aléas et les changements qui peuvent survenir. La capacité d'intégration de *la flexibilité est donc une caractéristique essentielle de ces processus et de leurs outils de coordination*.

Pour synthétiser, nous constatons que la gestion des processus de conception est soumise à des exigences liées aux formalismes de modélisation de ces processus (*i.e.* besoin en termes de formalismes simples et expressifs) et d'autres liés au support à l'exécution des processus (*i.e.* besoin en termes d'outil de communication et synchronisation, de réutilisation et automatisation de la collaboration et de flexibilité de l'outil). Face à cette complexité, des standards permettant la modélisation des processus ont vu le jour. De même, des solutions permettant de faciliter la collaboration entre les différents acteurs de la conception et de garantir le bon déroulement du processus ont vu le jour. En effet, la coordination des activités du processus à travers l'automatisation de l'enchaînement des activités, l'association des données aux activités du processus, la génération automatique des documents, l'automatisation de la notification du début d'une activité et la diffusion de documents et de données, permet de prendre en compte l'ensemble des exigences liées au support à l'exécution des processus. Cela offre plusieurs avantages à savoir :

- limiter les saisies inutiles de données et fournir l'information à la bonne personne au bon moment ;
- limiter le transfert physique de papier et les erreurs d'envoi des mauvaises versions de documents (par exemple, à la fin d'une activité, l'état de l'objet de l'activité est modifié automatiquement et le résultat sera envoyé à l'acteur responsable de l'activité suivante);
- avoir une meilleure gestion de projet de développement produit, en faisant bien et le plus vite possible ;
- garder trace des opérations et des décisions. Tracer les tâches en notifiant la bonne personne au bon moment et en stockant toutes les informations liées au démarrage et terminaison de chaque activité ;
- réduire les délais en déclenchant plusieurs opérations à la fois ce qui permet et de diminuer le délai de conception de produit et de mise sur le marché ;
- accroître la rigueur dans les processus en intégrant toutes les règles métier et les contraintes au plus tôt dans le processus. Ceci permettra de vérifier automatiquement le respect des contraintes et le bon respect des règles métier à des points bien déterminés du processus.

L'automatisation des processus est donc un besoin crucial qui présente plusieurs avantages [Stohr 97]. Toutefois, aucun système n'est actuellement satisfaisant, qui soit adapté aux spécificités du contexte industriel. Dans la section suivante nous allons examiner les systèmes de gestion des processus de conception pour voir s'ils permettent de répondre aux exigences ci-dessous mentionnées (en termes de modélisation et support à l'exécution).

3. Systèmes de gestion des processus de conception : Etat de l'art

Comme nous avons pu le voir, les processus de conception de produits manufacturés sont au cœur des activités du domaine de la conception de produits manufacturés. Nous nous intéressons dans cette section à la façon avec laquelle les processus de conception sont gérés actuellement.

Comme tout processus métier, les processus de conception doivent être gérés. La gestion des processus constitue un des grands sujets de préoccupation des entreprises. Elles y sont poussées par

leur constante nécessité d'optimiser leur performance pour rester compétitives sur leurs marchés, voire pour être présentes sur des nouveaux marchés. Dans la littérature, nous trouvons des définitions qui se recouvrent pour décrire la gestion des processus. Certains auteurs parlent de l'ingénierie des processus métier -BPM (*Business Process Management en anglais*) et d'autres de gestion de workflow - WFM (*Workflow Management en anglais*). Une différence est faite par [vanderAalst 04], qui considère le WFM comme étant un sous ensemble du BPM. Le WFM se concentre sur la partie basse du cycle de vie du BPM, plutôt sur l'exécution, ceci est une vision orientée technologie. D'autre part, le BPM est une vision plus orientée métier qui permet de donner à une entreprise une vision de bout en bout de ces processus. Différentes définitions existent dans la littérature nous retenons celle de [Smith 02], *le BPM correspond à l'ingénierie des processus de l'organisation, ou processus métier, à l'aide des technologies de l'information. Il a pour vocation de modéliser, déployer, exécuter et optimiser de manière continue les différents types de processus et ainsi d'améliorer l'agilité d'une organisation.* Comme le dit la définition, dans le cadre du BPM, la gestion des processus s'associe à trois grandes disciplines ; l'analyse des processus (modélisation), l'automatisation des processus (exécution) et le pilotage des processus (optimisation). D'ailleurs beaucoup d'entreprises se sont lancés dans la cartographie de leurs processus et leur modélisation pour mieux les comprendre, les gérer et les automatiser.

Il ressort de l'état de l'art sur la gestion de processus que cette gestion consiste en la modélisation et la traduction du modèle métier vers un modèle permettant l'automatisation du support. Comme nous l'avons constaté, les processus de conception sont caractérisés par leur nature collaboratif et leur structure complexe et évolutif. Ils impliquent des formalismes de modélisation adaptés mais aussi le recours à des outils supports. Ainsi le système support doit prendre en compte les objectifs stratégiques mais aussi économiques et organisationnels du domaine de conception étant donnés les spécificités de ses processus collaboratifs, évolutifs/émergents et tendus. Les exigences économiques concernent les coûts de gestion et de modification des processus. Le coût peut être de différentes natures (de modification, des ressources humaines/matériels impliquées dans la mise en place du changement, de communication et d'échange d'informations entre les personnes métier et les informaticiens, du retard dû au dysfonctionnement des processus, etc.). La flexibilité est donc une caractéristique essentielle des processus de conception. Ceci nécessitera d'avoir des modèles de processus flexibles mais aussi un système support flexible permettant de répondre rapidement aux besoins de changement. Selon [Daoudi 06] pour être flexible, une méthode de modélisation de processus doit être assez riche, permettant de modéliser tous les concepts nécessaires et de rajouter des nouveaux concepts s'il y a besoin. Quant aux exigences organisationnelles, elles concernent la prise en compte de la nature des acteurs métiers et le respect de leurs intentions. L'expressivité et la simplicité sont donc des caractéristiques essentielles des processus de conception. En effet, les acteurs métier doivent pouvoir représenter les processus en entier. Toutefois, le modèle doit rester compréhensible par ces acteurs métier. Ceci nécessitera d'avoir une notation simple et de pouvoir représenter tous les concepts métier du domaine.

3.1. Modélisation des processus de conception de produits manufacturés dans l'industrie

La modélisation des processus consiste à représenter la structure et le fonctionnement du processus avec un certain niveau de détail. Si l'on s'interroge sur les raisons qui poussent à modéliser un processus, on doit distinguer deux grandes catégories de processus : répétitif ou unique [Berthier 06]. Un processus unique, dont l'exemple principal est le projet, n'est exécuté qu'une seule fois. Sa modélisation peut être faite soit pour des raisons de planification, soit parce qu'on cherche à représenter une structure générique pouvant servir de support à un outil de collaboration autour d'un

processus, ou car le processus est complexe. Un processus répétitif est amené à être exécuté à de multiples reprises et sa description sous forme de modèle présente un caractère normatif : les activités sont supposées être effectuées conformément à leur description en tâches et à leur ordonnancement. Les processus répétitifs caractérisent le comportement global de l'entreprise tels que les processus de conception de produits manufacturés.

Dans ce qui suit, nous présentons les standards de description des processus industriels. Nous n'étudions pas les langages déclaratifs, car contrairement aux langages impératifs, la description du flux de contrôle est implicite ce qui rend les processus difficilement compréhensibles par les acteurs métier. De plus, l'objectif de la section 3 étant d'examiner les systèmes de gestion des processus en termes de modélisation et puis d'outils support à l'exécution dans le but de voir s'ils permettent de répondre aux exigences mentionnées dans la section 2.3. Nous nous focalisons, dans cette première section, uniquement sur les standards les plus répandus pour la modélisation des processus industriels. Nous examinerons les travaux de recherche dans ce domaine si les standards ne permettent pas de répondre aux exigences des processus de conception en termes de flexibilité, expressivité et simplicité du modèle de processus.

La méthode SADT, résultat de fusion des deux techniques IDEF0 (Integration Definition Language 0) et IDEF1 [Ross 77] a été auparavant utilisée dans la modélisation des processus industriels tels que les processus de conception. Elle se compose d'une arborescence de diagrammes. Un diagramme peut être décomposé en plusieurs diagrammes "enfants", chacun correspondant au détail d'une fonction. SADT, permet de modéliser deux types d'analyse. L'analyse par des actigrammes (boîtes d'actions) et l'analyse par des datagrammes (boîtes de données). Sur des actigrammes, les actions sont reliées entre elles par des flux de données alors que dans les datagrammes se sont les données qui sont reliées entre-elles par des flux d'activités. Un processus de conception de produits manufacturés peut donc être modélisé comme une arborescence d'actions. Partant de l'action la plus abstraite telle que l'analyse fonctionnelle du produit à celle plus détaillée telle que la validation des fonctions du produit. Un langage plus récent est IDEF3 [Mayer 95]. Les schémas de processus d'IDEF3 ressemblent beaucoup aux actigrammes d'IDEF0. IDEF3 permet de réunir sous le même qualificatif, "unité de comportement" (UDC), une activité ou un ensemble d'activités. Les UDC sont reliées par des liens dont la représentation varie en fonction de la contrainte sur le lien tel que les "contraintes de précedence" entre UDC, etc. IDEF3 abandonne tous les autres types de flèches indiquant les contraintes sur les activités ou les ressources allouées, ce qui a le mérite d'augmenter la lisibilité du modèle. Par ailleurs, IDEF3 introduit des symboles de contrôle de flux appelés "jonctions". Il s'agit d'opérateurs logiques classiques : AND, OR et XOR. D'autres entreprises ont recours au diagramme d'activité du langage UML (Unified Modeling Language), standard de l'OMG (*Object Management Group*), pour modéliser leurs processus de conception de produits. Dans ses versions 1.x, UML proposait un ensemble de diagrammes permettant de modéliser des processus métier, tel que le diagramme d'activité. Dans sa version 2.0 UML propose "d'augmenter" sa notation grâce au concept de stéréotype. Un stéréotype permet de créer de nouveaux types qu'il est possible d'ajouter au méta-modèle, ces stéréotypes étant propres à chaque modélisateur. Ceci permettra au modélisateur d'exprimer l'ensemble des concepts métiers dont il a besoin dans son modèle de processus. Un autre langage très utilisé dans les industries de nos jours est BPMN (*Business Process Modeling Notation*) [OMG 08a], l'initiative du BPMI (*Business Process Management Initiative*). BPMN est un langage de conception de processus et non pas d'exécution [Tudor 06]. Il permet de décrire les processus de conception de produits manufacturés sous forme graphique. La notation de BPMN est divisée en cinq catégories : les activités, les connecteurs, les événements, les couloirs d'activités et les artefacts. Il offre une notation qui permet de représenter plusieurs concepts métier des processus métier. Enfin, le

langage XPDL (*XML Process Definition Language*) est proposé comme un langage standardisé de description des processus par le WfMC (*Workflow Management Coalition*) [WFMC 12]. XPDL est construit à partir du langage XML (*Extensible Markup Language*) et décrit la représentation graphique des éléments du processus de manière textuelle. Il définit les activités, les transitions, les partenaires et les interactions entre eux sous forme textuelle (coordonnées 'XY', taille des nœuds, etc.). La première version de XPDL définit un ensemble minimal des concepts adoptés par la plupart des moteurs de workflow.

Étant donné que notre objectif est de permettre une gestion flexible des processus de conception tout en offrant une expressivité qui permet de prendre en compte tous les éléments d'un processus métier mais aussi en garantissant la simplicité du modèle, nous allons présenter une comparaison entre les standards présentés ci-dessous. Cette comparaison est basée sur trois critères qui sont au cœur de nos préoccupations à savoir la flexibilité, l'expressivité et la simplicité du modèle. Nous résumons les caractéristiques de ces langages dans le tableau ci-dessous:

Tableau 1 : Comparaison des langages de modélisation des processus de conception

	SADT	IDEF3	BPMN	UML	XPDL
Simplicité de la Notation	Concepts redondants, utilisation difficile	Modèle difficilement compréhensible	Notation graphique simple mais le nombre énorme de concepts est parfois source d'erreurs	Notation graphique mais modèle Peu intuitif	Notation basé sur XML, incompréhensible pour des acteurs métier
Expressivité du langage	Langage pauvre et utilisation de même notation pour représenter des concepts différents	flux de contrôle de base et Notation pauvre; tout les concepts sont réunis sous le nom d'UDC	Permet de représenter les flux de contrôle de base. Notation riche en concepts métier	Permet de représenter les flux de contrôle les plus utilisés. Possibilité de rajouter des nouveaux concepts en UML 2	Permet de représenter les flux de contrôle les plus utilisés. Notation pas assez riche
Flexibilité	NON	NON	NON	NON	NON

Si nous reprenons la méthode SADT, nous constatons que la représentation des ressources et des contraintes est dès lors redondante, donc source d'erreurs. Par conséquent, la clarté du modèle et sa lisibilité s'en ressentent quand le nombre d'entités augmentent. Quant au modèle IDEF3, le principal inconvénient est le groupement de plusieurs concepts tels que la représentation des données, des personnes et les UDC dans un même schéma. En fait, il est possible de lier un objet manipulé par les transitions à un objet "personne" pour exprimer que cet objet "personne" correspond au rôle qui peut agir sur cette donnée. Ceci rend le modèle incompréhensible par les acteurs du domaine de la conception. Les diagrammes UML restent néanmoins peu intuitifs pour les personnes du domaine de conception, ce qui rend difficile la communication autour de ces diagrammes ainsi que leur analyse. L'introduction de stéréotypes dans les diagrammes UML2 permet de représenter tout les concepts métier du domaine ce qui apporte de la richesse à ce formalisme. Néanmoins, elle rend l'exploitation de ceux-ci plus difficile notamment dans les outils d'exécution de processus [Tudor 06]. La notation BPMN possède une représentation graphique simple mais elle a le défaut de ses qualités : le nombre énorme d'objets (type messages, types d'évènements, etc.) qu'elle propose peut être une source d'erreurs. Toutefois BPMN offre une notation qui couvre plus correctement les besoins en modélisation de processus métier qu'UML. Ceci est confirmé par l'étude effectuée par [Zur Muehlen 08]. En effet, son plus grand avantage par rapport à UML se situe au niveau de la spécialisation des activités, des structures de contrôle et des événements. De plus, Le travail de White dans [White 04] présente une comparaison des capacités des deux langages UML et BPMN à exprimer les patrons de flux de contrôles. Cette comparaison montre que BPMN supporte un plus grand nombre de ces patrons qu'UML. Finalement, XPDL n'offre pas une facilité d'utilisation pour définir les processus par les acteurs métier. Cela est dû au fait qu'il hérite des concepts et de la syntaxe du langage XML (langage

à base de balise et non pas graphique). De plus, l'analyse faite par [vanderAalst 03a] montre que XPDL ne supporte que 11 patrons de base des flux de contrôle tels que le parallélisme, la synchronisation et le choix exclusif. Cet ensemble minimal ne permet pas de représenter directement tout les flux de contrôle présentés pratiquement dans la plupart des modèles de processus rencontré dans la pratique ni de présenter des processus complexes. Enfin, en utilisant les langages impératifs tels que UDEF, BPMN, UML et XPDL, les scénarii d'exécution sont décrits et liés aux activités dès la phase de conception, ceci compromet la flexibilité du modèle.

Pour conclure, Nous constatons que les modèles de processus produits par ces langages se situent toujours dans un cadre axé sur l'instrumentation en décrivant les ressources nécessaires et les événements, en passant parfois sous silence les aspects ayant trait à l'acteur humain à savoir la simplicité et la non-ambiguïté du modèle de processus. De plus, les standards et langages étudiés, et qui sont utilisés dans l'industrie, ne répondent pas au besoin d'expressivité des processus à savoir la possibilité de représenter tout les concepts métiers.

3.2. Outils support à l'exécution des processus de conception de produits manufacturés dans l'industrie

Cette section présente les principaux outils permettant de supporter l'exécution des processus de conception de produits dans les entreprises et de dégager leurs limites vis-à-vis des exigences liés au support à l'exécution des processus de conception (*i.e.* outil de communication, automatisation de la collaboration et flexibilité de l'outil).

Certaines entreprises utilisent des progiciels intégrés pour le support de leur processus. Les progiciels intégrés sont une solution "clé en main" qui fournit une solution complète pour un domaine précis tel que la conception de produits manufacturés, la gestion du projet, etc. Du fait, que la mise en place d'un tel progiciel demande un long projet de paramétrage et d'adaptation, l'entreprise doit alors s'adapter aux processus définis par le progiciel et non l'inverse [Tomas 00]. De plus, pour modifier un processus il est nécessaire de personnaliser la plateforme. Cela peut s'avérer trop chère et problématique. Généralement, il n'est pas facile d'adapter un processus pour répondre à un changement au niveau des besoins métier même s'il est urgent ou critique.

Nous citons aussi le *groupware* (collecticiels) comme outil support aux activités de la conception de produits manufacturés. Il englobe un nombre important de technologies et applications. D'ailleurs, une des définitions du *groupware* le considère comme « *un système qui intègre du matériel et du logiciel existants en vue d'améliorer le travail en équipe* » [Khoshafian 98]. Ainsi, de nombreux produits qui participent à la mise en oeuvre des processus de conception de produits peuvent être classés comme produits de *groupware*. Il existe trois catégories principales d'applications *groupware* à savoir. Le *groupware* orienté documents et formulaire tels que la messagerie qui permet d'expédier des informations et documents. Le *groupware* orienté gestion transactionnelle de gros volume d'informations tels que les systèmes de GED (gestion électronique de documents), les moteurs de recherche d'information ou encore les SGBD (systèmes de gestion de base de donnée) et le *groupware* orienté communication horizontale tel que l'agenda du groupe, la visioconférence, la conférence électronique, etc. Ceci étant, le *groupware* en tant qu'outil support aux activités de la conception a pour objectif premier de supporter les tâches du processus sans qu'il soit nécessaire de le définir ni de l'automatiser en entier. Ainsi, le *groupware* n'apporte pas de solution aux problèmes d'exécution des processus de conception de produits manufacturés et peu même induire a une distribution injuste des tâches et activités.

Comme tout processus métier, les processus de conception peuvent être gérés par les systèmes de workflow. Le Workflow est défini comme étant « *The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules* » [WfMC 99]. Les workflow étaient alors utilisés pour la gestion des documents et le support du travail collaboratif. L'idée était l'automatisation des tâches administratives afin de diminuer le temps de leur réalisation. En effet, le workflow permet de faire circuler les lots de travaux³ relatifs à un processus de conception donné d'un acteur ou un groupe d'acteurs vers un autre sous forme de "paquets" organisés selon une logique de découpage du processus qui correspond aux différents sous-objectifs du processus de conception de produits considéré. Cette vision du workflow est essentiellement procédurale : toutes les activités ainsi que leur enchaînement sont prédéfinis et toute information ne peut être transférée d'un acteur à l'autre que si elle a changé d'état. Ce workflow peut être organisé manuellement mais dans la pratique la plupart des workflow sont organisés dans le cadre d'un système informatique à savoir le système de gestion de workflow (en anglais *WorkFlow Management systems* –WFMS). Le WFMS est défini comme étant « *A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of Information-Technology tools and applications*» [WfMC 99]. Le moteur de workflow a pour objectif de transférer des documents entre les acteurs d'un processus en leur assignant des tâches à réaliser (valider le document, effectuer une modification, etc.). Ceci permet de faciliter la collaboration entre les différents acteurs.

Enfin, les activités des processus de conception de produits manufacturés sont de nos jours supportées par des applications et systèmes logiciels favorisant la collaboration entre les différents acteurs du processus. Les systèmes de gestion de cycle de vie de produit (PLM) intègrent ces applications et logiciels pour supporter les processus de conception. Après l'apparition de cet acronyme dans les années 2000 [Lamouri 06], le PLM est devenu un marché important en volume et en valeur [CIMdata 09]. Les Systèmes PLM sont, en première analyse, relatifs à la gestion du cycle de vie des produits. Selon [Sääksvuori 05] le PLM joue un rôle essentiel dans la gestion des données produites tout au long du cycle de vie du produit (conception, industrialisation, maintenance, etc.). Bien que le besoin initial du PLM fût de permettre le développement simultané et collaboratif des produits en prenant en compte toutes les phases de leur cycle de vie, nous pensons que le PLM ne peut être réduit à sa seule dimension de SGBD. En effet, le PLM en anglais *Product Lifecycle Management*, désigne simultanément la gestion des structures des produits, des données techniques et documents associés, et des flux de travaux sous-jacents aux processus de conception. D'ailleurs selon CIMData⁴ le PLM est considéré comme *une approche stratégique d'entreprise, qui applique un ensemble de solutions pour soutenir dans un mode collaboratif la création, la gestion, la dissémination et l'utilisation de l'information de définition des produits, en entreprise étendue, du concept à la fin de vie en intégrant les acteurs, les processus, les systèmes et l'information*. Ces systèmes PLM intègrent des bouts prédéfinis de processus de conception de produits et dirigent le processus de conception en se basant sur les données du produit (telles que les états des informations du produit). Pour assurer ses fonctionnalités, le noyau du PLM offre des moyens nécessaires permettant l'intégration et la communication avec des outils classiques de bureautique (Word, Excel, Access), des outils de Conception Assistée par Ordinateur (CAO, DAO, etc.) et des outils de simulation (Abaqus, Ansys, etc.). Ces outils aident le concepteur à créer des données techniques en fonction de la tâche qui lui a

³ Un ensemble de documents, d'informations et de tâches à effectuer pour atteindre un objectif métier donné du processus considéré [Gzara 00]

⁴ <http://www.cimdata.com/plm/definition.html>

été affectée afin de définir le produit. Le PLM permet aussi l'organisation, le partage et la conservation des données créées à travers la gestion de données techniques. Ceci en fournissant un méta-modèle des données et un gestionnaire de la structure du produit. Ce gestionnaire organise et stocke les informations du produit. Il offre aussi un moteur de gestion des processus qui, pour une structure produit donnée, permet d'envoyer la bonne donnée à la bonne personne et au bon moment en se basant sur les bouts de processus intégrés. D'autres moyens sont également disponibles, comme l'intégration avec un serveur de messages électroniques, qui notifie les utilisateurs d'un événement relatif à un document, du début ou la fin d'une activité du processus pour lequel la personne est impliquée. Cependant, une des spécificités des processus de conception est leur évolution ; ce qui fait que les processus intégrés dans le PLM seront amenés à changer plusieurs fois. Pour réaliser ces changements, on entre alors dans une boucle de modélisation et intégration du nouveau modèle dans le PLM. Ceci engendre des coûts considérables et conduit dans beaucoup de cas à l'échec du processus dû à la lenteur du développement et d'intégration des nouveaux processus dans le PLM. En effet, le développement des processus nécessite deux étapes. Une étape de modélisation par une personne du domaine métier et une étape de traduction par une personne ayant des connaissances techniques. Pour cela certains systèmes PLM font recours à des outils permettant la prise en compte de modèles de processus en vu de leur automatisation tel que les moteurs de workflow.

Nous présentons, dans le tableau ci-dessous, une comparaison des principaux outils supports à l'exécution des processus de conception de produits manufacturés présentés ci-dessus. Cette comparaison est basée sur trois critères qui sont au cœur de nos préoccupations à savoir le besoin en termes de communication et de synchronisation, le besoin en terme d'automatisation de la collaboration et le besoin en terme de flexibilité de l'outil de collaboration.

Tableau 2: Comparaison des outils supports à l'exécution des processus de conception

	Logiciels Intégrés	Groupware	Workflow système	PLM
Communication	OUI	OUI	OUI	OUI
Automatisation	OUI	NON	OUI	OUI
Flexibilité	NON	NON	NON	NON

Cette présentation des principaux outils supports à l'exécution des processus de conception de produits manufacturés montre qu'il y'a trois types d'outils. Le premier type est celui qui n'intègre pas le modèle du processus tels que le groupware et les systèmes PLM qui n'intègrent pas des moteurs de workflows. Ce genre d'outils facilite le déroulement du processus, à travers la mise à disposition d'outils de communication et de planification des tâches, mais il ne permet pas d'automatiser le processus. Le fait de ne pas intégrer le modèle de processus induit, la plupart du temps, à une mauvaise conduite des activités. Le deuxième type d'outils intègre des modèles de processus prédéfinis auxquels l'entreprise doit s'adapter tel est le cas des logiciels intégrés qui offrent des solutions 'clé en main'. Ceci satisfait les deux contraintes d'automatisation et de communication mais ne remplit pas le critère de flexibilité. En effet ces systèmes nécessitent un effort énorme qui doit être produit par l'entreprise. De plus, il peut s'avérer trop coûteux d'aligner les processus de l'entreprise aux processus prédéfinis. Finalement, le troisième type est celui qui intègre le modèle de processus tel que les moteurs de workflows et les systèmes PLM qui font recours aux systèmes de workflows. Vu la complexité des processus de conception et la multitude des acteurs, cette automatisation des processus

à travers l'utilisation de moteurs de workflows est nécessaire. Toutefois, l'intégration des deux outils (PLM et moteur de workflow) alourdit le système et le rend plus complexe. L'argumentation de dire c'est mieux d'avoir un moteur d'automatisation de processus dans le PLM est une grosse erreur car on revient dans les problèmes d'une approche intégrée comme les ERP (*Enterprise Resource Planning*, ou en français, PGI pour Progiciels de Gestion Intégrée) qui a montré ces limites [Gilbert 04]. En effet, le tout intégré conduit à une perte d'efficacité du système global. Il est difficile de synchroniser correctement les applications intégrées, puisque les versions n'évoluent pas toutes au même rythme [Jomaa 07]. De ce fait nous constatons qu'il faut séparer l'outil de workflow du PLM et nous proposons que ces deux outils puissent être utilisés de façon complémentaire sans être intégrés. De plus, les technologies workflow utilisées actuellement pour le support des processus de conception de produits, ne permettent pas de supporter certains processus évolutifs, ni d'aménager les processus déjà définis. Les entreprises sont donc confrontées à plusieurs obstacles, notamment, la mise en œuvre limitée des nouvelles méthodes de travail.

Pour conclure, la question de flexibilité des processus de conception impacte nécessairement l'outil support de ses activités. Cet outil doit disposer d'une capacité qui lui permet d'agir et être réactif afin de répondre aux demandes de changement. De même, les modèles de processus qu'utilisent ces outils sont rigides. Les systèmes workflows utilisent généralement un modèle de processus trop simpliste, voire incohérent avec la réalité opérationnelle. Les systèmes et outils logiciels mis en place pour supporter les activités de conception ne permettent pas de faire face aux changements imprévus ni d'intégrer les changements sans devoir redéfinir les processus dès le début. Ceci, nous conduit à étudier les travaux sur les approches de mise en œuvre de processus flexibles dans la section suivante afin de voir en quoi ces approches sont utilisables, pertinentes et peuvent être adaptées aux processus de conception de produits manufacturés.

4. Approches pour la mise en œuvre de processus flexibles

À l'issue des différents constats et analyses que nous avons menés jusqu'à ici, nous pouvons conclure que les entreprises doivent avoir une réflexion approfondie sur la modélisation et l'automatisation de leur processus afin d'assurer la flexibilité et l'agilité nécessaire. Plusieurs travaux ont essayé de résoudre le problème de rigidité des processus et de proposer des nouvelles approches pour leur gestion. Cette section présente une évaluation d'une sélection de ces travaux. Nous avons choisi ces travaux, d'une part pour leur intérêt à la notion de flexibilité et d'autre part parce qu'ils forment un ensemble représentatif de l'état de l'art. Cette partie présente ces approches et les situe par rapport à une taxonomie de la flexibilité que nous proposons en début de cette section et aux exigences du domaine.

4.1. Cadre de référence pour la flexibilité des processus

La flexibilité désigne 'la capacité de l'entreprise à agir et l'aptitude à répondre aux changements d'une manière dynamique et efficace' [McCoy 06]. Dans le contexte des systèmes d'information, [Chelli 03] définit la notion de flexibilité comme étant la déclinaison technique de la notion d'agilité des entreprises. Cette notion de flexibilité est souvent perçue en termes de la capacité des processus d'une entreprise et de ses technologies support de s'adapter aux changements métier [Snowdon 07] [Daoudi 07]. D'autres considèrent la flexibilité du point de vue opposé, c'est à dire, ils se concentrent sur les parties du processus qui restent inchangés, plutôt que de se concentrer sur les parties qui doivent être changées [Regev 07]. Ils considèrent un processus comme flexible que s'il est possible de mettre en œuvre les changements en ne modifiant que les parties qui ont besoin d'être changées tout en gardant la stabilité des autres parties du processus [Regev 06]. La flexibilité constitue aujourd'hui une

préoccupation majeure puisqu'il permettra aux entreprises de s'adapter aux changements et de supporter les processus émergents.

Plusieurs travaux ont tenté de proposer une taxonomie de la flexibilité [van der Aalst 00], [Carlsen 97], [Heinl 99], [Schonenberg 07] et [Regev 06]. Dans ce contexte, deux grandes taxonomies ont été proposées. La première taxonomie est celle proposée par [Schonenberg 07], il s'intéresse à la flexibilité du point de vue opérationnel. Cette taxonomie définit quatre types :

- la flexibilité par conception : concerne la possibilité d'exprimer des chemins alternatifs lors de la modélisation du processus et cela en permettant d'exprimer le parallélisme entre les activités, le choix de l'exécution d'une activité parmi plusieurs, l'instanciation multiple d'une activité dans la même exécution ou encore l'annulation de l'exécution d'une activité ;
- la flexibilité par spécification partielle : concerne la possibilité d'exécuter un processus partiellement spécifié c-à-d la possibilité de définir certaines structures de processus dans la phase d'exécution en permettant la sélection tardive d'un fragment de processus parmi plusieurs fragments alternatives ;
- la flexibilité par déviation : concerne la possibilité de dévier l'exécution d'une instance du processus de sa définition initiale en permettant par exemple de défaire ou de court-circuiter certaines activités ;
- la flexibilité par changement : ce type concerne la possibilité de modifier la définition du processus en permettant aux instances du processus de migrer vers la nouvelle définition.

La deuxième taxonomie est celle proposée par [Regev 06] qui s'intéresse aux changements qui peuvent survenir tout au long du cycle de vie du processus. Cette taxonomie comprend trois dimensions de changement (cf. Figure 3) :

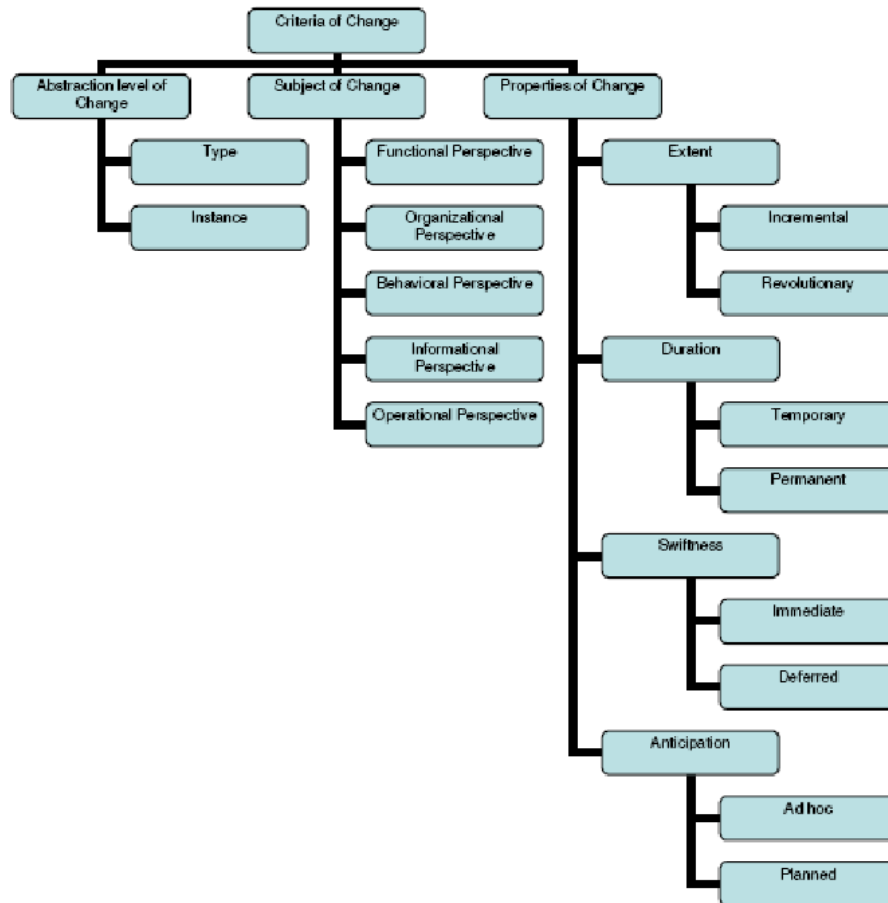


Figure 3 : Taxonomie de la flexibilité selon [Regev 06]

- le niveau du changement : concerne le niveau d'application du changement dans un processus. Le changement peut concerner la définition (i.e. modèle) ou l'instance (i.e. une exécution donnée) du processus.
- l'objet du changement : cette dimension concerne les différents aspects du processus qui sont sujets du changement. Le changement peut concerner les activités du processus (dimension fonctionnelle), l'enchaînement ou ordonnancement des activités (dimension structurelle), les données du processus (dimension informationnelle) ou les protocoles utilisés dans le processus tels que ceux de routage d'information (dimension opérationnelle).
- les propriétés du changement : concerne le degré du changement qui peut être partiel pour changer une partie du processus, total/radical pour créer un nouveau processus. Cette dimension concerne aussi la durée du changement qui peut être soit immédiate ou différée, et l'anticipation du changement qui peut être soit planifiée ou ah doc.

Dans notre contexte les processus de conception de produits manufacturés doivent pouvoir changer au niveau fonctionnel et structurel (i.e. objet du changement). Ces critères sont importants par rapport à notre problématique car la nature dynamique de l'environnement dans lequel se déroulent les processus de conception de produits rend les éléments de ce dernier susceptibles d'être modifiés, réorganisés ou encore supprimés. De plus, pour être flexible, un processus de conception de produits manufacturés doit pouvoir intégrer un changement partiel (i.e. degré de changement) quel que soit l'objet du changement. Ce critère est important car nous voulons intégrer le changement sans avoir besoin de redéfinir les processus dès le début. Enfin, le changement doit pouvoir être introduit au niveau de la définition du processus mais aussi au niveau des instances (i.e. niveau du changement).

Le Tableau 3 résume les exigences citées dans la section 2.3 et les critères de comparaison que nous retenons pour la flexibilité.

Tableau 3: Critères de comparaison des approches de mise en œuvre de la flexibilité

		Critère		Approche
Contraintes liées au Modèle		Syntaxe simple		
		Expressivité		
Contraintes liées au support	Communication et synchronisation			
	Automatisation			
Flexibilité	Niveau du changement	Définition		
		Instance		
	Objet du changement	Fonctionnel		
		structurel		
	Degré du changement	Partiel		
		radical		

4.2. Approches de flexibilisation des processus

Cette partie présente les approches de flexibilisation de processus et les situe par rapport à notre problématique. Un accent sera mis sur le critère de flexibilité.

Agilium est un produit commercial pour la gestion des processus métier. Il a été utilisé pour les processus industriels. L'architecture fonctionnelle d'Agilium est structurée autour de quatre modules ; modélisation, intégration, pilotage, exécution et supervision [CXP 06]. Le module de modélisation est un outil graphique basé sur Eclipse (ViFlow + Agilium) [Bitlog 12]. Il permet la modélisation des processus simples en utilisant des concepts classiques d'activités et de flux de contrôle. Le module d'exécution est basé sur le moteur d'exécution Crystal, un produit du CERN (centre européen de recherche nucléaire), qui suit une démarche BPM et qui permet de faire des changements dynamiques des processus métiers [Gaspard 03]. Ceci est possible en insérant des points de mesures ayant comme rôle le monitoring des activités. Des activités prédéfinies détaillent les actions à entreprendre en cas de besoin de changement (annulation d'activité, ajout d'une nouvelle activité, etc.). Une fois un évènement de déviation est détecté une activité prédéfinie est choisie et exécuté. Les actions des activités prédéfinies peuvent être modifiées au moment de l'exécution [Estrella 03]. Les détails du moteur permettant de mettre en œuvre les changements dynamiques ne sont pas bien expliqués vu que c'est un produit commercial.

Le langage YAWL (*Yet Another Workflow Language*) [van der Aalst 04a]. Ce langage est destiné à représenter les processus métier en étendant la syntaxe des réseaux de Pétri pour qu'elle supporte tous les patrons de flux de contrôle. En réalité, YAWL est un langage à la fois graphique et d'exécution. Un moteur d'exécution a été implémenté pour ce langage [Glatard 09]. Une caractéristique de YAWL c'est qu'il permet d'exécuter plusieurs fois une instance d'activité. Le nombre d'instance peut être fixé en différé ce qui permet une certaine flexibilité. De plus, il supporte les patrons d'annulation de fragment de processus ce qui permet d'apporter du changement sur les instances de processus. YAWL offre aussi des mécanismes de flexibilité sous forme de patrons d'exceptions [Michael 07]. Une

exception peut être interne ou externe (déclenché par l'utilisateur). Dix types d'exceptions ont été identifiés dont sept ont été implémentés par l'interpréteur de YAWL [Michael 07]. Quand une exception est déclenchée, son type est identifié et un fragment de processus (exlet) est défini et déclenché pour le traitement de l'exception. L'exlet peut être exécuté selon plusieurs actions (annulation du fragment, suspension du fragment, ajout du fragment ou *re-starting* du fragment). Une évolution est ainsi vue comme le traitement d'une exception. Bien que plusieurs éditeurs de BPMS s'intéressent à ce nouveau langage, ce dernier est encore dans sa phase de recherche et il a besoin d'un peu plus de temps pour se stabiliser.

Dans ses travaux de recherche, Chaâbane a utilisé la notion de versionnement afin de répondre au besoin de flexibilité des processus métier [Chaâbane 09]. Plus précisément, il a proposé un méta-modèle, appelé *Versioned Business Process Meta-model* (VBP-M), pour exprimer comment un processus métier flexible peut être modélisé en utilisant des versions de processus et comment ces versions peuvent être manipulées (choisies, instanciées et exécutées). Sa proposition met l'accent sur les activités qui doivent être exécutées et la façon avec laquelle elles sont coordonnées. Pour ce faire, il a introduit une perspective intentionnelle qui permet d'identifier quand une version d'un processus métier peut être utilisée. Dans son travail, la perspective intentionnelle a été introduite dans le but de définir le contexte d'utilisation des versions de processus, ce qui permettra d'aider les utilisateurs à choisir, au moment de l'exécution, parmi les différentes versions possibles. De ce fait, le concepteur de processus doit préciser, lors de la définition d'une version au moment de la conception, le contexte d'utilisation de cette version (*i.e.* les différentes conditions qui doivent être satisfaites pour pouvoir utiliser la version). Cette proposition permet d'introduire une certaine flexibilité au contexte à condition que toutes les versions nécessaires aient été prévues.

Les travaux de Lezoche proposent une approche orientée activité basée sur des règles de transition ou de choix entre activités [Lezoche 08]. Ces règles expriment les règles métier de l'entreprise. Son approche propose de voir le processus métier comme un ensemble de tâches coordonnées entre elles à l'aide de ces règles. Cette modélisation basée sur les règles permettra par la suite de déterminer l'ordre d'exécution des activités en fonction de l'évaluation dynamique des conditions des règles à l'aide d'un moteur de règles. Le moteur de règle permet d'interpréter les règles métier qui définissent le processus et de les exécutés. Le moteur permet aussi d'identifier les processus impactés par un changement de règles métier. A chaque fois qu'une règle est modifiée la base de processus est parcourue pour vérifier si cette règle fait partie d'un des processus existants. Il permet ainsi de supporter l'évolution des processus impactés par le changement à travers l'intégration des nouvelles règles ou la modification de celles ci.

Les travaux de Morley proposent d'enrichir l'expression des modèles de processus et d'ouvrir la voie vers la modélisation par composants dynamiques [Morley 05-b]. Dans le travail de Morley, la notion de flexibilité est assurée à travers la prise en compte des deux concepts ; 'But' et 'Agent'. Son approche adopte une perspective orientée but. Elle propose de modéliser le processus comme un ensemble d'enchaînements entre activités décrites par des buts qui peuvent être atteints par des agents ou bien par des exécutants. En effet, une activité peut être attribuée à un exécutant, défini comme un acteur qui effectue des tâches en fonction de leur description. Elle peut être aussi attribuée à un agent si elle est décrite par un but et que son but correspond à l'un des objectifs que l'agent peut assumer. Ainsi, le concept d'agent étend le concept d'acteur qui a la responsabilité d'exercer certaines activités dans le processus. Un agent est défini comme un acteur avec une capacité à effectuer des activités de façon autonome, sans avoir besoin d'indiquer le mode de fonctionnement (l'autonomie est ici celle des moyens et non pas celle des fins). Le concept de 'but' est utilisé aussi bien pour caractériser une

activité (une activité visant à atteindre un objectif) ou un agent (un agent a la capacité d'atteindre certains objectifs). De ce fait, en cas d'un besoin de changement, la possibilité de choisir parmi les agents qui peuvent atteindre le but avec des moyens différents permettra de faire face à ce changement. Le choix de l'agent se fait d'une façon dynamique à l'aide d'un système multi-agents. L'introduction de ces concepts et leurs associations permet d'apporter de la flexibilité au niveau des activités. Ceci est dû au fait que les agents sont autonomes et peuvent atteindre l'objectif de plusieurs façons.

D'autres travaux adoptent une perspective orientée service en vue de proposer des activités réutilisables et flexibles [Boukadi 09a]. Dans le travail de Boukadi, la notion de flexibilité est assurée à travers la prise en compte des deux concepts ; Contexte et Service. Les activités proposées par son approche sont matérialisées par des services adaptables selon le contexte d'utilisation. Ces services sont enregistrés dans un catalogue de service et seront découverts au moment de l'exécution selon le contexte d'utilisation. En effet, le processus est défini initialement, pendant la phase de modélisation, sous la forme d'un schéma abstrait de processus. Le schéma abstrait contient les différents *Goal-Templates* qui correspondent aux tâches à accomplir afin d'atteindre les objectifs du processus. Un *goal-Template* spécifie ce que l'on souhaite réaliser en sortie, ce que l'on peut fournir en entrée et l'ensemble de paramètres de contexte qui le définissent. Ces paramètres de contexte correspondent aux paramètres contextuels des services disponibles. Le *goal-Template* exprime aussi une requête permettant de faire la découverte dynamique des services enregistrés dans le catalogue de service. Le processus exécutable constitue une instanciation d'un schéma abstrait de processus. La transformation du processus abstrait vers un processus exécutable est faite à travers le remplacement des *goal-Templates* par les services découverts au moment de l'exécution. Nous considérons que la prise en compte du contexte lors de la modélisation des services et des processus abstrait, en plus que le 'matching' retardée des services et *goal-Templates* apporte une certaine flexibilité à la démarche. En cas d'un besoin de changement dans la façon de faire d'une activité, ceci peut être résolu à travers la publication d'un nouveau service possédant les mêmes paramètres contextuels demandés mais qui implémente l'activité selon la nouvelle façon de faire. Toutefois, comme la proposition de Morley, la vision de flexibilité traitée par cette approche est limitée au niveau des activités, plutôt que sur le niveau structurel. En effet, en définissant l'ordre d'enchaînement entre activités, ceci réduit considérablement la flexibilité du processus.

Les travaux de Pourraz visent à fournir une réponse quant à l'évolution dynamique d'une orchestration de services Web en cours d'exécution [Pourraz 07a] [Pourraz 07b]. L'orchestration de services Web permet d'exprimer un processus exécutable. Ses travaux ont pour but de proposer des orchestrations de services agiles et dynamiquement modifiables. L'approche proposée, nommée Diapason, adresse ce challenge en offrant un langage d'orchestration nommé π -Diapason. π -Diapason est formellement défini (basé sur le langage formel π -calcul), exécutable et offre la possibilité de faire évoluer dynamiquement une orchestration de services web grâce au concept de mobilité introduit par le π -calcul. En effet, la notion de mobilité supportée par le π -Calcul, permet de faire transiter un processus au travers d'un canal, afin de fournir un nouveau comportement à un processus en cours d'exécution. Dans son approche, il utilise ce concept de mobilité pour faire évoluer dynamiquement une orchestration. Cette évolution dynamique peut être planifiée ou non. L'évolution dynamique planifiée peut se faire à travers la prise en compte d'une remontée d'alerte par l'orchestration elle-même (une demande d'évolution) ou encore à travers la substitution d'une partie du comportement de l'orchestration (déterminée pendant la phase de conception). Ces points d'évolution planifiée sont gérés au niveau du langage. Ces deux cas d'évolution sont formalisés de la façon suivante. Il faut insérer à des endroits, au sein de la description de l'orchestration, où l'on pense que l'orchestration

peut évoluer des points d'évolution. Ceci est fait pendant la phase de conception. La modification est possible grâce à un modèleur graphique qui gère l'ensemble des points d'évolution comme des opérateurs, au même titre que les patrons de Workflow. L'évolution non planifiée est gérée au niveau de l'interpréteur π -calcul. Pour ce type d'évolution seul l'architecte de l'orchestration pourra décider de faire évoluer ou non une orchestration. Il peut alors utiliser le modèleur graphique pour modifier la description de l'orchestration de manière à prendre en compte les changements et puis envoyer la nouvelle orchestration à la machine virtuelle qui décidera de la réalisation effective du changement si les évolutions apportées à l'orchestration interviennent après l'état courant de l'exécution. La réalisation effective se traduit aussi par le mécanisme de substitution d'orchestrations.

4.3. Analyse des approches de flexibilisation des processus

En se basant sur l'étude des approches de flexibilisation des processus, nous allons situer ces approches par rapport aux exigences des processus de conception citées dans la section 2.3. Un accent sera mis sur les critères de flexibilité fixés au début de cette section (cf. Tableau 4).

Tableau 4: Synthèse des approches de flexibilisation vis-à-vis de la problématique

		[Pourraz 07a]	[Lezoche 08]	[Chaâbane 09]	[Morley 05-b]	[Boukadi 09]	[van der Aalst 04a] YAWL	[CXP 06] Agilium	
Contraintes liées au Modèle	Syntaxe simple							+	
	Expressivité	+					+		
Contraintes liées au support	Communication et synchronisation	+					+	+	
		+		+		+	+	+	
	Niveau du changement	Définition	+	+	+				
		Instance	+			+	+	+	+
	Objet du changement	Fonctionnel	+			+	+		
		structurel	+	+	+			+	+
	Degré du changement	Partiel	+			+	+	+	+
radical		+	+	+					

Contraintes liées au modèle : Les travaux de recherche tels que le travail de [Morley 05-b] [Lezoche 08], [Chaâbane 09] et [Boukadi 09a] ne supportent pas tous les flux de contrôles nécessaires pour la représentation des processus de conception. De même, les flux de contrôle d'Agilium ne sont pas nombreux (Et, Ou, Boucle et Rendez-vous). De plus, comme tous les travaux de recherche cités ci-dessus, Agilium ne possède pas de couche métier qui permet de faciliter la modélisation des processus de conception par des acteurs métier ni de représenter les concepts du domaine métier. L'approche de Lezoche est basée sur un langage impératif à base de règles qui n'est pas facile à utiliser par les personnes métier.

Contraintes liées à la synchronisation et l'automatisation : Les propositions de [Morley 05-b] [Lezoche 08], [Chaâbane 09] et [Boukadi 09a] sont dédiées à la modélisation graphique des processus et doivent être traduits vers un langage d'exécution pour que le processus soit implémenté. À l'inverse de ces approches, Agilium et YAWL permettent l'automatisation des processus en proposant des moteurs d'exécution. π -Diapason est aussi un langage exécutable qui permet l'automatisation des processus.

Contraintes liées à la flexibilité :

- **Objet du changement :** nous remarquons que certaines approches parmi celles étudiées permettent d'introduire de la flexibilité au niveau fonctionnel du processus (cf. L'objet du changement : Fonctionnel dans le Tableau 4) tels que ceux de [Morley 05-b] et [Boukadi 09a].

Morley propose de modéliser les activités par des buts et fournit des traitements réutilisables implémentés par des agents permettant d'atteindre les buts. Ainsi, son travail permet d'introduire du changement au niveau des activités en sollicitant des agents qui répondent au but de l'activité. Quant au travail de Boukadi, il permet de modifier les activités en utilisant des services adaptables selon le contexte d'utilisation. Le changement proposé par ces deux approches est bien au niveau fonctionnel. Cependant ceci n'est pas suffisant. Pour être flexible, les processus doivent pouvoir changer pas uniquement les parties fonctionnelles mais aussi la partie structurelle (enchaînement entre les activités). En fait, en définissant l'enchaînement entre les activités, ceci compromet la flexibilité du modèle de processus. Cette pratique rend les processus rigides et difficiles à maintenir tant que la nature dynamique de l'environnement industriel, rend les éléments du processus susceptibles d'être modifiés, réorganisés ou encore supprimés. D'autres travaux tels que ceux de [Lezoche 08], [Chaâbane 09] [van der Aalst 04a], [Pourraz 07a] et Agilium [Gaspard 03] permettent d'introduire de la flexibilité au niveau structurel du processus (cf. L'objet du changement : Structurel dans le Tableau 4). Cependant, la flexibilité introduite au niveau structurel par [Lezoche 08], [Chaâbane 09] est prévue au moment de la conception. Dans le travail de Lezoche le changement de l'enchaînement des activités est basé sur des règles de transition qui permettent de choisir entre les activités du processus. Ces règles sont prédéfinies et le changement ne peut être fait si aucune règle ne correspond au besoin du changement. Quant au travail de Chaâbane, le changement au niveau structurel est assuré en utilisant des versions contextuelles du processus. Ceci dit, si jamais aucune version ne répond au besoin du changement, le changement ne peut pas être mis en œuvre. La flexibilité introduite par l'évolution planifiée dans le travail de Pourraz, permet de mettre en œuvre le changement au niveau structurel à travers la substitution de bout d'orchestration mais nécessite de planifier les points d'évolution au moment de la conception. Cette planification lors des phases de conception reste contraignante. En effet, tout scénario d'évolution doit être pensé avant l'exécution, ce qui n'est pas forcément faisable sur des gros processus. Toutefois, l'évolution non planifiée proposée par Pourraz, les mécanismes d'annulation et de traitement d'exception proposés dans YAWL et ceux proposés par le moteur CRISTAL de la solution Agilium constituent une réponse au besoin de changement dynamique au niveau structurel des processus.

- **Niveau du changement** : nous constatons que la majorité des approches étudiées permettent d'introduire de la flexibilité au niveau de la définition du processus (cf. Niveau du changement : Définition dans le Tableau 4). Ceci permettra d'introduire le changement au moment de la définition du processus. En effet, nous constatons que la flexibilité introduite au niveau structurel du processus par [Lezoche 08] et [Chaâbane 09] est prévue au moment de la modélisation. Ce qui fait que le seul moyen d'introduire du changement, en utilisant leurs approches, consiste à choisir une des versions ou une des adaptations prévues selon le contexte d'utilisation. Ceci implique le besoin d'anticiper tous les scénarii d'utilisations possibles, dès la phase de modélisation, ce qui n'est jamais envisageable. Cette limite a été partiellement résolue dans les travaux proposés par [Morley 05-b] et [Boukadi 09a]. Ces deux approches permettent d'introduire le changement au niveau de l'instance du processus à travers la spécification partielle du processus (cf. Niveau du changement : Instance dans le Tableau 4). En fait, le travail de Morley propose d'affecter les agents qui permettent de mettre en œuvre les activités décrites par des buts qu'au moment de l'exécution. De même, pour le travail de Boukadi qui permet de modéliser un processus abstrait en décrivant les activités par leur contexte d'utilisation et en découvrant dynamiquement les services adéquats au contexte qu'au moment de l'exécution du processus. Les travaux de [Pourraz 07a], [van der Aalst 04a] et du moteur CRISTAL d'Agilium permettent aussi d'introduire de la flexibilité au niveau de l'instance du processus. Ces travaux se distinguent des autres approches car contrairement aux autres approches ils proposent une évolution dynamique. Le travail de Pourraz le permet en intégrant des adaptations non planifiées à base de mécanismes de substitution. De même, YAWL le permet sous forme de traitement d'exception. Enfin, Agilium le permet en se basant sur les mécanismes d'adaptation dynamique du moteur CRISTAL. Ceci dit, que le

changement introduit au niveau de l'instance par ces travaux ce produit au moment que le processus initial est toujours en cours d'exécution.

- **Degré de changement** : Certaines approches telles que [Lezoche 08] et [Chaâbane 09] permettent de changer radicalement le processus. Nous constatons que le degré de changement au niveau de ces approches est radical (cf. Degré du changement : Radical dans le Tableau 4). Ce qui fait, qu'en utilisant ces approches, il est obligatoire de redéfinir le processus dès le début chaque fois qu'il y'a un besoin de changement non prévu. Par exemple, dans le cas du travail de Chaâbane, il faudra définir une nouvelle version du processus qui répond au changement. Quant au travail de Lezoche, il faudra redéfinir les règles d'enchaînement pour intégrer les besoins du changement. D'autres travaux tels que [Morley 05-b] et [Boukadi 09a] permettent de changer partiellement les processus. En fait, leurs travaux permettent de définir du comportement (activité) réutilisable. Dans le travail de Morley les activités à intégrer sont décrites par des buts et automatisés par des agents autonomes. Dans le travail de Boukadi, le comportement réutilisable est défini par des services découverts dynamiquement selon le contexte d'utilisation. Ceci permet de réduire le degré de changement du processus de radical vers partiel (cf. Degré du changement : Partiel dans le Tableau 4). De même, le travail de Pourraz [Pourraz 07a] permet de changer partiellement l'orchestration à travers la substitution de bout d'orchestration avec l'orchestration initiale. Ceci dit que qu'on peut aussi tout changer en utilisant son approche. Le travail de [van der Aalst 04a] permet aussi de changer partiellement le processus en traitant les changements sous forme d'exceptions. Toutefois, si le type d'exception n'existe pas, le changement ne peut être intégré proprement. Enfin, le moteur CRISTAL d'Agilium offre aussi des mécanismes permettant de changer partiellement le processus.

4.4. Synthèse

En se basant sur les différentes constatations développées ci-dessus, la piste qui nous paraît pertinente pour mettre en œuvre des processus de conception flexibles et dynamique est celle basée sur les approches à base de service. En effet, les approches à base de services constituent une avancée permettant d'organiser et d'utiliser des savoirs-faires sous forme de services dont les caractéristiques principales sont le faible couplage et la réutilisation [Pourraz 07]. La propriété de couplage faible implique qu'un service n'appelle pas directement un autre service. En effet, les services seront composés et gérés par une fonction d'orchestration [Collet 06]. La réutilisation d'un service est alors plus facile, du fait qu'il n'est pas directement lié aux autres services de la composition. L'utilisation du concept de service permettra ainsi d'exprimer les activités des processus de conception de produits manufacturés sous formes de services réutilisables. Ainsi, ces services peuvent être réutilisés, organisés et rapidement composés pour définir des processus de conception de produits manufacturés. De plus, les services composés, appelés services composites, constituent des services plus complexes à forte valeur ajoutée [Casati 01] [Benatallah 02]. Ces services peuvent participer à la définition d'autres processus tout en masquant la complexité des sous processus.

Par ailleurs, il existe plusieurs langages d'orchestration permettant la définition d'application ou de processus par composition de services existants. Certains de ces langages sont dynamiques et permettent de mettre en œuvre des processus évolutifs. En effet, l'évolution dynamique des orchestrations de services présente un réel challenge [Papazoglou 06] permettant de faire face au triptyque coût-qualité-délai et de favoriser l'agilité de l'entreprise et la flexibilité de ses processus qui semblent incontournables. Ainsi, en plus d'une approche orientée services, nous favorisons l'utilisation des mécanismes d'orchestration dynamique tels que ceux définis dans [Pourraz 07a]. Ceci, nous permettra d'intégrer facilement de nouveaux services au processus de conception de produits manufacturés pour prendre en compte les changements imprévus. Ce qui nous permettra en contre

partie de changer pas uniquement la partie fonctionnelle mais aussi bien la partie structurelle des processus.

5. Conclusion

Ce premier chapitre a permis de situer le contexte de notre travail. Nous avons pu parcourir le métier de la conception de produits manufacturés et constater que ce domaine est caractérisé par sa complexité, le nombre énorme de ces intervenants et la diversité de leurs connaissances. Nous avons pu voir que ces caractéristiques impactent directement les processus de ce domaine, à savoir les processus de conception de produits manufacturés. D'où le besoin de les automatiser afin de faciliter la communication entre les différents intervenant et de gérer le flux d'informations de ces processus. Une autre caractéristique, qu'on a pu soulever du domaine de la conception est l'évolution continue des contraintes et exigences de ce domaine. Ceci induit un besoin de support flexible aux activités des processus de conception de produits manufacturés afin qu'ils puissent s'adapter facilement aux changements imprévus. De plus, nous avons soulevé une exigence liée aux acteurs métier de ce domaine, responsable de la définition et l'exécution de ces processus. Ces acteurs métier ont besoin de manipuler des modèles de processus simples et expressifs en utilisant des concepts métier.

L'analyse de la pratique industrielle en termes de modélisation et de support de ces processus a conclut que les modèles actuels ne sont pas assez expressifs et compréhensibles par les personnes du domaine métier. Cette analyse a souligné aussi que les systèmes utilisés comme support aux activités des processus de conception souffrent d'un manque de flexibilité qui permettra de doter les processus de l'entreprise de l'agilité nécessaire. Il en résulte un ensemble de difficultés méthodologiques, organisationnelles, économiques et techniques auxquelles doivent faire face les industriels. Ceci nous a mené à étudier, dans la section 4, les travaux offrant un support à la problématique de flexibilité des processus, afin de dégager leurs avantages et limites vis-à-vis de notre problématique et principalement aux critères de flexibilité que nous avons fixées au début de la section 4. L'analyse de ces approches nous a permis aussi de tirer les bonnes pratiques que nous pouvons réutiliser pour la résolution de notre problématique de mise en œuvre de processus flexible dans les systèmes PLM. Ainsi, la piste qui permet d'apporter des réponses à notre problématique est celle des approches à base de services. Cette piste nous semble intéressante car elle permet d'une part de présenter les activités du domaine de la conception sous forme de services. Ceci permettra de favoriser l'agilité de l'entreprise à travers la réutilisation de traitements définis sous forme de services faiblement couplés. D'autre part, elle permettra l'automatisation des processus sous forme d'orchestration de services prêts à l'emploi. De plus, l'approche orienté service permettra de prendre en compte des changements imprévus à travers l'utilisation de langages d'orchestration évolutifs.

C'est l'objet du chapitre suivant (chapitre II) de développer les concepts liés aux approches à base de services (*i.e* service et composition de services) et de décliner leurs usage vis-à-vis de notre problématique. Ceci nous permettra d'approcher la problématique et de présenter notre approche.

Chapitre II :

Etat de l'art et Démarche pour la Gestion flexible des Processus de Conception de produits manufacturés

1. Introduction

Comme nous l'avons noté dans le chapitre précédent (Chapitre I), la gestion des processus de conception de produits manufacturés reste un sujet complexe et souffre de deux problèmes majeurs. Le premier concerne la définition des processus de conception de produits manufacturés. En effet, l'expressivité des modèles de processus n'est pas vraiment celle adaptée aux acteurs métiers du domaine de la conception. Les spécificités des processus de conception impliquent des formalismes de modélisation simples et bien définis. Le deuxième problème est celui lié au support à l'exécution des processus. Le domaine de la conception nécessite un support permettant l'automatisation des processus associés. De plus, l'exécution souffre d'un manque de réutilisation des tâches répétitives et de flexibilité qui doit doter l'entreprise de l'agilité nécessaire.

L'analyse des approches de flexibilisation des processus, dans le chapitre précédent, a mis en avant la pertinence du concept de service pour répondre aux problèmes cités là-haut. Le concept de service permet d'organiser et d'utiliser des savoirs-faires sous forme de services dont les caractéristiques principales sont le faible couplage et la réutilisation [Pourraz 07]. Ces services sont composés et gérés par une fonction d'orchestration [Collet 06]. Ceci permettra d'adresser les besoins en termes de réutilisation des tâches répétitives et de flexibilité d'enchaînement des tâches vu qu'un service n'est pas directement lié aux autres services de la composition. De plus, les services composés constituent des services plus complexes à forte valeur ajoutée [Benatallah 02] qui peuvent participer à la définition d'autres processus tout en masquant la complexité des sous processus. Ceci, permettra de définir des processus simples tout en masquant la complexité des sous processus.

C'est l'objet de ce chapitre de développer les approches orientées services afin de vérifier leurs pertinences vis-à-vis des spécificités et exigences des processus de conception de produits manufacturés mentionnées ci-dessus. Ce chapitre est organisé en trois parties. La première partie (§2) a pour objectif de présenter un état de l'art sur les approches orientées services afin de décliner leurs pertinence par rapport à notre problématique. La deuxième partie (§3) a pour objectif de se focaliser sur la notion d'orchestration de services comme support flexible aux processus de conception de produits manufacturés. Nous terminons ce chapitre par la présentation de notre démarche à base de services pour la flexibilisation de processus de conception de produits manufacturés (§4).

2. État de l'art sur l'orientation services

2.1. L'approche orientée services

L'une des approches les plus courantes aujourd'hui est l'approche orientée services connue sous l'appellation SOA. Cette approche, permettant de faire l'usage de fonctionnalités réutilisables dont le but de les rassembler pour répondre à divers besoins, est de nos jours utilisée dans plusieurs domaines. Dans le domaine de génie logiciel, cette approche est utilisée pour développer des applications agiles. L'application à développer est ainsi décomposée en modules indépendants, et ceux-ci sont par la suite rassemblés dans une plate forme d'exécution pour répondre aux besoins initiaux. Dans le domaine de E-Commerce, l'SOA est considérée comme une architecture qui vise à transformer le Web en une énorme plate-forme de services faiblement couplés et automatiquement intégrables. Elle offre un modèle à base de service en ligne permettant de construire une application par composition de services définis par des interfaces uniformes et standardisées. En même temps, l'acceptation et la popularité croissantes de l'approche SOA font émerger une demande d'extension du paradigme logiciel au monde de l'entreprise pour automatiser par composition de services, les processus intra et inter-entreprises tels que les ventes, les chaînes de production, les ressources humaines entre autres. En effet, le concept de service a trouvé son essor dans la construction d'applications à base de services et ne cesse d'attirer les entreprises industrielles [Ramus 03]. Ceci permettra la définition de processus collaboratifs, en permettant de présenter les services et sous-processus des entreprises sous forme de composants réutilisables appelés services.

Ce paradigme n'est pas nouveau. Les services sont apparus comme une suite logique des composants logiciels et des approches d'intégration d'applications à base de composants [Pfister 96]. L'idée des approches à base de composants était aussi la construction d'applications complexes par l'assemblage de blocs préfabriqués, appelés composants [Szyperski 02]. L'architecture à base de composants a permis le développement rapide et évolutif d'applications complexes et distribuées. D'ailleurs, beaucoup de travaux ont été dédiés à la modélisation de systèmes à base de composants [Wendt 05] [Hepner 06]. Cependant, cette architecture impose sa propre infrastructure et une liaison forte et figée entre les fonctionnalités offertes par les composants et leurs clients. Les applications construites à base de cette architecture sont donc monolithiques. Ce problème est dû principalement au fort couplage qui provient de la définition explicite des liens entre les interfaces fournies et requises des composants constituant l'application. C'est cette caractéristique qui différencie aujourd'hui les approches à base services des précédentes approches. Les approches à base de services s'appuient sur les principes des approches à base de composants tout en y ajoutant de nouveaux principes. Dans les approches à base de services une application est réalisée par composition de services qui sont mis à disposition par divers fournisseurs [Papazoglou 03]. Un des principaux bénéfices de cette approche par rapport à celle à composants est le faible couplage entre les unités de composition appelées services. Les liens exprimés entre les interfaces fournies et requises de l'approche à base de composants, tels que ceux définis dans les applications de contrôle industriel [Khalgui 12] [Rockwell 06], disparaissent dans l'approche à base de services. Ainsi un service est indépendant de l'état et du contexte d'exécution des autres services.

Notons que les approches orientées services ne représentent pas une technologie mais une façon de concevoir et de déployer ses applications et processus. Plus précisément, il s'agit de structurer ses applications selon une approche basée sur le principe de "services" et non plus, comme par le passé, sur la base d'applications monolithiques. Le concept de service est considéré comme la brique de base de l'architecture orientée services. Cette architecture organise les fonctionnalités élémentaires contenues dans un système en services qui peuvent être rapidement combinés et réutilisés pour répondre à divers besoins.

Une fois fournis, les services doivent être publiés pour pouvoir être découverts et utilisés. Ceci est assuré grâce à un cadre d'interaction définie par l'architecture orientée service (SOA) (cf. Figure 4). En fait, celui qui définit les services, appelé fournisseur, met à disposition des utilisateurs des services, appelés consommateurs, les informations nécessaires (*i.e.* interface de service) pour pouvoir les utiliser. La publication des services se fait auprès d'un registre de services (ou catalogue de service) qui joue un rôle d'intermédiaire entre le fournisseur et les consommateurs de services. Cette pratique dote l'approche à base de service d'avantages importants. D'abord les services sont stockés dans un catalogue et prêts à l'emploi si besoin. De plus, comme le catalogue de service centralise l'information des services, les clients n'ont pas besoin de s'adresser au fournisseur pour les chercher. La liaison avec le fournisseur n'est faite qu'une fois le service à utiliser est choisi. En plus, seule la description est partagée entre les utilisateurs. Ainsi, l'utilisateur n'a pas à se soucier de la réalisation technique du service lors du choix du service dans le catalogue.

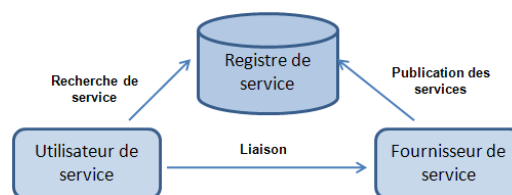


Figure 4: Cadre d'interaction selon SOA

En choisissant certaines fonctionnalités d'un ensemble de services rendus disponibles dans le catalogue, ces services peuvent être composés pour développer des processus⁵ ou d'autres services plus riches en termes de fonctionnalités.

2.2. Les concepts principaux de l'orientation services

2.2.1 Concept de service

Au cœur du domaine SOA se trouve le concept de service. Le concept de service est actuellement le sujet de définitions très variées. Historiquement, ce concept été de nature purement logiciel. Ensuite, il est devenu applicable à la fois au niveau logiciel et métier. En effet, nous dégageons de l'étude de la littérature portant sur la notion de services, deux visions principales⁶ : la vision logicielle et la vision métier.

En ce qui concerne la vision logicielle, la définition donnée par [Papazoglou 03] est très largement référencée dans la littérature. Elle définit le service comme *un ensemble d'applications modulaires auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées. Un service peut effectuer des actions allant de simples requêtes à des processus métiers complexes*. Celle de [OGSA 05] définit un service comme *A software component participating in a service-oriented architecture that provides functionality and/or participates in realizing one or more capabilities*. Un service est défini dans [OASIS 06] comme *a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface*. Une autre définition du service technique est celle donnée par [Fareghzadeh 08], *is a technology specific functionality, provide reusable functions related to processing data within new or legacy application*. [Nurcan 09] le définit comme *a special kind of interface for an encapsulated unit of software*. Selon [Papazoglou 07], un service est défini comme suit: *Services, which are well defined, self-contained modules that provide standard business functionality and are independent of the state or context of other services*. Dans [Fremantle 02], l'auteur liste les caractéristiques d'un service, à savoir :

- la modularité : un service est modulaire et réutilisable. Il est possible de construire des applications ou des services plus complexes à partir d'autres services,
- la disponibilité : un service est disponible est utilisable par des clients,
- la description : un service possède une description qui est lisible. La description sert à spécifier l'interface du service et sa localisation,
- l'indépendance d'implémentation : un service a une claire séparation entre sa description (ou interface) et son implémentation.
- la publication : la description du service est publiée dans un annuaire qui sera consultable par les clients du service.

De ces définitions, nous ressortons une idée principale, à savoir qu'un service logiciel est une entité logicielle qui permet d'exposer une ou plusieurs fonctionnalités réutilisable et faiblement couplées. Selon ces définitions, cette entité possède les propriétés suivantes ; elle est bien définie, auto-contenue, indépendante de l'état des autres services, ainsi qu'elle est caractérisée par l'indépendance de sa description (i.e. interface) par rapport à son implémentation. En effet, les descriptions de services sont publiées dans un annuaire pour pouvoir être utilisés par les utilisateurs de services. L'utilisateur du service n'a donc pas à connaître le détail de la réalisation technique du service. Ce découplage entre la

⁵ Un processus ici décrit comment les services peuvent interagir entre eux pour remplir une fonction complexe.

⁶ Certains distinguent une troisième vision de service : la vision interactionnelle. D'un point de vue interactif, un service est défini comme *une collection de tâches utilisateur. Ces tâches sont regroupées de manière à permettre l'exécution de l'activité requise par un utilisateur* [Chevrin 06].

description du service et sa réalisation technique masque la complexité due aux langages d'implémentation et à l'infrastructure de déploiement qui est souvent complexe.

D'un point de vue métier, le concept de service est actuellement le sujet de définitions très variées. Une première définition métier du concept service est énoncée par [Pallos 01], elle définit un service comme *un groupement logique de composants requis pour satisfaire une demande métier*. [Vernadat 07] définit un service métier comme *a discrete piece of functionality that appears to be atomic and self contained from a point of view of service caller. Uniquely identified within the entreprise and has an owner*. [Kaabi 07a] présente une nouvelle vision pour définir un service du niveau métier à savoir la vision intentionnelle. Elle définit le service du niveau métier comme *un service intentionnel qui permet d'atteindre l'intention ou le but d'un acteur dans un contexte donné*. Une autre proposition [Cauvet 08] définit le service métier comme *une unité réutilisable qui encapsule un ou plusieurs fragments d'un processus métier et qui vise à satisfaire des buts métiers*. [Boukadi 09a] considère qu'un service métier offre une fonction métier. Cette fonction peut être de moyenne granularité telle que les fonctionnalités liées à la gestion d'une commande client, calcul du montant d'une commande, etc. Elle peut aussi encapsuler des sous processus voire des processus métiers (c'est-à-dire la logique de l'enchaînement des étapes du processus). Celle proposée par [Han 09] considère un service comme *une abstraction du niveau métier implémentant les processus métiers de l'entreprise*.

Nous constatons, en se basant sur ces définitions qu'il n'y a pas une définition ni une vision claire de ce que peut être un service au niveau métier. Selon ces définitions, le service métier peut satisfaire un but métier, encapsuler un fragment du processus ou encore implémenter un processus métier. Les buts et sous processus peuvent avoir des granularités différentes et par conséquent les services métiers peuvent avoir aussi des granularités différentes. En effet, un service implémentant un sous processus peut faire partie d'un autre processus plus complexe. Ceci constitue l'un des avantages des approches à base de services qui permettent de masquer la complexité de processus sous-jacents et de construire d'autres processus plus élaborés sans se soucier de la complexité des sous-processus exposés sous forme de services [Casati 01] [Benataallah 02]. Le niveau de granularité du service métier n'est donc pas bien défini. Nous pouvons conclure que cette multitude de définitions proposées pour le service métier, montre d'une part que le concept de service métier n'a pas encore atteint sa maturité. D'autre part, elle montre que la définition du service métier dépend fortement de la vision de celui qui l'énonce. Notre constatation est confirmée par la vision de [Kohlborn 09] qui considère que le service métier peut exister avec plusieurs niveaux de granularité. Nous proposons notre vision du service du niveau métier dans le chapitre III de ce manuscrit. Toutefois, nous constatons que les définitions du service métier regroupent pratiquement les mêmes caractéristiques que celles des services logiciels telles que la réutilisation, la modularité et l'autonomie (i.e. faible couplage).

2.2.2 Composition de services

Dans cette section, nous nous intéressons au deuxième concept de base de l'orientation services, la composition de services. La composition de services a été définie comme étant : *la capacité d'offrir des services à valeur ajoutée en combinant des services existants* [Casati 00]. Le résultat d'une composition de services peut être une application ou un autre service nommé service composite [Alonso 03]. Cette propriété fait que la composition de services est récursive ou hiérarchique, c'est-à-dire que des services atomiques ou composites peuvent être intégrés pour implémenter la logique d'autres services composites [Khalaf 03]. La composition de services est valable pour les services logiciels ou métier. Au niveau métier, la composition de services sert à décrire la collaboration entre les fonctionnalités de certains services choisis pour atteindre un objectif métier (ou définir un processus). Au niveau logiciel, elle permet d'atteindre un objectif particulier, par l'intermédiaire de primitives de contrôles (boucles, test, traitement d'exception, etc.) [Kellert 06]. La spécification de la

composition est faite à ce niveau dans un langage interprété par un moteur d'exécution qui permet d'invoquer les services dans l'ordre spécifié.

La composition de services permet donc de combiner les fonctionnalités de plusieurs services dans un même service (ou processus métier) afin de répondre à des demandes complexes qu'un seul service ne peut pas réaliser. La composition de services nécessite les étapes suivantes. Tout d'abord il faut découvrir les services qui peuvent répondre aux besoins de la composition. Ensuite, il faut définir l'ordre d'invocation des différents services et organiser l'interaction entre les services qu'on veut composer : il s'agit alors d'orchestration et de chorégraphie. En effet, La composition de services englobe deux concepts différents : la chorégraphie et l'orchestration [Peltz 03]. La chorégraphie permet de décrire la collaboration entre plusieurs services en termes de messages échangés et de règles d'interaction entre les participants d'un processus. Elle décrit, d'un point de vue global, le comportement observable des différents services au sein d'un processus métier [Papazoglou 06]. L'orchestration, quant à elle, fait référence aux processus exécutables ; elle décrit comment les services peuvent interagir entre eux selon une perspective opérationnelle, avec des structures de contrôle, incluant l'ordre d'exécution des interactions [Pourraz 07].

Notion de chorégraphie

La chorégraphie est, par nature, collaborative. Un service peut avoir plusieurs descriptions afin de modéliser les différents scénarios collaboratifs auxquels il participe. En effet, la chorégraphie permet la description du flux des messages échangés par un service, lors de son interaction avec d'autres services. Elle ne concerne donc qu'un seul service et permet la collaboration point-à-point entre plusieurs services (cf. Figure 5) [Esper 10]. Elle décrit les différents messages qui transitent entre les différents acteurs (les services) d'un processus et donne ainsi une vision abstraite des échanges au sein du processus. Elle permet de définir le comportement externe, en termes de messages échangés, d'un ensemble de services qui collaborent pour atteindre un objectif commun. En d'autres termes, la chorégraphie décrit les interactions possibles entre les différentes opérations d'un service.

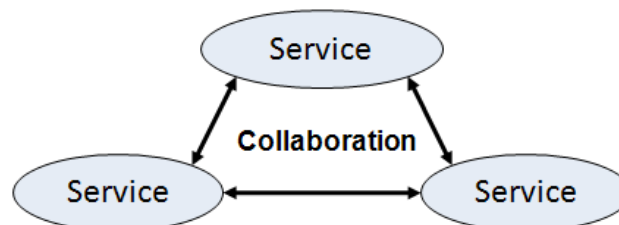


Figure 5 : Chorégraphie de services

La description de la chorégraphie est faite indépendamment de la logique opérationnelle. Elle est orientée message et ne définit en aucun cas une description exécutable d'un processus comme le permet l'orchestration de services [Pourraz 07]. Cependant, elle sert de première spécification au processus concret (i.e. orchestration) à réaliser.

Notion d'orchestration

L'orchestration décrit comment les services peuvent interagir entre eux selon une perspective opérationnelle, avec des structures de contrôle, incluant l'ordre d'exécution des interactions (souvent qualifiée de logique métier) [Pourraz 07]. Les interactions décrites dans la description de l'orchestration peuvent s'étendre à plusieurs applications (exposées sous forme de services) et/ou organisations (qui offrent leur métier sous forme de services) et résultent en un processus transactionnel de longue durée [Peltz 03]. L'identité des services participant à l'orchestration est alors

connue. L'orchestration décrit aussi et les messages échangés et les conditions d'activation des opérations de services. Elle donne ainsi une vision concrète qui permet l'expression d'un processus exécutable (cf. Figure 6). Nous dégagons ainsi le rôle principal d'une orchestration, à savoir la description de la logique métier et l'ordre d'exécution des opérations de services.

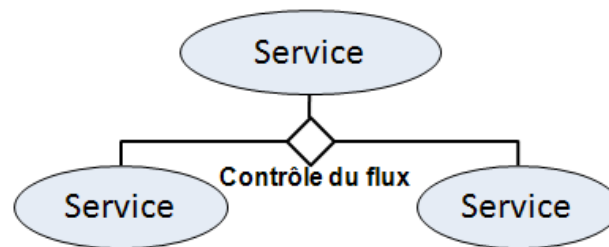


Figure 6 : Orchestration de services

Un service complexe peut lui-même, être vu comme le résultat d'une orchestration. En d'autres termes, le processus exécutable décrit par le biais d'une orchestration peut être déployé comme une opération de service et ainsi faciliter son utilisation et sa réutilisation au sein d'autres orchestrations. Le nombre de niveaux d'imbrication est alors illimité.

2.3. Synthèse

La présentation de l'approche orientée services et les concepts qui lui sont liés (service et composition de services), met en avant plusieurs caractéristiques qui semblent pertinents pour notre problématique.

Comme nous l'avons pu voir dans cette première section, la réutilisation des fonctionnalités prêtes à l'emploi est l'une des caractéristiques majeures qui motive l'utilisation des approches orientées services. De façon similaire aux approches par objet et par composant, l'approche par service cherche à fournir un niveau d'abstraction encore supérieur, en encapsulant des fonctionnalités dans des services et en permettant la réutilisation de services déjà existants [Collet 06]. En effet, les services offrent des fonctionnalités réutilisables. Ces fonctionnalités peuvent être réutilisées et combinées pour définir de nouveaux processus de conception de produits manufacturés ou participer à l'adaptation de processus existants. D'ailleurs, plusieurs auteurs comme [Vernadat 07] et [Cauvet 08] affirment la pertinence des approches à base de service pour assurer une flexibilité considérable du système d'information de l'entreprise. En effet, l'architecture orientée services organise les fonctionnalités élémentaires contenues dans les systèmes d'informations en des services réutilisables qui peuvent être rapidement combinés et réutilisés pour répondre aux besoins métier [Papazoglou 07].

De plus, comme nous l'avons énoncé au début de cette section, les services sont caractérisés par leur faible couplage. Le faible couplage a pour objectif d'introduire le minimum de dépendances entre les services pour permettre d'assembler ceux-ci aisément [Collet 06]. Il s'agit donc de favoriser la réutilisation de services existants, ainsi que leur combinaison afin de répondre rapidement et à faible coût à de nouveaux besoins métier. Le maître mot est ici le découplage. Cette notion peut prendre diverses connotations telles que réduire le couplage entre services, pour une meilleure réutilisation ou encore réduire le couplage entre l'utilisateur d'un service et une implémentation spécifique de ce service, pour une meilleure adaptabilité [Margerie 05]. Le faible couplage est ainsi la deuxième caractéristique qui motive l'utilisation des approches orientées services pour résoudre notre problématique. En d'autres termes le faible couplage permet de réduire la dépendance entre les différents services constituant le processus et permettra ainsi une meilleure flexibilité et adaptabilité des processus de conception définis à base de composition services. La réutilisation ou le remplacement d'un service est alors plus facile, du fait qu'il n'est pas directement lié aux autres services.

Notons aussi que les services sont caractérisés par l'indépendance de leur description par rapport à leur implémentation. Comme nous l'avons dit précédemment, ceci permet de réduire le couplage entre l'utilisateur d'un service et la façon de mise en œuvre du service, ce qui permettra en retour une meilleure adaptabilité [Margerie 05] et une simplicité dans l'utilisation de service. Les interfaces et les données exhibées par les services peuvent être exprimées en termes métiers (propres à un domaine d'application) [Esper 10]. L'utilisateur du service manipule donc des interfaces faciles et compréhensibles par des acteurs n'ayant pas des connaissances techniques. Ainsi les personnes métier du domaine de la conception peuvent utiliser facilement les services pour définir leurs processus de conception de produits manufacturés.

Mis à part ces trois caractéristiques, l'approche orientée service offre des mécanismes de composition de service. Comme nous l'avons présenté, la composition de services permet de combiner les fonctionnalités de plusieurs services dans un même service (ou processus métier) afin de répondre à des demandes complexes qu'un seul service ne peut pas réaliser [Alonso 03]. Cette composition de service est facilitée par la modularité des services. Cette caractéristique de modularité rend possible la définition des services plus complexes ou encore des processus à partir d'autres services [Fremantle 02]. La composition de service permet ainsi de masquer la complexité des sous processus en les exposants sous forme de services composites. Ces sous processus peuvent à leur tour participer à la définition de nouveau processus plus complexe. Selon [Esper 10], la possibilité de publication de fragments de processus sous forme de services composite permet de diffuser des traitements répétitifs et d'organiser rapidement une chaîne de services répondant aux besoins exprimés.

Nous constatons de la confrontation des caractéristiques des approches orientées services avec les besoins du domaine de la conception de produits manufacturés, que le mécanisme de composition de services et les caractéristiques principales, par lesquelles le concept de service est défini, permettent de résoudre les problèmes liés :

- à la lenteur de développement des processus de conception de produits manufacturés et d'intégration du changement. Ceci peut être assuré grâce à **la propriété de réutilisation** qui caractérise le concept de service. **La capacité de réutilisation et d'automatisation des tâches répétitives et de l'enchaînement des activités** peut être assurée à travers l'utilisation des approches orientées services.
- au manque de support pour l'exécution des processus de conception de produits manufacturés, dans les systèmes PLM, qui soit flexible et facilement configurable afin de pouvoir intégrer facilement les changements imprévus. Ceci peut être assuré grâce à **la propriété de faible couplage** qui caractérise le concept de service. La réutilisation ou le remplacement d'un service dans une composition de services est alors plus facile, du fait qu'il n'est pas directement lié aux autres services. **La flexibilité des processus et de leurs outils support** peut être assurée à travers l'utilisation des approches orientées services.
- au manque de modèle permettant la définition de processus de conception de produits, qui soit à la fois flexible, expressif et facilement manipulable par des acteurs métiers. Ceci peut être assuré grâce à **la propriété de découplage de la description de service par rapport à son implémentation**. L'utilisateur du service peut donc manipuler des interfaces faciles et compréhensibles par des personnes métier. De plus, la **composition de services** permet de masquer la complexité des processus composites. Ces processus peuvent être réutilisés pour la définition de processus plus complexes. Le besoin en termes de **formalismes de représentations simples et expressifs** peut être assuré à travers l'utilisation des approches orientées services.

Nous pouvons conclure que l'approche à base de service est pertinente pour notre problématique. Elle permet aux personnes métier de définir des processus simples et expressifs. De plus, elle permet au

système support aux activités de la conception d'acquérir une certaine flexibilité en lui offrant la possibilité de répondre aux besoins du marché d'une manière agile. Ceci à travers la réutilisation de fonctionnalités et de sous processus exposés sous forme de services réutilisables et faiblement couplés. Cette vision permettra d'offrir au système PLM, support aux activités des processus de conception de produits manufacturés, la flexibilité nécessaire pour pouvoir supporter les processus de conception à caractère évolutif. Elle permettra la réutilisation de services existants pour mettre en œuvre le changement en profitant de la caractéristique du faible couplage entre les services et des mécanismes de composition de services.

Comme nous l'avons dit précédemment la composition de services englobe deux concepts différents. La chorégraphie permettant de décrire, d'un point de vue global, le comportement observable des différents services au sein d'un processus métier [Papazoglou 06]. L'orchestration permettant de décrire comment les services peuvent interagir entre eux selon une perspective opérationnelle, avec des structures de contrôle, incluant l'ordre d'exécution des interactions [Pourraz 07]. Nous nous focalisons dans la section suivante sur les approches pour l'orchestration puisqu'on s'intéresse aux processus exécutables. En effet, les processus de conception définis, au niveau métier, sous forme de composition de services doivent avoir un support à l'automatisation pour pouvoir être exécutés (cf. section (§2.3) du chapitre I). L'orchestration le permet puisqu'elle donne une vision concrète du processus et permet l'expression d'un processus exécutable.

3. Orchestration de services

Nous focalisons dans cette section sur la notion d'orchestration de services comme support aux processus de conception de produits manufacturés. Nous présentons les différentes approches et langages qui permettent l'automatisation de l'orchestration.

3.1. Présentation des approches et langages d'orchestration

La spécification de la composition sous forme d'orchestration est faite dans un langage interprété par un moteur d'orchestration. La responsabilité du moteur d'orchestration est d'invoquer les services dans l'ordre spécifié, de faire le routage des données et de maintenir et gérer l'état des activités et du service composite [Pourraz 07]. À cet égard, certains travaux ont été proposés. Ces travaux proposent des langages permettant de décrire les orchestrations de services. Les langages proposés possèdent des structures de contrôle qui ont pour vocation de décrire le déroulement des processus. Par ailleurs, une caractéristique importante que peu de langages supportent aujourd'hui est l'adaptation dynamique des orchestrations de services. On entend par adaptation ou évolution dynamique, la capacité de modifier l'orchestration au cours de son exécution sans avoir besoin de l'arrêter. Certains des langages proposés supportent cette caractéristique et d'autres non. Nous présentons dans ce qui suit les deux types de langages d'orchestration.

Standards et langages d'orchestration ne supportant pas l'évolution dynamique de la composition

Plusieurs standards proposent des langages permettant de décrire l'orchestration des services. Les plus connus sont XLANG, WSFL, BPEL4WS et BPML.

XLANG est un langage qui a été créé par Microsoft [Thatte 01]. Cette spécification était dédiée à la création de processus métier sous forme d'orchestration de services Web. XLANG supportait le séquençement, la mise en parallèle ou encore les branchements conditionnels pour décrire comment combiner plusieurs services web afin de produire un processus métier multi-participant. Il s'appuie sur la spécification WSDL (Web Service Description Language) pour décrire l'interface des différents services utilisés au sein d'un processus.

WSFL (Web Service Flow Language) est un langage proposé par IBM. Il est basé sur une syntaxe XML permettant de décrire des orchestrations de services web. Il permet de décrire à la fois le modèle de flux et le modèle global d'un processus [Leymann 01]. Le modèle de flux correspond au séquençement des activités du processus alors que le modèle global permettait la corrélation entre une activité et une instance d'opération de service Web. Une description WSFL peut être exposée par le biais d'une interface WSDL afin de permettre une décomposition récursive d'un processus (utilisation d'une description WSFL comme activité d'une autre description WSFL).

WS-BPEL (Web Services Business Process Execution Language) ou **BPEL4WS** (Business Process Execution Language for Web services) est considéré aujourd'hui comme le standard de la description des orchestrations de services [TC 07]. WS-BPEL est basé sur XML permettant de décrire la logique de contrôle requise pour la coordination entre les services participants dans un processus métier. Cette grammaire peut ensuite être interprétée et exécutée par un moteur d'orchestration. Ce moteur coordonne les activités du processus et fournit des mécanismes de compensation en cas d'erreur. Un processus décrit en WS-BPEL comprend deux types d'activités : des activités basiques et des activités structurées. Les activités basiques décrivent les étapes élémentaires du comportement du processus comme l'invocation d'une opération de service (Invoke), la réception de message (Receive), l'assignation d'une variable (Assign), etc. Les activités structurées définissent la logique de contrôle du processus. Elles permettent de spécifier la manière dont le processus est exécuté en orchestrant les activités basiques. La séquence, le choix conditionnel (if) et le parallélisme (flow) sont des exemples d'activités structurées. Enfin, tout processus BPEL4WS peut être exposé comme un service Web et donc peut être utilisée dans la description d'une autre orchestration.

BPML (Business Process Management Language) est quant à lui un méta-langage de description de processus métier exécutables [Arkin 03]. BPML a été proposé par BPMI (Business Process Management Initiative). Basé sur une syntaxe XML, ce langage peut être utilisé pour décrire les implémentations privées (l'orchestration) d'un processus. BPML partage de nombreux concepts communs avec BPEL4WS. Il supporte les activités (basiques et structurées), les variables, la notion de rôle et des mécanismes de gestion des transactions et des exceptions. Les concepts de BPML comme l'activité permettant d'instancier un processus et attendre sa terminaison (call) ou l'activité permettant d'instancier un processus sans attendre sa terminaison (spawn) n'ont pas d'homologue dans BPEL4WS [Mendling 03].

Standards et langages d'orchestration supportant l'évolution dynamique de la composition.

Plusieurs auteurs considèrent que l'évolution dynamique de la composition (orchestration) devient un réel challenge [Papazoglou 06a]. L'évolution dynamique permet de changer l'orchestration en cours d'exécution. Ce critère apporte de la flexibilité à la démarche de composition et lui permet de faire face à de nouveaux changements ou à des scénarios d'évolution. Nous présentons dans ce qui suit les tentatives en termes de langages supportant l'évolution dynamique.

π -Diapason [Pourraz 08] est un langage d'orchestration (basé sur le langage formel π -calcul), exécutable et offre la possibilité de faire évoluer dynamiquement une orchestration de services web grâce au concept de mobilité introduit par le π -calcul. En effet, la notion de mobilité supportée par le π -Calcul, permet de faire transiter un processus au travers d'un canal, afin de fournir un nouveau comportement à un processus en cours d'exécution. Cette évolution dynamique peut être planifiée ou non [Verjus 08]. L'évolution dynamique planifiée peut se faire à travers la prise en compte d'une remontée d'alerte par l'orchestration elle-même (une demande d'évolution) ou encore à travers la substitution d'une partie du comportement de l'orchestration (déterminée pendant la phase de

conception). Afin de formaliser ces deux cas d'évolution, il faut insérer, à des endroits où l'on pense que l'orchestration peut évoluer, des points d'évolution au sein de la description de l'orchestration pendant la phase de conception. Ces différents points d'évolution dynamique planifiée sont gérés au niveau du langage.

Process eXtensible Language (PXL) [Verjus 11] est un langage d'orchestration algébrique basé sur le π -calcul, exécutable et qui a été développé essentiellement pour fournir une sémantique opérationnelle formelle (sans ambiguïté). Ceci permet d'avoir une interprétation exacte de l'orchestration par le moteur d'exécution. PXL est le successeur de π -Diapason. Il introduit de nouveaux mécanismes supportant l'évolution dynamique non planifiée des orchestrations de services [Pourraz 07b]. PXL est un langage structuré en six couches (cf. Figure 7). Cette architecture en couches permet à l'architecte de modéliser plus facilement les processus avec des construits de plus haut niveau d'abstraction. Chaque couche d'un niveau supérieur est exprimée en termes des couches inférieures. Par conséquent, une orchestration exprimée dans la couche AOSW peut être traduite systématiquement dans la couche π -calcul fournissant ainsi une description formelle du modèle de l'orchestration.

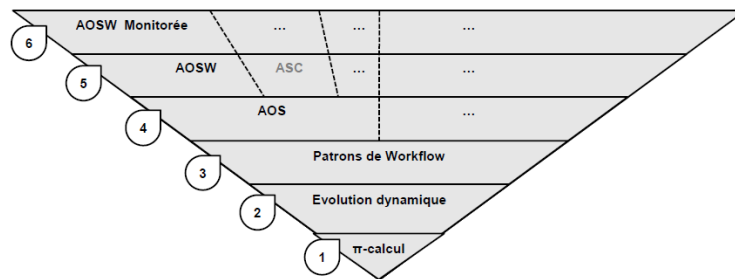


Figure 7 : Les différentes couches du langage PXL [Verjus 11]

Les travaux de [Ben Halima 08] ont proposé des mécanismes d'évolution dynamique planifiée pour WS-BPEL. Ils définissent une architecture prédictive auto-adaptative des services web. L'approche se base sur la supervision et l'analyse des attributs qualité. Une fois, une dégradation de qualité est détectée, la meilleure stratégie de réparation est identifiée et un plan d'évolution sera créé et inséré dans la base des plans. Une demande d'évolution est ainsi envoyée au module de reconfiguration qui cherche le plan d'évolution dans la base et l'exécute. Les actions d'adaptation possibles sont : la substitution du service responsable de la dégradation par un autre service équivalent ou la duplication du service initial avec un autre service équivalent (distribution de la charge entre le service initial et un autre service équivalent) [Pegoraro 08]. Cette duplication peut être matérialisée par l'exécution de deux substitutions. En fait, deux threads seront créés, un pour appeler le service initial et un autre pour appeler le service candidat à la substitution. Le service le plus rapide en réponse à la requête sera choisi.

Le travail de [Qiao 09] a proposé des mécanismes d'évolution dynamique planifiée pour WS-BPEL en enrichissant le langage par de nouveaux concepts. Il présente une architecture pour l'adaptation dynamique des orchestrations de services. L'approche consiste à changer dynamiquement le chemin d'exécution de l'orchestration lorsque cela s'avère nécessaire. Pour ce faire, les auteurs ont introduit un nouveau constructeur dans BPEL appelé « flexpath ». Ce constructeur permet de définir des chemins d'exécution multiples. Pendant la phase de conception, l'architecte définit les parties de l'orchestration qui sont susceptibles d'être modifiées au cours de l'exécution. Dans chacune de ces parties, un ensemble de chemins d'exécution alternatifs (flexpath) est spécifié. Durant l'exécution, l'adaptation est déclenchée au moment où les valeurs des attributs qualité sont jugées inacceptables.

Le mécanisme d'adaptation se charge de trouver le prochain constructeur flexpath et choisit le meilleur chemin d'exécution.

3.2. Comparaison des approches et langages d'orchestration

Les travaux présentés abordent plusieurs axes de recherche, à savoir :

- La composition des services afin de réaliser et créer de nouvelles fonctionnalités plus avancées,
- L'exécution des compositions ainsi formées et la synchronisation entre les différents services participant à la composition,
- L'évolution de ces compositions en cours d'exécution.

Afin d'analyser les travaux pertinents au regard de notre cadre de travail, nous les comparons (cf. Tableau 5) en fonction des critères et exigences que nous avons dégagés initialement (voir chapitre I), à savoir, la simplicité du modèle, l'automatisation et la synchronisation des tâches de processus et la flexibilité du système support.

Tableau 5: Comparaison des approches d'orchestration

	Simplicité et Expressivité	Communication /Synchronisation	Automatisation /Exécution	Flexibilité
XLANG	Expressive mais pas simple vu l'absence de couche métier	x	x	
WSFL	Expressive mais pas simple vu l'absence de couche métier	x	x	
WS-BPEL	Expressive mais pas simple vu l'absence de couche métier	x	x	
BPML	Expressive mais pas simple vu l'absence de couche métier	x	x	
π-Diapason	Expressive mais pas simple vu l'absence de couche métier	x	x	Evolution dynamique planifiée et non planifiée x
PXL	Expressive mais pas simple vu l'absence de couche métier	x	x	Evolution dynamique planifiée et non planifiée x
[Qiao 09] Extention WS-BPEL	Expressive mais pas simple vu l'absence de couche métier	x	x	Evolution dynamique planifiée
[Ben Halima 08] Extention WS-BPEL	Expressive mais pas simple vu l'absence de couche métier	x	x	Evolution dynamique

Simplicité et expressivité : les langages d'orchestration présentés offrent la possibilité de définir un processus complexe par composition de sous-processus exposés sous forme de services à part entière. Ceci permettra de masquer la complexité des sous processus et donc d'augmenter la lisibilité et la simplicité du modèle final de processus. De plus, ils offrent plusieurs flux de contrôles permettant d'exprimer la collaboration entre les services participant à l'orchestration. Ainsi, l'expressivité des processus de conception représentés sous forme d'orchestration de services peut être assurée. Néanmoins, nous constatons que ces langages n'offrent pas une couche proprement métier. Toutefois, vu que la composition se fait sur la base des interfaces des services et que ceux-ci sont caractérisés par l'indépendance de leur description par rapport à leur implémentation, il est possible de définir une telle couche métier. L'interface du service peut être ainsi définie en utilisant des termes métier ce qui permettra de simplifier la tâche du concepteur du processus.

Communication et synchronisation : nous remarquons que tous les langages et approches proposés permettent la spécification de la composition sous forme d'orchestration en précisant l'ordre d'exécution et les conditions d'activation des services. La synchronisation et la communication

(transfert de données) entre les différents services participant à l'orchestration peuvent être ainsi assurées en utilisant ces langages d'orchestration.

Automatisation / Exécution : nous constatons que tous les langages et approches étudiés sont couplés avec des moteurs d'orchestration permettant l'exécution des compositions spécifiés. Il est donc possible de définir des processus de conception de produits manufacturés exécutables en utilisant l'un de ces langages d'orchestration.

Flexibilité : Le critère de flexibilité exige le faite de pouvoir changer l'orchestration au cours de l'exécution pour faire évoluer le processus. En se basant sur l'état de l'art présenté dans la section précédente, nous distinguons deux types d'évolution possibles. L'évolution dynamique planifiée et l'évolution dynamique non planifiée [Cîmpan 05]. L'évolution dynamique planifiée prévoit d'avance (c'est-à-dire pendant la conception de l'orchestration) les changements à apporter sur l'orchestration. De plus, les instants où une évolution potentielle peut avoir lieu, sont fixés en avance. L'évolution dynamique non planifiée [Pourraz 07], quant à elle, consiste à modifier le modèle de l'orchestration en cours d'exécution à n'importe quel moment en intégrant n'importe quel scénario d'évolution. L'analyse des travaux présentés dans la section suivante, nous permet de constater que cette caractéristique de flexibilité est native dans le langage PXL [Verjus 11] puisqu'il supporte l'évolution dynamique non planifiée des orchestrations. Les travaux de [Qiao 09], [Ben Halima 08] supportent uniquement l'évolution dynamique planifiée. Ceci permettra une certaine flexibilité mais ne permet en aucun cas de remédier à un aléa imprévisible qui survient au cours de l'exécution. Enfin, le travail de [Pourraz 08] supporte les deux types d'évolution dynamique.

3.3. Synthèse sur les approches et langages d'orchestration

Nous avons vu précédemment (voir chapitre I) que les processus de conception de produits manufacturés évoluaient dans un contexte dynamique et évolutif. Ceci nécessite d'avoir une certaine agilité afin de réagir dynamiquement à des potentiels changements. La capacité de réutilisation et d'automatisation des tâches répétitives et de l'enchaînement des activités est une des exigences essentielles des processus de conception et leur système support. Comme nous l'avons noté dans la section 2 de ce chapitre, cette exigence peut être assurée en utilisant les approches orientées services. L'objet de cette troisième section est de se focaliser plus particulièrement sur la notion d'orchestration de services comme support aux processus de conception exécutables.

L'état de l'art que nous avons mené dans cette section, montre que la notion d'orchestration de services représente une réponse crédible au besoin de support exécutable pour les processus de conception. En effet, la spécification de la composition sous forme d'orchestration est faite dans un langage interprété par un moteur d'orchestration. La responsabilité du moteur d'orchestration est d'invoquer les services dans l'ordre spécifié, de faire le routage des données et de maintenir et gérer l'état des activités et du service composite [Pourraz 07]. Comme le montre le Tableau 5, tous les langages d'orchestration peuvent être interprétés et exécutés par un moteur d'orchestration et par conséquent de supporter les processus de conception spécifiés sous formes d'orchestration de services.

Une autre exigence des processus de conception de produits manufacturés implique que le système support aux processus de conception, doit être assez flexible pour pouvoir mettre en œuvre les changements imprévus. L'analyse que nous avons menée dans cette section a permis de montrer que certains langages d'orchestration supportent l'adaptation dynamique des orchestrations de services [Qiao 09] [Ben Halima 08] [Pourraz 08] et [Verjus 11]. Cette caractéristique est importante et nécessaire dans l'environnement imprévisible dans lequel évoluent les processus de conception. L'exigence qui porte sur le besoin de flexibilité des solutions support au processus de conception peut

être ainsi vérifiée en utilisant les langages permettant l'évolution dynamique non planifiée tels que PXL [Verjus 11] et π -Diapason [Pourraz 08]. Nous pouvons ainsi conclure que la notion d'orchestration va améliorer la flexibilité des processus de conception définis selon une approche orientées services.

Enfin, le Tableau 5 montre qu'aucun travail ni aucune approche ne remplit aujourd'hui les quatre critères de la problématique. Les personnes métiers du domaine de la conception ont besoin de formalismes simples et expressifs pour manipuler facilement les processus. Nous avons pu voir, à travers l'analyse des langages d'orchestration que ces derniers sont expressifs. Ils possèdent des structures de contrôle qui permettent de décrire le déroulement des processus de conception sous forme d'orchestration de services. Néanmoins, la simplicité du modèle n'est pas celle adaptée aux personnes métier du domaine de la conception. La plupart de ces langages sont basés sur une syntaxe XML pour décrire l'orchestration (WSFL [Leymann 01], BPML [Arkin 03], WS-BPEL [TC 07], etc.).

Dans la suite de ce chapitre, nous présentons les fondements de la démarche à base de services que nous proposons pour répondre à la problématique de support flexible de processus de conception de produits manufacturés.

4. Notre démarche pour la gestion flexible des processus de conception de produits manufacturés dans le PLM

4.1. Objectif de la thèse

Les limitations et problématiques que nous avons soulevées dans le premier chapitre servent de base pour l'élaboration de notre contribution. Comme nous l'avons souligné dans les deux premières sections de ce chapitre, les concepts de base des approches orientées services sont pertinents pour notre problématique et apportent des réponses aux problèmes d'automatisation et de flexibilité des processus de conception. Ainsi, nos travaux ont pour objectifs de :

- se concentrer sur la partie informatisée du SI support aux processus de conception de produits manufacturés afin de la rendre flexible. Dans ce cadre, le PLM, support à l'exécution des activités des processus de conception, joue un rôle majeur. Il doit être doté de l'agilité nécessaire qui lui permettra de définir et d'adapter les processus de conception de produits manufacturés aux différents besoins de changement. Cela revient à ré-exploiter l'existant. En effet, le nombre de processus définis dans le PLM résulte de l'empilement de générations successives de processus et fragments de processus, comportant des redondances et manquant de cohérence. La complexité croissante de ce nombre de processus génère des difficultés de plus en plus grandes pour faire évoluer les processus existants afin de les mettre en adéquation avec les différentes circonstances et attentes de clients. Ainsi, la définition des fonctionnalités du PLM sous forme de services réutilisables et leur composition permettra de résoudre ce problème,
- mettre en œuvre des processus de conception de produits manufacturés flexibles et dynamique pour favoriser l'agilité de l'entreprise et permettre la prise en compte des changements imprévus. Cela revient à utiliser des moteurs d'orchestration permettant d'automatiser les processus définis sous forme de composition de services, en s'appuyant sur des langages permettant de mettre en œuvre des orchestrations évolutives.
- masquer la complexité des sous processus et fournir au concepteur un cadre méthodologique avec des concepts simples, bien définis et non ambigus qui lui permettent de définir facilement les processus de conception de produits manufacturés. Cela revient à réutiliser des services prêts à l'emploi et de les composer en utilisant des concepts simples et compréhensibles par des personnes métier. Ainsi, les activités et besoins métier du domaine de la conception de

produits manufacturés peuvent être traduits par des services et modèles d'orchestration qui peuvent être compris par les analystes métiers,

La suite de ce chapitre détaille l'approche que nous proposons. Nous présentons par la suite, l'architecture conceptuelle de notre approche (§ 4.2) et précisons la pertinence de notre démarche par rapport à la problématique adressée.

4.2. Notre approche

Notre approche considère le service comme un concept central. Nous proposons de considérer les ressources et les fonctionnalités d'une entreprise comme des services qui sont faiblement couplés et qui peuvent être composés à la demande en fonction des besoins. Ainsi, nous proposons de cartographier les fonctionnalités de l'application logicielle tel que le système PLM, support aux activités des processus de conception de produits manufacturés, en services réutilisables. L'orchestration de ces services faiblement couplés permettra le support flexible des processus de conception de produits manufacturés. Ceci permettra de mettre en place un outil support à l'exécution des processus de l'entreprise qui soit flexible. Cette proposition favorisera aussi la mise en place du changement et la résolution, en temps réel, des problèmes liés au contournement des processus en se basant sur les langages d'orchestrations évolutives. En fait, s'il y'a besoin de changement, des services (composites ou non) peuvent être ajoutés, supprimés ou remplacés par d'autres services de la composition (cf. Figure 8).

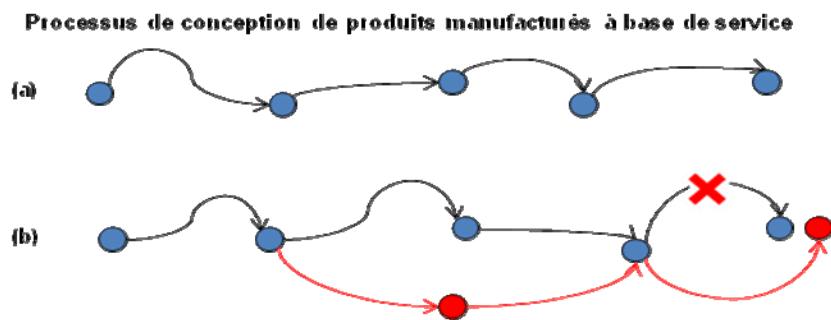


Figure 8: Changement de la composition dans un processus de conception de produits manufacturés

Cette proposition au niveau technique de notre approche apporte des réponses en termes de support flexible pour l'exécution des processus de conception de produits manufacturés. Ainsi, nous proposons de la généraliser au niveau métier afin de simplifier et faciliter la définition des processus à travers l'appel à des services réutilisables et prêts à l'emploi. Cette généralisation permettra non seulement l'automatisation des processus de conception par réutilisation de services déjà existants mais aussi leur définition par cette pratique. Ainsi, nous proposons de cartographier les activités du domaine de la conception de produits manufacturés et de les fournir aux concepteurs de processus sous forme de services réutilisables. Ces services faiblement couplés sont facilement composables avec d'autres services. Ceci peut être réalisé grâce à la composition de services. Les services à valeur ajoutée, issus de cette composition, constituent des sous-processus de conception de produits manufacturés ou encore des processus à part entière. Ces sous-processus peuvent participer à la définition d'autres processus complexes (cf. Figure 9).

Processus de conception de produits manufacturés à base de service

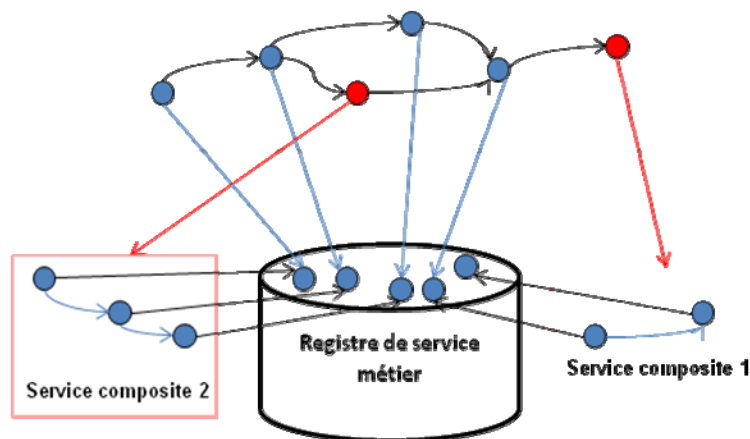


Figure 9: Processus de conception de produits manufacturés à base de composition de services

Par ailleurs, la définition des processus de conception de produits manufacturés à base de service doit faire abstraction de leur réalisation effective. Pour ce faire, nous proposons un niveau intermédiaire, fonctionnel, qui regroupe un ensemble de services fonctionnels PLM. Les services du niveau métier sont ainsi définis sur la base des services de ce niveau fonctionnel. Les services du niveau fonctionnel représentent les activités du PLM indépendamment de toute implémentation logicielle. De plus, la distinction entre services fonctionnels PLM et service logiciels PLM permettra la réutilisation des orchestrations métier (*i.e.* processus de conception de produits manufacturés), définies en termes de services du niveau métier sur des systèmes PLM différents (tels que Windchill, Audros, TeamCenter, etc.). En effet, une fois un modèle de processus à base de services est déployé sur un nouveau système PLM, l'éditeur devra uniquement définir les correspondants de services fonctionnels PLM en termes de services logiciels PLM adéquats (cf. Figure 10).

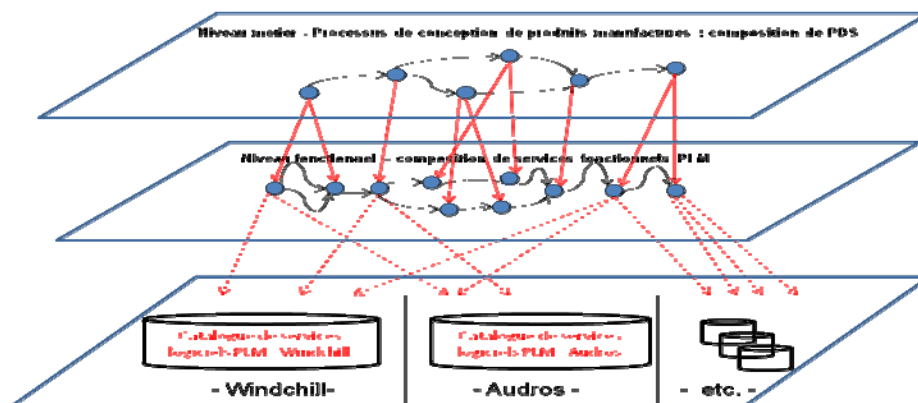


Figure 10: Réutilisation d'un processus de conception de produits manufacturés à base de service dans plusieurs systèmes PLM

Enfin, nous proposons de faire un alignement entre les différentes orchestrations de services en proposant un cadre méthodologique permettant le déploiement de l'orchestration du niveau métier vers une orchestration du niveau fonctionnel. De même pour l'alignement entre le niveau fonctionnel et le niveau technique. Ceci permettra de mettre en cohérence les besoins métier de l'entreprise avec le système utilisé pour le support de ses activités.

4.3. Architecture conceptuelle de notre approche

Dans cette section nous présentons l'architecture conceptuelle de notre approche. Cette architecture résume les concepts de base de l'approche proposée et facilite en retour la compréhension de la démarche d'alignement que nous proposons.

Cette architecture suit une démarche d'ingénierie dirigée par les modèles (IDM) ou *Model Driven Engineering* (MDE). En effet, notre objectif est de partir d'une spécification métier du processus de conception de produits manufacturés, sous forme d'orchestration d'opérations du domaine métier, pour retrouver à la fin un modèle de processus sous forme d'orchestration de service logiciels. Ceci en proposant un cadre méthodologique permettant le passage d'un niveau de l'architecture vers un autre. La démarche IDM a permis plusieurs améliorations significatives dans le développement de systèmes complexes en permettant de se concentrer sur une préoccupation plus abstraite que la programmation classique. L'IDM vise à fournir un grand nombre de modèles pour exprimer séparément chacune des préoccupations des concepteurs, des architectes, etc. [Miller 03]. Un modèle est une abstraction, une simplification d'un système qui est suffisante pour comprendre le système modélisé et répondre aux questions que l'on se pose sur lui [Combemale 09]. L'intérêt grandissant d'une démarche IDM se justifie par le fait que les niveaux proposés par cette démarche s'apparentent à notre problématique. De plus, son utilisation permettra de factoriser certains aspects tels que la modification dans une orchestration. En effet, il suffit d'effectuer les modifications dans un niveau et les mécanismes de transformation, offerts par la démarche IDM, s'occupent de leurs propagations sur les autres niveaux de la démarche.

L'architecture conceptuelle de notre travail comprend trois niveaux (cf. Figure 11).

- Le niveau métier : contient les représentations du domaine de la conception de produits manufacturés et en particulier, les processus de conception de produits manufacturés. Ces représentations sont exprimées par deux concepts : l'orchestration métier (OM) et les SCP (PDS en anglais). Les SCP représentent l'ensemble des fonctionnalités du domaine de conception de produits manufacturés exposées sous forme de services métier (SCP- *Services Conception Produit*, PDS - *Product Design Services* en anglais) et opérations de service. Les orchestrations métier représentent les processus de conception de produits manufacturés sous forme de composition des opérations de SCP. Le niveau métier correspond au niveau CIM (*Computer Independent Model*) de l'approche IDM.
- Le niveau fonctionnel : contient l'ensemble de fonctionnalités que l'on trouve dans un système PLM, sous forme de services (services fonctionnel PLM) et opérations de service. Ces opérations sont génériques et indépendantes de tout système logiciel PLM spécifique. Ce niveau comprend aussi des orchestrations de services fonctionnels PLM (OF) abstraites qui constituent, une projection des orchestrations du niveau métier. On entend par orchestration abstraite, l'orchestration des services fonctionnels PLM indépendamment de leur réalisation technique. Ce niveau fonctionnel correspond au niveau PIM (*Platform Independent Model*) de l'approche IDM.
- Le niveau technique : regroupe l'ensemble de services logiciels PLM et des opérations de services, proposés par les éditeurs de systèmes PLM. Ces services représentent une implémentation particulière des services fonctionnels PLM dans un système PLM spécifique. En d'autres termes, ce niveau comprend l'API du système PLM utilisé. Ce niveau comprend également les orchestrations de services logiciels PLM (OT). Les opérations de ce niveau sont orchestrées conformément aux modèles d'orchestrations définis dans le niveau fonctionnel qui, eux-mêmes, sont conformes aux orchestrations du niveau métier. Le niveau technique représente le niveau PSM (*Platform Specific Model*) de l'approche IDM.

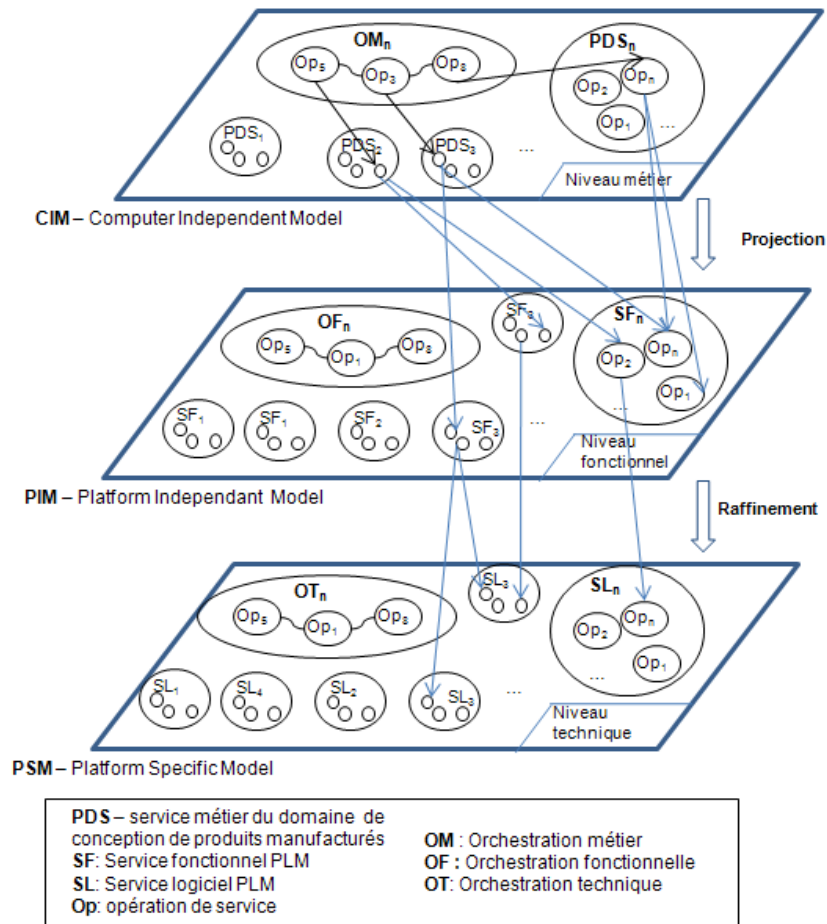


Figure 11: Architecture conceptuelle de l'approche proposée

Pour résumer cette architecture, nous trouvons au niveau métier, les processus de conception de produits manufacturés qui sont définis à travers l'orchestration des opérations de SCP. Les opérations de SCP, qui définissent le processus, seront accomplies en faisant appel aux opérations de services fonctionnels PLM. De même, les modèles d'orchestration de SCP seront projetés vers des modèles d'orchestration de services fonctionnels PLM. Notons, que ces orchestrations du niveau fonctionnel PLM réalisent une partie des processus définis au niveau métier. En effet, les processus définis au niveau métier peuvent faire appel à des services fonctionnels d'autres logiciels tels que les logiciels de CAO, de calcul, etc. (cf. Figure 12).

Les orchestrations du niveau fonctionnel PLM seront à leur tour implémentées par des orchestrations de services du niveau technique. Ce passage du niveau fonctionnel au niveau technique consiste principalement au raffinement de l'orchestration fonctionnel PLM. Le raffinement consiste à préciser les services logiciels PLM qui doivent être exécutés pour mettre en œuvre les services fonctionnels PLM.

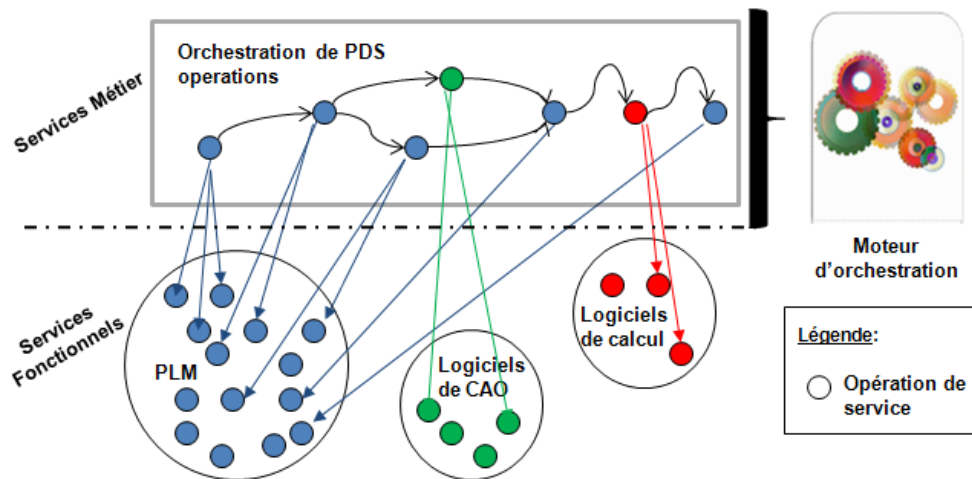


Figure 12 : Processus de conception de produits manufacturés sous forme d'orchestration de SCP

La transformation du modèle d'orchestration du niveau métier vers un modèle d'orchestration du niveau fonctionnel est faite suivant une démarche méthodologique que nous proposons (cf. Figure 13). Cette démarche d'alignement, appelée aussi de déploiement assure l'alignement entre le niveau métier et le niveau fonctionnel en utilisant des mécanismes, illustrés à travers des concepts, modèles et méta-modèles. Ce déploiement métier suit un méta-modèle de déploiement, nommé Méta-Modèle de déploiement métier. Le Méta-Modèle de déploiement métier définit les relations qui existent entre les concepts de la couche métier (service métier, opération de service métier, orchestration métier, etc.) et ceux de la couche fonctionnelle (service fonctionnel, opération de service fonctionnel, orchestration fonctionnelle, etc.). Comme nous l'avons dit précédemment, chaque opération de SCP se projette au niveau fonctionnel sous forme d'une orchestration de services fonctionnels PLM. Les orchestrations fonctionnelles qui implémentent les différentes opérations de SCP seront orchestrées conformément au modèle d'orchestration du niveau métier. Pour faciliter cet alignement, nous définissons un catalogue de déploiement métier qui regroupe l'ensemble des lignes directrices permettant de projeter une opération de SCP en une orchestration d'opérations de services fonctionnelle PLM.

Cette démarche de déploiement peut être utilisée aussi, par les personnes responsables du déploiement du processus défini à base de SCP, dans plusieurs systèmes PLM afin d'assurer l'alignement entre le modèle d'orchestration du niveau fonctionnel et le modèle d'orchestration du niveau technique. Par analogie, le déploiement fonctionnel contient l'ensemble des lignes directrices permettant de raffiner ou encore implémenter une opération de services fonctionnels PLM en une orchestration d'opérations de services logiciels PLM. Ce déploiement fonctionnel suit un méta-modèle de déploiement, nommé Méta-Modèle de déploiement fonctionnel, qui définit les relations qui existent entre les concepts de la couche fonctionnelle (service fonctionnel, opération de service fonctionnel, orchestration fonctionnelle, etc.) et ceux de la couche technique (service logiciel, opération de service logiciel, orchestration technique, etc.) (cf. Figure 13). Les techniques qui permettent d'assurer ces alignements sont décrites dans le chapitre 5 de ce manuscrit.

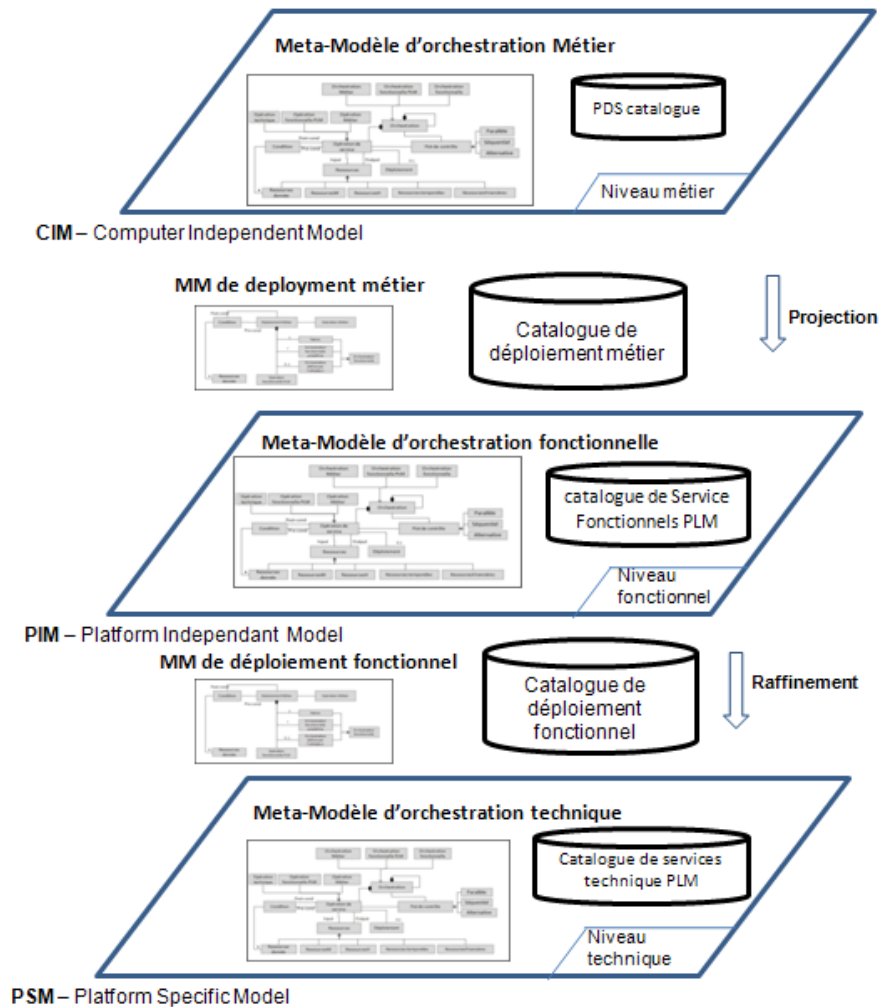


Figure 13: Alignement entre les différents niveaux de notre approche

4.4. Pertinence de notre approche vis-à-vis de nos objectifs

L'approche que nous proposons organise les fonctionnalités du domaine de la conception et du système support aux processus de conception de produits manufacturés en fonctions élémentaires exposés sous forme de services réutilisables et faiblement couplés. Cette proposition permettra la rapidité de construction de processus de conception par orchestration dynamique de services. Ceci dotera le domaine de la conception et notamment son système PLM, d'une meilleure réactivité. Cette réactivité permettra en retour d'accélérer le temps de réponse de l'entreprise et de son système vis-à-vis des changements de son environnement.

De plus, la spécification des processus sous forme d'orchestrations quelque soit au niveau métier ou logiciel permettra de mettre en œuvre des processus de conception de produits manufacturés flexibles et dynamique. L'utilisation de moteurs d'orchestration pour exécuter les orchestrations de services permet d'automatiser les processus définis sous forme de composition de services.

Enfin, du fait que les activités et besoins métier du domaine de la conception de produits manufacturés sont traduits par des SCP et des orchestrations métier, ceci fournit au concepteur du processus un cadre méthodologique avec des concepts métier qui lui sont familiers et compréhensibles. De plus, la possibilité d'utilisation de sous processus définis sous forme d'orchestration à part entière lors de la définition des processus, permet au concepteur métier de définir des processus complexes sans ce

soucier de la complexité des sous processus. Ceci fournit au concepteur un cadre méthodologique avec des concepts simples et des modèles non ambigus.

Dans l'état actuel des choses, plusieurs obstacles contraignent la mise en place de cette démarche. Il s'agit fondamentalement du manque de méthode qui permet de définir les services nécessaires pour notre démarche. Ainsi, la construction des différents catalogues de services nécessaires pour notre démarche est un processus qui pose un certain nombre de défis et de risques. En effet, si les principes d'une approche à base de services sont connus, il n'y a pas encore d'approche méthodologique permettant la construction, d'une façon efficace, les catalogues de services nécessaires pour notre démarche à base de services. La construction des différents catalogues correspondant aux différentes catégories de services nécessite donc la mise en œuvre d'une méthode efficace et robuste afin d'assurer la bonne identification et spécification des services en vue de permettre, par la suite, leur utilisation. Ainsi, nous avons identifié plusieurs verrous à lever, entre autres :

- qu'est-ce qu'un service métier dans notre cas ?
- comment l'identifier, le formaliser ?
- comment l'aligner avec le service logiciel ?

5. Conclusion

Comme nous l'avons précisé au chapitre précédent, les processus de conception de produits manufacturés souffrent d'un manque de support pour la flexibilité. Pour faire face à cette problématique, notre choix s'est focalisé sur l'utilisation d'une approche orientée services. En effet, la revue de littérature a montré clairement que l'utilisation des concepts de service et de composition de services apporte l'agilité nécessaire aux processus de conception de produits manufacturés et qu'elle constitue une réponse pertinente en termes de flexibilité du système support aux processus.

Nous avons étudié dans ce chapitre la pertinence des approches orientées services pour les processus de conception de produits manufacturés et leur système support. D'une part, nous avons souligné l'importance de l'approche orientée services pour les processus de conception de produits manufacturés. L'utilisation du concept central de cette approche, le service, permettra une définition agile des processus de conception de produits manufacturés à travers la réutilisation et la composition de services du niveau métier prêts à l'emploi. D'autre part, nous avons souligné l'importance de la notion d'orchestration de services pour assurer une flexibilité du système support aux activités du domaine de la conception de produits manufacturés. Elle permettra ainsi l'automatisation des processus de produits manufacturés d'une façon flexible en s'appuyant sur des langages d'orchestration qui permettent de mettre en œuvre des orchestrations évolutives et à travers la réutilisation de fonctionnalités élémentaires et réutilisables, exposées sous formes de services faiblement couplés, du système PLM. De plus, l'utilisation d'une approche orientée service permettra d'alléger le système PLM en découplant le moteur d'orchestration des services définissant le processus du système PLM.

Par ailleurs, nous avons présenté notre démarche orientée service pour le support flexible des processus de conception de produits manufacturés. Nous avons présenté l'architecture conceptuelle de notre démarche. Cette architecture suit une démarche d'ingénierie dirigée par les modèles. En effet, cette architecture qui repose sur un ensemble de concepts et de méta-modèles permet de partir d'une spécification métier du processus de conception de produits manufacturés, sous forme d'orchestration d'opérations de SCP, pour retrouver à la fin un modèle de processus sous forme d'orchestration de service logiciels PLM. Ceci en proposant un cadre méthodologique permettant le passage d'un niveau

de l'architecture vers un autre. Nous avons proposé une typologie de services qui distingue trois types de services. Des services métier du domaine de la conception de produits manufacturés (SCP). Ces services permettent la définition des modèles de processus de conception de produits manufacturés sous forme de composition de services. Des services fonctionnels PLM qui permettent de mettre en œuvre les opérations des services du niveau métier. Des services logiciels PLM qui permettent la réalisation technique des services fonctionnels PLM dans un système PLM spécifique. L'un des avantages de la typologie de service que nous proposons, est le fait qu'elle permettra la réutilisation des processus définis par les SCP et les services fonctionnels PLM dans plusieurs systèmes PLM.

Ce chapitre clôt ainsi le cadre de réflexion de ce travail de thèse. Les chapitres suivants ont pour objectif d'illustrer les étapes permettant la mise en œuvre de la démarche que nous avons proposée dans ce chapitre. Ainsi, il ressort de l'analyse de notre démarche que nous devons aborder, dans les chapitres suivants, principalement trois aspects qui sont :

- La définition d'une approche permettant l'identification des services nécessaires pour la définition et l'automatisation des processus de conception de produits manufacturés. Ce premier aspect doit répondre aux questions en relation avec le concept de service, à savoir ; que peut être un service du niveau métier, fonctionnel et technique ? Comment les identifier ? Comment les organiser ?
- La mise en place d'une démarche permettant l'utilisation des services identifiés. Cette démarche doit couvrir l'ensemble de couches de services de notre approche en proposant les modèles nécessaires pour chaque couche. Ce deuxième aspect doit apporter des réponses, entre autres, à la question comment définir un processus de conception de produits manufacturés en utilisant des services du niveau métier ?
- La mise en place d'un cadre méthodologique permettant d'assurer un continuum de transformations entre les modèles élaborés aux différents niveaux de notre démarche. Ce cadre doit être facilement exploitable par les personnes métier du domaine de la conception de produits manufacturés. Ce troisième aspect doit répondre à la question comment lier les modèles de haut niveau aux modèles de bas niveau ?

Chapitre III :

Définition des Catalogues de Services du niveau métier, fonctionnel et technique

1. Introduction

L'architecture conceptuelle de notre démarche de flexibilisation du support informatisé des processus de conception de produits manufacturés repose sur trois niveaux d'abstraction [Hachani 13]. Chacun de ces niveaux manipule une catégorie de service. Le niveau métier manipule les services métier du domaine de la conception de produits manufacturés (*PDS* en anglais). Le niveau fonctionnel manipule les services fonctionnels PLM. Enfin, le niveau technique manipule les services logiciels PLM. Ainsi, nous devons tout d'abord proposer les trois catalogues de services correspondant aux trois catégories de services nécessaires pour notre démarche.

Cependant, comme nous l'avons signalé auparavant, les services logiciels PLM sont des services spécifiques à un système PLM donné. Ils sont des services qui implémentent les fonctions PLM telles qu'exposées sous forme de services du niveau fonctionnel. La définition du catalogue de services logiciels PLM dépend du système PLM considéré. Dans cette thèse, nous n'avons pas vocation à traiter un système PLM commercial particulier. De ce fait, nous ne proposons pas de services logiciels PLM. Toutefois, nous donnons une idée sur ce que peut être un catalogue de services logiciels PLM (section 3.3). L'objet de la section 3.3 n'est pas de proposer un catalogue de services logiciels PLM mais de définir une sorte de cahier des charges pour ce catalogue. D'ailleurs des standards orientés services ont déjà été proposés pour le PLM [OMG 08] [ISO 05]. Par contre, aucun standard ou initiative n'a été proposé en termes de services métiers du domaine de la conception de produits manufacturés ou de services fonctionnels PLM.

Pour le catalogue de services métiers du domaine de la conception de produits manufacturés et le catalogue de services fonctionnels PLM, la démarche a consisté à :

- établir une cartographie des services métier du domaine de la conception de produits manufacturés : pour cela, les fonctionnalités du domaine de la conception sont recensées et exposées sous forme d'opérations de service génériques ;
- établir une cartographie des services fonctionnels PLM : pour cela, les fonctionnalités offertes par les systèmes PLM sont recensées et exposées sous forme d'opérations de services génériques ;

Pour ces fins, une démarche d'identification des services de chacune de ces deux catégories doit être définies. Cela revient à définir pour chaque catégorie de service (*SCP* et services fonctionnels PLM) les étapes, techniques et moyens nécessaires pour la définition et l'identification des services. Certaines techniques et bonnes pratiques existantes peuvent servir comme base à la démarche d'identification que nous souhaitons proposer. L'objectif de ce chapitre est de présenter la démarche mise en place pour identifier les services métiers du domaine de la conception et les services fonctionnels PLM. Dans un premier temps, nous passerons en revue les approches et techniques d'identification de services proposées dans la littérature. Cette revue nous permettra d'analyser les approches existantes et de délimiter nos choix sur les techniques d'identification les mieux adaptées (§2). Ensuite, nous nous focalisons sur la démarche que nous proposons pour l'identification des services du niveau métier (i.e. *SCP*) et des services fonctionnels PLM. Nous donnerons également un aperçu de ce que peut être un catalogue de services logiciels PLM.

Nous présentons enfin les catalogues de services de conception de produits manufacturés (*SCP*), de services fonctionnels PLM et de services logiciels PLM obtenus à l'aide des démarches d'identification proposées.

2. Méthodes d'identification de services : État de l'art

Dans cette première partie du chapitre, nous passons en revue les méthodes et techniques d'identification de services proposées dans la littérature. Nous soulignons l'ensemble de bonnes pratiques et les bonnes techniques nécessaires pour l'identification des catalogues de services nécessaires pour notre approche.

L'analyse des forces et faiblesses des méthodes étudiées nous permettra de délimiter nos choix sur les techniques d'identification de services qui seront utilisées pour outiller notre approche.

2.1. Méthodes d'identification des services existantes

Cette section présente une revue de littérature des travaux de mise en place d'une AOS. Nous nous intéressons principalement dans cette section à la phase d'identification de services comme c'est l'objet de ce chapitre. Nous avons choisi les travaux qui forment un ensemble représentatif de l'état de l'art.

Nous citons en premier lieu la méthode SOMA d'IBM [Arsanjani 04] [Arsanjani 08] pour l'identification, la modélisation et la spécification des services. Elle consiste en trois phases : la phase d'identification des services, la phase de spécification des services et la phase de réalisation des services. (1) La phase d'identification se base sur trois démarches : une démarche descendante, une démarche ascendante et une démarche hybride. La démarche ascendante analyse les systèmes logiciels existants afin d'identifier les services de bas niveau. La démarche descendante offre une cartographie des cas d'utilisation du système réel à travers l'analyse des processus. Elle a pour but d'étudier les composants métiers responsables de la réalisation des services. Tandis que la démarche hybride consiste en une modélisation des services basée sur les buts afin d'identifier les services qui n'ont pas été identifiés dans les deux autres démarches. (2) La deuxième phase, est celle de spécification, consiste à filtrer l'ensemble des services candidats en se basant sur des règles bien déterminées telles que l'élimination des redondances, la réutilisation des services par les différents processus et la facilité de l'implémentation. Dans cette phase, les auteurs proposent d'attribuer une note pour les services sur une échelle de 1 à 5. Les services qui auront les scores les plus élevés seront considérés comme des candidats pour la réalisation. (3) La phase de réalisation se charge d'implémenter l'ensemble des services identifiés. Comme nous l'avons dit précédemment nous nous intéressons principalement à la phase d'identification. Etant donnée que notre objectif est l'identification des deux catalogues de services du niveau métier et fonctionnel, nous soulignons l'importance de l'utilisation des deux démarches d'identification, ascendante et descendante, citées là-haut.

[Papazoglou 06] propose une méthode de développement des services Web. Cette méthode consiste en six phases allant de la planification de la mise en place d'une AOS jusqu'à l'exécution des services. La phase de planification comprend l'analyse des besoins métier, l'étude des technologies actuelles et de leurs capacités. Elle comprend aussi une analyse financière et une estimation des coûts et des bénéfices d'un projet de développement d'une AOS. La phase d'analyse et de spécification : comporte deux étapes : (1) l'étape d'analyse consiste en la préparation des modèles de processus métier qui existent actuellement et ceux qui doivent exister. Cette étape examine l'ensemble des services existants dans le but de savoir quels sont les services qui sont déjà en place. (2) L'étape de spécification de services se préoccupe de la présentation de la structure du service, de son comportement et de ses politiques. La phase de construction et de test se concentre sur l'implémentation des services et la définition de leurs interfaces. Par la suite, il s'agit de vérifier la compatibilité des services obtenus par rapport aux objectifs fixés. La phase de *provisionnement* a pour objectif le contrôle du comportement d'un service durant son utilisation. La phase de déploiement assure la publication des interfaces des services. La phase d'exécution et de supervision consiste en la découverte, l'invocation et l'évaluation des opérations de services. Notons ici que les deux premières phases sont principalement ceux qui permettent l'identification des services. Ceci est assuré principalement à l'aide de l'étude des besoins métier et des processus de l'entreprise et leur confrontation avec les services existants dans le patrimoine applicatif. Ainsi, l'analyse des processus métier du domaine est une technique primordiale par laquelle il faut passer pour identifier les services d'une entreprise.

Dans la méthode SOAD (Service-Oriented Analysis and Design) proposée par [Zimmermann 04], les auteurs considèrent que les approches de modélisation des processus et la conception orientée objet sont des approches pertinentes qui permettent d'identifier des services suivant deux niveaux d'abstraction : métier et technique. Les auteurs proposent d'utiliser une approche hybride pour l'identification des services qui combine à la fois l'analyse des cas d'utilisation afin de définir les services nécessaires à leur réalisation et l'analyse de l'existant pour déterminer les fonctions existantes du système d'information. Deux types de services d'entreprise sont définis, à savoir : les services métier (business service) et les services logiciels (software service). Ces derniers peuvent être des services atomiques ou des services composites. Par la suite, les services sont décrits grâce à des

politiques qui enrichissent la description de leurs interfaces avec des spécifications de qualité (QoS). Un autre élément de base de l'approche SOAD consiste en la gestion des connaissances autour des services d'entreprise. Il s'agit de construire une culture de service qui permet par la suite de juger la pertinence d'un service. En effet, un service qui n'est pas réutilisable est un service non pertinent et ne devra pas exister dans le catalogue de services de l'entreprise. Notons que de même que pour les deux premières approches l'analyse du domaine, ici à travers l'analyse des cas d'utilisation, est nécessaire pour l'identification des services de haut niveau.

D'autres travaux [Chang 07] présentent une méthode pour le développement des services adaptables composée de six phases. La première phase est celle de définition des services dans laquelle les auteurs procèdent à une analyse des services existants et l'identification des processus métiers cibles. La deuxième phase s'intéresse à la définition des services unitaires, à partir des processus métier définis dans la phase précédente, en se basant sur l'analyse du domaine. Cette phase consiste aussi à définir les interfaces des services unitaires et à spécifier leur modèle structurel et comportemental. Les auteurs définissent des variabilités de services selon leurs comportements ainsi que les situations qui les déclenchent afin d'assurer l'adaptation des services. La troisième phase définit une typologie de services ; les services extraits à partir des applications existantes, les services qui doivent être développés et les services qui doivent être découverts à partir des registres de service externes. La quatrième phase est la phase d'acquisition qui s'intéresse au développement des deux premiers types de services issus de la classification des services. Les auteurs identifient des contraintes telles que les contraintes fonctionnelles et non fonctionnelles pour les services qui doivent être découverts. Concernant les services qui doivent être extraits à partir du patrimoine applicatif, les auteurs proposent d'ajouter des façades ou des médiateurs pour encapsuler ces systèmes et obtenir les services. La cinquième phase se charge du développement des nouveaux services qui seront développés selon des techniques existantes comme la conception orientée objet ou le développement basé sur les composants. La dernière phase est la phase de composition des services unitaires obtenus à l'issue des phases précédentes. Notons ici que les services unitaires permettant de mettre en œuvre les processus métiers de l'entreprise sont principalement ceux qui encapsulent les fonctionnalités des applications existantes. Donc nous pouvons conclure que l'identification de services de bas niveau permettant la mise en œuvre des services de haut niveau doit nécessairement passer par l'analyse du patrimoine applicatif.

Dans son livre intitulé « Service-oriented Architecture: Concepts, Technology, And Design », [Erl 05] présente les étapes du cycle de vie de la construction d'une AOS. Nous nous intéressons à la première phase qui présente une méthode de définition des services à savoir : l'identification et la conception des services. Trois types de services ont été identifiés par cette méthode: les services métier, les services applicatifs et les services hybrides. Le service métier encapsule la logique métier de l'entreprise et il est issu de l'analyse de ses différents processus. Quant au service applicatif, il expose des fonctionnalités de fines granularités, qui sont sollicitées par les services métier. La particularité de ce travail consiste à proposer des services de type hybrides. Ces services encapsulent à la fois la logique métier et la logique applicative. Par contre, l'auteur ne propose aucune démarche permettant de les spécifier. Nous nous intéressons principalement à la méthode d'identification des services métier et applicatifs. La Figure 14 illustre le processus proposé pour l'identification des services applicatifs et métier.

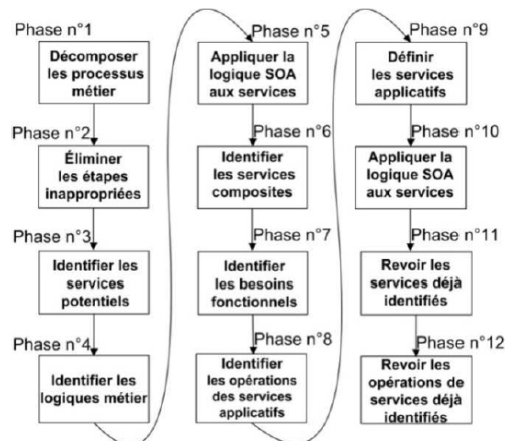


Figure 14: Processus d'identification des services selon [Erl 05]

Il s'agit d'une approche descendante comportant douze phases à réaliser dans l'ordre. La première phase consiste à décomposer les processus métiers déjà cartographiés en un ensemble d'étapes. Une fois identifiées, il s'agit par la suite d'éliminer les étapes qui ne peuvent pas être encapsulées par des services. Par la suite, les services potentiels sont identifiés en regroupant les étapes qui semblent appartenir à un même contexte. Ensuite, la logique métier issu de l'étude des processus est identifiée. L'objectif d'une telle identification est de sélectionner les étapes éligibles au rang d'opérations de services. La phase suivante propose d'appliquer la logique de la AOS aux services potentiellement définis, à savoir : la réutilisation et l'autonomie. La sixième phase consiste à définir les services composites. Quant à la septième phase, elle se charge d'étudier les besoins fonctionnels de chaque opération d'un service. Ces besoins seront pris en considération dans la huitième phase qui permet d'identifier les opérations des services applicatifs à partir des besoins d'implémentation exprimés dans la phase précédente. Par la suite, il s'agit de regrouper les opérations appartenant à un même contexte afin d'identifier les services applicatifs. Ces derniers vont être vérifiés dans la phase suivante afin de s'assurer qu'ils sont réutilisables et autonomes. Les deux dernières phases du processus d'identification consistent à revoir les services déjà identifiés ainsi que leurs opérations. Notons ici que l'auteur fait la différence entre service et opération de services. En effet, dans la littérature un service peut être considéré comme un groupement d'opérations ou encore comme une seule opération composite ou unitaire [Erl 05]. Dans le cas où le service est considéré comme un groupement d'opérations, tel est le cas pour cette approche, des critères de groupement d'opérations en services doivent être utilisés. Ici l'auteur a favorisé le contexte d'utilisation comme critère de groupement mais il y'en a d'autres qui sont propres à la logique AOS à savoir la cohésion et le faible couplage que nous allons définir et prendre en compte dans notre approche d'identification. Enfin, nous notons que la technique d'analyse des processus métier est appliquée par cet auteur à la fois pour l'identification des services unitaires et ceux de haut niveau (à travers la logique métier du processus).

[Kohlborn 09b] propose une méthode d'identification et d'analyse de services qui couvre deux types de services ; métier et logiciel. Cette méthode se présente sous la forme de deux parties principales, chacune avec des phases différentes. La première partie traite de l'identification et l'analyse des services métier et elle comporte quatre phases (cf. Figure 15, à gauche). La première phase est celle de préparation qui sert de fondement pour la démarche d'identification des services métier. Dans cette phase les domaines clés de l'entreprise, une ontologie du vocabulaire métier et les données métier tel que les documents etc. sont identifiés. La deuxième phase, celle d'identification, mène à la dérivation d'un ensemble de services métier. Pour se faire, l'auteur utilise des techniques telles que l'analyse du domaine, des capacités métier et des acteurs et leurs interactions. Par la suite, il regroupe les services identifiés par service de domaine pour avoir une forte cohésion au sein d'un même service et un faible couplage entre les différents services. La troisième phase est celle de détail. Elle consiste à définir les interactions entre les différents services métier ainsi que les interactions entre les domaines identifiés, afin de définir les processus sous-jacents à chaque service métier. La dernière phase est celle d'hierarchisation qui permet de construire une classification des services selon leur importance. La deuxième partie de cette méthode traite de l'identification et l'analyse des services logiciels et

comporte trois phases (cf. Figure 15, à droite). La première phase est celle de préparation. Elle permet de préciser la portée de l'identification des services logiciels. Elle se base sur le détail des services métier identifiés, à savoir les processus qui leurs sont sous-jacents. Afin de fournir une base pour l'identification des services logiciels, les processus liés aux services métier doivent être décomposés et enrichis par le résultat de la phase de préparation de la première partie, à savoir les documents et base de données. Le résultat de cette phase est un ensemble de processus décomposés en des étapes susceptibles d'être définies comme opérations de service logiciel. La deuxième phase de la 2^{ème} partie est celle d'identification. Elle consiste en la dérivation de services logiciels en se basant sur les modèles de données pour identifier les services entités et les modèles de processus décomposés pour en extraire la logique d'enchaînement et définir des groupements de services. La dernière phase est celle de détail. Cette phase consiste à vérifier que les services existants ne se chevauchent pas avec ceux découverts. Elle permet aussi de mapper les services identifiés à la couche d'application afin d'identifier les fonctionnalités manquantes et donc rajouter des services logiciels supplémentaires. Notons que l'auteur de cette démarche fait recours à plusieurs techniques d'identification quelque soit pour l'identification des services de bas niveau ou ceux du haut niveau. En effet, nous constatons que la combinaison de plusieurs techniques peut amener à un catalogue plus complet de services. De même que pour le travail de [Erl 05], les services identifiés consistent en un groupement d'opérations. Le groupement d'opérations ici est basé principalement sur les deux critères de couplage et cohésion mais aussi sur l'étude de la logique d'enchaînement des traitements unitaires au sein du domaine métier.

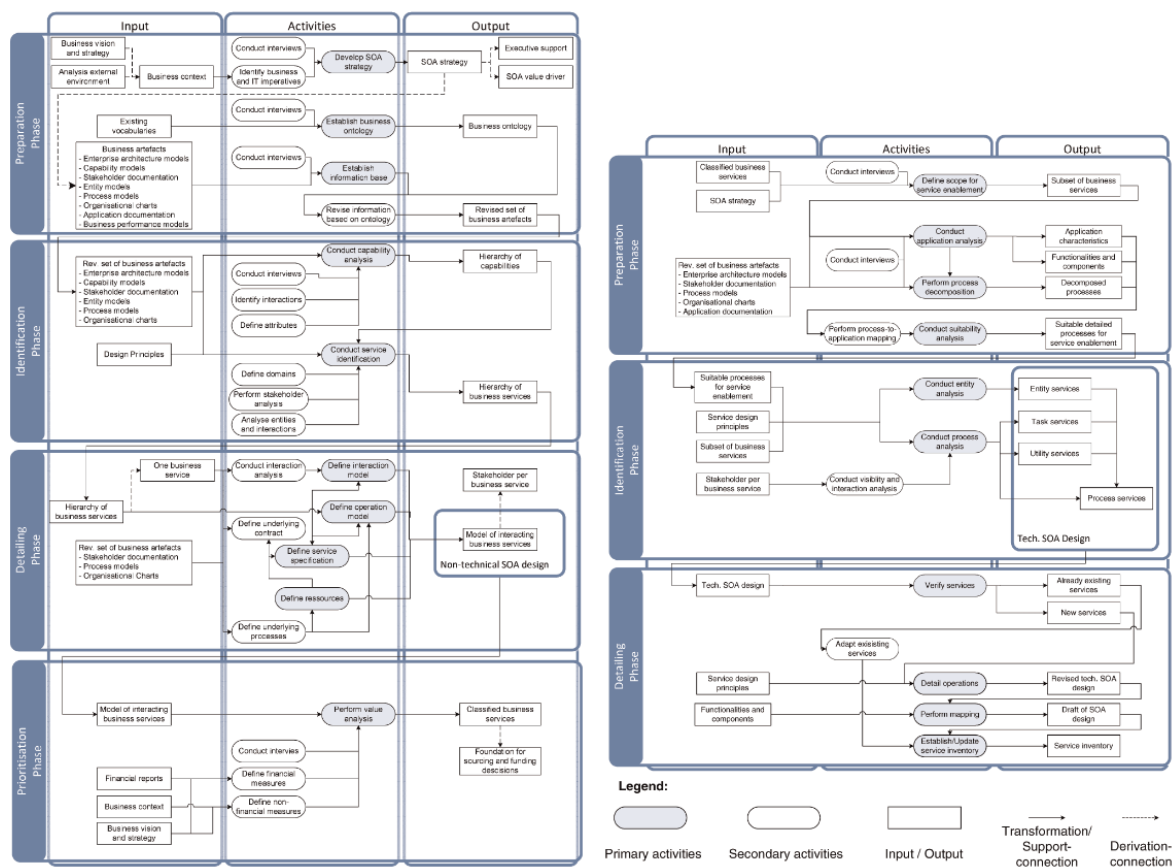


Figure 15 : Méthode d'identification des services de haut et bas niveau [Kohlborn 09b]

[Kaabi 07b] propose une méthode pour l'identification de services intentionnels (de haut niveau) et de services logiciels (de bas niveau). Pour l'identification des services intentionnels, elle propose de commencer par l'identification des besoins utilisateurs. Ces besoins sont décrits avec un modèle orienté buts appelé « carte ». La carte exprime les besoins des utilisateurs par des buts et des stratégies pour les atteindre. Dans ce système de représentation, un but est ce qu'on cherche à atteindre. Une stratégie est une manière de réaliser un but. L'objectif de cette pratique est d'identifier les services

intentionnels à partir du modèle de la carte. Chaque section (triplet but, stratégie, but) de la carte identifie un service. Notons qu'une stratégie permet de passer d'un but à un autre. Les relations entre les sections permettent ainsi de composer les différents services. Quant à la démarche d'identification des services logiciels, elle comporte trois étapes. Elle débute par l'écriture des scénarios de base et la découverte des exceptions. Cette étape a pour objectif de construire les scénarios permettant de décrire le comportement d'un service intentionnel dans la satisfaction du but associé. La deuxième étape est celle d'analyse des scénarios. L'objectif de cette étape est de découvrir les éléments constituant le service logiciel à partir des scénarios obtenus lors des étapes d'écriture des scénarios. La troisième étape est celle d'implémentation des services identifiés. Les services intentionnels identifiés par cette méthode sont des services de haut niveau. Leur démarche d'identification proposée, à travers l'identification des besoins des utilisateurs (étude des besoins métier) renforce l'idée de nécessité d'utilisation d'une approche descendante pour l'identification des services de haut niveau.

2.2. Évaluation des méthodes d'identification

Afin d'être en mesure d'évaluer les techniques et méthodes d'identification de services, nous proposons des critères permettant d'évaluer ces méthodes. En effet, nous cherchons à définir principalement deux catalogues de services. Un pour le niveau métier et un pour le niveau fonctionnel. Nous pouvons alors synthétiser les caractéristiques des méthodes d'identification de services à l'aide de quatre critères :

- **Concepts SOA** : l'objectif de la démarche d'identification peut être la dérivation de services de haut niveau (SM – services métier), de services de bas niveau fonctionnels ou techniques (SL – services logiciel), ou des deux types de services (SM / SL). Ce critère permettra ainsi d'analyser quel concept de services est concerné par la méthode d'identification. Nous avons choisi le critère *concept SOA* en premier afin de fixer l'objectif de la méthode étudiée au regard de la démarche d'identification et des techniques utilisées.
- **Démarche d'identification** : il existe trois démarches d'identification de services dans la littérature. La première est la démarche descendante dans laquelle la logique métier des processus existants est examinée afin d'identifier les services. À l'inverse, la démarche ascendante débute par l'analyse du patrimoine applicatif afin de déterminer les fonctions existantes du système et dresser une cartographie de fonctions qui sont susceptibles de faire partie d'un service. Enfin, l'approche hybride consiste à mener à la fois une approche descendante et une approche ascendante. Ce critère est important pour notre problématique car il nous permet de savoir quelle méthode est mieux adaptée pour l'identification de telle ou telle catégorie de services. D'ailleurs ce critère est toujours pris en compte par plusieurs auteurs lors de l'analyse des approches de mise en place de AOS [Kohlborn 09b] et [Ramollari 07].
- **Techniques utilisées** : plusieurs techniques d'identification existent dans la littérature, nous pouvons citer parmi ces techniques ; l'analyse des cas d'utilisation, l'analyse des besoins et des processus du domaine, la modélisation orientée but, l'analyse du patrimoine applicatif, etc. Ce critère permet de renseigner si la méthode étudiée propose la réutilisation de techniques existantes et de préciser quelles techniques peuvent être utilisées pour l'identification de telle ou telle catégorie de services.
- **Degré de clarté et facilité de prise en main de la démarche** : une méthode d'identification et d'analyse de services peut être prescriptive et définit les étapes avec beaucoup de détail. Elle peut aussi décrire des étapes d'une façon moins détaillée et non structurée. Ce critère permet ainsi d'évaluer la qualité des méthodes proposées en termes de clarté de la démarche et de facilité de prise en main de la démarche. En effet, dans notre cadre de travail, les systèmes PLM peuvent évoluer (ajout de nouveaux modules, etc.) ainsi que les besoins métier du domaine. La démarche doit être claire et facilement exploitable par les personnes métier du domaine si jamais ils veulent enrichir les catalogues proposés. Nous avons choisi de les classer en utilisant les symboles (+, ++, +++).

Le tableau ci-dessous (Tableau 6) récapitule les principales méthodes que nous avons étudiées précédemment et les compare selon les 4 critères présentés ci-dessus :

Tableau 6: Comparaison des méthodes d'identification de services

	Concepts SOA	Démarche d'identification	Techniques utilisées	Degré de clarté et facilité de prise en main
[zimmermann 04]	SM	Descendante	Analyse des cas d'utilisation	+
	SL	Ascendante	Analyse du patrimoine applicatif	
[Arsanjani 04]	SL	Hybride	Analyse des processus et du patrimoine applicatif	++
[Erl 05]	SL	Descendante , Hybride	<i>Analyse des processus et des modèles de données du système</i>	++
[Papazoglou 06]	SL	Hybride	<i>Analyse de domaine et du patrimoine applicatif</i>	+
[Chang 07]	SL	Hybride	<i>Analyse du domaine et du patrimoine applicatif</i>	++
[Kaabi 07b]	SM	Descendante	<i>Modélisation orientée but</i>	+
	SL			
[khlborn 09]	SM	Descendante	<i>Décomposition des processus et du domaine</i>	++
	SL	Hybride	<i>et analyse du patrimoine applicatif</i>	

L'analyse du Tableau 6 permet de dégager un ensemble de constats :

- **Concepts SOA** : Peu nombreux sont les travaux de recherche qui se sont intéressés aux approches d'identification des services du niveau métier. La plupart des méthodes présentées s'intéressent à l'identification des services de bas niveau (ST). D'ailleurs, un reproche du travail présenté dans [Chang 07] tient au fait que les services définis sont de très fines granularités. Le service du niveau métier n'a pas été traité et uniquement une couche de services techniques est disponible. Même dans le travail de [zimmermann 04], le service métier est utilisé pour distinguer entre les services qui encapsulent la logique métier et ceux qui encapsulent la logique d'application.
- **Démarche d'identification** : La majorité des méthodes de mise en place d'une SOA proposent des démarches hybrides pour l'identification des services. Nous pensons que l'approche hybride est l'approche la plus réaliste au regard des contraintes industrielles. Il est intéressant de mener en parallèle à la fois une approche descendante et une approche ascendante. La première permet de définir des services de haut niveau nécessaires à la réalisation des processus tandis que la deuxième permet de cartographier l'existant applicatif de l'entreprise afin d'identifier les services de bas niveau. Nous concluons de l'analyse des approches proposées, que l'approche descendante est l'approche la mieux adaptée pour l'identification des services de haut niveau. Quant à l'approche ascendante est la mieux adaptée pour l'identification des services de bas niveau. D'ailleurs l'une des reproches de la méthode proposée par [kaabi 07b] est qu'elle propose une démarche descendante à la fois pour l'identification des services de haut et de bas niveau.
- **Techniques utilisées** : Nous constatons que la majorité des méthodes étudiées utilisent les mêmes techniques d'identification de services même s'ils ne s'accordent pas sur les mêmes étapes d'identification. L'originalité de la méthode proposée par [Zimmermann 04] réside dans l'utilisation de plusieurs techniques existantes. De plus, nous constatons que les techniques d'identification sont liées à la catégorie de service à identifier (SM/ST). En fait, pour l'identification des services de bas niveau, la technique souvent utilisée est celle de l'analyse du patrimoine applicatif. Cependant, pour l'identification des services de haut niveau le point de départ n'est pas toujours le même pour toutes les méthodes. Certaines proposent l'utilisation de la technique d'analyse des cas d'utilisation [Zimmermann 04], d'autres des processus ou encore de l'analyse du domaine [Kohlborn 09]. Par contre ils s'accordent tous

sur le fait que la technique d'analyse du patrimoine applicatif ne peut pas être utilisée pour l'identification des services métier.

- **Degré de clarté et facilité de prise en main de la démarche** : mis à part le travail de [Erl 05], les méthodes présentées n'offrent pas une démarche complète, claire et précise permettant d'atteindre le but d'une manière efficace et facilement exploitable par les personnes métier. Par exemple, un reproche qui peut être adressé au travail de [Zimmermann 04] tient au fait que la SOAD ne propose pas de démarche méthodologique pour l'identification des services. En effet, elle propose un ensemble d'éléments et de concepts qui ressemblent plutôt à des bonnes pratiques qu'à des directives d'aide pour l'identification des services. De même, la méthode proposée par [Papazoglou 06] présente un cycle complet pour le développement des services. Néanmoins, peu de détail concernant la méthode a été présenté ce qui peut poser des questions sur sa faisabilité. L'approche SOMA [Arsanjani 04] est une approche qui ne propose pas de description ouverte de sa méthode ce qui rend difficile l'analyse de ses capacités réelles et de son application si jamais y'a des nouveaux besoins et donc des nouveaux services à identifier.

De l'étude de diverses méthodes d'identification de services, nous concluons sur l'inexistence d'une méthode qui couvre à la fois les deux catégories de services (*i.e.* de haut et de bas niveau) et qui soit à la fois compréhensible et facilement exploitable par des personnes métier du domaine de conception (concepteurs, industrialisateurs, chefs de projets, etc.). Nous identifions le besoin de proposer une méthode qui couvre tous les niveaux d'abstraction de services et qui facilite à la fois l'utilisation des techniques existantes, tout en utilisant les bonnes pratiques soulignées à travers cette analyse, à savoir :

- le choix d'une approche descendante pour l'identification des services du niveau métier. Ceci en mobilisant les techniques les plus adaptées telles que l'analyse du domaine et des processus. D'ailleurs aucune des approches analysées n'incite à la définition des services du niveau métier en utilisant une approche ascendante.
- Le choix d'une approche ascendante pour l'identification des services fonctionnels PLM. Ceci en utilisant la technique d'analyse du patrimoine applicatif. En effet, la plupart des organisations ont déjà des systèmes PLM en place qui ne peuvent être facilement remplacés. Nous proposons donc de partir des exemples de systèmes PLM existants pour identifier des services fonctionnels PLM génériques.

Notons ici que l'approche d'identification ne peut pas être la même pour les deux catégories de services. En fait, ils exposent des informations de type différent : besoins du domaine métier pour les services de haut niveau et les fonctionnalités du système PLM pour les services de bas niveau. Nous devons donc proposer des critères d'identification différents, qui reflètent l'orientation service. Nous présentons dans la partie suivante la démarche d'identification de différentes catégories de services que nous jugeons nécessaires pour la mise en œuvre de notre approche de flexibilisation des processus de conception de produits manufacturés.

3. Démarche proposée pour l'identification de différentes catégories de services

Comme indiqué plus haut (cf. Introduction), nous voulons définir deux catalogues de services (le catalogue de services contient un ensemble de services) :

- Un catalogue de services du niveau métier du domaine de la conception de produits manufacturés (*i.e.* SCP). Ce catalogue exprime les besoins métier en termes de contenu / structure des processus de conception de produits manufacturés.
- Un catalogue de services fonctionnels PLM qui permet de décliner les opérations des SCP à l'aide des fonctions proposées dans les systèmes PLM (telles que créer un document, créer une nomenclature, rattacher un document CA, etc.)

Enfin, concernant le catalogue de services logiciels PLM, nous proposons un cahier des charges sur le contenu de ce catalogue. En effet, les services de ce catalogue doivent permettre la mise en œuvre et

l'implémentation des services fonctionnels PLM. Pour le catalogue de services du niveau technique PLM, nous ne déployons pas la démarche d'identification de services car cette catégorie de services dépend du système PLM considéré. Chaque éditeur pourrait implémenter le catalogue de services fonctionnels que nous proposons selon l'architecture technique de son système. Nous proposons toutefois d'analyser les standards du domaine qui proposent des services techniques et qui pourraient être réutilisés pour définir le catalogue de services logiciels.

Pour l'identification des services du niveau métier du domaine de la conception de produits manufacturés et ceux du niveau fonctionnel PLM, nous proposons une démarche d'identification inspirée des bonnes pratiques tirées des méthodes d'identification de services proposées dans la littérature (cf. §2). Ainsi, nous proposons une démarche méthodologique à deux phases mettant en exergue une double démarche d'identification de services : l'identification de services du niveau métier et l'identification des services du niveau fonctionnel PLM. Chaque phase de notre démarche comporte plusieurs étapes et peut être réalisée soit par un intervenant métier, soit par un intervenant IT soit par les deux au même temps (cf. Figure 16). Nous allons présenter en détail les différentes étapes de notre démarche d'identification en soulignant pour chaque étape l'ensemble des livrables.

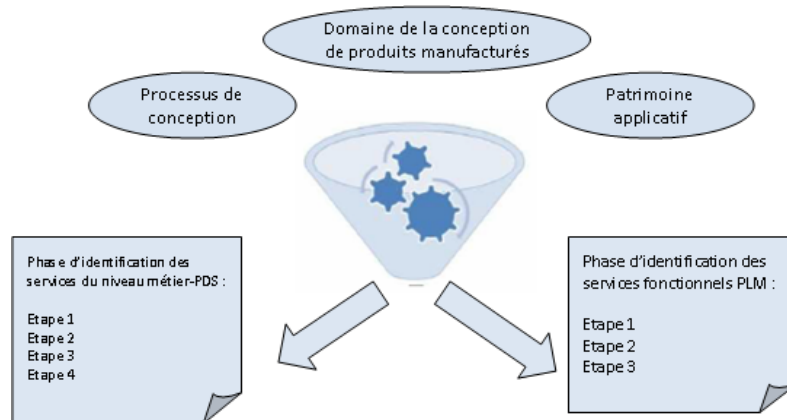


Figure 16 : Démarche d'identification des services du niveau métier et du niveau fonctionnel PLM

3.1. Identification des services du niveau métier – services de conception de produits manufacturés (SCP)

Cette section définit les services du domaine de la conception de produits manufacturés. Elle présente la démarche d'identification adoptée et les résultats qui en sont issus mais avant cela, nous nous attardons sur le concept de service métier (*SCP*) afin de donner une définition précise de ce que nous souhaitons identifier.

3.1.1 Définition de services du niveau métier

Les services de conception de produits manufacturés sont des services qui font sens pour les acteurs métier du domaine de la conception de produits manufacturés. Ils sont issus de l'analyse du domaine de la conception de produits manufacturés et des processus de ce domaine.

Nous définissons un service de conception de produits manufacturés (*SCP*) comme suit ; un service de conception de produits manufacturés est un groupement d'opérations métiers du domaine de la conception de produits manufacturés. Ces opérations métier traduisent des solutions aux besoins métier du domaine de la conception de produits manufacturés [Hachani 12b]. Ce sont des opérations qui permettent d'implémenter une partie ou la totalité de la notion des tâches / activités composants les processus de conception de produits manufacturés. Ils encapsulent la logique du sous-processus qu'ils exposent.

Les opérations des services de conception de produits manufacturés représentent des traitements récurrents du domaine de la conception de produits manufacturés. Ils sont de grande granularité comme par exemple des opérations qui répondent à un besoin de validation d'un document ou encore une conception d'une pièce 2D. Par conséquent, chaque opération consiste en une action métier faite

par les acteurs des processus de conception de produits manufacturés, qui opère sur un ou plusieurs objets métier du domaine de la conception de produits manufacturés.

- Un objet métier du domaine de la conception de produits manufacturés est un artefact qui fait sens pour les acteurs du domaine de la conception de produits manufacturés. Il correspond aux entités manipulées par les acteurs de ce domaine lors de la pratique de leur métier de conception pour spécifier, décrire une partie ou la totalité du produit qu'ils sont en train de concevoir (par exemple ; un dessin 2D d'une pièce, la liste de matériaux entrant dans la composition d'un produit, un modèle CAO d'un assemblage, etc.).
- Une action métier du domaine de la conception de produits manufacturés est un acte qui peut être accompli par un acteur participant au processus de conception dans le but de parvenir à un résultat particulier (par exemple ; création, validation, etc.)

Une fois déployées dans un système PLM, ces opérations peuvent être mobilisées dans divers processus de conception de produits manufacturés. En fait, elles pourront être choisies par les concepteurs de processus et participer par la suite à la définition d'un ou plusieurs processus.

3.1.2 Démarche d'identification de services du niveau métier

Comme nous l'avons dit précédemment, les services du niveau métier (*SCP*) doivent être identifiés et définis suivant une logique métier, indépendamment de la technologie. Pour ce faire, nous proposons une démarche d'identification descendante qui consiste à analyser le domaine de conception de produits manufacturés ainsi que ses processus. Cette démarche part des caractéristiques des processus et du domaine afin d'identifier les différents services nécessaires à la définition et la formalisation de ces processus.

La démarche que nous proposons est basée sur la technique de décomposition du domaine métier [Arsanjani 08], qui consiste en l'analyse des domaines fonctionnels, et sur la technique de décomposition des processus du domaine [Erl 05]. L'analyse des domaines fonctionnels consiste à décomposer le domaine métier en domaines fonctionnels afin de fournir des frontières entre les différentes activités du domaine métier. La décomposition des processus du domaine consiste à décomposer les processus en sous-processus puis des tâches pour avoir une première liste de services candidats. De plus, cette démarche propose des étapes compréhensibles et adaptées aux personnes métier du domaine ainsi que des nouveaux critères spécifiques au domaine pour le groupement des opérations métier en services de conception de produits manufacturés.

Quatre étapes constituent la démarche que nous proposons pour l'identification des services du niveau métier – SCP :

- Étape 1 : Identification des objets métier du domaine de la conception de produits manufacturés ;
- Étape 2 : Identification des actions métier du domaine de la conception de produits manufacturés ;
- Étape 3 : Association des objets métier aux actions métier du domaine de la conception de produits manufacturés. Cela permet d'identifier les opérations de services métiers ;
- Étape 4 : Identification des E/S des opérations métier identifiées ;
- Étape 5 : Regroupement des opérations métier identifiées dans des services de conception de produits manufacturés.

Étape 1 : Identification des objets métier du domaine de la conception de produits manufacturés

Pour mener cette étape, deux sources d'informations ont été utilisées pour l'identification des objets métier du domaine de la conception de produits manufacturés :

- des études terrains permettant de collecter le vocabulaire partagé entre les différents acteurs du domaine.

- l'analyse des standards du domaine tels que les modèles de référence relatifs aux processus de conception proposés dans la littérature.

Analyse du domaine à base des études terrain

Afin d'analyser le domaine de conception de produits et déterminer les différents artefacts métier qui lui appartiennent, nous avons analysé des résultats d'entretiens menés avec quatre entreprises françaises, dans le cadre d'une enquête menée par le club PLM⁷. Le résultat des entretiens a conduit à une liste conséquente d'objets métier. L'analyse de cette liste révèle l'existence de plusieurs termes partagés entre tous les acteurs du domaine, quelle que soit l'entreprise considérée. D'autres termes sont plus spécifiques à certaines entreprises. Après avoir éliminé les synonymes, une liste d'objets métiers plus réduite a été arrêtée.

Analyse des modèles de processus de conception

Nous nous sommes référés à la littérature afin de confronter le vocabulaire partagé avec les travaux du domaine. Nous avons ainsi analysé des modèles de processus de conception [Scaravetti 04] et [Pahl 07] [Tolleneanere 98] afin de déterminer les différents livrables qui jalonnent les activités des processus de conception.

Enfin, nous avons organisé des rencontres avec des chercheurs des laboratoires G-SCOP et SYMME, qui travaillent sur le domaine de la conception de produits. Ce travail a été mené dans le but de valider et d'enrichir le vocabulaire et la liste d'objet métier que nous avons identifiés. Nous ne gardons que les objets métier qui font sens aux acteurs du domaine. Nous considérons qu'un concept métier est éligible au rang d'un objet métier du domaine de la conception de produits manufacturés s'il remplit les conditions suivantes :

- il doit marquer la transition entre deux étapes différentes du domaine de la conception. En d'autres termes, deux étapes qui n'ont pas le même objectif ne peuvent pas avoir comme résultat le même objet métier ;
- il est souvent associé à un livrable pour valider la transition entre deux étapes du processus ;
- il permet de répondre à un des sous objectifs du processus dans lequel il a été crée ;
- il présente une forte valeur ajoutée pour le déroulement de la suite du processus de conception auquel il appartient.

Le Tableau 7 présente la liste définitive des objets métier du domaine de la conception de produits manufacturés Ces objets métier représentent une abstraction du produit désiré. Ils constituent également un vecteur de coopération entre les acteurs du domaine (cahier des charges, cahier des charges technique, note de calcul, etc.).

Tableau 7 : Objets métier du domaine de la conception de produits manufacturés

Cahier des charges client	<i>un document qui vise à préciser les exigences relatives au produit, laissant les détails de conception et de mise en œuvre pour le concepteur. Le cahier des charges sert à formaliser les besoins et à les expliquer aux différents acteurs pour s'assurer que tout le monde soit d'accord. Dans ce document, on peut spécifier des informations sur les performances attendues du produit, les objectifs de quantité, prix, délai souhaité, etc.</i>
Plan client	<i>une première représentation graphique (dessin) du produit tel qu'il est souhaité par le client.</i>
Planning	<i>décrit la planification du projet. Il défini le délai de déroulement de chaque étape et l'ordre d'enchaînement entre les étapes.</i>
Affectation ressources	<i>décrit l'affectation des ressources humaines aux différentes étapes du projet. Il cadre la</i>

⁷ Le club PLM est un club d'industriels et d'acteurs économiques et de la recherche qui a pour objectif le partager d'expériences en gestion de projet PLM, d'avoir une réflexion commune sur le plan stratégique de l'approche PLM et d'échanger sur les différentes utilisations des logiciels PLM [Club PLM 13].

humaines et matérielles	<i>mission de chaque acteur, dont celles du directeur de projet, du concepteur, et/ou du chef de projet, etc. Il permet aussi d'allouer les ressources matérielles nécessaires pour chaque étape.</i>
Cahier des charges fonctionnel	<i>un document qui détaille les différentes fonctions attendues d'un produit ainsi que les contraintes qui peuvent contraindre le bon fonctionnement du produit.</i>
Cahier des charges technique	<i>un document qui spécifie toutes les informations sur la faisabilité technique du produit, y compris le coût, la quantité, le délai, etc.</i>
Rapport d'étude du marché	<i>Cet objet métier représente un document qui vise à préciser la situation du marché, les besoins du marché et les opportunités qui peuvent être formulées sous la forme d'offre.</i>
Analyse dysfonctionnelle (AMDEC)	<i>un document qui énumère les modes de défaillance possibles avec la la probabilité d'occurrence et les conséquences associées. Il décrit la ou les mesures correctives qui peuvent être mises en œuvre en produisant la plus grande valeur rajoutée.</i>
Projet	<i>représente le projet de la conception de produits manufacturés.</i>
Procédure type	<i>décrit les procédures devant être suivies pour concevoir le produit.</i>
Normes	<i>décrit les normes à respecter lors de la conception du produit.</i>
Plan de test - Validation	<i>un document qui définit les différents tests à réaliser sur le produit.</i>
Matériaux	<i>décrit le choix de matériaux pour une pièce ou la totalité du produit.</i>
Principe de solution	<i>décrit un principe de solution qui respecte les exigences décrites dans le cahier des charges et qui remplit les fonctionnalités décrites dans la spécification fonctionnelle.</i>
Devis	<i>estime le coût total du projet.</i>
Liste de composants standards	<i>décrit le choix de composants standard. Ce choix de composants standard peut s'opérer à deux niveaux dans l'entreprise : par le service achats qui arrête la liste de Composants Standards utilisables dans l'entreprise et par le BE qui choisit un Composant Standard pour la nomenclature du produit à sa charge (parmi la liste de CS utilisables).</i>
Nomenclature	<i>décrit une liste des pièces, des assemblages et des matières premières qui composent un produit. Il définit la quantité de chaque pièce, et les regroupements entre les différents composants du produit.</i>
Modèle 3D du produit	<i>représente un modèle numérique du produit (CAO) tel qu'il est conçu par le concepteur. Ce modèle regroupe tout les composants du produit et leurs assemblages (dimension, hauteur, etc..).</i>
Modèle 3D pièce	<i>représente une modélisation numérique d'une partie du produit (CAO).</i>
Plan d'ensemble	<i>Cet objet métier décrit une représentation graphique du produit tel qu'il est conçu par le concepteur. Il s'agit d'un plan d'ensemble contenant tout les composants du produit et son équipement. Ce plan contient peu d'informations, seules les parties techniques, désignés pour le client.</i>
Plan détail	<i>décrit une représentation graphique d'un ou plusieurs composants du produit tel qu'il est conçu par le concepteur. Il s'agit d'un plan détaillé contenant toutes les informations nécessaires pour la validation de la conception. Il est plus détaillé que le dessin d'ensemble.</i>
Note de calcul	<i>décrit le résultat d'un calcul effectué sur une partie ou la totalité du produit.</i>
Prototype	<i>représente un modèle physique du produit à petite échelle. Il est utilisé pour visualiser et tester des formes et des idées, sans encourir le coût et l'effort de production d'un produit à grande échelle.</i>

Rapport de validation

décrit le résultat de l'application du plan de test sur un produit.

Étape 2 : Identification des actions métier du domaine de la conception de produits manufacturés

De même que pour l'identification des objets métier, nous proposons l'utilisation de plusieurs types de sources d'information pour l'identification des actions métier du domaine de la conception de produits manufacturés. Nous basons l'identification des actions métier sur l'analyse d'exemples réels de processus de conception de produits manufacturés. Ceci permettra de décomposer des exemples de processus en des tâches candidates au rang d'une action métier tel est le cas dans la démarche proposée par [Arsanjani 04] et [Kohlborn 09b]. Nous considérons que cette première analyse permet de dégager une liste conséquente d'actions candidates. Cette analyse de processus est par ailleurs couplée à une analyse du domaine qui consiste à décomposer des modèles de processus issus de la littérature en domaines fonctionnels. Ceci permettra de délimiter les différents types d'actions métiers du domaine de conception de produits manufacturés.

Analyse d'exemples de processus de conception de produits manufacturés

Le choix de l'exemple de processus de conception de produits manufacturés utilisé pour identifier des actions métiers est important. Il doit être suffisamment complet (couvre plusieurs aspects de la conception) pour que le nombre d'activités métier qui le composent soit significatif. Ces activités devraient correspondre à des fonctions réutilisables, donc candidates à devenir des services du niveau métier. Par ailleurs, l'exemple est d'autant plus intéressant que le nombre d'activités à automatiser au sein de ce processus est important. Ceci traduit une volonté de dégager des traitements récurrents.

Ainsi, nous avons retenu deux exemples de processus industriels. Le premier décrit un processus de réalisation des plans d'un produit selon des spécifications du client. Il inclut des activités de réalisation de modèle 3D, de plan 2D, de revue de conception avec le client, de modification de plan, etc. Le deuxième exemple décrit un processus de gestion des modifications. Il inclut des activités de demande de changement, de vérification d'état d'avancement de la réalisation d'un produit, de recherche de cas d'emploi du produit impacté par le changement, etc. Nous avons décomposé les exemples de processus en actions élémentaires (tâches). Le résultat de cette étape a conduit à une liste significative d'opérations élémentaires réutilisables. L'analyse de cette liste révèle l'existence d'opérations ayant des appellations différentes mais qui concourent à la réalisation du même objectif. Nous avons ainsi pu éliminer certaines actions redondantes et arrêter une première liste d'actions candidates. Toutefois, cette analyse d'exemples de processus ne peut pas mener à une liste exhaustive d'actions métier du domaine de la conception. En effet, on ne peut pas être sûr de la complétude de la liste d'actions identifiée à l'aide de la technique de décomposition d'exemples de processus. D'ailleurs, comme nous l'avons dit précédemment, plusieurs auteurs comme [Erl 05] et [kohlborn 09b] favorisent l'utilisation de plusieurs sources ou encore la combinaison de plusieurs techniques d'identification afin d'identifier une liste complète de services. Pour compléter cette étude, une analyse du domaine a été réalisée afin d'enrichir la liste des tâches que nous avons identifiées.

Analyse du domaine et de modèles de processus

De nombreux travaux décrivent les activités menées par les acteurs de la conception. La notion d'activité rend compte de l'accomplissement des tâches des acteurs du processus dans le but de décrire la structure réelle du processus de conception. Nous citons en particulier les travaux de Micaelli [Micaelli 02], Amhed [Amhed 02] et de Robin [Robin 04]. Ces travaux présentent différentes classifications possibles des activités menées par les acteurs de la conception. Les activités correspondent à des catégories génériques menées par les acteurs de la conception (cf. Tableau 8).

Tableau 8: Catégories d'activités du processus de conception

Analyser le problème,	Discuter de l'approche du processus.
S'informer sur le problème,	Identifier les critères.
Evaluer le problème,	Définir les critères.
Reporter le problème.	Pondérer les critères.
Proposer une solution,	Clarifier des principes de fonctionnement.
Clarifier la solution,	Clarifier l'environnement du produit.
Revenir sur une ancienne solution,	Délibérer de sous-problèmes.

Prendre une décision de conception, Consulter des informations externes, Reporter l'action de conception, Se projeter vers l'avenir, Regarder vers le passé. Analyser la solution proposée, Justifier la solution proposée, Calculer la solution proposée, Reporter l'analyse de la solution proposée,	Obtenir des informations extérieures. Avancer une preuve, une justification. Déterminer ou évaluer des performances Traduire l'intuition en classement. Contrôler le processus. Évaluer la solution proposée. Se rapporter aux connaissances existantes, Se rapporter aux domaines de compétences, Se rapporter à la stratégie de conception.
--	---

Nous nous sommes basés sur ces classifications d'activités et sur la liste d'opérations élémentaires que nous avons identifiées pour arrêter la liste finale d'actions métier du domaine de la conception de produits manufacturés.

Nous avons par la suite organisé des entretiens avec des chercheurs du laboratoire G-SCOP qui travaillent sur le domaine de la conception de produits. Ce travail a été mené afin de valider la liste d'actions métier que nous avons identifiées. Une action candidate est éligible au rang d'une action métier si elle correspond à un traitement réutilisable dans plusieurs exemples de processus de conception de produits manufacturés. Par ailleurs, elle ne doit pas posséder la même logique métier qu'une autre action.

Suite à ces analyses et entretiens nous avons proposé une liste d'actions métier du domaine de la conception de produits manufacturés contenant des actions génériques qui peuvent opérer sur les différents objets métier que nous avons identifiés dans la première étape de notre démarche d'identification de services du niveau métier (cf. Tableau 9 **Erreur ! Source du renvoi introuvable.**).

Tableau 9 : Liste d'action métier du domaine de la conception de produits manufacturés

Action PRODUCTION :	Action DEMANDE :
- <i>Information</i>	- <i>Information</i>
- <i>Solution</i>	- <i>Solution</i>
- <i>Evaluation</i>	- <i>Evaluation</i>
- <i>Validation</i>	- <i>Décision</i>

Nous avons pu classer les différentes actions métier issues de la décomposition d'exemples de processus de conception de produits manufacturés selon huit types d'actions (Production Information, Production solution, Production évaluation, Production validation, Demande Information, Demande solution, Demande évaluation et Demande validation). Ces actions sont représentatives de l'intention des concepteurs et de l'interprétation que ceux-ci font de leurs tâches. Elles fournissent un modèle des tâches élémentaires du processus de conception.

Étape 3 : Association des objets métier et des actions métier du domaine de la conception de produits manufacturés

Cette étape se base sur les deux livrables des étapes précédentes. Munis des deux listes d'objets métier et d'actions métier du domaine de la conception de produits manufacturés, nous entamons l'identification de la liste des opérations de services de conception de produits manufacturés. En effet, comme nous l'avons défini précédemment (cf. 3.1.1), une opération de services du niveau métier est définie comme une action métier qui opère sur un objet métier. Ainsi, son intitulé peut être vu comme un croisement d'une action métier avec un objet métier. Le résultat de ce croisement mène à une liste d'opérations génériques qu'on peut réutiliser dans plusieurs modèles de processus de conception de produits manufacturés (cf. Figure 17).

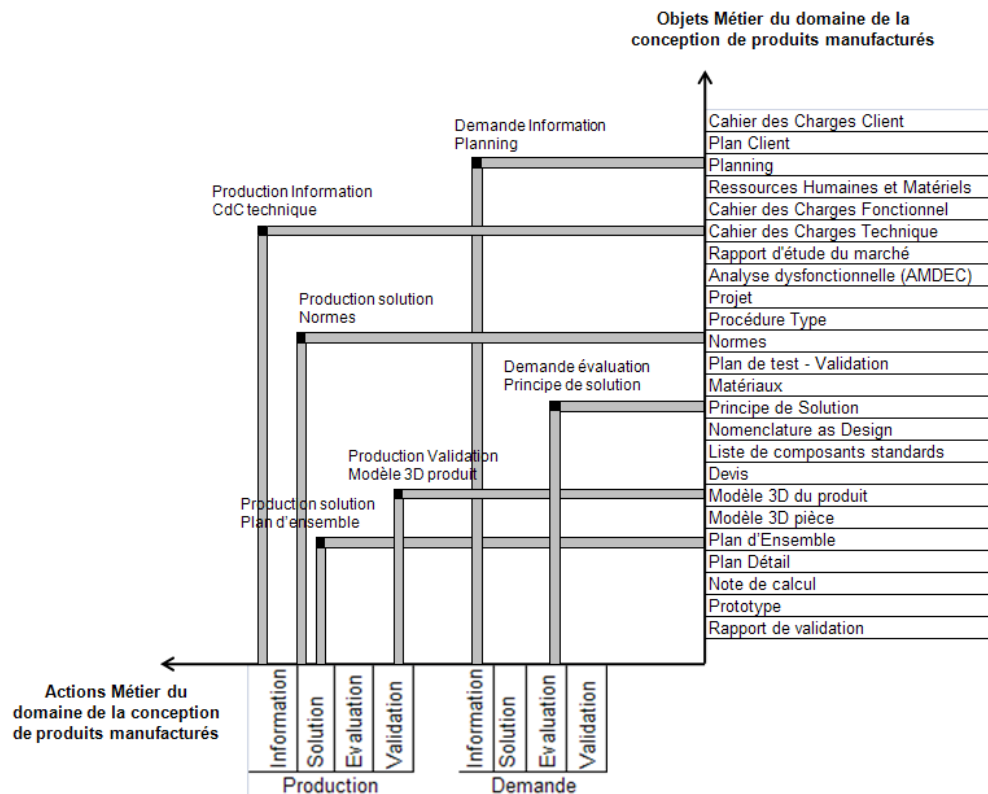


Figure 17: Croisement entre actions métier et objets métier de la conception de produits manufacturés

Ce croisement conduit à une liste d'opérations métier candidates pour faire partie des services de conception de produits manufacturés. Cependant, nous ne gardons pas toutes les opérations issues de l'application du croisement. Le résultat du croisement des deux concepts Action métier et Objet métier doit faire du sens aux acteurs du domaine de la conception du produits manufacturés (pour qu'elle soit significative vis-à-vis des acteurs du domaine de conception). Certaines opérations issues de ce croisement n'avaient pas de sens dans le domaine de conception de produits manufacturés. Par exemple, l'opération métier *Production Solution Norme* issue du croisement de l'action métier *Production Solution* et de l'objet métier *Norme* n'a pas de sens dans ce domaine, dû au fait que la liste des Normes utilisables dans l'entreprise est déjà arrêtée et les normes sont déjà créées. Le concepteur n'a donc qu'à choisir une norme parmi celles disponibles ou imposées. De même pour l'opération métier *Production Solution Plan Client* issue du croisement de l'action métier *Production Solution* et de l'objet métier *Plan Client* puisqu'un plan client est produit par le client.

Après l'élimination des opérations n'ayant pas de sens dans le domaine de conception de produits manufacturés, la liste finale consiste en un ensemble d'opérations métier génériques. Cette liste sera utilisée par les personnes métier de ce domaine pour définir plusieurs modèles de processus de conception de produits manufacturés. Ainsi, les appellations des opérations doivent être compréhensibles par les acteurs métier. Nous proposons donc de faire une révision de la liste finale d'opérations métier afin de la raffiner en proposant des appellations significatives aux personnes métier du domaine de la conception de produits manufacturés (Tableau 10 et Tableau 11).

Tableau 10 : Liste révisée des d'opérations métier de type Production

Objet Metier \Action M	P. solution	P.Information	P.Evaluation	P.Décision
Cahier des Charges Client		Diffuser le cahier des charges client	Evaluer le cahier des charges client	Valider le cahier de charges client
Cahier des Charges fonctionnel	Elaborer la specification fonctionnelle	Diffuser la specification fonctionnelle	Evaluer la specification fonctionnelle	Valider la specification fonctionnelle
Cahier des Charges technique	Elaborer la specification Technique	Diffuser la specification technique	Evaluer la specification technique	Valider la specification technique
Rapport de validation	Elaborer le rapport de validation	Diffuser le rapport de validation	Evaluer le rapport de validation	Valider le rapport de validation
Rapport d'etude du marche	Elaborer l'étude de marché	Diffuser le rapport d'étude de marché	Evaluer les résultats d'étude de marché	Valider l'étude de marché
Analyse dysfonctionnelle	Elaborer l'analyse dysfonctionnelle	Diffuser les résultats de l'analyse dysfonctionnelle	Evaluer la specification dysfonctionnelle	Valider la specification dysfonctionnelle
Plan de test	Elaborer plan de test	Diffuser le plan de test	Evaluer Plan de test	Valider plan de test
Plan d'ensemble	Creer plan d'ensemble	Diffuser plan d'ensemble	Evaluer plan d'ensemble	Valider plan d'ensemble
Plan de détail	Creer plan de détail	Diffuser plan de détail	Evaluer plan de détail	Valider plan de détail
Modele 3D Produit	Créer un modèle 3D de produit	Diffuser modèle 3D produit	Evaluer modèle 3D produit	Valider modèle 3D produit
Modele 3D pièce	Créer un modèle 3D du pièce	Diffuser modèle 3D pièce	Evaluer modèle 3D pièce	Valider modèle 3D pièce
Plan Client		Diffuser Plan client	Evaluer Plan Client	Valider le Plan Client
Procédure Type	Elaborer procedure de travail	Diffuser Procedure de travail	Evaluer Procedure de travail	valider une procedure du travail
Normes		Diffuser la norme	Evaluer Norme (applicabilité / répercussions)	Choisir Norme à appliquer
Ressources Humaines et Matériels	Affecter les ressources au projet	Diffuser l'affectation des ressources du projet	Evaluer l'affectation des ressources du projet	Valider l'affectation des ressources du projet
Planning	Planifier le projet	Diffuser le planning du projet	Evaluer le planning du projet	valider le planning du projet
Nomenclature as design	Créer la BOM	Diffuser la BOM	Evaluer la BOM	Valider la BOM
Devis	Elaborer devis	Diffuser devis	Evaluer Devis	Valider Devis
Liste de composants standards	Choisir composant standard*	Diffuser choix composant standard	Evaluer choix composant standard	valider choix composant standard
Principe de Solution	Proposer un principe de Solution	Diffuser un principe de solution	Evaluer un principe de solution	valider un principe de solution
matériaux	Choisir un matériau	Diffuser choix matériau	Evaluer choix matériau	valider choix matériau
Prototype	Creer prototype	Diffuser prototype	Tester le prototype	Valider le prototype
Projet / Affaire	créer un projet (lancement)	Notifier projet	Evaluer projet (pour son lancement / sa cloture)	valider projet (pour son lancement / sa cloture)
Note de calcul	Effectuer un calcul (simulation)	Diffuser un calcul	Evaluer un calcul	Valider un calcul

Tableau 11 : Liste révisée des opérations métier de type Demande

Objet Metier \Action M	D.solution	D.Information	D.Evaluation	D.Décision
Cahier des Charges Client		Consulter le cahier des charges	Demander une évaluation du cahier de charges client fonctionnelle	Demander une validation cahier des charges
Cahier des Charges fonctionnel	Lancer l'étude fonctionnelle	Consulter la spécification fonctionnelle	Demander une évaluation de la spécification fonctionnelle	Demander une validation de la spécification fonctionnelle
Cahier des Charges technique	Lancer l'étude technique	Consulter la spécification technique	Demander une évaluation de la spécification technique	Demander une validation de la spécification technique
Rapport de validation	Lancer la validation produit	consulter le rapport de validation	Demander une évaluation du rapport de validation	Demander une validation du rapport de validation
Rapport d'étude du marché	Lancer une étude de marché	Consulter le rapport d'étude de marché	Demander une évaluation de l'étude de marché	Demander une validation de l'étude de marché
Analyse dysfonctionnelle	Lancer l'analyse dysfonctionnelle	Consulter la spécification dysfonctionnelle	Demander une évaluation de la spécification dysfonctionnelle	la spécification dysfonctionnelle
Plan de test	Demander plan de test	Consulter plan de test	Demander une évaluation du plan de test	Demander une validation du plan de test
Plan d'ensemble	Demander mise en plan d'ensemble	Consulter plan d'ensemble	Demander une évaluation du plan d'ensemble	Demander une validation du plan d'ensemble
Plan de détail	Demander mise en plan de détail	Consulter plan de détail	Demander une évaluation du plan de détail	Demander une validation du plan de détail
Modele 3D Produit	Demander modélisation 3D produit	Consulter modèle 3D produit	Demander une évaluation du modèle 3D produit	Demander une validation du modèle 3D produit
Modele 3D pièce	Demander modélisation 3D pièce	Consulter modèle 3D pièce	Demander une évaluation du modèle 3D pièce	Demander une validation du modèle 3D pièce
Plan Client		Consulter Plan Client	Demander une évaluation du Pla	Demander une validation du Plan Client
Procédure Type	Demander nouvelle Procédure	Consulter procédure de travail	Demander une évaluation de la Procédure de travail	Demander une validation de la procédure de travail
Normes		Consulter norme	Demander l'évaluation d'une norme	Demander décision sur une norme
Ressources Humaines et Matériels	Demander l'affectation de ressources au projet	Consulter les ressources affectées au projet	demander l'évaluation de l'affectation ressources/projet	demander une validation de l'affectation ressources/projet
Planning	Demander la planification du projet	Consulter le planning du projet	Demander l'évaluation du planning de projet	Demander une validation du planning de projet
Nomenclature as design	Demander BOM	Consulter BOM	Demander l'évaluation dun BOM	Demander la validation dun BOM
Devis	Demander devis	Consulter devis	Demander l'évaluation d'un devis	Demander la validation d'un devis
Liste de composants standards	Demander un choix de composant standard	Consulter C.S.	Demander une évaluation du choix de composant standard	Demander validation choix de composant standard
Principe de Solution	Demander un principe de solution	Consulter un principe de solution	Demander une évaluation d'un principe de solution	Demander une validation d'un principe de solution
matériaux	Demander un choix de matériau	Consulter matériau	Demander une évaluation du choix de matériau	Demander une validation du choix de matériau
Prototype	Lancer creation du prototype	Demander le prototype	Demander une évaluation du prototype	Demander une validation du prototype
Projet / Affaire	Demander création projet	Consulter projet	Demander une évaluation de projet (pour son lancement / sa cloture)	Demander une validation de projet (pour son lancement / sa cloture)
Note de calcul	Demander un calcul	Consulter un calcul	Demander une évaluation du calcul	Demander une validation du calcul

Étape 4 : Identifications des paramètres d'entrée et de sortie des opérations identifiées

Après avoir identifié cet ensemble d'opérations de service, nous nous sommes focalisés sur l'identification de leurs paramètres d'entrée/sortie. Nous avons conclu que les entrées et les sorties correspondent aux objets métier sur lesquels agit l'opération pour réaliser le traitement en question. Il s'agit essentiellement de représentations de produit (par exemple document) qui peuvent changer d'état à la sortie d'une opération de service suite à la modification de leur contenu ou bien des informations qui les définissent (par exemple, l'attribut responsable de l'objet document, etc.). Selon le type d'entrées/sorties, nous distinguons deux types d'opérations :

- les opérations de services qui agissent sur les objets métier (création, modification de l'état, etc.). Ce sont des opérations qui créent ou transforment un état d'entrée (de l'objet métier en question), sous l'influence d'objets de contrôle⁸ et sous réserve de disponibilité de ressources

⁸ les objets dont dépend l'objet métier en question tel que le cahier des charges par exemple

nécessaires⁹ en un état de sortie (nouvel état de l'objet métier en question). Ces opérations correspondent aux opérations de service de type production.

- les opérations de services porteuses d'informations. Ce sont des opérations qui ne produisent pas de données mais qui sollicitent des informations qui déclenchent l'exécution d'une autre opération de service. Ces opérations correspondent aux opérations de service de type Demande.

Ainsi nous avons proposé des E/S génériques pour chaque type d'opération (cf. Tableau 12). Nous avons par la suite appliqué cette proposition générique sur l'ensemble des opérations métier spécifiques que nous avons identifiées (cf. Tableau 10 et Tableau 11). Ce travail a permis d'identifier 480 paramètres d'entrée/sortie pour les 194 opérations de services que nous avons identifiées. Les paramètres d'entrée/sortie ainsi que les opérations de service que nous avons identifiées, ont été décrits par le biais de fiches descriptifs (cf. quelques exemples de ces fiches dans l'annexe A). De cette manière, le niveau métier de notre approche est indépendant de toute implémentation particulière. Les services de ce niveau (opérations de services) sont exprimés en langage naturel avec des termes familiers pour les utilisateurs du domaine de conception de produits manufacturés.

Tableau 12: Entrée / sortie génériques des opérations métier du domaine de la conception

Type de l'opération	Entrée	Sortie
Production information	personne à qui on veut envoyer l'information	l'objet à envoyer (sujet de l'action de l'opération)
Production Décision	objet à approuver	objet à approuver sujet avec le nouvel état
Production évaluation	objet à évaluer	objet évalué avec le nouvel état
Production Solution	Deux cas se présentent : - Pour la création : objet auquel dépend le sujet de l'action tel que Cahier des Charges - Pour la modification : ordre De Modification	objet avec nouvel état (créé/ modifié)
Demande évaluation	personne a qui on veut envoyer la demande d'évaluation, objet à évaluer	
Demande solution	Deux cas se présentent : - Demande de création : personne à qui on veut envoyer la demande, objet auquel dépend l'objet à créer - Pour la modification : personne à qui on veut envoyer la demande et ordre de modification	
Demande Information	- objet à demander	
Demande décision	- personne n'a qui on veut envoyer la demande, l'objet à approuver	

Étape 5 : Regroupement des opérations métier identifiées en services de conception de produits manufacturés

Cette dernière étape de la démarche d'identification de services du niveau métier consiste à regrouper les opérations métier identifiées en services de conception de produits manufacturés. Pour mener à bien cette dernière étape, notre démarche réutilise des critères de groupement souvent utilisés par les approches orientées services à savoir les deux critères de dépendance fonctionnelle [Credle 08] et de cohérence fonctionnelle [Hao 03].

Le critère de dépendance fonctionnelle consiste à grouper les opérations qui sont fortement couplées dans un même service. Ceci favorise le faible couplage entre les services [Credle 08]. L'objectif est d'introduire le minimum de dépendances entre les services pour permettre d'assembler ceux-ci aisément. Il s'agit notamment de favoriser la réutilisation de services existants et déjà déployés, ainsi que leur combinaison, afin de répondre rapidement et à faible coût à de nouveaux besoins métier.

⁹ ressources humaines qui réalisent l'opération de service

Le critère de cohérence fonctionnelle consiste à grouper les opérations qui concourent à la réalisation du même objectif dans un même service. Ceci favorise la forte cohésion entre les opérations du même service [Hao 03]. La cohésion est une règle qui traduit le degré de proximité fonctionnelle des opérations exposées par un service [Hao 03]. En d'autres termes, elle vise à favoriser la facilité de compréhension et la réutilisation d'un service en y regroupant des opérations homogènes constituant une unité métier.

Pour ce faire, nous proposons de dresser une matrice de dépendance entre les différents objets métier. L'objectif de cette matrice est de déterminer des groupes d'objets dépendants. Les objets dépendants sont regroupés dans un même groupe (cf. Figure 18).

	CdC Client	Plan Client	Rapport d'étude du marché	Normes	Procédure Type	Devis	CdC technique	Projet	R. H/M	Planning	spécification dysfonctionnelle	CdC fonctionnel	Plan de test	Design Solution	Composants standards	Note de Calcul	Matériaux	Nomenclature as Design	Modèle 3D produit	Modèle 3D pièce	Plan Ensemble	Plan détail	Prototype	Rapport de validation	
CdC Client	*																								
Plan Client	*																								
Rapport d'étude du marché	*																								
Normes	*																								
Procédure Type	*			*																					
Devis	*		*	*	*	*																			
CdC technique	*	*	*	*	*	*																			
Projet	*																								
R. H/M	*						*	*	*	*															
Planning	*						*	*	*	*															
spécification dysfonctionnelle	*						*				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
CdC fonctionnel	*						*				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Plan de test	*						*				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Design Solution											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Composants standards											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Note de Calcul							*				*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Matériaux											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Nomenclature as Design											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
Modèle 3D produit																		*	*	*	*	*	*	*	*
Modèle 3D pièce																		*	*	*	*	*	*	*	*
Plan Ensemble																		*	*	*	*	*	*	*	*
Plan détail																		*	*	*	*	*	*	*	*
Prototype																		*	*	*	*	*	*	*	*
Rapport de validation													*	*	*	*	*	*	*	*	*	*	*	*	*

Figure 18 : Matrice de dépendance entre les objets métier du domaine de la conception

Par conséquent les opérations qui opèrent sur ces objets sont regroupées dans un même service. Ceci favorise le faible couplage entre les services et permet de prendre en compte le critère de dépendance fonctionnelle. Ainsi, nous obtenons une première ébauche de regroupement d'opérations métier en six services de conception de produits manufacturés :

- Un service *Gestion Produit / offre* qui regroupe les opérations qui opèrent sur les objets métier du groupe coloré en jaune dans la Figure 18.
- Un service *Gestion projet* qui regroupe les opérations qui opèrent sur les objets métier du groupe coloré en vert.
- Un service *BE Analyse fonctionnelle* qui regroupe les opérations qui opèrent sur les objets métier du groupe coloré en violet.
- Un service *BE Conception préliminaire* qui regroupe les opérations qui opèrent sur les objets métier du groupe coloré en bleu.
- Un service *BE Conception Détaillée* qui regroupe les opérations qui opèrent sur les objets métier du groupe coloré en rouge.
- Un service *BE prototypage* qui regroupe les opérations qui opèrent sur les objets métier du groupe coloré en Orange.

Cette première ébauche de groupement d'opérations métier en services conduit à une liste de services contenant plusieurs opérations. Nous proposons donc de prendre en compte d'autres critères de groupement afin de raffiner le groupement. Nous nous référons à la technique de décomposition du domaine afin de déterminer les corps de métier du domaine de la conception de produits manufacturés. L'objectif de raffinement du groupement des opérations métier par corps de métier est d'avoir des services moins grands et plus parlant aux personnes du domaine (cf. Figure 19). Ceci permettra en retour de faciliter la découverte des opérations lors de la définition d'un processus de conception ou encore au moment du changement.

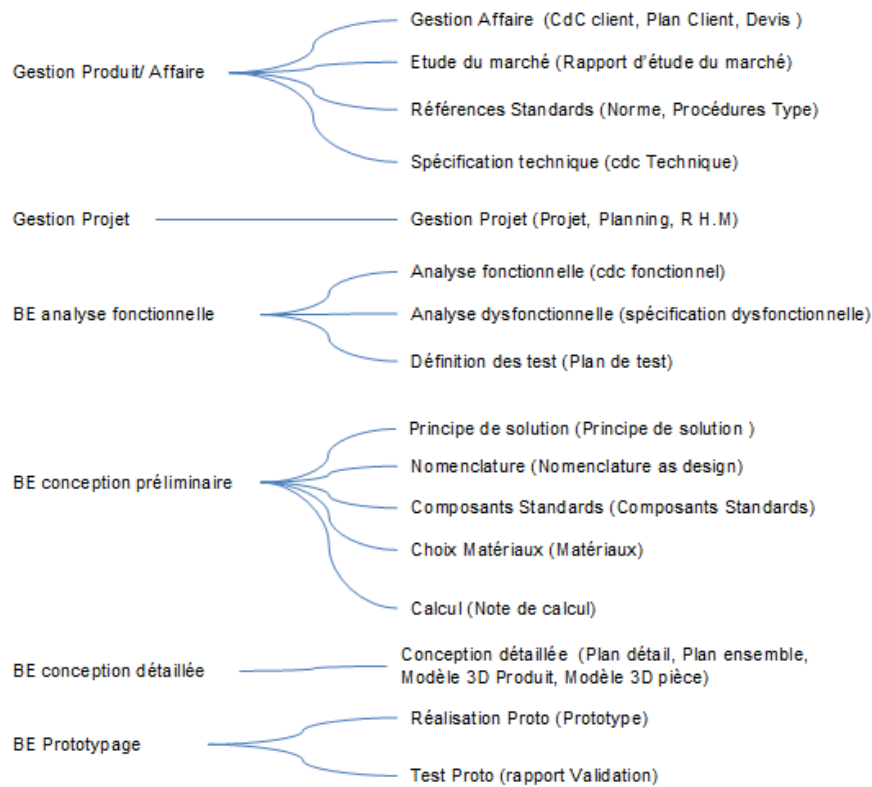


Figure 19 : Raffinement du groupement des opérations métier par décomposition en corps de métier

A part ces deux critères de groupement, nous proposons de prendre en compte, certaines règles métier du domaine de la conception de produits manufacturés. En effet, certaines opérations métier, comme celles du domaine *Test proto* (cf. Figure 19), sont souvent utilisées avec d'autres opérations, en l'occurrence les opérations métier de type Demande qui opèrent sur l'objet métier Prototype. Nous proposons donc de déplacer ce type d'opérations dans le service *Test Proto*. Nous faisons de même, pour les deux services *Calcul* et *Conception Détaillée*. En effet, les opérations du service calcul, pour être utilisées, ont besoin des opérations du service *Conception Détaillée*. Nous proposons donc de regrouper ces deux services.

Au final, le catalogue de services de conception de produits manufacturés proposé comprend 194 opérations de service regroupées dans 16 services (cf. Figure 20) (cf. Annexe A pour la version en anglais des services de conception de produits).

fonctionnels PLM (§ 3.2.2), nous nous attardons sur la définition du concept de Service fonctionnel PLM (§ 3.2.1).

3.2.1 Définition des services fonctionnels PLM

Les services fonctionnels PLM sont des services qui font sens aux utilisateurs des systèmes PLM. Ils reflètent les fonctions attendues d'un système PLM [Hachani 11].

Un service fonctionnel PLM est un groupement d'opérations fonctionnelles PLM. Chaque opération fonctionnelle PLM traduit le concept de tâche et expose une des fonctionnalités offertes par un système PLM telle qu'elle est vue par les utilisateurs de la partie informatisée du système PLM. Ils représentent donc les activités du système PLM indépendamment de toute réalisation technique (comparer deux nomenclatures de produits, créer un lien entre deux articles, etc.). Les services fonctionnels PLM sont des services de granularité moyenne, ceux permettant de lancer une simulation, ou encore liées au calcul de similarité entre deux nomenclatures de produits.

Une fois déployé, les services fonctionnels PLM participent à la réalisation de plusieurs opérations de services de conception de produits manufacturés. Les opérations de services fonctionnels PLM vont être choisies par les concepteurs de processus pour mettre en oeuvre le flux de travail associé à une opération de service du niveau métier (*SCP*).

Les opérations de services fonctionnels PLM représentent toutes les manipulations et traitements possibles sur les objets techniques enregistrés dans la base de données du système PLM. Par conséquent, identifier les opérations de services fonctionnels PLM revient à identifier tous les objets techniques qu'on trouve dans un système PLM et les différentes fonctionnalités offertes par le système afin de créer, modifier ou faire des traitements sur ces objets techniques.

3.2.2 Démarche d'identification de services fonctionnels PLM

Nous proposons une démarche ascendante pour l'identification des services fonctionnels PLM.

Cette démarche est basée sur la technique d'analyse du patrimoine applicatif [Arsanjani 08] – système PLM dans notre cas. La technique d'analyse du patrimoine applicatif consiste à analyser les applications de l'entreprise afin d'identifier les services permettant de mettre en oeuvre les processus du domaine [Arsanjani 08]. L'objectif de cette technique est généralement d'identifier des services de bas niveau tels que les services techniques ou les services entités qui permettent d'agir sur le cycle de vie des entités principales du domaine.

Comme ressources supplémentaires, notre démarche d'identification est principalement basée sur les études de terrain. En effet, à part l'analyse de systèmes PLM existants, nous avons effectué des entretiens avec des utilisateurs PLM. Nous avons également effectué une revue de littérature pour l'identification des objets techniques qu'on peut trouver dans les systèmes PLM.

Quatre étapes constituent la démarche d'identification de services fonctionnels PLM proposée :

- Étape 1 : Identification des différentes catégories de données gérées dans les systèmes PLM
- Étape 2 : Identification des opérations fonctionnelles PLM de chaque objet PLM
- Étape 3 : Identification des E/S des opérations fonctionnelles identifiées
- Étape 4 : Regroupement des opérations fonctionnelles en services fonctionnels PLM

Étape 1 : Identification des différentes catégories de données gérées dans le PLM

Cette première étape de la démarche d'identification de services fonctionnels PLM, consiste à identifier les différentes catégories d'objets techniques PLM. Un Objet technique PLM est une représentation de tout ou partie d'un objet métier du domaine de la conception de produits manufacturés. Ce sont les objets techniques enregistrés dans la partie informatisée du PLM et sur lesquels des traitements peuvent être faits à travers les fonctionnalités de l'interface du système PLM. Pour l'identification de ces objets techniques PLM, plusieurs types de sources d'information sont utilisés. Nous considérons dans notre approche, que l'identification des objets techniques PLM doit

être basée d'une part sur l'analyse d'exemples de systèmes PLM du marché. Ceci consiste à analyser le patrimoine applicatif afin de dégager les informations nécessaires pour l'identification des services fonctionnels PLM. D'autre part, nous proposons d'analyser le modèle de données proposés dans la littérature pour représenter le produit dans les systèmes PLM. Ces modèles sont souvent proposés en vue de spécifier des systèmes PLM répondant aux besoins métiers en matière de gestion des données produit. L'analyse de ces modèles permet de déduire l'ensemble d'objets PLM qu'on peut trouver dans un système PLM. Cette double analyse est suivie par une série d'entretiens avec des utilisateurs de systèmes PLM afin de valider et enrichir de la liste d'objets PLM identifiés.

Analyse d'exemples de systèmes PLM

Nous proposons de commencer par examiner deux exemples de systèmes PLM du marché en se référant à la technique d'analyse du patrimoine applicatif dans le but d'identifier les données qui peuvent être manipulés dans les systèmes PLM par les acteurs des processus de conception de produits manufacturés. Ces données techniques constituent les différentes informations liées au produit. Elles sont structurées dans un modèle de produit. Comme nous l'avons défini dans le premier chapitre, le modèle de produit est considéré comme un réservoir d'informations des données du produit. Nous avons ainsi choisi deux systèmes PLM du marché : Audros [Audros 13], principalement utilisé dans des petites et moyennes entreprises, et Windchill [Windchill 13], principalement utilisé dans de grands groupes. Ceci nous permettra de dégager les différentes catégories de données pouvant être traitées dans ces deux systèmes et d'identifier les données techniques PLM de chaque catégorie.

L'analyse de ces deux systèmes PLM, nous a permis de dégager trois catégories de données PLM (cf. Figure 21).

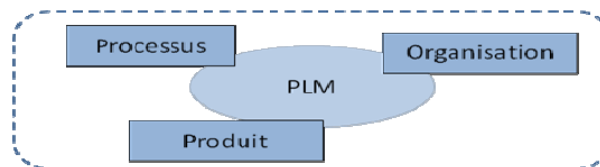


Figure 21 : Catégories de données PLM

En effet, la base de données couplée avec le système PLM gère des données en lien avec le produit mais aussi des données en lien avec l'organisation de l'entreprise et ses processus. Ceci s'explique par le fait que le PLM gère les données représentant le produit mais aussi tous les traitements effectués sur ces données, i.e. les processus métiers dont les activités permettent de créer et transformer des données produit, ainsi que l'organisation habilitée à manipuler ces données (les permissions et les droits d'utilisation des informations du produit par les utilisateurs de l'organisation). De plus, les processus sont définis en s'appuyant sur le modèle produit et organisationnel du PLM. La dualité de la gestion des données produit et des processus associés permet de savoir, entre autres, pour quel produit un processus est déclenché, etc.

Chacune de ces catégories encapsule un ensemble de données PLM. Le tableau ci-dessous présente une vue d'ensemble de ces données (cf. Tableau 13).

Tableau 13 : Objets PLM par catégorie

Catégorie	Donnée PLM	Description
Organisation	Participant	<i>Cet objet PLM définit les informations concernant un utilisateur donné du système PLM.</i>
	Action	<i>Cet objet PLM décrit une autorisation de type consultation, modification, création ou encore suppression qui peut être appliquée sur un objet PLM donné.</i>
	Rôle	<i>Cet objet PLM définit une collection d'actions sous le nom de rôle. Ce rôle peut être affecté à un ou plusieurs participants.</i>

Processus	Processus	<i>Cet objet PLM décrit les informations concernant un processus donné.</i>
	Etape	<i>Cet objet PLM décrit les informations en lien avec une étape qui peut faire partie d'un processus.</i>
	Indicateur	<i>Cet objet encapsule des informations qui synthétisent les informations en lien avec un processus donné.</i>
Produit	Produit	<i>Cet objet PLM permet de décrire toutes les informations qui permettent de différencier une instance donnée de produit.</i>
	Article	<i>Cet objet PLM permet de décrire les informations définissant un article faisant partie d'un produit.</i>
	Document	<i>Cet objet PLM permet de pointer sur un fichier (norme, procédure de travail, etc..) et défini toutes les informations permettant de différencier un document d'un autre.</i>
	Article CAO	<i>Cet objet PLM permet de référencier un fichier CAO et de grouper toutes les informations permettant de l'identifier et de le consulter</i>
	Nomenclature	<i>Cet objet PLM décrit les différents types de lien que peut avoir un produit avec les autres types d'objets techniques PLM.</i>

Une réunion avec des utilisateurs de systèmes PLM a permis de valider la liste d'objets PLM identifiés. Des membres des laboratoires SYMME et G-SCOP ont participé à cette réunion.

Étape 2 : Identification des opérations fonctionnelles PLM pour chaque objet PLM

Cette deuxième étape de la démarche d'identification de services fonctionnels PLM, consiste à identifier les différentes opérations fonctionnelles PLM. Une opération fonctionnelle PLM est une fonctionnalité qui opère sur un ou plusieurs objets techniques PLM. Ces fonctionnalités sont offertes par plusieurs systèmes PLM mais peuvent être implémentées différemment d'un système PLM à un autre. Cette étape utilise le résultat de la première étape de la démarche d'identification de services fonctionnels PLM, à savoir la liste d'objets techniques PLM. Nous proposons d'identifier pour chaque objet technique issu de la première étape la liste des opérations fonctionnelles PLM, qui peuvent opérer sur cet objet. Pour ce faire, nous proposons d'analyser les deux systèmes PLM que nous avons déjà utilisé dans la première étape de cette démarche. L'analyse de ces deux systèmes PLM nous a permis d'identifier plusieurs groupes d'opérations fonctionnelles PLM :

- des opérations permettant la création d'objets PLM. La création peut être faite à partir d'un modèle qui définit tous les attributs de l'objet. Pour un objet de type document le modèle peut même définir une trame du fichier attaché à l'instance de l'objet document.
- des opérations permettant la consultation des différentes informations qui définissent un objet PLM. En effet, dans un système PLM les objets sont définis par des attributs tels que la référence de l'objet, la date de création, la version de l'objet, etc. (cf. Figure 22).

Attributs	Valeurs
Référence	REN1349_OP40
Désignation	Assemblage final
Créé le	05/09/2008
Version	1
Statut	Initial
Numero_OP	OP40
CentredeCharge	100 Assemblage
Code_Article	REN1349
Designation_Article	amortisseur

Figure 22 : Les attributs d'un objet PLM

- des opérations permettant la suppression des objets PLM.
- des opérations permettant la modification des attributs d'un objet PLM.
- des opérations permettant de protéger les objets PLM. En effet, le PLM propose des fonctions qui permettent de protéger les documents en modification. Ainsi la récupération d'un document par un utilisateur (check out) doit être précédée par un verrouillage de l'instance sur le serveur de données afin qu'aucune autre personne ne puisse modifier le document avant que celui-ci soit réintégré (check in).
- des opérations permettent de relier les objets PLM entre eux. Cela permet de structurer le produit en termes de nomenclature (cf. Figure 23).

Lien	Structure	Référence	Désignation
	ArticleENS	PEU2405	accouplement 300x200
—	Devis	2008-38	accouplement 300x200
—	Planning	PEU2405	accouplement 200x300
—	PlanClient	PEU2405	accouplement 300x200
—	ArticleCOMPO	PEU2405-01	Flector élastique
—	FicheTechAchat	PEU2405-01	Flector élastique
—	PlanClient	PEU2405-01	Flector élastique
—	ArticleCOMPO	PEU2405-02	Moyeu rep 2
—	Norme	E01.45.415.G	Fonte perlitique à graphit
—	PlanClient	PEU2405-02	Moyeu rep 2

Figure 23: Structuration d'objets PLM

- des opérations permettant d'effectuer des recherches ou des analyses à partir d'une nomenclature. Par exemple une opération permettant de chercher les cas d'emploi d'un composant retourne tous les produits avec lesquels le composant à un lien.
- des opérations permettant de visualiser, stocker et télécharger des données PLM qui sert comme documentation à d'autres objets PLM. En effet, certaines informations produit sont archivés dans la base de données du système PLM grâce aux classes d'objet et aux attributs mais d'autres informations comme les fichiers CAO, les fichiers de photos numérisées, etc. ne sont que référencées. Le PLM offre donc des opérations permettant d'assurer l'import et l'export de ces objets.

La plupart des catégories d'opérations que nous avons identifiées s'applique à tous les objets PLM que nous avons identifiés. D'autres ne s'appliquent que sur un objet PLM donné.

Enfin, nous avons validé la liste d'opérations fonctionnelles PLM que nous avons identifiées avec des utilisateurs PLM. Ainsi, nous avons effectué une réunion avec des membres des laboratoires SYMME et G-SCOP. Le résultat de cette réunion nous a permis de valider et d'enrichir la liste d'opérations fonctionnelles PLM.

Étape 3 : Identification des paramètres d'entrée et de sortie des opérations identifiées

Après avoir identifié cet ensemble d'opérations fonctionnelles PLM, nous nous sommes focalisés sur l'identification de leurs paramètres d'entrée/sortie. Les paramètres d'entrées et de sorties des opérations fonctionnelles PLM constituent les attributs des objets PLM manipulés par ces opérations (par exemple *refrence_document* et *Id_Editeur* pour l'objet PLM *Document*).

Cette étape a permis d'identifier 188 paramètres d'entrée/sortie pour les 111 opérations fonctionnelles PLM que nous avons identifiées (cf. Tableau 14). De même que pour les opérations de services métier, les paramètres d'entrée/sortie ainsi que les opérations de service fonctionnels PLM que nous avons identifiées, ont été décrits par le biais de fiches descriptifs (cf. quelques exemples de ces fiches dans l'annexe A). De cette manière, le niveau fonctionnel PLM est aussi indépendant de toute implémentation particulière.

Tableau 14: Extrait des entrées et des sorties des opérations fonctionnels PLM

Operation fonctionnelle PLM	Entrée	Sortie
DefinirNouveauDocument	nom_document	reference_document
AfficherDocumentEditeur	reference_document	nom_editeur
DetruireDocument	reference_document	
CreerParticipant	nom_du_participant	id_participant
DefinirInfosProduit	reference_produit	
CreerProduitDeModele	reference_modele_produit, libelle_produit	reference_produit
ModifierInfosProduit	reference_produit,	
NouvelleVersionProduit	reference_produit,	numero_version_produit
ChangerstatutProduit	reference_produit, statut_produit	
NouvelleRevisionProduit	reference_produit	
GenererDocumentationProduit	reference_produit	reference_document
ExporterProduitCommeModele	reference_produit	reference_modele_produit
ChercherProduct	reference_produit	URL
AfficherInfosProduit	reference_produit	
AfficherAttachmentProduit	reference_produit	reference_attachement
VoirEditeurProduit	reference_produit	nom_editeur
VoirHistoriqueProduit	reference_produit	
DetruireProduit	reference_produit	
AfficherVersionProduit	reference_produit	numero_de_version
CheckInProduit	reference_produit	
CheckOutProduit	reference_produit	
AfficherEtapesTerminer	reference_process	reference_etapes
AfficherEtapeCourante	reference_process	reference_etape
AfficherStatutEtapeCourante	reference_process	statut_etape
AfficherRoleParticipantEtape	reference_etape	ref_role
CopierStructureArticleCommeModele	reference_article	ref_modeleStructure_article
CopierStructureProduitCommeModele	reference_produit	ref_modeleStructure_produit
DefinirArticle	libelle_article	reference_article
DefinirInformationArticle	reference_article, id_editeur, date_creationArticle, etc	
CreerArticleDeModele	ref_modele_article, libelle_article	reference_article
ModifierInfosArticle	reference_article, id_editeur, date_creationArticle, etc	
NouvelleVersionArticle	reference_article	numero_de_version

Étape 4 : Regroupement des opérations fonctionnelles PLM en services

La dernière étape de notre approche d'identification de services fonctionnels PLM, consiste à regrouper les opérations fonctionnelles PLM, issues de l'étape 2, dans des services fonctionnels PLM. Pour mener à bien cette dernière étape, notre démarche réutilise les deux critères de groupement dépendance fonctionnelle et cohérence fonctionnelle. Comme nous l'avons défini précédemment, le critère de dépendance fonctionnelle consiste à grouper les opérations qui sont fortement couplées dans un même service ce qui permettra de favoriser le faible couplage entre les services fonctionnels PLM. Quant au concept de cohérence fonctionnelle, il consiste à grouper les opérations qui concourent à la réalisation du même objectif dans un même service ce qui permettra de favoriser la forte cohésion entre les opérations du même service fonctionnel PLM. Contrairement à la démarche de groupement d'opérations métier, on se limite à l'utilisation de ces deux critères de groupement.

Le groupement est ainsi réalisé en construisant deux matrices carrées CF et DF appartenant à M_{111} avec 111 le nombre d'opérations fonctionnelles PLM identifiées. L'objectif de DF est de tester les dépendances fonctionnelles entre les 111 opérations fonctionnelles PLM identifiées. Nous considérons qu'il existe une dépendance fonctionnelle entre une opération_i et une opération_j avec $0 < i < 112$ et $0 < j < 112$ si les deux opérations sont souvent utilisées ensemble (l'une après l'autre). Le fait de regrouper

ces deux opérations permettra de réduire le couplage entre les services fonctionnels PLM. L'objectif de CF est de tester le critère de cohérence fonctionnelle entre les différentes opérations fonctionnelles PLM. En effet, il existe une cohérence fonctionnelle entre une opération_i et une opération_j avec $0 < i < 112$ et $0 < j < 112$ si les deux opérations concourent à la réalisation d'un objectif commun. Le fait de les grouper dans un même service, permettra de favoriser la forte cohésion entre les opérations de service fonctionnel PLM. Pour remplir la matrice DF, nous proposons d'identifier pour chaque opération les cas d'utilisation dans lesquelles l'opération peut être utilisée (cf. Tableau 15).

Tableau 15 : Extrait de l'identification des cas d'utilisation des opérations fonctionnelles PLM

Opérations	Cas d'utilisation
LierArticleProduit	DéfinitionNomenclature,
LierDocumentProduit	DéfinitionNomenclature,
DetruireLienArticleProduit	ModificationdunArticle, DéfinitionNomenclature
DefinirQuantiteLien	DéfinitionNomenclature,
AjouterSignatureElectronique	ModificationDocument,
CheckOutDocument	ModificationDocument,
CheckInDocument	ModificationDocument,
DownloadAttachement	ModificationDocument, ConsultationDocument
EditerAttachement	ModificationDocument, CréationDocument
ModifierInfosDocument	ModificationDocument

Pour remplir la matrice CF, nous proposons d'identifier pour chaque opération l'objectif principal d'utilisation de l'opération. Cet objectif correspond à un besoin de haut niveau. Ce qui fait que les opérations nécessaires à l'accomplissement de ce besoin seront chorégraphiées ensemble (cf. Tableau 16).

Tableau 16 : Extrait de l'identification des objectifs des opérations fonctionnelles PLM

Opération	Objectif de haut niveau
GenererDocumentationArticle	Répondre à une demande de client, Demande de Validation
DefinirAction	Lancement de Projet, Modification de droit, Gestion de RH
ChercherVersionArticle	Répondre à une demande de client
LierActionsAuRole	Lancement de Projet
VoirHistoriqueArticle	Répondre à une demande de client
CreateArticleDeModele	Répondre à une demande de client,
CreerRole	Lancement de Projet, Gestion RH

Nous dressons les deux matrices de la façon suivante. Les en-têtes des colonnes des matrices sont les mêmes que les en-têtes des lignes. Ces en-têtes correspondent aux intitulés des opérations fonctionnelles PLM que nous avons identifiées. Nous remplissons les éléments des deux matrices comme suit : l'élément a_{ij} de la matrice, avec $0 < i < 112$ et $0 < j < 112$, décrit la relation entre l'opération_i et l'opération_j en valeurs binaires (0 ou 1). Si les deux opérations opération_i et opération_j sont dépendantes (i.e. ont un cas d'utilisation en commun) (cf. Figure 24) / cohérentes (i.e. ont le même objectif) (cf. Figure 25) l'élément $a_{ij} = 1$ sinon l'élément $a_{ij} = 0$.

	Detruire Produit	Affecter Participant AEtape	Afficher Version Produit	Modifier Participant Etape	CheckIn Produit	Notifier Participant ParE-Mail	CheckOut Produit	Afficher Notification SurBureau	Creer Participant
DetruireProduit	1	0	1	0	1	0	1	0	0
AffecterParticipantAEtape	0	1	0	1	0	1	0	1	0
AfficherVersionProduit	1	0	1	0	1	0	1	0	0
ModifierParticipantEtape	0	1	0	1	0	1	0	1	0
CheckInProduit	1	0	1	0	1	0	1	0	0
NotifierParticipantParE-Mail	0	1	0	1	0	1	0	1	0
CheckOutProduit	1	0	1	0	1	0	1	0	0
AfficherNotificationSurBureau	0	1	0	1	0	1	0	1	0
CreerParticipant	0	0	0	0	0	0	0	0	1

Figure 24: Extrait de la matrice DF

	Definir Article	Enlever Action DuRole	Definir Information Article	Definir Action	CreerArticle DeModele	Annuler Action	Modifier InfosArticle
DefinirArticle	1	0	1	0	1	0	1
EnleverActionDuRole	0	1	0	1	0	1	0
DefinirInformationArticle	1	0	1	0	1	0	1
Definir Action	0	1	0	1	0	1	0
CreerArticleDeModele	1	0	1	0	1	0	1
AnnulerAction	0	1	0	1	0	1	0
ModifierInfosArticle	1	0	1	0	1	0	1

Figure 25 : Extrait de la matrice CF

La décision finale de groupement des opérations fonctionnelles PLM en services est faite à travers la superposition des deux matrices (*i.e.* union des deux matrices CF et DF). Ceci nous a permis de définir le catalogue de services fonctionnels PLM. Il comprend 111 opérations de service regroupées dans neuf services fonctionnels PLM (cf. Figure 26) (cf. Annexe A pour version la en anglais des services fonctionnels PLM).

Document	Produit	Nomenclature	Article
DefinirNouveauDocument	DefinirNouveauProduit	LierArticleProduit	DefinirArticle
DefinirInfosDocument	DefinirInfosProduit	LierDocumentProduit	DefinirInformationArticle
AjouterAttachement	CreerProduitDeModele	LierDocumentArticle	CreateArticleDeModele
CreerDocumentDeModele	ModifierInfosProduit	ComparerStructuresProduits	ModifierInfosArticle
EditerAttachement	NouvelleVersionProduit	DefinirQuantiteLien	NouvelleVersionArticle
ModifierInfosDocument	ChangerstatutProduit	ModifierQuantiteLien	NouvelleRevisionArticle
ExporterDocumentCommeModele	NouvelleRevisionProduit	AfficherStructureArticle	ModifierStatutArticle
AfficherInfosDocument	GenererDocumentationProduit	AfficherStructureProduit	GenererDocumentationArticle
NouvelleVersionDuDocument	ExporterProduitCommeModele	AfficherCasD'emploiArticle	ExporterArticleCommeModele
ChangerStatutDocument	ChercherProduit	AfficherCasD'emploiDocument	ChercherArticle
NouvelleRevisionDocument	AfficherInfosProduit	DetruireLienArticleProduit	AfficherInfosArticle
AfficherVersionDocument	AfficherAttachmentProduit	DetruireLienDocumentProduit	AfficherAttachmentArticle
AfficherHistoriqueDocument	VoirEditeurProduit	DetruireLienDocumentArticle	VoirEditeurArticle
AfficherInfosAttachement	VoirHistoriqueProduit	CopierStructureArticleCommeModele	VoirHistoriqueArticle
OuvrirAttachement	DetruireProduit	CopierStructureProduitCommeModele	DetruireArticle
SupprimerAttachement	AfficherVersionProduit	CopierStructureProduitCommeModele	AfficherVersionArticle
DownloadAttachement	CheckInProduit		CheckInArticle
UploadAttachement	CheckOutProduit		CheckOutArticle
AjouterSignatureElectronique		Processus et Etape	
CheckOutDocument		AfficherEtapesTerminer	
CheckInDocument	Participant	AfficherEtapeCourante	
ChercherDocument	CreerParticipant	AfficherStatutEtapeCourante	Indicateur
ImprimerAttachement	LierParticipantAuRole	AfficherRoleParticipantEtape	AfficherTempsExecutionDuProcessus
ConvertirAttachmentPDF	RajouterInfosParticipant	AffecterParticipantAEtape	ChercherTempsEtapeProcessus
ExporterDocumentCommeModele	EnleverRoleDeParticipant	ModifierParticipantEtape	ComparisonInstanceEtape
AfficherDocumentEditeur	EnleverUnParticipant	NotifierParticipantParE-Mail	AfficherNombreParticipantProcessus
DetruireDocument	ModifierInformationParticipant	AfficherNotificationSurBureau	
	AfficherInformationParticipant	SupprimerParticipantEtape	Role et action
		ChercherParticipantAEtapeStatut	CreerRole
	CAO	LierDocumentProcessus	DetruireRole
	RecupererStructureAssemblage	EnvoyerPopUpPourNotification	LierActionsAuRole
	RecupererQuantiteD'Article		EnleverActionDuRole
	VisualiserModeleCAO		Definir Action
			AnnulerAction

Figure 26 : Catalogue de services fonctionnels PLM

3.3. Catalogue de services logiciels PLM

Un service logiciel PLM expose un ensemble d'opérations qui devront être implémentées par les systèmes PLM. Les opérations de ces services représentent une implémentation particulière des fonctions offertes par un système PLM particulier. Le catalogue du niveau technique est donc propre à l'éditeur du système PLM. Il s'en sert pour implémenter les services fonctionnels PLM et par conséquent il permet de mettre en œuvre les processus de conception de produits manufacturés.

Ainsi, contrairement au deux sections précédentes, l'objet de cette section n'est pas l'identification d'un catalogue de services logiciels PLM mais plutôt de donner une impression/idée sur le genre d'opérations qu'on peut trouver dans un catalogue de services logiciels PLM. Cela permettra de guider les éditeurs lors de la définition de leur catalogue de services logiciels PLM et d'éviter les compromis concernant la différence entre les opérations du niveau fonctionnel et ceux du niveau technique.

Par ailleurs, des standards pour les services PLM existent déjà dans la littérature. Ces standards été proposer dans le but de faciliter la communication entre plusieurs systèmes PLM. Nous avons identifié deux standards du domaine ; PLM 2.0 Services [OMG 08] et le PLCS web Services [ISO 05] qui ont

été proposés pour répondre au problème d'intégration en ligne des systèmes PLM hétérogènes afin de mettre en œuvre leurs processus collaboratifs. Les services proposés par ces deux standards sont d'un niveau technique. Ils constituent des tentatives de catalogues de services logiciels PLM. Certains systèmes PLM sont déjà basés sur ces deux standards PLM, nous proposons donc de les analyser et de dégager à partir de ces deux standards les catégories ou genres de services logiciels PLM qui regroupent des opérations logiciels pouvant être implémentées dans un système PLM.

PLCS (Product Life Cycle Support) est un ISO STEP standard (ISO 10303-239) [PLCS 12] qui propose des services permettant la création et la gestion de données de produit selon le modèle de donnée proposé par ce standard. Les services de PLCS sont organisés en deux catégories de modules. Un module fonctionnel et un module non fonctionnel. Le module non fonctionnel propose des services permettant la gestion de la connexion avec le système PLM. Il propose aussi un service permettant la gestion des requêtes envoyées au système (pour la récupération des résultats et leur conversion selon le schéma de donnée proposé par le standard ISO 10303-239). Le module fonctionnel est composé de sept services chacun contenant un ensemble d'opérations de services permettant la manipulation de données exprimées dans le modèle de donnée proposé par ISO 10303-239. Ce modèle de donnée est énorme et générique afin qu'il puisse être adapté aux données de tout système PLM.

OMG PLM Services [OMG 08] est un standard de l'OMG (Object Management Group) qui fournit des services permettant l'accès aux systèmes PLM via les interfaces des services qu'il propose. Il permet ainsi la connexion à un système PLM donné puis la lecture et l'écriture de données de produit dans ce système. Le standard propose aussi des connecteurs qui permettent la traduction des requêtes (i.e. services) selon le schéma de donnée du système PLM interrogé (cf. Figure 27). Les services proposés par ce standard se résument en un service de gestion de la connexion avec le système PLM interrogé et un service qui permet l'écriture des messages à envoyé et leurs envois sous forme de requêtes. Les opérations principales du service permettant la manipulation des données sont : le service *Query* qui prend en paramètre un message décrivant la donnée à chercher. Le service *Write* qui prend en paramètre le message décrivant le type de la donnée à sauvegarder et ces attributs. Enfin le service *import_data* qui prend en paramètre une URL et permet de télécharger les documents décrivant les données PLM.

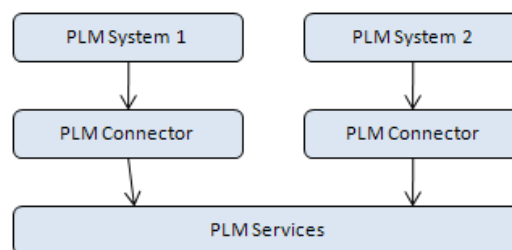


Figure 27: Interrogation de système PLM en utilisant PLM services

Suite à l'analyse de ces deux standards nous constatons qu'ils proposent un ensemble de services complémentaires et que certains services tels que ceux permettant la création d'objets PLM ou encore ceux qui permettent leurs gestion (i.e. modification et suppression) sont définis dans les deux standards étudiés. Cette catégorie de services est à notre avis nécessaire et doit nécessairement faire partie du catalogue de services logiciels PLM. Nous remarquons aussi que ces deux standards définissent des services qui permettent l'établissement et la gestion de la connexion entre deux systèmes PLM. Nous soulignons que cette catégorie de services ne nous intéresse pas et ne répond à aucun objectif de notre travail. Toutefois, nous remarquons l'absence d'une catégorie de services logiciels PLM que nous jugeons nécessaire pour notre travail. Il s'agit d'une catégorie de services qui permet la mise en œuvre des services fonctionnels PLM nécessitant un lien avec d'autres applications tels que les logiciels CAO, la suite Microsoft Office, etc. Ainsi, nous distinguons deux catégories de services pour le catalogue de services logiciels PLM.

Une première catégorie qui regroupe des services permettant la gestion des données techniques PLM tels que la création, consultation, modification, suppression et recherche selon certains critères. Se sont les services qui se focalisent sur les objets clés du système d'information de l'entreprise. Il s'agit

essentiellement des services qui permettent l'accès aux informations relatives aux objets stockés dans la base de données du système. Typiquement, il s'agit des services qui réalisent des opérations de création, lecture, écriture et suppression. Cette catégorie de services logiciels PLM permettra l'implémentation réelle des opérations de services fonctionnels PLM.

La deuxième catégorie de services regroupe des services transversaux liés à la mise en œuvre des services fonctionnelles PLM nécessitant un lien avec d'autres applications. Ces services permettent ainsi d'encapsuler le patrimoine applicatif. L'objectif des services est de mettre en place des façades homogènes au-dessus des applications informatiques existantes. À titre d'exemple, nous pouvons considérer des services qui encapsulent l'application informatique permettant la connexion à la messagerie électronique, l'accès aux outils CAO, DAO, de calcul, etc. Cette deuxième catégorie de services logiciels PLM permettra de soutenir l'exécution des opérations des services fonctionnels PLM.

4. Conclusion

Ce chapitre a présenté la démarche proposée pour l'identification des services nécessaires pour notre démarche de flexibilisation des processus de conception de produits manufacturés. La démarche d'identification proposée comporte deux phases : (1) une phase d'identification des services du niveau métier et (2) une phase d'identification des services du niveau fonctionnel. En effet, il ressort de l'analyse bibliographique des approches d'identification de services que la démarche d'identification ne peut être la même pour les services du niveau métier et ceux du niveau fonctionnel. Les services du niveau métier ne peuvent être identifiés que selon une approche descendante. Quant aux services du niveau fonctionnel, il est conseillé d'utiliser une approche ascendante pour leur identification.

La démarche proposée comble certaines limites des démarches d'identification étudiées et répond à certains objectifs de notre problématique concernant la proposition de traitement générique et réutilisable sous forme de service ainsi que la facilité de prise en main de la méthode par les personnes métier du domaine de conception. Précisément, notre démarche a proposé un enrichissement des techniques d'identifications de services à travers des techniques propres aux personnes du domaine métier tels que l'analyse des modèles de données pour l'identification des services de bas niveau et l'analyse des actions des modèles de processus pour l'identification des services de haut niveau. De même, concernant les critères de groupement, notre démarche a proposé un enrichissement en proposant des critères propres au domaine métier tel que l'utilisation des règles métier et le raffinement du groupement par décomposition du domaine en corps de métier. Ceci facilitera la recherche de services par des personnes métier. Enfin, notre démarche prend en compte l'aspect humain et propose une description, des services découverts, compréhensible par les personnes métier. En effet, nous avons défini les opérations du niveau métier en terme de concepts du domaine à savoir les objets métier et les actions métier du domaine de la conception de produits manufacturés. Cette approche d'identification est caractérisée par la facilité de prise en main par les personnes métier et donc elle peut être réutilisée pour enrichir les catalogues proposés. Le niveau de détail des étapes de cette approche lui permet d'être reproduite à d'autres domaines.

Le référentiel de services obtenu à l'issue de cette démarche d'identification organise les services en deux catalogues de services. Un catalogue de services de conception de produits manufacturés et un catalogue de services fonctionnels PLM. Ce chapitre a présenté ainsi une démarche pour la réutilisation (i.e. identification des services réutilisables). Il reste à définir une démarche permettant l'utilisation des services identifiés afin de définir des processus de conception de produits manufacturés, tenant compte ainsi de la spécificité de ces processus et les exigences des personnes métier. Cette démarche par réutilisation (i.e. orchestration et alignement des services) fera l'objet du chapitre suivant.

Chapitre IV :

Usage des services: définition du processus par composition de services et alignement entre les orchestrations métier, fonctionnelle et technique

1. Introduction

La démarche d'identification des services nécessaires pour la mise en œuvre de notre approche, que nous avons proposée dans le chapitre III, donne lieu à un ensemble de services métier (SCP), fonctionnels PLM et logiciels PLM [Hachani 12b]. L'objet de ce chapitre est de proposer un cadre permettant l'utilisation des services identifiés. Ce cadre doit permettre (i) la définition des processus de conception de produits manufacturés et (ii) leur mise en œuvre.

Ce chapitre est organisé en plusieurs sections. La prochaine section (§2) donne un aperçu du cadre méthodologique permettant l'utilisation des services identifiés. Elle introduit les deux aspects d'usage des services, à savoir la définition des processus par composition des services du niveau métier et la mise en œuvre des processus ainsi définis par alignement entre des orchestrations définies aux différents niveaux (métier, fonctionnel et logiciel). Les sections suivantes détaillent ces aspects. La section 3 focalise sur la définition de processus par composition des services du niveau métier. Elle présente d'abord nos critères pour le choix de l'approche de composition de services au regard des besoins spécifiques au domaine de la conception de produits manufacturés (§3.1) et décrit ensuite les concepts nécessaires pour la composition des services du niveau métier sous forme de méta-modèle (§3.2). Les sections 4 et 5 se focalisent sur la mise en œuvre du processus de conception défini au niveau métier dans les deux niveaux fonctionnel et technique de notre approche. Dans la section 4, nous définissons les méta-modèles du niveau métier, fonctionnel PLM et technique PLM formalisant les concepts qui permettent les orchestrations au niveau métier, fonctionnel PLM et technique PLM. Dans la section 5, nous nous focalisons sur l'alignement entre les différentes orchestrations. Nous présentons les relations entre les différents modèles d'orchestration que nous proposons et la démarche permettant le passage de l'orchestration du niveau métier vers l'orchestration du niveau fonctionnel PLM et puis vers l'orchestration du niveau technique PLM. Une synthèse qui présente les principales contributions de notre approche par rapport aux objectifs fixés clora ce chapitre (§6).

2. Cadre pour la définition et la mise en œuvre des processus de conception de produits manufacturés

Comme nous l'avons annoncé dans la section 3.2 du chapitre II, nous souhaitons partir d'une spécification métier du processus de conception de produits manufacturés, sous forme d'orchestration d'opérations de SCP, pour aboutir à un modèle de processus sous forme d'orchestration de services logiciels PLM.

Pour ce faire, notre approche s'appuie sur les principes de l'IDM (Ingénierie Dirigée par les Modèles). Comme expliqué dans la section §4.3 du chapitre II, l'IDM propose la notion de modèle comme artefact principal durant tout le cycle de vie du système (conception, spécification, implémentation, maintenance/évolution, etc.). Il propose l'utilisation systématique de méta-modèles, de modèles et de transformations suffisamment précises et formelles, pour être interprétées par des machines [Combemale 09]. La Figure 28 illustre les liens qui existent entre le système réel, le modèle et le méta-modèle.

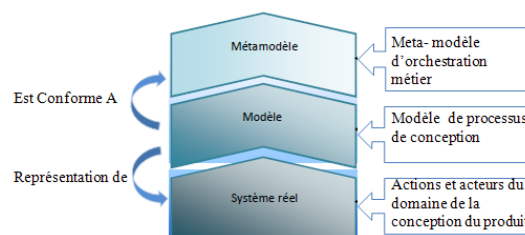


Figure 28 : Système réel, modèle et méta-modèle

Si l'on considère le niveau métier de notre approche (i.e. domaine de la conception), le système réel correspond au déroulement habituel des actions de conception et les acteurs qui réalisent ces actions métiers. Le niveau modèle correspond à la représentation du système réel sous forme de modèles de processus de conception. Ce modèle doit être conforme à un méta-modèle qui définit les concepts permettant de représenter le modèle [Bézivin 04] qui est le méta-modèle d'orchestration métier.

Les transformations représentent l'ensemble des étapes partiellement ordonnées permettant l'obtention d'un modèle à partir d'un premier ensemble de modèles. La transformation prend des modèles en entrée et génère un modèle en sortie. Ainsi, pour chaque modèle de processus de conception (m_{PC}) conforme à son méta-modèle (MM_{PC}), il existe un unique modèle fonctionnel (m_F) support, résultat de la transformation, et qui est conforme à son méta-modèle fonctionnel (MM_F).

L'intérêt d'une telle approche basée sur les modèles est de pouvoir représenter les activités du domaine de conception durant tout le cycle de vie du processus (i.e. de sa modélisation jusqu'à son implémentation). Ceci facilitera la compréhension du processus et le suivi de son déroulement par les acteurs métier du domaine de conception. En effet, les personnes métier auront une vision sur le processus métier (modèle du niveau métier) mais aussi une vision sur comment le processus est implémenté (à travers le modèle du niveau fonctionnel) sans besoin d'avoir des connaissances techniques. De plus, l'utilisation des transformations entre modèles augmente l'agilité du déploiement du processus modélisé sur un système support. En effet, comme les règles de transformations sont toujours définies à l'aide des concepts figurants dans les méta-modèles source et destination [Dolques 09], la transformation de modèles est générique et ne dépend pas d'un modèle de processus particulier. Ceci étant, à chaque déploiement, l'exécution des transformations permet de générer automatiquement le modèle d'implémentation spécifique au processus à déployer.

Suivant une démarche IDM, nous proposons de définir les concepts spécifiques à chaque niveau de notre approche (métier, fonctionnel, technique) dans un méta-modèle qui représente le langage d'expression qui sera utilisé pour tous les modèles du niveau en question.

Le premier niveau de notre approche étant le niveau métier. Il contient la spécification (i.e. modèle) du processus de conception de produits manufacturés telle qu'elle est définie par les acteurs métier du domaine de la conception. Cette spécification métier est obtenue à l'aide de l'orchestration d'opérations de services métier du domaine de la conception (cf. OM en Figure 29). Afin que les acteurs métier du domaine de la conception puissent définir cette spécification et procéder à l'orchestration métier, nous proposons un Meta-modèle d'orchestration métier regroupant un ensemble de concepts et de formalismes nécessaires à cette définition. Cette première étape de définition de processus (modélisation) constitue la première phase du cadre méthodologique que nous proposons pour le déploiement agile des processus de conception de produits manufacturés (cf. phase 1 dans la Figure 30).

Une fois défini, ce modèle métier (sous forme d'orchestration d'opérations de SCP) doit pouvoir être exécuté. Comme discuté dans le chapitre II, ce modèle métier est supporté par des opérations de services fonctionnels PLM qui seront mises en œuvre, à leur tour, par appel aux opérations de services logiciels PLM (cf. OF en Figure 29). Les services logiciels PLM sont ceux qui implémentent totalement ou partiellement les services fonctionnels PLM dans un système PLM particulier (cf. OT en Figure 29).

De même que pour le niveau métier, nous proposons de définir les concepts et méta-modèles d'orchestration associés aux niveaux fonctionnel et technique de l'approche proposée.

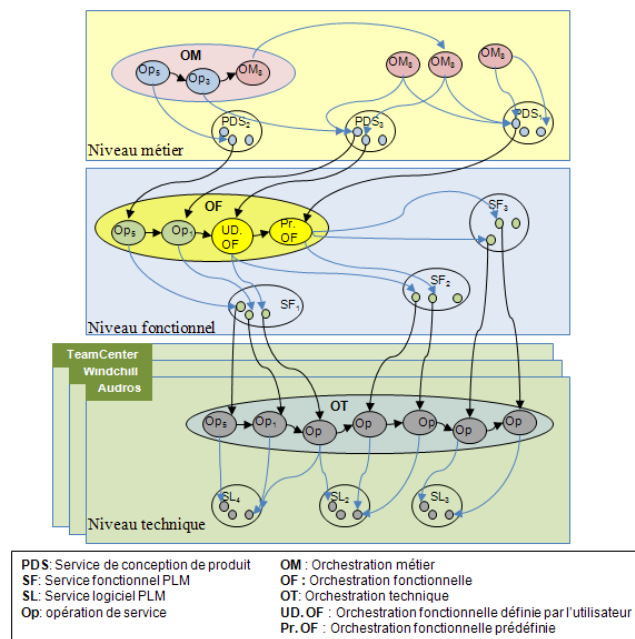


Figure 29: Aperçue de l'approche proposée

La mise en œuvre du processus ainsi défini consiste en la transformation du modèle d'orchestration du niveau métier en un modèle d'orchestration du niveau fonctionnel qui joue le rôle de relais entre le domaine métier et le système PLM utilisé (i.e. entre l'orchestration métier et l'orchestration technique). Ce passage du niveau métier au niveau fonctionnel constitue la phase de déploiement métier (cf. phase 2 dans la Figure 30) et revient à un choix des opérations fonctionnelles PLM qui permettent la mise en œuvre de chaque opération métier. Enfin, pour une exécution du processus à l'aide d'un système PLM donné, la transformation du modèle d'orchestration du niveau fonctionnel en un modèle d'orchestration du niveau technique est nécessaire. Ce passage du niveau fonctionnel au niveau technique constitue la phase de déploiement fonctionnel (cf. phase 3 dans la Figure 30) et consiste à projeter les opérations fonctionnelles PLM sur les opérations de services logiciels PLM associées au système PLM utilisé. Notons aussi que comme nous pouvons le voir à travers la Figure 30, une orchestration du niveau métier peut être basée sur une autre orchestration métier (fragments d'orchestration). De même, une orchestration fonctionnelle peut faire appel à des orchestrations fonctionnelles prédéfinies (cf. Pr.OF dans la Figure 29) ou définies par l'utilisateur (cf. UD.OF dans la Figure 29)

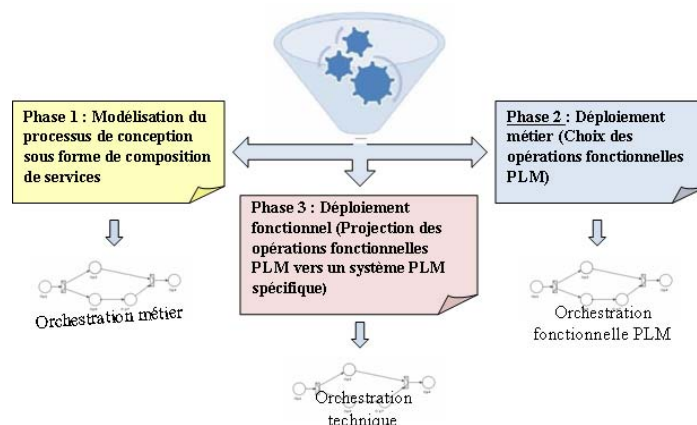


Figure 30 : Phases du cadre méthodologique pour la définition et mise en œuvre des processus de conception de produits manufacturés

Les deux phases de déploiement métier et fonctionnel nécessitent la mise en place d'un cadre méthodologique permettant d'assurer un continuum de transformations et un alignement entre les modèles élaborés aux différents niveaux. Ce cadre doit être facilement exploitable par les personnes métier du domaine de la conception de produits manufacturés. Pour cela, nous proposons de définir d'abord les relations qui existent entre les concepts nécessaires pour la description à la définition métier du processus de conception (i.e. orchestration Métier) et ceux qui permettent l'expression du processus en termes de services fonctionnels PLM (i.e. orchestration fonctionnelle), puis de services logiciels PLM (i.e. orchestration technique). Ensuite, nous définissons les mécanismes nécessaires pour la transformation entre les modèles des différents niveaux de notre approche.

Dans la section suivante (§3), nous présentons la démarche permettant la définition des processus de conception sous forme d'orchestration de services du niveau métier. Cette démarche de composition doit satisfaire certains critères des processus de conception (cf. §2.3 du chapitre I) ainsi que les objectifs de notre thèse (cf. §4.1 du chapitre I. Nous allons décrire les concepts nécessaires pour la composition des services du niveau métier sous forme de méta-modèle.

3. Définition du processus de conception par composition de services

La définition du processus de conception de produits manufacturés consiste à (cf. Figure 31):

- découvrir, à partir du catalogue de *SCP*, l'ensemble des services métier et les opérations de services qui répondent aux différents besoins du processus à modéliser. En effet, les opérations de services du niveau métier permettent d'exprimer les objectifs poursuivis par le processus. Des sous processus déjà définis et exposés sous forme d'orchestrations à part entière (i.e. fragments d'orchestration) peuvent aussi participer à la définition du processus final.
- composer les fragments d'orchestration métier et/ou les opérations de services métier choisis, en spécifiant l'ensemble des flux de contrôles qui les relient. Les flux de contrôles permettent d'exprimer l'enchaînement entre les différentes opérations de *SCP* choisies.

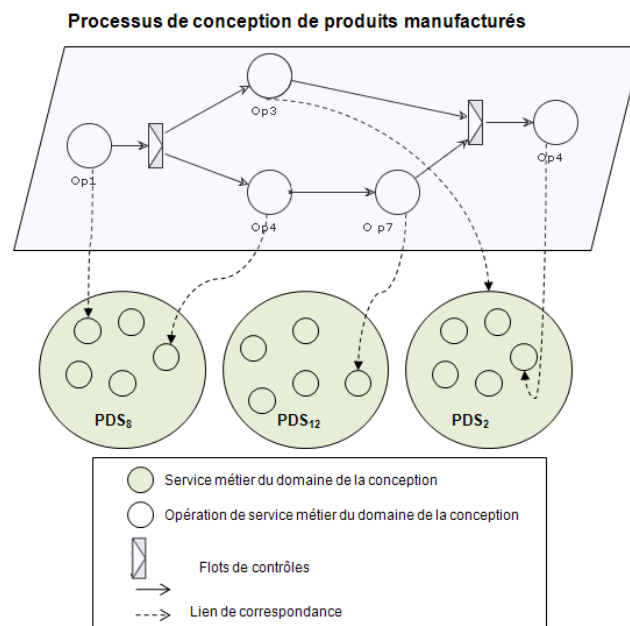


Figure 31 : Processus de conception : Enchaînement entre les opérations des SCP

Le processus de conception de produits manufacturés est ainsi perçu comme une composition des orchestrations et opérations de services métier du domaine de la conception (SCP). Le concept de service métier (SCP), largement étudié dans le chapitre précédent, constitue ainsi la brique de base de la démarche de définition du processus de conception. Toutefois, les opérations de SCP ne suffisent pas à eux seules à définir un processus de conception. Nous examinons dans la suite les autres éléments de modélisation qui peuvent participer à la définition du processus de conception et les caractéristiques que doit satisfaire l'approche de composition tenant compte des spécificités du domaine de la conception de produits.

3.1. Caractéristiques de notre approche de composition de services

Comme nous l'avons défini au chapitre II, l'orchestration de services permet d'assembler des services afin d'atteindre un objectif particulier, par l'intermédiaire de primitives de contrôles (boucles, test, traitement d'exception, etc.) [Kellert 06] ou flot de contrôle. Nous proposons dans ce chapitre une approche permettant la définition des processus de conception de produits manufacturés sous forme d'orchestration de services.

Une question primordiale se pose : quel type d'approche de composition serait le plus adapté pour représenter les processus de conception de produits manufacturés ? Pour répondre à cette question, il est nécessaire de fixer nos choix sur les caractéristiques de l'approche de composition que nous souhaitons développer.

En effet, différentes approches de composition de services ont vu le jour. Chacune de ces approches s'intéresse à un aspect particulier de la composition. Les auteurs [Casati 01] [Foster 03] distinguent plusieurs types de composition, que nous pouvons caractériser selon trois critères : le niveau d'abstraction de la démarche, le type de découverte des services et le degré d'automatisation de la composition.

- **Le type de découverte des éléments de la composition (dynamique, statique)**

[Casati 01] considère que la composition des services peut être qualifiée de statique ou dynamique. La composition statique passe par une phase de spécification durant laquelle les services sont identifiés, interconnectés et déployés pour être utilisés. Quant à la composition dynamique, elle implique la capacité de sélectionner, composer et faire inter-opérer des services à partir d'une spécification de haut niveau des objectifs. Différentes approches ont été proposées dans la littérature pour réaliser la découverte dynamique de services [Medjahed 05], [Benatallah 03] [Ben Halima 08].

Dans la plupart des cas, la découverte dynamique est requise dans un environnement où les services sont accessibles via Internet pour remédier au problème de disponibilité des services. Elle permet de choisir et de remplacer plus facilement un service par un autre équivalent. Dans le cadre de nos travaux, la question de la sélection dynamique de service n'a pas été abordée et est complémentaire de ce que nous proposons. Nous considérons pour l'instant que les services que nous avons identifiés dans le chapitre III seront déployés dans le système PLM et disponibles à tout moment. Ainsi, la découverte des services participant à la composition restera centrée utilisateur dans le sens où ce dernier identifie et choisit les services dont il a besoin.

- **Le degré d'automatisation de la démarche (manuelle, automatique)**

Certains auteurs comme [Foster 03] distinguent deux types de composition ; la composition manuelle et la composition automatique. La composition manuelle suppose que l'utilisateur génère la composition à la main par exemple via un éditeur graphique. Quant à la composition automatique, elle prend en charge tout le processus de composition et le réalise automatiquement, sans qu'aucune intervention de l'utilisateur ne soit requise. Nous soulignons que la composition automatique peut aller

de l'automatisation de la conception jusqu'à l'automatisation de l'exécution de la composition. Nous parlons ici de l'automatisation de la conception dans laquelle, en fonction d'un objectif, la sélection automatique de services vise à automatiser la conception et générer un schéma de composition. Ces dernières années, de nombreux travaux ont porté sur l'automatisation de la composition des services [Milanovic 04] [Bourdon 07]. Le travail de [Benatallah 03] adopte aussi l'approche automatique de la composition. Partant d'un objectif à atteindre, son approche permet de déterminer le schéma de composition ainsi que les services qui le composent.

La volonté d'automatiser complètement la composition n'est pas une voie réaliste dans notre cas. En effet, dans le cadre de notre travail, de nombreuses tâches doivent rester à la charge des acteurs métier. Nous considérons, dans nos travaux, qu'il est important que les acteurs métier construisent eux même leurs processus par composition des services par exemple via un éditeur graphique. Ceci est justifié par le fait que chaque entreprise possède ses propres modes de travail et donc elle utilise des services métier bien déterminés. De plus, selon le contexte, un processus peut être modélisé d'une façon ou d'une autre (en utilisant des enchaînements différents de services). La composition automatique ne constitue pas donc une caractéristique essentielle de notre approche de composition de services.

- **Le niveau d'abstraction de la démarche de composition (conceptuel, technique)**

Ce critère précise si la démarche de composition se préoccupe de définir des mécanismes pour l'exécution de la composition de services ou plutôt pour la définition conceptuelle de la composition de services. Peu sont les travaux qui adoptent une vision conceptuelle de la composition [Boukadi 09a] [Castro 07]. La plupart des travaux sur la composition adoptent une vision technique et se préoccupent essentiellement de définir des mécanismes pour l'exécution des services au niveau de l'implémentation de la composition et non pour la définition conceptuelle de la composition [Medjahed 05] [Pourraz 08] [Verjus 11].

Nous considérons que la composition des services est loin d'être un problème purement technique (i.e. au niveau de l'implémentation), surtout dans un domaine où ceux qui définissent la composition sont des acteurs dont le métier est loin du génie logiciel. Certes, les mécanismes techniques pour la composition sont fondamentaux, mais la composition doit être accompagnée par une démarche conceptuelle (i.e. comment définir la composition au niveau conceptuel). Ainsi, nous considérons que la vision conceptuelle est une caractéristique essentielle de notre approche de composition de services. Nous voulons fournir une démarche simple permettant de décrire les processus de conception sous forme de composition de services. Quoique, nous envisageons de proposer des transformations du modèle de composition de services du niveau métier vers des langages d'orchestration permettant de mettre en œuvre des compositions évolutives tels que PXL [Verjus 11] et π -Diapason [Pourraz 08] mais aussi dans des plateformes spécifiques ou propriétaires.

Synthèse sur les caractéristiques de l'approche de composition

Fort de ces éléments de classification et au regard des spécificités du domaine de la conception de produits manufacturés, nous synthétisons dans ce qui suit les caractéristiques de l'approche de composition de services que nous ciblons :

- nous nous focalisons sur une vision conceptuelle de la composition de services. Nous proposons de fournir une démarche simple permettant de décrire des processus métier en utilisant des concepts peu nombreux, bien définis, descriptifs, intelligibles. La description du processus doit être exprimée en termes de concepts métier du domaine, ce qui fait qu'elle est générique et indépendante de toute plateforme ou système d'exécution spécifique. Ceci

simplifie la tâche des acteurs métier et rend le modèle de processus plus simple et compréhensible,

- toutefois, nous proposons de faire recours à des mécanismes d'orchestration qui permettent l'adaptation dynamique des processus définis au niveau conceptuel sous forme de composition de services au niveau de l'implémentation de la composition. L'automatisation de l'orchestration en utilisant des langages d'orchestration qui supportent l'évolution dynamique de la composition permet non seulement d'automatiser les processus mais aussi d'effectuer des changements au niveau structurel du processus (ajout ou suppression de services ou fragments d'orchestration) s'il y'a besoin de changement,
- nous proposons une démarche de découverte statique des services participant au processus. La sélection dynamique de service n'est pas abordée dans le cadre de nos travaux et elle est complémentaire de ce que nous proposons. Le concepteur du processus aura à parcourir le catalogue de services et choisir les services dont il a besoin. Ceci simplifie la tâche du concepteur du processus qui n'aura pas à passer par une étape de spécification de l'objectif du service recherché,
- nous favorisons dans notre travail une solution de composition manuelle de telle façon que le concepteur conservera le contrôle sur le processus. C'est à dire qu'il pourra décider quels services et quels enchaînements sont mieux adaptés à son entreprise et sa situation. De plus, il aura à manipuler un outil de modélisation et des concepts de composition simples.

3.2. Éléments de modélisation du processus de conception

Comme nous l'avons annoncé au début de cette section, l'élément central de modélisation du processus de conception de produits manufacturés est l'unité de décomposition du processus à savoir les opérations de services ou encore les sous-processus définis par composition de services et exposés sous forme de services à part entière (fragments d'orchestration). Reste à identifier les autres éléments qui participent à la définition et la mise en œuvre du processus de conception. Nous regroupons ces éléments par la suite dans un méta-modèle générique qui définit les éléments nécessaires pour la modélisation et la mise en œuvre du processus de conception de produits manufacturés. Nous commençons tout d'abord par déterminer les éléments caractérisant ce méta-modèle.

L'analyse des langages de modélisation utilisés dans les entreprises industriels, présentée dans la section 3.1 du chapitre I, a montré que le langage BPMN est, de nos jours, le plus utilisé dans les entreprises industrielles [Zur Muehlen 08], même s'il comporte un nombre important de concepts. En effet, les langages de modélisation, BPMN entre autres, se sont complexifiés au fil des années. Cette complexification est justifiée par le fait que plus le processus est finement détaillé, plus la transformation vers le modèle d'implémentation correspondant est facile. Cependant, nous considérons que cela augmente la difficulté de transformation de l'orchestration du niveau métier vers les orchestrations du niveau fonctionnel et technique. En effet, la multiplication des concepts utilisés pour la modélisation du modèle métier induit une augmentation (1) au niveau du nombre des règles de transformations nécessaires pour passer au modèle d'implémentation correspondant et (2) au niveau des délais de transformation. De plus, cette multiplication de concepts nuit à la compréhension du modèle et à la difficulté que vont avoir les acteurs métier pour réaliser des modèles conformes à ce langage de modélisation. Nous rappelons que l'un des objectifs de notre travail est de faciliter l'emploi des concepts pour le concepteur métier afin qu'il puisse modéliser des processus de conception de produit d'une manière simplifiée.

Afin de permettre une représentation plus simple du processus de conception, nous favorisons l'utilisation d'un méta-modèle qui comprend les concepts juste nécessaires et suffisants pour la représentation des processus de conception. Ce méta-modèle doit permettre tout de même la représentation de tous les besoins en termes de modélisation de processus du domaine de la conception de produits manufacturés. [Zur Muehlen 08a] a réalisé une étude portant sur 126 modèles BPMN, issus de plusieurs domaines métier. L'objectif de cette étude était de déterminer quels sont les concepts de BPMN qui sont les plus utilisés, et à quelle fréquence. Le résultat de cette étude est représenté sur la Figure 32. Il en ressort que sur les 50 concepts définissant BPMN 1.2, moins de 20% est en réalité régulièrement utilisé. Seulement cinq éléments sont utilisés à plus de 50% par les acteurs métier : l'activité de type tâche, la représentation du flux normal (séquence), les événements de début et de fin d'une séquence d'activité, la porte logique XOR (OU exclusif) et l'élément Pool (groupement par acteur).

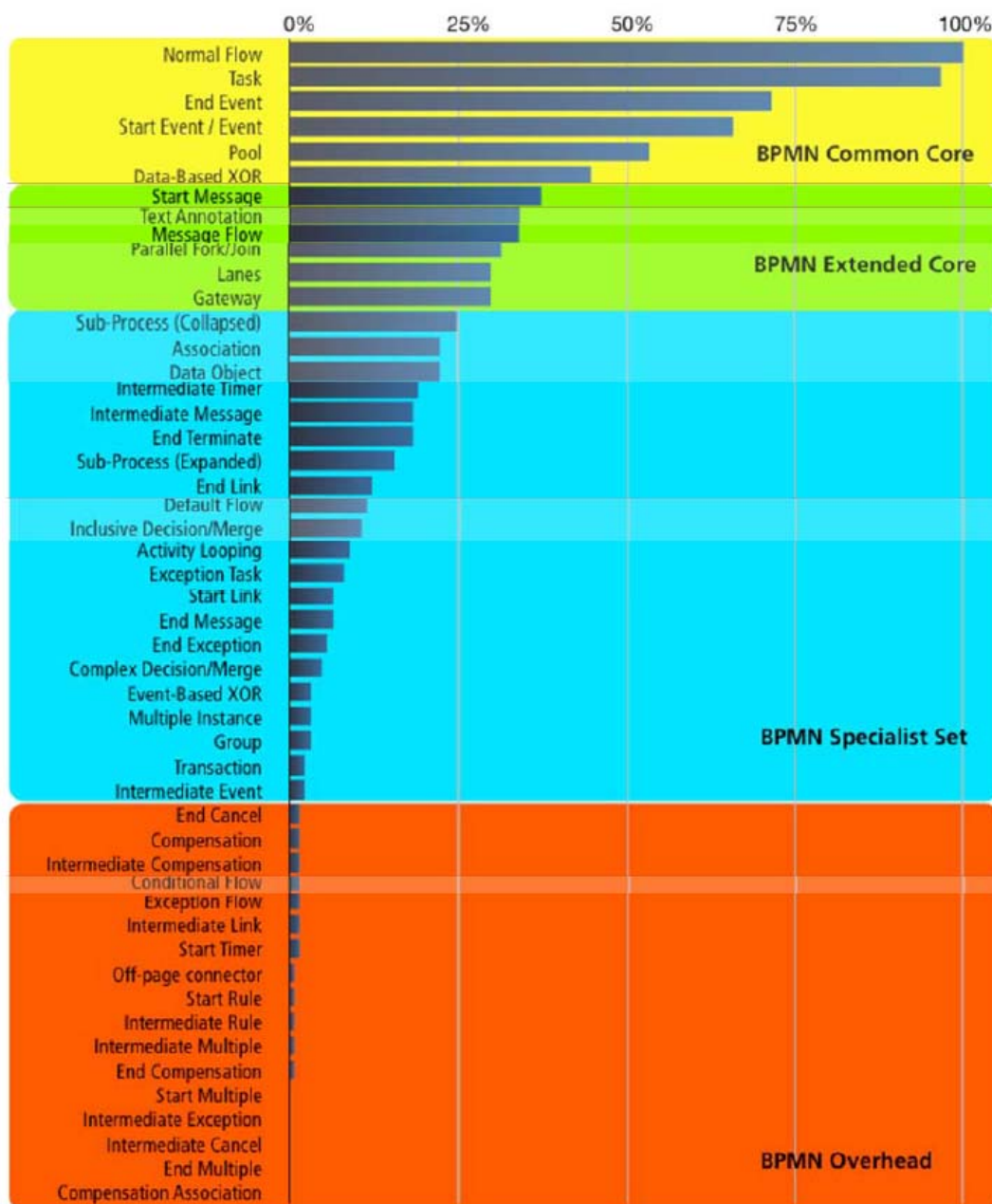


Figure 32 : Fréquence d'occurrence de concepts BPMN [Zur Muehlen 08a]

L'analyse de cette étude, montre que l'ensemble des concepts des deux groupes *BPMN Common Core* et *BPMN Extended Core*, ainsi qu'une partie des concepts du groupe *BPMN specialist Set* sont les plus utilisés en entreprise. D'ailleurs nous trouvons dans ces deux groupes les flots de contrôles basiques (Séquence, Parallel Split, Synchronization, Exclusive Choice, Simple Merge) tels que définis par Russell et Van der Aalst dans la spécification de patrons de workflow [Russell 06] [Van der Aalst 03b].

Nous proposons d'utiliser un nombre minime mais suffisant des concepts les plus utilisés en entreprise afin de présenter et mettre en œuvre les processus de conception, à savoir les concepts *Normal Flow* (la représentation du flux normal : séquence), *Parallel fork and Join*, *Data-Based XOR* (OU exclusif), *task* (l'activité de type tâche), *sub-process* (sous-processus) et *data object*. Ce choix permet d'assurer une transformation du modèle métier vers les modèles du niveau fonctionnel et technique de notre approche qui soit simple et sans ambiguïté (la réduction du nombre des concepts implique une réduction du nombre de règles de transformations). Ce nombre restreint de concepts permet néanmoins de modéliser et mettre en œuvre la plupart des processus rencontrés en industrie, entre autres les processus de conception, à condition que nous rajoutions certains concepts spécifiques à notre approche et au domaine étudié.

L'élément *tâche*, par exemple, qui représente les actions qui constituent le processus sera traduit dans notre approche par le concept *d'opération de service*. De même, l'élément *sub-process* sera traduit dans notre approche par le concept *d'orchestration de service* (i.e. *composition d'opérations de services*). De plus, le domaine de conception de produits manufacturés exige de spécifier les ressources nécessaires pour chaque étape du processus. Nous devons alors augmenter la liste des éléments choisis par les concepts qui permettent de décrire les ressources liées au domaine de conception (*ressources*).

La Figure 33 illustre les concepts que nous retenons pour notre démarche de composition à l'aide d'un diagramme de classes, décrit en UML, appelé méta-modèle d'orchestration.

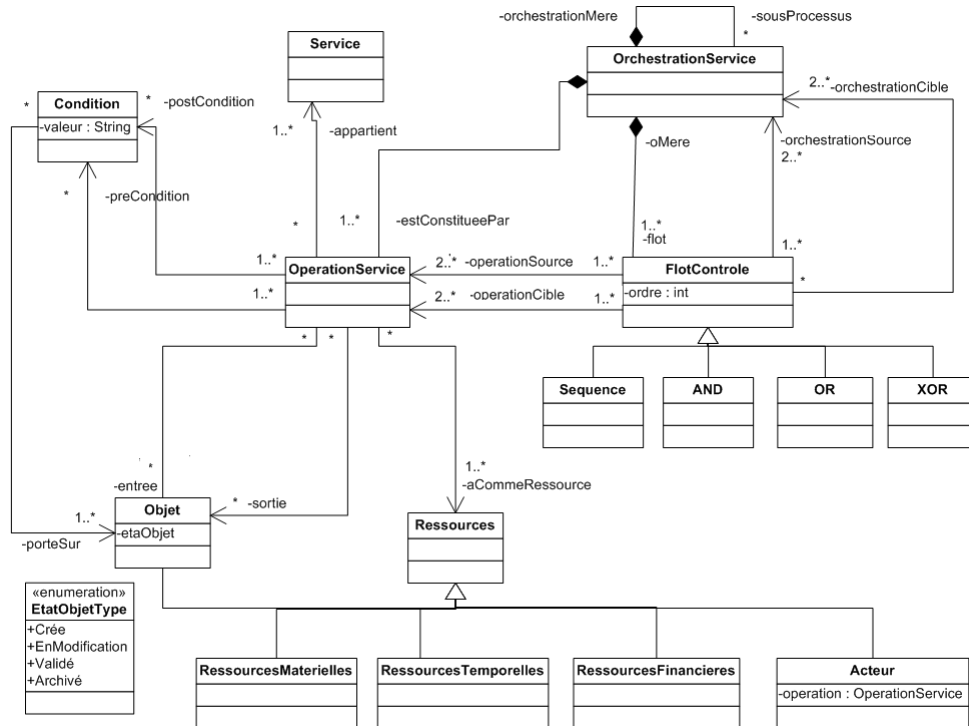


Figure 33 : Méta-modèle d'orchestration

Chacun des éléments du méta-modèle d'orchestration ci-dessus possède les attributs Désignation (i.e. correspondant au nom de l'élément) et Id (i.e. correspondant à l'identifiant de l'élément, donc unique) :

Tableau 17: Attributs communs à tous les éléments du méta-modèle d'orchestration

Nom de l'attribut	Type	Description
Designation	string	Nom de l'element
Id	string	Id de l'element (unique)

Le méta-modèle présenté ci-dessus est générique, il contient les éléments nécessaires pour exprimer une orchestration de services à tous les niveaux de notre approche (métier, fonctionnel et technique). Ce méta-modèle d'orchestration générique permet d'avoir une vision complète et concise des concepts nécessaires pour exprimer la composition de services mais aussi les relations structurelles qui peuvent exister entre ces différents concepts. Nous définissons dans ce qui suit les différents concepts de ce méta-modèle et montrons, dans la section suivante, comment ce méta-modèle générique peut être utilisé à tous les niveaux de notre approche.

L'élément central du méta-modèle d'orchestration proposé est la classe *OrchestrationService*. Ce concept englobe l'ensemble des autres éléments exprimés dans la Figure 33. En effet, le concept d'*OrchestrationService* correspond à la composition de services que nous avons proposées dans le chapitre III. Il correspond, au niveau métier de notre approche, au modèle de processus de conception de produits manufacturés exprimé sous forme de composition de services. Dans le niveau fonctionnel de notre approche, il correspond à la composition de services fonctionnels PLM permettant de mettre en œuvre les services du niveau métier. Comme nous pouvons le voir sur la Figure 33, ce concept d'orchestration peut être composé par d'autres orchestrations ou encore par des instances de la classe *OperationService*. Cela veut dire que l'orchestration peut faire appel à d'autres fragments d'orchestration déjà définis. Par exemple, au niveau métier, le processus de conception de produits manufacturés (i.e. *OrchestrationdeServices*) peut faire appel à des sous processus déjà définis. L'orchestration peut aussi être composée par des opérations de services appartenant aux services que nous avons proposés dans le chapitre III. Les relations structurelles que nous venons de définir entre les instances de la classe *OrchestrationService* et la classe *OperationService* facilitent la tâche du concepteur du processus et simplifient le modèle final. Cette proposition permettra de définir des processus (*OrchestrationService*) complexes mais non ambigus, du fait que la complexité des sous-processus est masquée par les fragments d'orchestration composant l'orchestration globale.

L'élément *Ressources* regroupe la famille de ressources que peut avoir une instance de la classe *OperationService*. En effet, afin qu'elles puissent être exécutées, les instances de la classe *OperationService* peuvent avoir besoin de certaines ressources. Par exemple, au niveau fonctionnel de notre approche, l'opération fonctionnelle PLM portant sur la conception d'un modèle 2D aura besoin d'une instance de la classe *Ressource* de type acteur qui sait faire des modèles CAO. Un autre type de ressource nécessaire pour notre approche est la classe *Objet*. Les instances de cette classe représentent les objets informationnels ou physiques manipulés ou générés par les acteurs au cours de la réalisation de l'*OpérationService*. Les instances de cette classe sont caractérisées par leurs états (Crée, enModification, validé, Archivé, etc.) qui permettent d'exprimer des pré/post conditions sur les instances de la classe *OperationService* en termes d'objets qu'ils ont en entrée et en sortie. Ces relations structurelles entre *OperationService*, *Objet* et *Condition* permettent d'apporter de la flexibilité au processus. Le chemin du processus peut être altéré en changeant les conditions sur les états des objets liés aux opérations de services participant à l'orchestration.

Enfin, l'ordre d'exécution des opérations de services et des orchestrations participant à l'orchestration composite est contrôlé à l'aide d'un ensemble de *FlotContrôle*. Il existe plusieurs sortes de flots de contrôles dans la littérature [Russell 06] [VanDerAalst 03a] où chaque type de branchement a une sémantique distincte. Nous utilisons l'essentiel de types de branchement qui suffit pour représenter les besoins du domaine de la conception de produits manufacturés. Comme nous l'avons vu précédemment, les types de branchement les plus utilisés dans l'entreprise sont les flots de type séquence, AND, OR et XOR ;

- Le flux de séquence est utilisé pour représenter l'ordre d'exécution des services participants à la composition de services. Ce flux n'a qu'une seule origine et qu'une seule destination. Il peut connecter au maximum deux opérations de services, deux orchestrations ou encore une orchestration avec une opération de service.

- Le flux de contrôle de type Parallèle (And) permet de créer et de synchroniser des flux parallèles reliant des opérations de services, des orchestrations métier ou encore des orchestrations métier avec des opérations de services. Ce flux n'a qu'une seule origine mais peut avoir plusieurs destinations. Il peut connecter au minimum trois opérations de services, trois orchestrations ou encore une combinaison d'ordre de trois d'orchestration et opération de service.

- Le flux de contrôle de type Exclusif (XOR) permet de spécifier la possibilité de choix entre plusieurs alternatives. Ce flux n'a qu'une seule origine et peut avoir plusieurs destinations alternatives. Chacune des alternatives est conditionnée par un test. Les tests conditionnant les branchements alternatifs sont évalués dans l'ordre et dès que l'un d'eux est vérifié, l'opération de service ou le fragment d'orchestration sélectionné et répondant à la condition peut être exécuté.

- Le flux de contrôle de type Inclusif (OR) permet de spécifier la possibilité de choix multiple entre plusieurs alternatives. Ce flux n'a qu'une seule origine et peut avoir plusieurs destinations. Chacune des alternatives est conditionnée par un test. Cependant, le fait qu'une alternative soit validée (i.e. condition vérifiée) n'exclue pas l'évaluation des autres alternatives.

L'utilisation de flot de contrôles permettra d'exprimer l'enchaînement des opérations de services et des fragments d'orchestration. Chaque flot de contrôle peut avoir comme source et destination des instances des deux classes *OperationsService* ou *OrchestrationsService*. Ces relations structurelles entre *OperationService*, *OrchestrationService* et *FlotContrôle* permettent d'apporter de la flexibilité au niveau structurel de l'orchestration (i.e. du processus). En effet, en modifiant les sources et destinations des flots de contrôles qui définissent la structure du processus (i.e. l'orchestration), des opérations de services ou encore des fragments d'orchestrations peuvent être remplacés par d'autres opérations de service ou fragments d'orchestration.

Nous avons ainsi présenté les éléments constituant le méta-modèle d'orchestration générique. Comme nous pouvons le voir à travers la Figure 33 et les explications qui lui suivent, l'approche de composition est générique et indépendante de toute plateforme ou système d'exécution particulier. Dans la section suivante, nous montrons comment ce méta-modèle d'orchestration générique est spécialisé à chaque niveau de notre approche.

4. Méta-modèles d'orchestration spécifiques au niveau métier, fonctionnel et technique

Nous détaillons, dans cette section, la spécialisation des concepts du méta-modèle générique et des relations entre les concepts spécifiques pour chaque niveau de notre approche. En effet, comme nous l'avons annoncé dans la deuxième section de ce chapitre, pour pouvoir être exécuté, le processus de conception défini sous forme d'orchestration de services du niveau métier (orchestration métier) doit

être supporté par une orchestration des opérations de services fonctionnels PLM (orchestration fonctionnelle). A son tour, l'orchestration fonctionnelle sera mise en œuvre via une orchestration des opérations de services logiciels PLM spécifiques au système PLM sur lequel le processus sera déployé. Dans cette section, nous proposons de définir les concepts propres à l'orchestration métier permettant la modélisation du processus de conception sous forme de composition de services. Nous, définissons de même les concepts spécifiques à chacune des orchestrations fonctionnelle et technique permettant le support du processus ainsi défini.

4.1. Méta-modèle d'orchestration métier

La Figure 34 illustre la spécialisation du méta-modèle d'orchestration générique au niveau métier de notre approche, appelé méta-modèle d'orchestration métier. Ce méta-modèle contient les concepts nécessaires pour modéliser un processus de conception de produits manufacturés [Hachani 12a]. Il est utilisé lors de la première phase du cadre méthodologique que nous proposons (cf. Figure 30), à savoir la définition du processus de conception par un acteur métier.

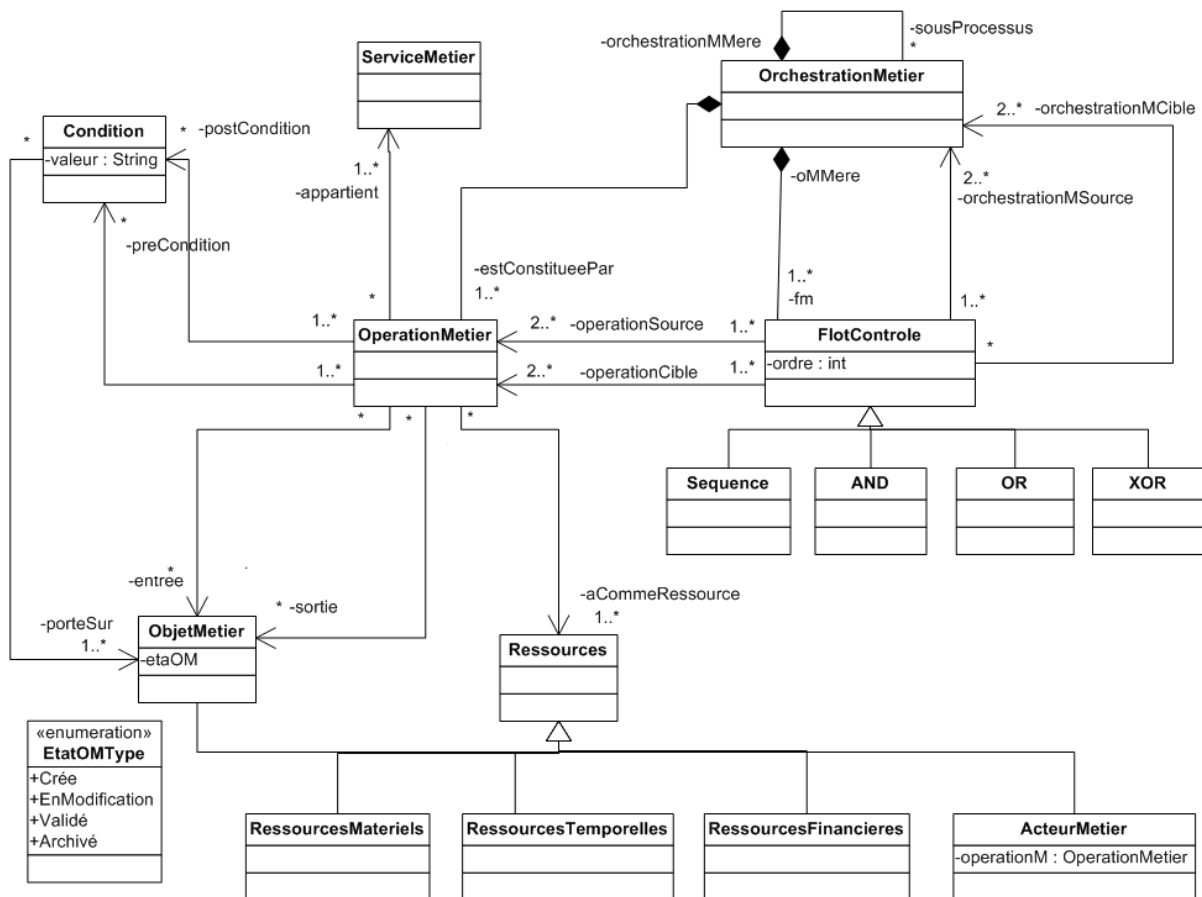


Figure 34 : Méta-modèle d'orchestration métier

Comme le montre la Figure 34, l'élément *OrchestrationMétier*, qui représente le processus de conception de produits manufacturés, est composée par une ou plusieurs opérations (*OperationMetier*) de service de conception de produit (SCP). Cela veut dire que le processus doit avoir au moins une opération métier qui le définit. Cette classe *OrchestrationMetier* peut être aussi composée par zéro ou plusieurs fragments de processus (*OrchestrationMetier*). Un processus peut en effet faire appel à des sous processus déjà définis (fragments d'orchestration) ce qui permet de simplifier en retour le modèle de processus de conception en masquant la complexité des sous-processus. Ces relations structurelles

que nous venons de définir se traduisent, dans le méta-modèle métier à travers les attributs spécifiques de l'élément *OrchestrationMetier* (cf.

Tableau 18).

Tableau 18: Attributs spécifiques de l'élément OrchestrationMetier

Attribut	Type	Description
sousProcessus	Orchestration Metier	indique les éléments <i>Orchestration Metier</i> contenus dans l'élément parent <i>Orchestration metier</i>
estConstituePar	Operation Metier	Indique les éléments <i>Operation Metier</i> contenus dans l'élément <i>Orchestration Metier</i>

Les instances de la classe *OperationMetier* représentent les opérations métier du domaine de la conception (par exemple : diffuserPlanD'ensemble, elaborerPlanDeTest). Chaque opération peut appartenir à un ou plusieurs services du domaine de la conception, identifiés dans le chapitre III de ce manuscrit (exemple ; service GestionAffaire, service GestionNomenclature, etc.). L'ordre d'exécution des opérations métier et des fragments de processus participant au processus de conception de produits manufacturés est contrôlé à l'aide du concept de *FlotControle* défini dans la section précédente. Les différentes relations qui existent entre les concepts *FlotControle*, *OperationMetier* et *OrchestrationMetier* sont traduites à l'aide des attributs spécifiques de la classe *FlotControle* (cf. Tableau 19).

Tableau 19 : Attributs spécifiques de l'élément FlotControle

Attribut	Type	Description
operationSource	Operation Metier	Indique l'élément ou les éléments <i>Operation Metier</i> source du flot de controle
operationCible	Operation Metier	Indique l'élément ou les éléments <i>Operation Metier</i> cible du flot de controle
orchestrationMSource	Orchestration Metier	Indique l'élément ou les éléments <i>Orchestration Metier</i> source du flot de controle
orchestrationMCible	Orchestration Metier	Indique l'élément ou les éléments <i>Orchestration Metier</i> cible du flot de controle

La spécificité de la spécialisation de la classe *OperationService* au niveau métier par rapport aux deux autres niveaux (fonctionnel et technique) consiste dans le fait que chaque opération métier est conditionnée par les états des objets métier qu'elle peut avoir en entrée ou en sortie. La notion d'objet métier est celle que nous avons déjà définie dans la section 3.1 du chapitre III (par exemple : CahierDesChargesClient, PlanClient, Planning, etc.). Elle représente les objets informationnels ou physiques manipulés par les acteurs métier du domaine de la conception au cours de la réalisation de leurs activités habituelles (i.e. des *OperationMetier*). Les instances de cette classe sont caractérisées par leurs états (Crée, enModification, validé, Archivé, etc.). Elles peuvent être créées par une *OperationMetier* (l'objet est produit comme sortie d'une opération métier). Des instances qui existent déjà de cette classe *ObjetMetier* peuvent être utilisées par une instance de la classe *OperationMetier*. L'opération peut ainsi les consulter ou encore les modifier (Modification de l'état de l'objet, le faire passer de l'état *enModification* à l'état *validé* par exemple). Les conditions définies sur la base des états de ces objets métier sont formalisées sous forme de pré et post conditions qui sont liées aux *OperationsMetier* (cf. Figure 34). Ces pré/post conditions régissent l'exécution de l'opération métier en question. En effet, si jamais une pré-condition n'est pas vérifiée l'opération métier ne peut être exécutée. Ce mécanisme de condition permet d'apporter des changements sur le processus sans toucher à sa structure. Il suffit de changer les conditions liées aux opérations métier ou encore de changer les états des objets sur lesquels les conditions sont définies. Cela permet d'apporter de la flexibilité au niveau structurel du processus. Enfin, l'opération métier peut être exécutée par un ou plusieurs *ActeurMetier* et peut avoir une ou plusieurs ressources (cf. Figure 34) que l'on ne trouve pas dans les deux autres niveaux de notre approche tels que les *RessourcesMaterielles*, *RessourcesTemporelle* et *RessourcesFinancieres*.

Comme nous l'avons expliqué plus haut, le modèle de processus de conception de produits manufacturés, conforme au méta-modèle d'orchestration métier que nous venons de présenter, sera défini à l'aide de termes métiers familiers aux acteurs du domaine de la conception. De plus, les

concepts nécessaires pour la définition du processus de conception ne sont pas nombreux et sont suffisamment simples et expressifs pour une personne du domaine de la conception.

La création du processus de conception se base essentiellement sur la sélection d'une ou de plusieurs opérations métier et/ou orchestrations métier et de préciser ensuite les flux de contrôles qui les relient. La composition des différentes opérations et orchestrations métier constitue le schéma du processus de conception de produits manufacturés. Nous avons choisi d'exprimer ce schéma de processus dans un langage orienté utilisateur et ceci en utilisant une notation visuelle inspirée de BPMN. L'aspect visuel permet de faciliter la tâche pour la personne métier dans le sens où elle ne sera pas amenée à connaître les détails techniques des différents services (opérations de SCP et orchestrations métier), ni à se soucier de leurs implémentations. Nous proposons l'utilisation d'un sous ensemble de concepts du langage BPMN. Comme nous l'avons dit dans la section 3.2, nous utilisons les concepts juste nécessaires et suffisants pour la représentation des processus de conception, en les simplifiant afin de permettre une interprétation plus simple du processus de conception par les acteurs métier. Nous favorisons ainsi pour les symboles de flot de contrôle une représentation textuelle plus parlante aux acteurs métier que les symboles BPMN classiques.

Les symboles graphiques associés aux concepts du méta-modèle d'orchestration métier sont illustrés dans la figure suivante.

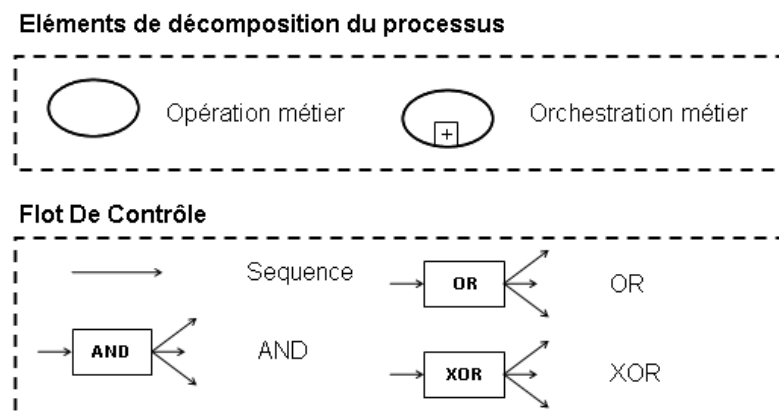


Figure 35 : Formalisme de représentation des concepts du méta-modèle d'orchestration métier

La Figure 36 présente un exemple de processus de conception construit à partir de quarts opérations de SCPs et deux orchestrations métier reliés par un ensemble de flux de séquence et d'un flot de contrôle de type XOR.

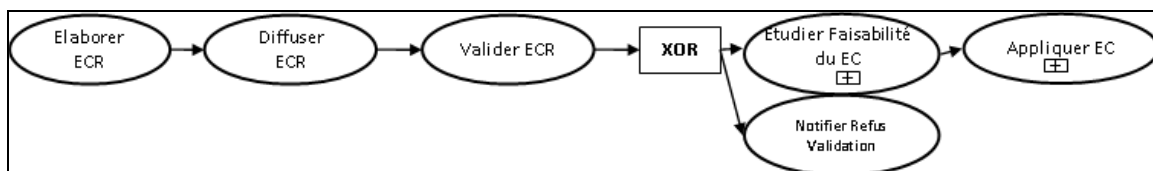


Figure 36 : Exemple de processus de conception défini selon le formalisme proposé

4.2. Méta-modèle d'orchestration de services fonctionnels PLM

Comme pour le niveau métier, nous détaillons la spécialisation du méta-modèle d'orchestration générique pour le niveau fonctionnel, appelée méta-modèle d'orchestration fonctionnelle (cf. Figure 37). Ce méta-modèle capitalise les concepts permettant la définition de l'orchestration du niveau fonctionnel PLM et décrit les éléments qui définissent le modèle de traitement au sens SI du processus de conception de produits manufacturés à l'issue de la phase du déploiement métier (cf. Figure 30). La

phase de déploiement métier correspond à l'étape où le concepteur (ou acteur métier du domaine de la conception) sélectionne les opérations fonctionnelles PLM nécessaires pour l'accomplissement des opérations métiers qui définissent le processus à l'aide d'un système PLM.

Comme le montre la Figure 37, la spécialisation du méta-modèle d'orchestration générique au niveau fonctionnel décrit comment le processus de conception de produits manufacturés est orchestré en termes de services fonctionnels PLM. Le concept d'*OrchestrationFonctionnelle* peut être assimilé à la projection de l'orchestration métier sous forme de compositions récursives d'opérations de services fonctionnels PLM (où chaque composition répond à une opération métier). Ainsi l'élément *OrchestrationFonctionnelle*, peut être composé d'une ou de plusieurs opérations (*OperationFonctionnellePLM*) de services fonctionnels PLM (*ServiceFonctionnelPLM*). Cette classe *OrchestrationFonctionnelle* peut être aussi composée de zéro ou plusieurs fragments d'orchestration fonctionnelle (*OrchestrationFonctionnelle*). Cela veut dire que l'orchestration peut faire appel à des sous-orchestrations fonctionnelles déjà définies. Ceci permet la réutilisation de traitements répétitifs et augmente l'agilité de la composition.

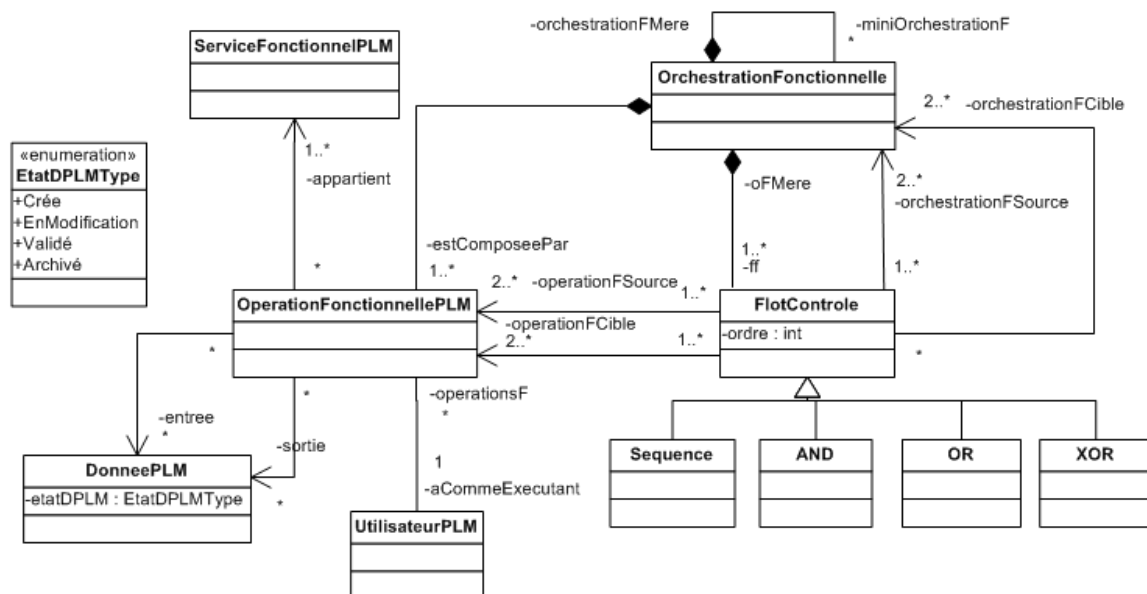


Figure 37 : Méta-modèle d'orchestration fonctionnelle

Ces relations structurelles que nous venons de définir se traduisent, dans le méta-modèle d'orchestration fonctionnelle à travers les attributs spécifiques de l'élément *OrchestrationFonctionnelle* (cf. Tableau 20).

Tableau 20: Attributs spécifiques de l'élément *OrchestrationFonctionnelle*

Attribut	Type	Description
miniOrchestrationF	OrchestrationFonctionnelle	Indique les éléments <i>OrchestrationFonctionnelle</i> contenus dans l'élément parent <i>OrchestrationFonctionnelle</i>
estComposeePar	OperationFonctionnellePLM	Indique les éléments <i>OperationFonctionnellePLM</i> contenus dans l'élément parent <i>OrchestrationFonctionnelle</i>

Les instances de la classe *OperationFonctionnellePLM* représentent les opérations des services fonctionnelles PLM (par exemple : *RecupererStructureAssemblage*, *ComparerStructureProduit*) où chaque opération peut appartenir à un ou plusieurs services fonctionnels PLM que nous avons identifiés dans le chapitre III (par exemple : *serviceNomenclature*, *service CAO*). L'ordre d'exécution des opérations fonctionnelles PLM et des fragments d'orchestrations fonctionnelles composites est contrôlé à l'aide du concept *FlotControle* défini dans la section 3 de ce chapitre. Les différentes relations qui existent entre les concepts *FlotControle*, *OperationFonctionnellePLM* et *OrchestrationFonctionnelle* sont traduites à l'aide des attributs spécifiques de la classe *FlotControle*

(cf. Tableau 21). Notons que le changement des attributs spécifiques à cette classe permet d'effectuer un changement au niveau fonctionnel du processus en remplaçant une opération fonctionnelle PLM par une autre ou un fragment d'orchestration fonctionnelle par un autre.

Tableau 21 : Attributs spécifiques de l'élément FlotControle

Attribut	Type	Description
operationFSource	OperationFonctionnellePLM	Indique l'élément ou les éléments <i>OperationFonctionnellePLM</i> source du flot de controle
operationFCible	OperationFonctionnellePLM	Indique l'élément ou les éléments <i>OperationFonctionnellePLM</i> cible du flot de controle
orchestrationFSource	OrchestrationFonctionnelle	Indique l'élément ou les éléments <i>OrchestrationFonctionnelle</i> source du flot de controle
orchestrationFCible	OrchestrationFonctionnelle	Indique l'élément ou les éléments <i>OrchestrationFonctionnelle</i> cible du flot de controle

La spécificité de la spécialisation de la classe *OperationdeService* (du métamodèle d'orchestration générique, cf. Figure 33) au niveau fonctionnel consiste au fait que chaque opération fonctionnelle PLM peut avoir besoin de certaines données PLM (cf. *DonneePLM* dans Figure 37) comme entrée et peut produire des données PLM en sortie. La notion de *DonneePLM* est celle que nous avons déjà définie dans la section 3.2 du chapitre III (par exemple : Produit, Article, Document). Elle représente les informations et données stockées dans le PLM et utilisées par les utilisateurs du système au cours de la réalisation de leurs tâches (i.e. des *OperationFocntionnellePLM*). Les instances de cette classe sont caractérisées par leurs états (Crée, enModification, Validé, Archivé, etc.). Ces données PLM peuvent être créées, modifiées ou consultées par une *OperationFonctionnellePLM*. La modification d'une donnée PLM par une *OperationFonctionnellePLM* se traduit par un changement de son état (passer de l'état *enModification* à l'état *Archivée* par exemple). Ces changements d'état peuvent être effectués par un ou plusieurs *UtilisateurPLM* lors de l'exécution de l'opération fonctionnelle PLM.

4.3. Méta-modèle d'orchestration de services logiciels PLM

Comme pour le niveau métier et fonctionnel, nous détaillons la spécialisation du métamodèle d'orchestration générique pour le niveau technique de notre approche, appelée métamodèle d'orchestration technique (cf. Figure 38). Ce métamodèle capitalise les concepts permettant la définition de l'orchestration du niveau technique et décrit les éléments qui définissent le modèle de traitement du processus de conception de produits manufacturés à l'issue de la phase du déploiement fonctionnel (cf. Figure 30). La phase de déploiement fonctionnel correspond à l'étape où le processus est déployé sur un système PLM spécifique. A ce moment, les opérations techniques PLM, nécessaires pour la mise en œuvre des opérations fonctionnelles PLM, doivent être sélectionnées et orchestrées.

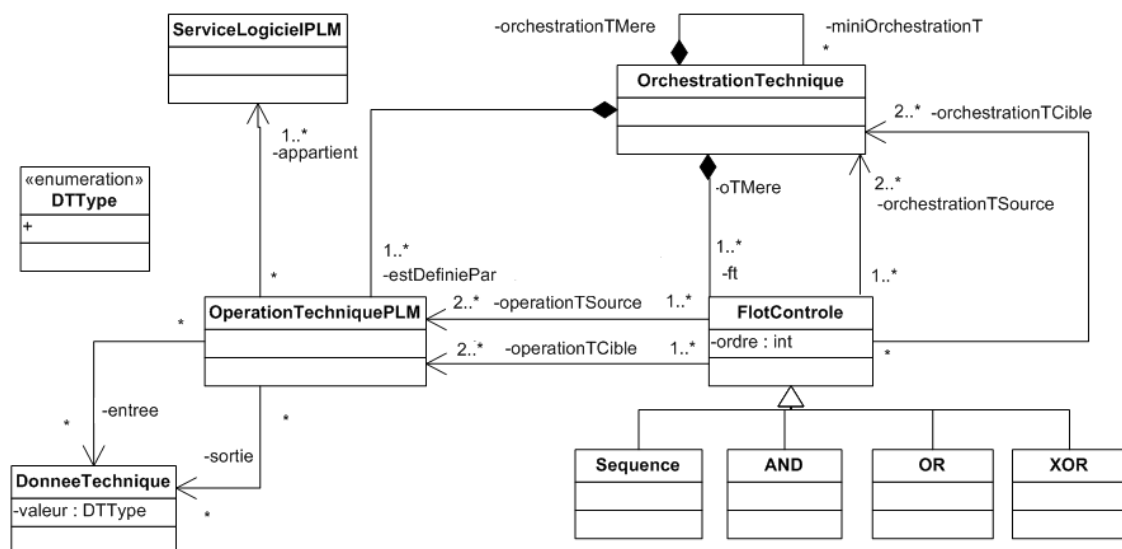


Figure 38 : Méta-modèle d'orchestration technique

Comme le montre la Figure 38, la spécialisation du méta-modèle d'orchestration générique au niveau technique de notre approche décrit comment le processus de conception de produits manufacturés est orchestré en termes de services logiciels PLM. Le concept d'*OrchestrationTechnique* peut être assimilé à la projection de l'orchestration fonctionnelle sous la forme de compositions récursives d'opérations de services logiciels PLM (où chaque composition implémente une opération fonctionnelle PLM). Ainsi l'élément *OrchestrationTechnique*, peut être composé d'une ou de plusieurs opérations (*OperationTechniquePLM*) de services logiciels PLM (*ServiceLogicielPLM*). Cette classe *OrchestrationTechnique* peut être aussi composée de zéro ou plusieurs fragments d'orchestration technique (*OrchestrationTechnique*). Cela veut dire que l'orchestration peut faire appel à des sous orchestrations techniques déjà définies pour implémenter une partie ou la totalité d'une opération fonctionnelle PLM. Cela permet la réutilisation de traitements répétitifs et augmente l'agilité de la composition au niveau technique de notre approche. Ces relations structurelles que nous venons de définir se traduisent, dans le méta-modèle d'orchestration technique, par les attributs spécifiques de l'élément *OrchestrationTechnique* (cf. Tableau 22).

Tableau 22: Attributs spécifiques de l'élément OrchestrationTechnique

Attribut	Type	Description
miniOrchestrationT	OrchestrationTechnique	indique les éléments <i>OrchestrationTechnique</i> contenus dans l'élément parent <i>OrchestrationTechnique</i>
estDefiniePar	OepartionTechniquePLM	Indique les éléments <i>OepartionTechniquePLM</i> contenus dans l'élément parent <i>OrchestrationTechnique</i>

Les instances de la classe *OpérationTechniquePLM* représentent les opérations des services logiciels PLM où chaque opération peut appartenir à un ou plusieurs services logiciels PLM (*i.e.* *ServiceLogicielPLM* dans la Figure 38). De même que pour les orchestrations des deux niveaux métier et fonctionnel, l'ordre d'exécution des opérations des services logiciels PLM et des fragments d'orchestrations technique composites est contrôlé à l'aide du concept de *FlotControle* défini dans la section 3 de ce chapitre. Les différentes relations qui existent entre les concepts *FlotControle*, *OperationTechniquePLM* et *OrchestrationTechnique* sont traduites à l'aide des attributs spécifiques de la classe *FlotControle* (cf. Tableau 23). Le changement des attributs spécifiques à cette classe permet d'effectuer un changement au niveau fonctionnel du processus en remplaçant une opération de service logiciel PLM par une autre ou un fragment d'orchestration technique par un autre.

Tableau 23 : Attributs spécifiques de l'élément FlotControle

Attribut	Type	Description
operationTSource	OperationTechniquePLM	Indique l'élément ou les éléments <i>OperationTechniquePLM</i> source du flot de controle
operationTCible	OperationTechniquePLM	Indique l'élément ou les éléments <i>OperationTechniquePLM</i> cible du flot de controle
orchestrationTSource	OrchestationTechnique	Indique l'élément ou les éléments <i>OrchestationTechnique</i> source du flot de controle
orchestrationTCible	OrchestationTechnique	Indique l'élément ou les éléments <i>OrchestationTechnique</i> cible du flot de controle

Enfin, la spécificité de la spécialisation de la classe *OperationService* au niveau technique consiste au fait que les opérations de services logiciels ne nécessitent qu'une ressource de type *DonneeTechnique*. Cette ressource peut être nécessaire à l'entrée d'une opération de service logiciel PLM ou peut être générée comme sortie d'une opération de services logiciels PLM. La notion de *DonneeTechnique* n'a aucun lien avec le domaine étudié et elle est caractérisée par sa *valeur*. Chaque donnée peut être de type différent et peut avoir une certaine valeur. Elle peut être de type entier, float, chaîne de caractères, etc. Cet objet consiste en une représentation informatique des *DonneePLM* du niveau fonctionnel telle qu'une adresse physique d'un document sous forme de chaîne de caractères ou encore une référence de produit sous forme de *DonneeTechnique* de type entier.

5. De la définition vers la mise en oeuvre du processus de conception de produits manufacturés

Nous présentons dans cette section la démarche que nous proposons pour assurer l'alignement entre les orchestrations aux différents niveaux de l'approche. Comme nous l'avons annoncé à la fin du chapitre II, notre approche suit une démarche de type IDM. Nous mettons en exergue, dans cette section, les efforts de formalisation qu'accompagne notre approche. Nous commençons d'abord, par définir la démarche de déploiement en général (§ 5.1). A cet effet, nous capitalisons les concepts nécessaires pour assurer le déploiement dans un méta-modèle de déploiement. Ensuite, nous détaillons (1) comment une opération métier peut être déployée sur le niveau fonctionnel PLM (§ 5.2) et (2) comment une opération fonctionnelle PLM peut être déployée sur le niveau technique de notre approche (§ 5.3). Enfin, nous exprimons les liens qui existent entre les concepts des modèles d'orchestration métier et fonctionnel et ceux qui existent entre les modèles d'orchestration fonctionnel et technique (§ 5.4). Nous présentons les règles de transformations qui permettent d'assurer l'alignement entre les modèles des différents niveaux de notre approche.

5.1. Modèle de déploiement générique

La mise en œuvre du processus défini au niveau métier, consiste en la transformation du modèle d'orchestration du niveau métier vers un modèle d'orchestration du niveau technique en passant par une transformation vers le niveau fonctionnel. Ceci nécessite la mise en place d'un cadre méthodologique permettant d'assurer une projection d'un niveau vers un autre niveau de notre approche (niveau métier, fonctionnel et technique). Ce cadre méthodologique se présente sous forme d'une aide pour l'utilisateur métier de l'approche afin qu'il puisse effectuer le déploiement des opérations métier composant l'orchestration métier. De même, pour assurer l'alignement entre le modèle d'orchestration du niveau fonctionnel et le modèle d'orchestration du niveau technique, nous proposons une aide pour les personnes responsables du déploiement des processus sur les systèmes PLM. Cette aide permet la projection d'une opération fonctionnelle PLM vers une orchestration d'opérations de services logiciels PLM.

Comme nous l'avons dit précédemment, chaque opération de *SCP* se projette au niveau fonctionnel sous forme d'une orchestration de services fonctionnels PLM. Ces orchestrations fonctionnelles seront orchestrées conformément au modèle d'orchestration du niveau métier (cf. Figure 39).

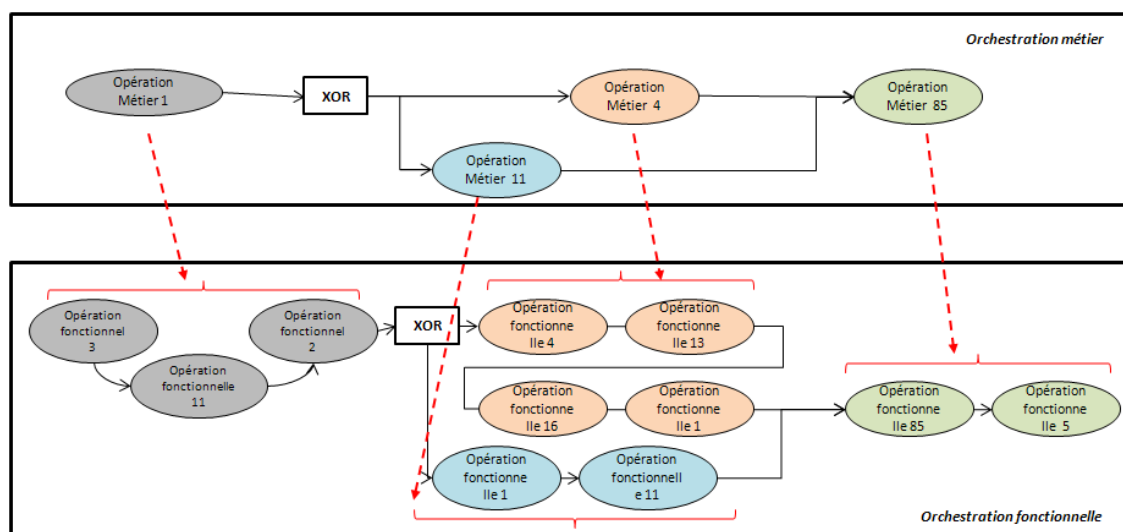


Figure 39 : Alignement entre une orchestration métier et une orchestration fonctionnelle

De même, chaque opération de services fonctionnels PLM se projette au niveau technique de notre approche sous forme d'une orchestration de services logiciels PLM. Ces orchestrations techniques qui implémentent les différentes opérations de services fonctionnels PLM seront orchestrées conformément au modèle d'orchestration du niveau fonctionnel (cf. Figure 40).

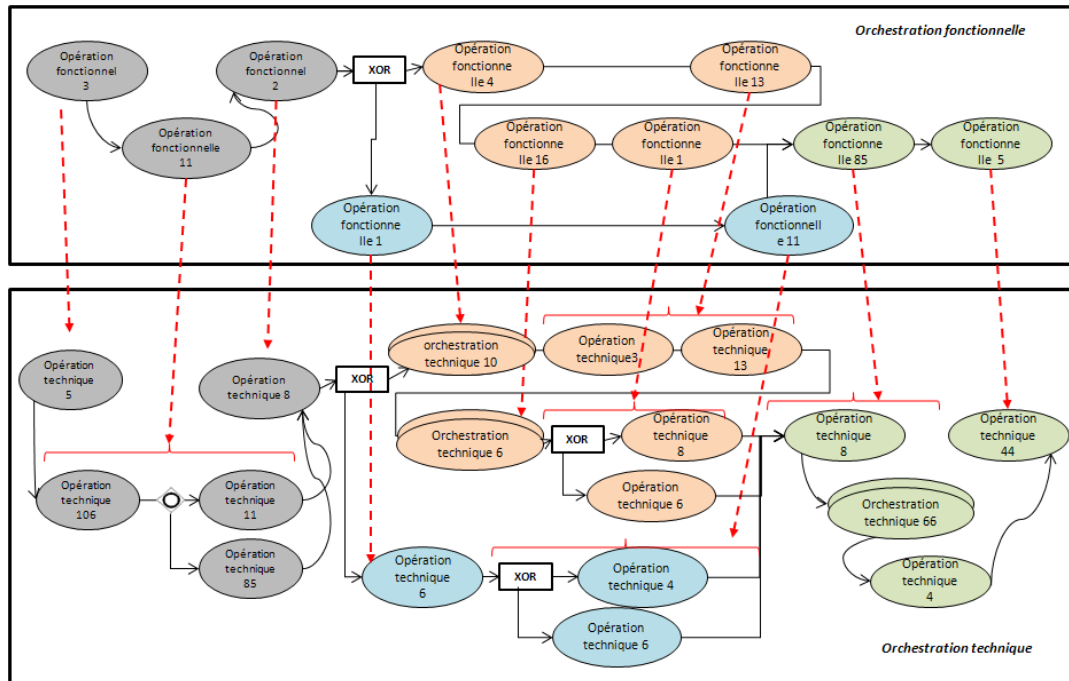


Figure 40 : Alignement entre une orchestration fonctionnelle et une orchestration technique

Pour faciliter cet alignement, nous définissons un catalogue de déploiement pour chaque passage métier/fonctionnel et fonctionnel/technique. L'idée du catalogue de déploiement est de regrouper l'ensemble des lignes directrices permettant le déploiement des opérations métiers et fonctionnelles PLM respectivement sur le niveau fonctionnel et technique. Ce catalogue définit :

- pour chaque opération du niveau supérieur (métier/fonctionnel) quelles sont les opérations du niveau inférieur (respectivement fonctionnel/technique) qui peuvent être utilisées pour sa mise en œuvre.
- les fragments d'orchestrations fonctionnelles ou techniques qui peuvent être utiles respectivement pour la mise en œuvre de plusieurs opérations métiers ou opérations fonctionnelles PLM.

Afin de faciliter le déploiement métier et favoriser la réutilisation des bonnes pratiques, nous proposons une orchestration type (une sorte de bonne pratique) pour le déploiement de chaque opération. Nous appelons cette orchestration type : patron. Une définition souvent associée aux patrons est : *une solution à un problème dans un contexte* [Fowler 97], [Lopistéguy 03]. Le contexte fait référence à un environnement caractéristique dans lequel le patron est applicable. Le problème exprime un ensemble de buts et contraintes qui ont lieu dans ce contexte. La solution fait référence à un modèle que l'on peut appliquer pour atteindre ces buts. Le terme de *patron* a été retenu car les orchestrations (fonctionnelle ou technique) proposées sous forme de patron (*i.e.* solution) traitent du problème récurrent (le but étant respectivement le déploiement d'une opération métier ou fonctionnelle PLM particulière). Les patrons présents dans la littérature sont généralement construits à partir d'expériences prouvées. Les orchestrations que nous jugeons comme patron pour le déploiement

des opérations métier obéissent à cette règle. Nous proposons de capitaliser ces traitements récurrents dans des orchestrations fonctionnelles réutilisables. Ainsi, le catalogue de déploiement que nous proposons définit en plus pour chaque opération, un patron d'orchestration. Ce patron représente une orchestration type qui résout le déploiement métier de l'opération.

Toutefois, une même opération peut avoir deux ou plusieurs déploiements possibles selon le contexte. Autrement dit, deux orchestrations fonctionnelles ou techniques différentes peuvent satisfaire ou mettre en œuvre respectivement une même opération métier ou fonctionnelle PLM mais de deux façons différentes. Nous proposons donc de laisser la possibilité de définir une nouvelle orchestration qui répond aux attentes de l'utilisateur et des contraintes du domaine. Ceci peut être aussi utile dans le cas où le patron fonctionnel ne convient pas aux attentes de l'utilisateur PLM ou bien que les opérations utilisées par le patron technique ne sont pas implémentées dans le système PLM considéré. En effet, chaque utilisateur peut avoir ses habitudes et sa façon de faire. La nouvelle orchestration sera alors définie sur la base des opérations et des fragments d'orchestrations, du niveau supérieur, fournies comme aide dans le catalogue de déploiement de l'opération en question. L'entreprise peut aussi adapter le déploiement à son contexte en partant du patron dans le but de l'enrichir. Ces orchestrations définies par l'utilisateur peuvent être sauvegardées comme des alternatives au patron et ainsi elles peuvent être utilisées par d'autres utilisateurs si le patron ne leur convient pas.

Nous proposons de formaliser les concepts de déploiement dans un méta-modèle de déploiement générique (cf. Figure 41). Ce modèle permet de relier les concepts de déploiement avec ceux des deux niveaux en relation avec le déploiement, à savoir le niveau supérieur (le niveau métier ou fonctionnel, le concept en gris clair) et le niveau inférieur (respectivement le niveau fonctionnel ou technique, les concepts en gris foncé). Les autres concepts sont propres à la démarche de déploiement que nous proposons (ceux en blanc) et n'appartiennent à aucun des deux niveaux sujets de l'alignement.

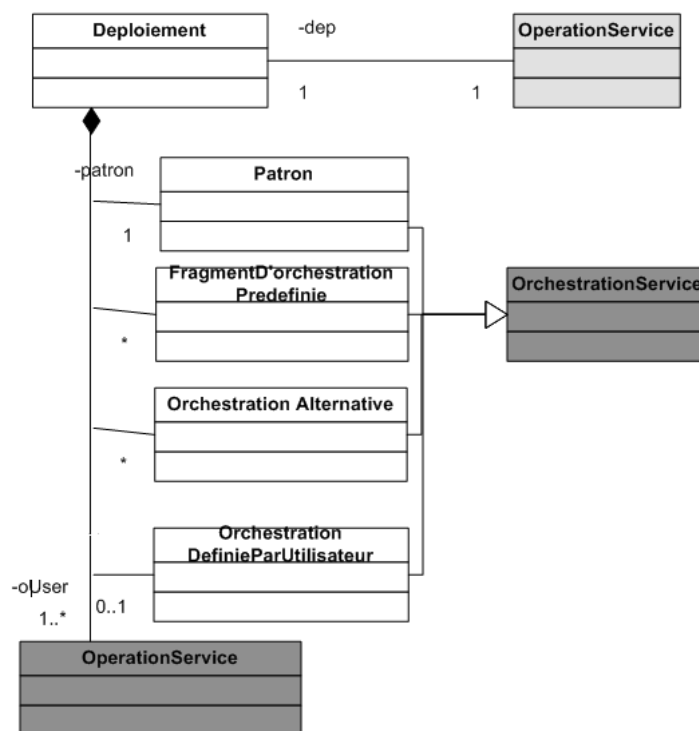


Figure 41 : Méta-modèle générique de déploiement

5.2. Déploiement métier des opérations de SCPs

Nous proposons dans cette section de présenter la spécialisation du méta-modèle de déploiement pour le passage du niveau métier au niveau fonctionnel, appelée méta-modèle de correspondance Métier/Fonctionnel (cf. Figure 42). Le catalogue de passage du niveau métier au niveau fonctionnel permettra de fournir une aide pour l'utilisateur métier de l'approche afin qu'il puisse effectuer le déploiement des opérations métier composant l'orchestration métier. Cette aide ou démarche de déploiement est à utiliser au cours de la phase 2 du cadre méthodologique que nous proposons pour la définition et la mise en œuvre des processus de conception (cf. Figure 30).

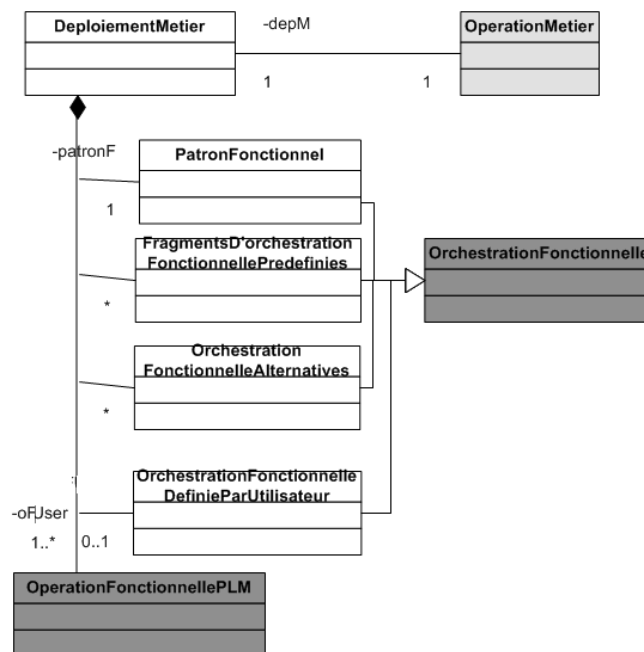


Figure 42 : Méta-modèle de correspondance Métier/Fonctionnel

Le déploiement d'une opération métier se traduit par une orchestration d'opérations de services fonctionnelle PLM permettant de mettre en œuvre l'opération de SCP. Ainsi, le catalogue de correspondance Métier/Fonctionnel que nous proposons définit pour chaque opération métier :

- la liste des opérations fonctionnelles PLM qui peuvent être utilisées pour sa mise en œuvre,
- la liste des fragments d'orchestrations fonctionnelles qui peuvent être utilisées ou composées avec les opérations fournies afin de mettre en œuvre l'opération métier en question,
- un patron d'orchestration fonctionnelle. Ce patron représente une orchestration fonctionnelle type qui résout le déploiement métier de l'opération,
- des orchestrations fonctionnelles définies par l'utilisateur et sauvegardées comme des alternatives au patron.

En effet, comme nous l'avons dit précédemment, les objectifs métiers d'une entreprise sont variables. Parfois, la réalisation d'une opération métier peut être conditionnée par une contrainte temporelle (par exemple, le modèle d'ensemble d'une pièce doit être prêt à une date donnée exigée par le client, ainsi au lieu de mobiliser une opération fonctionnelle de création de modèle CAO le modèle peut être dessiné par un concepteur à la main). Dans un autre temps elle peut plutôt avoir une contrainte sur la façon de faire (par exemple, l'utilisation d'une méthode de calcul bien précise (mobilisation d'une

opération fonctionnelle spécifique à l’outil de calcul)). Ainsi une même opération métier peut avoir deux ou plusieurs déploiements possibles selon le contexte. Nous prenons en compte le fait qu’il est possible de définir une nouvelle orchestration technique mis à part le patron proposé et ses alternatives. La nouvelle orchestration peut être définie sur la base des opérations fonctionnelles PLM et les fragments d’orchestrations fonctionnelles fournies dans le catalogue de déploiement de l’opération en question.

Nous capitalisons les informations concernant le déploiement d’une opération métier dans des fiches de correspondance Métier/Fonctionnel consultables et utilisables par les acteurs métier du domaine de la conception. Dans le tableau ci-dessous nous regroupons les différentes informations contenues dans les fiches détaillant le déploiement.

Tableau 24: Concepts de déploiement métier d'une opération métier

Nom du déploiement	L'intitulé du déploiement métier.
Opération Métier	Le nom de l'opération métier en relation avec le déploiement métier.
Opérations fonctionnelles PLM	La liste des opérations fonctionnelles PLM qui peuvent être utilisées dans le déploiement métier.
Patron	référence du modèle d'orchestration fonctionnelle type qui permet de mettre en œuvre l'opération métier.
Fragment (s) d'orchestration (s) fonctionnelle (s) prédéfinie (s)	La liste des fragments d'orchestrations fonctionnelle peuvent être composées avec des opérations fonctionnelle PLM ou d'autres type d'orchestration fonctionnelle (patron ou orchestration fonctionnelle prédéfinies) pour mettre en œuvre l'opération métier.
Orchestration (s) fonctionnelle (s) Alternatives (s)	Une liste des références des modèles d'orchestration fonctionnelles prédéfinies qui peuvent être utilisées comme alternative au patron.
Modèle d'orchestration fonctionnelle définie par l'utilisateur	C'est une orchestration fonctionnelle définie par l'utilisateur dans le cas où le patron et les orchestrations fonctionnelles prédéfinies ne lui conviennent pas.

Enfin, cette démarche de correspondance entre le niveau métier et le niveau fonctionnel possède un double objectif. D’une part, elle facilite la tâche de mise en œuvre du processus de conception de produits manufacturés par les acteurs métier du domaine de la conception. En effet, le passage du niveau métier au niveau fonctionnel est défini en utilisant les mêmes concepts utilisés par les acteurs métier pour la définition du processus (i.e. les opérations de services portant des noms familiers et les fragments d’orchestration simples et bien définis). D’autre part, le catalogue de déploiement métier constitue une source d’information réutilisable lors de l’alignement entre les deux niveaux métier et fonctionnel. Ceci augmente l’agilité de la démarche l’alignement.

De plus, la démarche facilite l’utilisation du système PLM par les nouveaux utilisateurs. Le fait de proposer pour chaque opération métier comment elle peut être déployée sur un système PLM, à travers l’utilisation des patrons, ceci constitue une sorte de tutoriaux d’apprentissage de l’utilisation du système PLM.

5.3. Déploiement fonctionnel des opérations de services fonctionnels PLM

De même que pour le passage du niveau métier au niveau fonctionnel, nous proposons de spécialiser le méta-modèle de déploiement pour le passage du niveau fonctionnel au niveau technique, appelée méta-modèle de correspondance Fonctionnel/Technique (cf. Figure 43). Le catalogue de passage du niveau fonctionnel au niveau technique permettra de fournir une aide pour les personnes responsables du déploiement des processus sur les systèmes PLM. Elle permet le déploiement des opérations fonctionnelles PLM composant l’orchestration fonctionnel sur un système PLM particulier. Cette aide

ou démarche de déploiement est à utiliser au cours de la phase 3 du cadre méthodologique que nous proposons pour la définition et la mise en œuvre des processus de conception (cf. Figure 30).

En effet, comme nous l'avons dit précédemment, une opération fonctionnelle PLM représente une action faite à travers un système PLM. Les opérations fonctionnelles PLM représentent une abstraction des fonctionnalités des systèmes PLM indépendamment d'une implémentation particulière [Hachani 11b]. Afin que cette action puisse être exécutée, elle doit être projetée vers les opérations de services logiciels PLM qui permettent son implémentation. Ainsi, le déploiement d'une opération de services fonctionnels PLM se traduit par une orchestration d'opérations de services logiciels PLM. Le catalogue de correspondance Fonctionnel/Technique que nous proposons définit pour chaque opération fonctionnelle PLM :

- la liste des opérations de services logiciels PLM qui peuvent être utilisées pour sa mise en œuvre,
- la liste des fragments d'orchestrations techniques qui peuvent être utilisés ou composés avec les opérations fournies afin de mettre en œuvre l'opération fonctionnelle PLM en question,
- un patron d'orchestration technique. Ce patron représente une orchestration technique type qui résout le mieux le déploiement fonctionnel de l'opération,
- des orchestrations techniques définies par l'utilisateur comme des alternatives au patron.

Notons qu'une même entreprise peut avoir plusieurs systèmes PLM (chacun sur un site ou plusieurs sur le même site). Une même opération fonctionnelle PLM peut ainsi avoir deux ou plusieurs déploiements fonctionnels possibles. De plus, le système PLM peut bouger tout au long du cycle de vie de l'entreprise. Des nouveaux modules peuvent être ajoutés (ajout d'un nouveau module qui permet l'intégration avec les outils CAO par exemple) et donc les déploiements fonctionnels de certaines opérations fonctionnelles PLM peuvent être enrichis ou modifiés. Nous prenons en compte le fait qu'il est possible de définir une nouvelle orchestration technique mise à part le patron proposé et ses alternatives. La nouvelle orchestration peut être définie sur la base des opérations techniques PLM et les fragments d'orchestrations techniques fournies dans le catalogue de déploiement de l'opération.

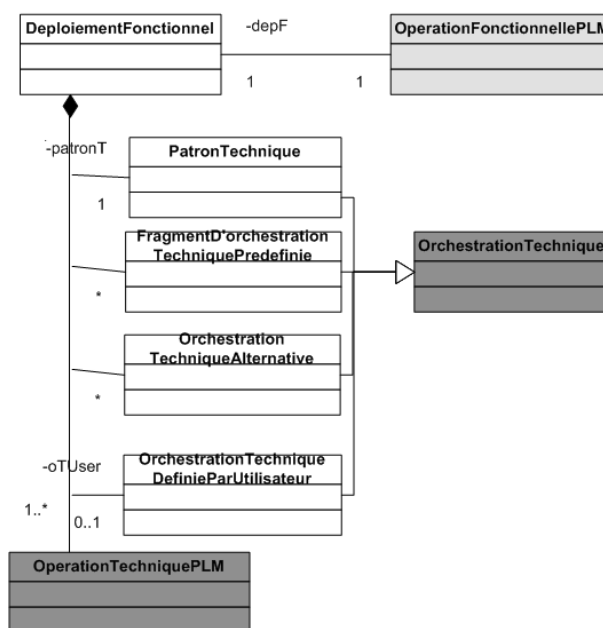


Figure 43 : Méta-modèle de correspondance Fonctionnel/Technique

Nous capitalisons les informations concernant le déploiement d'une opération fonctionnelle PLM dans des fiches de correspondance Fonctionnel/Technique. Dans le tableau ci-dessous nous regroupons les différentes informations contenues dans ces fiches.

Tableau 25: Concepts de déploiement fonctionnel d'une opération fonctionnelle PLM

Nom du déploiement	L'intitulé du déploiement fonctionnel.
Opération fonctionnelle PLM	Le nom de l'opération fonctionnelle PLM en relation avec le déploiement fonctionnel.
Opérations techniques PLM	La liste des opérations de services logiciels PLM qui peuvent être utilisées dans le déploiement fonctionnel.
Patron technique	un modèle d'orchestration technique type qui permet de mettre en œuvre l'opération fonctionnelle PLM.
Fragment (s) d'orchestration (s) technique (s)	La liste des fragments d'orchestrations technique qui peuvent être composés avec des opérations ou d'autres orchestrations techniques (patron technique ou orchestration technique prédéfinies) pour mettre en œuvre l'opération fonctionnelle PLM.
Modèle (s) d'orchestration (s) technique (s) prédéfinie (s)	Une liste de modèles d'orchestrations techniques prédéfinies qui peuvent être utilisés comme alternative au patron technique.
Modèle d'orchestration technique définie par l'utilisateur	C'est une orchestration technique définie par l'utilisateur ou ca ou le système PLM évolue ou dans le cas ou le processus est déployé dans un nouveau système PLM.

Cette démarche de correspondance entre le niveau fonctionnel et technique admet plusieurs avantages para rapport à notre problématique. Cette démarche intègre la flexibilité au cours du choix du déploiement fonctionnel (le déploiement d'une opération peut avoir plusieurs implémentations différentes ; il peut être implémenté par le patron ou encore par une orchestration définie par l'utilisateur). De plus, le déploiement fonctionnel montre la généralité de l'approche proposée. En effet, les orchestrations métier et fonctionnelles non pas à être ré-implémenter si on change de système PLM ou encore si on rajoute au système existant de nouveaux modules. Il suffit d'exécuter le déploiement fonctionnel correspondant au système PLM choisi. Enfin, le déploiement fonctionnel constitue une source d'information réutilisable lors de l'alignement entre le niveau fonctionnel et technique. Ceci évite de faire appel à des acteurs ayant des connaissances techniques afin de ré-implémenter le déploiement des opérations fonctionnelles PLM s'il y a un besoin de changement.

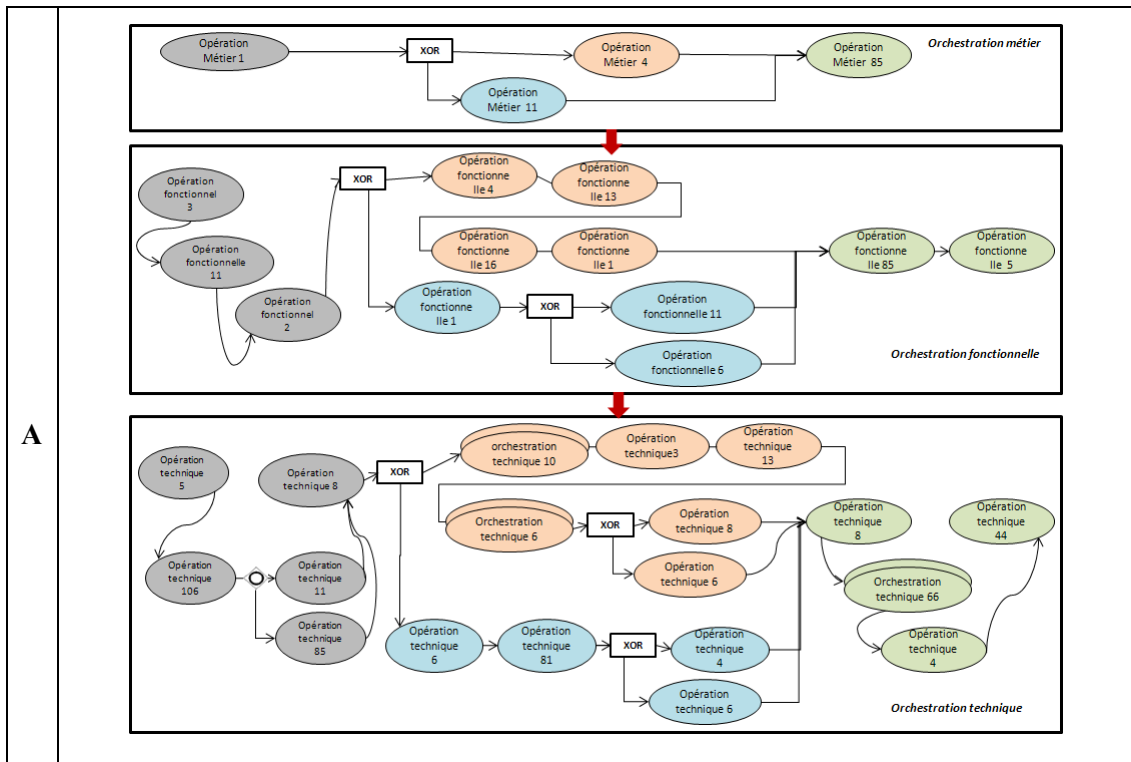
5.4. Alignement entre les différents modèles d'orchestration de notre approche

Nous avons jusqu'à ici présenté les différents méta-modèles nécessaires pour l'usage des services de notre approche. Le but de cette section est de présenter la démarche d'alignement entre les différents modèles conformes aux méta-modèles présentés. Cette démarche d'alignement se base principalement sur les transformations nécessaires pour aligner le modèle métier avec le modèle du niveau fonctionnel et ceux nécessaires pour aligner le modèle du niveau fonctionnel avec celui du niveau technique. Comme nous l'avons défini auparavant, les transformations représentent l'ensemble des étapes partiellement ordonnées qui permettent l'obtention d'un artefact inconnu à partir d'un ou plusieurs artefacts connus.

Nous proposons une démarche d'alignement et des règles de transformation génériques qui s'appliquent à tous les modèles de processus de conception. Cette démarche d'alignement possède deux intérêts : (1) Elle permet l'automatisation des processus à travers l'exécution des règles de transformation génériques. (2) Elle réduit les délais de mise en place du processus à travers l'utilisation de la méthode de déploiement des opérations, qui masque la complexité d'implémentation des opérations constituant le processus.

Avant de se pencher sur la définition des transformations entre les différents modèles de notre approche, nous nous sommes posé la question sur la meilleure démarche d'alignement. En effet, nous pouvons proposer deux démarches d'alignement différentes :

- une première démarche qui donne lieu à un seul modèle exécutable, c'est-à-dire contenant toutes les versions possibles d'instance exécutable du processus (cf. partie A de la Figure 44).
- une deuxième démarche qui donne lieu à plusieurs versions exécutables du processus de conception. Chacune de ces versions peut être utilisée selon le besoin (cf. partie B, Figure 44).



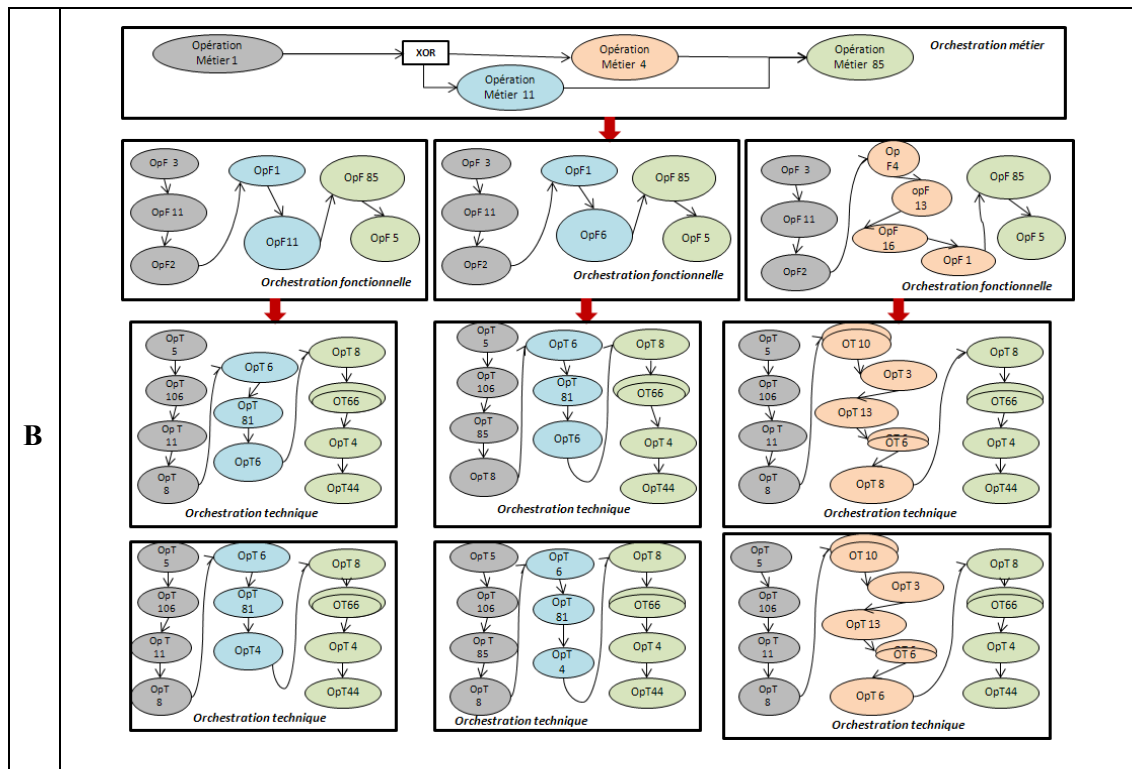


Figure 44 : Alignement : Génération d’une seule version exécutable Vs. Génération de toutes les versions exécutables du processus

Chacune de ces deux démarches d’alignement possède des avantages et des inconvénients. Nous synthétisons dans le Tableau 26 les avantages et inconvénients de ces deux démarches. Tableau 26

Tableau 26 : Avantages et inconvénients de chacune des deux démarches d’alignement

Génération d’un seul Modèle exécutable		Génération de plusieurs chemins d’exécution	
Avantages	Inconvénients	Avantages	Inconvénients
La vision globale du processus de conception est conservée.	Le modèle résultat de l’alignement est trop chargé.	Transformation du niveau fonctionnel vers le niveau technique peut être guidée par l’exécution. Cela à travers le calcul du temps d’exécution nécessaire pour chaque chemin possible.	Plusieurs transformations entre le niveau fonctionnel et le niveau technique.
Modification sur un seul modèle.		Possibilité de calcul des performances de chaque exécution. Cela peut être utile pour les prochaines exécutions du processus.	Perdre la vision globale du processus.
			Coût de modification énorme. Modification sur toutes les instances générées.

Comme le montre le Tableau 26, chacune des deux démarches d’alignement possède des avantages et inconvénients. Nous constatons que la deuxième façon de faire présente des avantages qui sont plutôt axés instrumentation et exécution. Elle permet de prédire le temps nécessaire pour chaque exécution. Toutefois, les inconvénients de cette deuxième façon de faire contredit les objectifs de notre travail. En effet, cette démarche permet de générer plusieurs versions exécutables pour un seul processus métier. Ainsi, si jamais il y’a un besoin de modification au niveau métier, cela nécessite de modifier chacune des versions exécutables (*i.e.* faire autant de modifications que le nombre de chemins d’exécution générés). Ceci augmente le délai nécessaire pour l’alignement entre les différents niveaux de notre approche (plusieurs transformations, une pour chaque version exécutable du processus) et réduit par conséquent l’agilité de notre approche. De plus, le fait d’avoir comme résultat plusieurs chemins d’exécution, cela ne donne pas une vision complète du processus. Cet inconvénient ne se présente pas

dans la première façon de faire vu qu'il y a génération d'un seul processus qui garde la vision globale du processus du niveau métier. De même pour le temps de modification, cet inconvénient est résolu vu que la modification ne sera faite que sur le seul modèle résultat de l'alignement. L'unique inconvénient de cette première façon de faire se résume dans le fait que le modèle résultat de l'alignement est trop chargé. C'est-à-dire qu'il orchestre plusieurs opérations et fragments d'orchestrations. En effet, chacune des opérations de *SCPs* qui composent le processus du niveau métier peut être traduite par des orchestrations fonctionnelles. Ces orchestrations fonctionnelles peuvent être traduites à leur tour par des opérations de services logiciels PLM ou encore des orchestrations techniques. Ceci augmente le nombre d'opérations orchestrées dans le modèle final. Toutefois, cet inconvénient ne contredit pas nos objectifs de thèse puisque les acteurs métier n'auront accès qu'au modèle du niveau métier. Ainsi, nous favorisons la première démarche d'alignement puisqu'elle ne contredit pas les objectifs de notre travail.

En suivant cette démarche, l'alignement entre les différents modèles de notre approche (modèles d'orchestration métier, fonctionnel et technique) est réalisé grâce à un ensemble de règles de transformation qui permettent de passer d'un ou plusieurs modèles source à un modèle cible. A chaque fois les modèles sources restent intacts alors qu'il y a génération d'un nouveau modèle, résultat de la transformation. Nous proposons un alignement qui se base principalement sur deux transformations ;

- une première transformation entre les deux niveaux métier et fonctionnel de notre approche (cf. T1 dans la Figure 45). Cette transformation a pour but de transformer le modèle d'orchestration du niveau métier vers un modèle d'orchestration du niveau fonctionnel (conforme au méta-modèle d'orchestration fonctionnelle). Pour cette première orchestration les modèles sources sont le modèle d'orchestration métier (conforme au méta-modèle d'orchestration métier) et le modèle de déploiement métier (conforme au méta-modèle de correspondance Métier/Fonctionnel). Le modèle de déploiement métier sert pour l'identification des orchestrations fonctionnelles permettant la mise en œuvre des opérations de *SCPs* composant le modèle d'orchestration métier. Le modèle d'orchestration métier sert pour l'identification des flots de contrôle qui orchestrent les opérations de *SCPs*. Ces flots de contrôle sont ceux qui vont permettre l'orchestration des orchestrations fonctionnelles permettant la mise en œuvre des opérations de *SCPs*. Enfin, le modèle cible de cette transformation est le modèle d'orchestration fonctionnelle (conforme au méta-modèle d'orchestration fonctionnelle).
- une deuxième transformation entre les deux niveaux fonctionnel et technique (cf. T2 dans la Figure 45). Cette deuxième transformation a pour but de transformer le modèle d'orchestration résultat de la première transformation (conforme au méta-modèle d'orchestration fonctionnelle) vers un modèle d'orchestration du niveau technique (conforme au méta-modèle d'orchestration technique). Pour cette deuxième orchestration les modèles sources sont le modèle d'orchestration fonctionnelle (conforme au méta-modèle d'orchestration fonctionnelle) et le modèle de déploiement fonctionnel (conforme au méta-modèle de correspondance Fonctionnel/Technique). Le modèle de déploiement fonctionnel sert pour l'identification des orchestrations techniques permettant la mise en œuvre des opérations fonctionnelles composant le modèle d'orchestration fonctionnelle. Le modèle d'orchestration fonctionnelle sert pour l'identification des flots de contrôles qui orchestrent les orchestrations techniques implémentant les opérations fonctionnelles PLM. Enfin, le modèle cible de cette transformation est le modèle d'orchestration technique (conforme au méta-modèle d'orchestration technique).

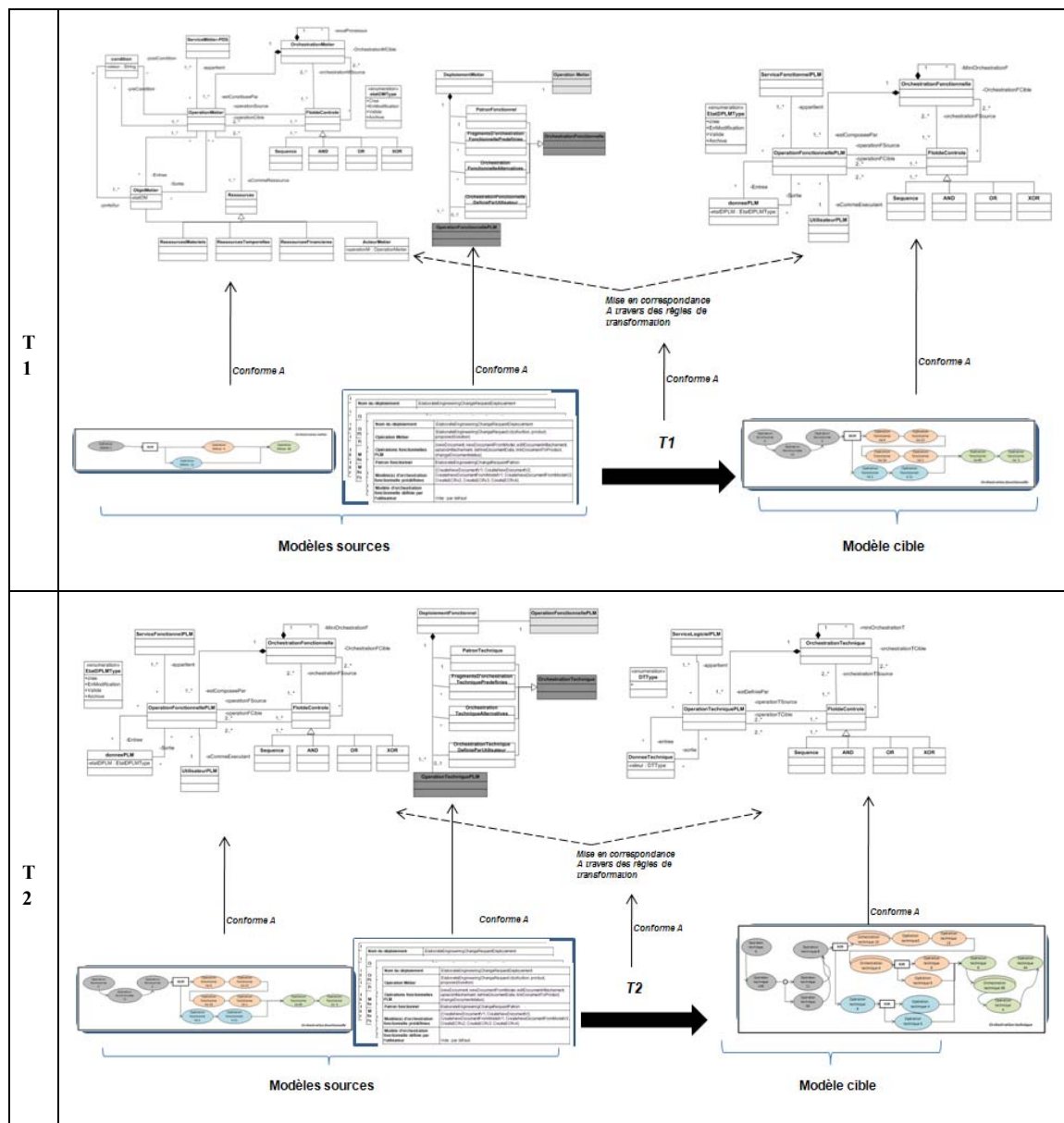


Figure 45 : Alignement : transformations entre les modèles d'orchestrations de services métier, fonctionnels et logiciels

Le passage entre les différents modèles est réalisé grâce à un ensemble de règles de transformation. Les règles de transformation de T1 et T2 peuvent être décrites de plusieurs façons. Selon [Miller 03], les règles de transformation peuvent être décrites en utilisant le langage naturel, ou un algorithme écrit en utilisant un langage d'action (*action language*). La description en langage naturel permet d'avoir une idée sur les règles pour pouvoir les comprendre ou encore les valider. La description en langage d'action permet d'automatiser les transformations. Ceci par le biais de programmes de transformation, qui sont dans la plupart des cas développés dans des langages généralistes (par exemple Java) ou dédiés (par exemple ATL [Jouault 05] et QVT [OMG 08b]). Dans ce qui suit nous exprimons les règles de transformation entre les modèles de notre approche en langage naturel pour qu'elles puissent être comprises. Toutefois afin que l'alignement entre les modèles d'orchestration de notre approche puisse être automatisé nous choisissons ATL (*ATLAS Transformation Language*) comme langage d'implémentation. ATL est le langage de transformation le plus courant et déjà utilisé [Allilaire 04].

Nous présentons dans le Tableau 27, d'une manière informelle (*i.e.* description en langage naturel), les règles que nous proposons pour assurer la transformation entre le niveau métier et fonctionnel de notre approche.

Tableau 27: Règles de transformation du modèle métier vers le modèle fonctionnel

Règle de Transformation – T1
<p>R0_: Chaque << OrchestrationMetier >> 'oM' contenue dans le Modèle d'orchestration métier et qui n'a pas d'attribut <<orchestrationMMere >> se transforme en une << OrchestrationFonctionnelle >> 'oF' dans le Modèle d'orchestration Fonctionnelle tel que (oF.id = oM.id, oF.designation = oM.designation, exécuter R1 avec comme entrée l'oM et sa correspondante oF);</p>
<p>R1 (oM,oF):</p> <ul style="list-style-type: none"> - R1.1: Chaque << OrchestrationMetier >> 'om' de <<oM.sousProcessus>> se transforme en une <<OrchestrationFonctionnelle>> 'of' dans le Modèle d'orchestration Fonctionnelle tel que (of.id = om.sousProcessus.id, of.designation= om.sousProcessus.designation, of.orchestrationFMere = oF, oF.MiniOrchestrationF= of, exécuter r1 avec comme entrée l'orchestration fonctionnelle courante om et l'of crée). - R1.2 : Chaque << OpérationMetier>> de <<oM.estConstitueePar >> se transforme en une <<OrchestrationFonctionnelle >> 'of' dans le Modèle d'orchestration Fonctionnelle tel que; (of.orchestrationFMere = oF, oF.MiniOrchestrationF = of) et affecter le déploiement métier de l'opération métier à 'of' de la façon suivante : vérifier dans le modèle de déploiement métier en utilisant l'identifiant de l'opération métier à alignée <<oM.estConstitueePar.id>> si l'opération métier à alignée 'opM' possède une orchestration fonctionnelle définie par l'utilisateur comme déploiement (c'est-à-dire que <<opM.depM.oFUser>> existe) alors : <ul style="list-style-type: none"> o affecter l'orchestration fonctionnelle définie par l'utilisateur ainsi que ses relations avec les classes OpérationFonctionnellePLM, Orchestrationfonctionnelle, flotDeContrôle, etc. à 'of' (of = opM.depM.oFUser). o affecter à 'of' l'identifiant de l'opération métier à alignée (of.id = oM.estConstitueePar.id). o affecter à 'of' la désignation de l'opération métier à alignée (of.designation= oM.estConstitueePar.designation). sinon, dans le cas ou le déploiement métier est définie par le patron fonctionnel, alors <ul style="list-style-type: none"> o affecter le patron fonctionnel et ces relations avec les classes OpérationFonctionnellePLM, OrchestrationFonctionnelle, flotDeContrôle, etc. à 'of' (of = opM.depM.patronF, etc.), o affecter à 'of' l'identifiant de l'opération métier à alignée (of.id = oM.estConstitueePar.id), o affecter à 'of' la désignation de l'opération métier à alignée (of.designation= oM.estConstitueePar.designation). - R1.3 : chaque flot de contrôle 'fm' <<oM.fm>>, lié à <<OrchestrationMetier>> 'oM' transformée en <<OrchestrationFonctionnelle>> 'oF', se transforme en un <<FlotDeControle>> dans le modèle d'orchestration fonctionnelle tel que: <ul style="list-style-type: none"> o Si 'fm' est de type Séquence, 'fm' se transforme en une classe 'ff' de type <<Sequence>> dans le modèle d'orchestration fonctionnelle et ses attributs se transforme comme suit: (ff.ordre = fm.ordre, ff.oFmere = oF, ff.id=fm.id, ff.designation=fm.designation), puis exécuter R2.1 pour affecter la source et cible du flot de contrôle. o Si 'fm' est de type AND, 'fm' se transforme en une classe 'ff' de type <<AND>> dans le modèle d'orchestration fonctionnelle et ses attributs se transforme comme suit: (ff.ordre = fm.ordre, ff.oFmere = oF, ff.id=fm.id, ff.designation=fm.designation), puis exécuter R2.2 pour affecter la source et les cibles du flot de contrôle. o Si 'fm' est de type XOR, 'fm' se transforme en une classe 'ff' de type << XOR >> dans le modèle d'orchestration fonctionnelle tel que (ff.ordre = fm.ordre, ff.oFmere = oF, ff.id=fm.id, ff.designation=fm.designation), puis exécuter R2.2 pour affecter la source et les cibles du flot de contrôle. o Si 'fm' est de type OR, 'fm' se transforme en une classe 'ff' de type << OR >> dans le modèle d'orchestration fonctionnelle tel que (ff.ordre = fm.ordre, ff.oFmere = oF, ff.id=fm.id, ff.designation=fm.designation), exécuter R2.2 pour affecter la source et les cibles du flot de contrôle.

R2.1 : aligne la cible et source d'un flot de contrôle 'fm' de type <<sequence>> contenu dans le modèle d'orchestration métier avec la cible et source du flot de contrôle 'ff' créée dans le modèle d'orchestration fonctionnelle.

o Si <<fm.operationMSource>> existe (c'est-à-dire le flot de séquence a comme source une classe de type <<OperationMetier>>) alors affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'opération métier source du flux de séquence <<fm.operationMSource.id>> à <<ff.orchestrationFSource>>.

o Si <<fm.orchestrationMSource>> existe (c'est-à-dire le flot de séquence a comme source une classe de type <<OrchestrationMetier>>) alors affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'orchestration métier source du flux de séquence <<fm.orchestrationMSource.id>> à <<ff.orchestrationFSource>>.

o Si <<fm.operationMCible>> existe (c'est-à-dire le flot de séquence a comme cible une classe de type <<OperationMetier>>) alors affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'opération métier cible du flux de séquence <<fm.operationMCible.id>> à <<ff.orchestrationFCible>>.

o Si <<fm.orchestrationMCible>> existe (c'est-à-dire le flot de séquence a comme cible une classe de type OrchestrationMetier) alors affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'orchestration métier cible du flux de séquence <<fm.orchestrationMCible.id>> à <<ff.orchestrationFCible>>.

R2.2 : aligne la cible et source d'un flot de contrôle 'fm' de type <<AND>>/<<OR>>/<<XOR>> contenu dans le modèle d'orchestration métier avec la source et les cibles du flot de contrôle 'ff' créée dans le modèle d'orchestration fonctionnelle.

o Si <<fm.operationMSource>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme source une classe de type <<OperationMetier>>) alors affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'opération métier source du flot de contrôle <<fm.operationMSource.id>> à <<ff.orchestrationFSource>>.

o Si <<fm.orchestrationMSource>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme source une classe de type <<OrchestrationMetier>>) alors affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'orchestration métier source du flot de contrôle <<fm.orchestrationMSource.id>> à <<ff.orchestrationFSource>>.

o Si <<fm.operationMCible>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme cible une ou plusieurs classe de type <<OperationMetier>>) alors pour chaque <<fm.operationMCible>>, affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'opération métier cible du flot de contrôle <<fm.operationMCible.id>> à <<ff.orchestrationFCible>>.

o Si <<fm.orchestrationMCible>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme cible une ou plusieurs classe de type <<OrchestrationMetier>>) alors pour chaque <<fm.orchestrationMCible>>, affecter l'orchestration fonctionnelle 'of' composite de l'orchestration fonctionnelle mère du flot de contrôle <<ff.oFMere.MiniOrchestrationsF>> qui a comme identifiant <<of.identifiant>> l'identifiant de l'orchestration métier cible du flot de contrôle <<fm.orchestrationMCible.id>> à <<ff.orchestrationFCible>>.

En somme, l'alignement entre le niveau métier et fonctionnel inclut un modèle intermédiaire pour aboutir enfin au modèle d'orchestration fonctionnelle, à savoir le modèle de déploiement métier. Ces règles permettent la transformation des concepts du méta-modèle d'orchestration métier en un ensemble de concepts du méta-modèle d'orchestration fonctionnelle. Étant donnée que ces règles ne dépendent pas d'un modèle donné d'orchestration métier ni de déploiement métier ceci les rends génériques, réutilisables et applicables pour l'alignement de n'importe quel processus de conception sur le niveau fonctionnel de notre approche.

Le Tableau 28 illustre les règles proposées pour assurer la transformation entre le niveau fonctionnel et technique de notre approche. Ces dernières suivent la même logique que celles proposées dans le Tableau 27. Ces règles permettent la transformation des concepts du méta-modèle d'orchestration fonctionnelle en un ensemble de concepts du méta-modèle d'orchestration technique. Cette

transformation inclut un modèle intermédiaire, le modèle de déploiement fonctionnel, pour aboutir à l'orchestration technique finale.

Tableau 28: Règles de transformation du modèle fonctionnel vers le modèle technique

Règle de Transformation
<p>R0 : Chaque << OrchestrationFonctionnelle >> 'oF' contenue dans le Modèle d'orchestration fonctionnelle et qui n'a pas d'attribut <<orchestrationFMere >> se transforme en une << OrchestrationTechnique >> 'oT' dans le Modèle d'orchestration Technique tel que (oT.id = oF.id, oT.designation = oF.designation, exécuter R1 avec comme entrée l'oF et sa correspondante oT);</p>
<p>R1 (oF,oT) :</p> <ul style="list-style-type: none"> - R1.1 : Chaque << OrchestrationFonctionnelle >> 'of' de <<oF.miniOrchestrationF>> se transforme en une <<OrchestrationTechnique >> 'ot' dans le Modèle d'orchestration Technique tel que (ot.id = of.miniOrchestrationsF.id, ot.designation= of.miniOrchestrationsF.designation, ot.orchestrationTMere = oT, oT.MiniOrchestrationT= ot, exécuter r1 avec comme entrée l'orchestration fonctionnelle courante of et l'ot créée). - R1.2 : Chaque << Opération Fonctionnelle PLM >> de <<oF.estDefiniePar>> se transforme en une <<OrchestrationTechnique >> 'ot' dans le Modèle d'orchestration Technique tel que; (ot.orchestrationTMere = oT, oT.MiniOrchestrationT = ot) et affecter le déploiement fonctionnelle de l'opération fonctionnelle PLM à 'ot' de la façon suivante : vérifier dans le modèle de déploiement fonctionnel en utilisant l'identifiant de l'opération fonctionnelle à alignée <<oF.estDefiniePar.id>> si l'opération fonctionnelle PLM à alignée 'opF' possède une orchestration technique définie par l'utilisateur comme déploiement (c'est-à-dire que <<opF.depF.oTUser>> existe) alors : <ul style="list-style-type: none"> o affecter l'orchestration Technique définie par l'utilisateur ainsi que ses relations avec les classes OpérationTechniquePLM, OrchestrationTechnique, flotDeContrôle, etc. à 'ot' (ot = opF.depF.oTUser). o affecter à 'ot' l'identifiant de l'opération fonctionnelle PLM à alignée (ot.id = oF.estComposeePar.id). o affecter à 'ot' la designation de l'opération fonctionnelle PLM à alignée (ot.designation= oF.estComposeePar.designation). sinon, dans le cas ou le déploiement fonctionnel est définie par le patron technique, alors <ul style="list-style-type: none"> o affecter le patron technique et ces relations avec les classes OpérationTechniquePLM, OrchestrationTechnique, flotDeContrôle, etc. à 'ot' (ot = opF.depF.patronT, etc.), o affecter à 'ot' l'identifiant de l'opération fonctionnelle PLM à alignée (ot.id = oF.estComposeePar.id), o affecter à 'ot' la désignation de l'opération fonctionnelle PLM à alignée (ot.designation= oF.estComposeePar.designation). - R1.3 : chaque flot de contrôle 'ff' <<oF.ff>>, lié à <<OrchestrationFonctionnelle>> 'oF' transformée en <<OrchestrationTechnique>> 'oT', se transforme en un <<FlotDeControle>> dans le modele d'orchestration technique tel que: <ul style="list-style-type: none"> o Si 'ff' est de type Séquence, 'ff' se transforme en une classe 'ft' de type <<Sequence>> dans le modèle d'orchestration technique et ses attributs se transforme comme suit : (ft.ordre = ff.ordre, ft.oTmere = oT, ft.id=ff.id, ft.designation=ff.designation), puis exécuter R2.1 pour affecter la source et cible du flot de contrôle. o Si 'ff' est de type AND, 'ff' se transforme en une classe 'ft' de type <<AND>> dans le modèle d'orchestration technique et ses attributs se transforme comme suit : (ft.ordre = ff.ordre, ft.oTmere = oT, ft.id=ff.id, ft.designation=ff.designation), puis exécuter R2.2 pour affecter la source et les cibles du flot de contrôle. o Si 'ff' est de type XOR, 'ff' se transforme en une classe 'ft' de type << XOR >> dans le modèle d'orchestration technique tel que (ft.ordre = ff.ordre, ft.oTmere = oT, ft.id=ff.id, ft.designation=ff.designation), puis exécuter R2.2 pour affecter la source et les cibles du flot de contrôle. o Si 'ff' est de type OR, 'ff' se transforme en une classe 'ft' de type << OR >> dans le modele d'orchestration technique tel que (ft.ordre = ff.ordre, ft.oTmere = oT, ft.id=ff.id, ft.designation=ff.designation), exécuter R2.2 pour affecter la source et les cibles du flot de contrôle.

R2.1 : aligne la cible et source d'un flot de contrôle 'ff' de type <<sequence>> contenu dans le modèle d'orchestration fonctionnelle avec la cible et source du flot de contrôle 'ft' créée dans le modèle d'orchestration technique.

o Si <<ff.operationFSource>> existe (c'est-à-dire le flot de séquence a comme source une classe de type <<OperationFonctionnellePLM>>) alors affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'opération fonctionnelle source du flux de séquence <<ff.operationFSource.id>> à <<ft.orchestrationTSource>>.

o Si <<ff.orchestrationFSource>> existe (c'est-à-dire le flot de séquence a comme source une classe de type <<OrchestrationFonctionnelle>>) alors affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'orchestration fonctionnelle source du flux de séquence <<ff.orchestrationFSource.id>> à <<ft.orchestrationTSource>>.

o Si <<ff.operationFCible>> existe (c'est-à-dire le flot de séquence a comme cible une classe de type <<OperationFonctionnelle>>) alors affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'opération fonctionnelle cible du flux de séquence <<ff.operationFCible.id>> à <<ft.orchestrationTCible>>.

o Si <<ff.orchestrationFCible>> existe (c'est-à-dire le flot de séquence a comme cible une classe de type OrchestrationFonctionnelle) alors affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'orchestration fonctionnelle cible du flux de séquence <<ff.orchestrationFCible.id>> à <<ft.orchestrationTCible>>.

R2.2 : aligne la cible et source d'un flot de controle 'ff' de type <<AND>>/<<OR>>/<<XOR>> contenu dans le modèle d'orchestration fonctionnelle avec la source et les cibles du flot de contrôle 'ft' créée dans le modèle d'orchestration technique.

o Si <<ff.operationFSource>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme source une classe de type <<OperationFonctionnelle>>) alors affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'opération fonctionnelle source du flot de contrôle <<ff.operationFSource.id>> à <<ft.orchestrationTSource>>.

o Si <<ff.orchestrationFSource>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme source une classe de type <<OrchestrationFonctionnelle>>) alors affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'orchestration fonctionnelle source du flot de contrôle <<ff.orchestrationFSource.id>> à <<ft.orchestrationTSource>>.

o Si <<ff.operationFCible>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme cible une ou plusieurs classe de type <<OperationFonctionnelle>>) alors pour chaque <<ff.operationFCible>>, affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'opération fonctionnelle cible du flot de contrôle <<ff.operationFCible.id>> à <<ft.orchestrationTCible>>.

o Si <<ff.orchestrationFCible>> existe (c'est-à-dire le flot de type <<AND>>/<<OR>>/<<XOR>> a comme cible une ou plusieurs classe de type <<OrchestrationFonctionnelle>>) alors pour chaque <<ff.orchestrationFCible>>, affecter l'orchestration technique 'ot' composite de l'orchestration technique mère du flot de contrôle <<ft.oTMere.MiniOrchestrationsT>> qui a comme identifiant <<ot.identifiant>> l'identifiant de l'orchestration fonctionnelle cible du flot de contrôle <<ff.orchestrationFCible.id>> à <<ft.orchestrationTCible>>.

6. Synthèse

L'approche de composition et le cadre de mise en œuvre des processus de conception que nous avons présentés respectent les contraintes spécifiques aux processus du domaine de la conception de produits manufacturés (cf. section 2.3 du chapitre I) et répondent aux objectifs de notre travail à savoir :

- Automatisation et dynamisme : les processus de conception de produits manufacturés sont tendus. La concurrence du marché induit une forte pression pour pouvoir mettre des produits sous le marché dans les plus brefs délais et avec des coûts réduits. Notre proposition permet la réutilisation des opérations des services métier du domaine de la conception (SCP) et l'automatisation de leur enchaînement sous forme d'orchestration de services. Ceci augmente la capacité de réutilisation et d'automatisation des tâches répétitives et de l'enchaînement des activités du processus. De plus, à travers l'exécution des règles de transformation génériques

et la manipulation d'une méthode de déploiement des opérations, qui masque la complexité d'implémentation des opérations constituant le processus, ceci permet de réduire les délais de mise en place du processus. Notre approche permet ainsi de réduire le temps de définition et mise en place du processus de conception de produits,

- Flexibilité : les processus de conception de produits manufacturés sont caractérisés par leur caractère évolutif et changement fréquent. Le cadre que nous proposons permet de répondre rapidement à un besoin de changement et de projeter la définition métier du processus vers un support flexible. Ceci est assuré à travers la modification de l'implémentation du déploiement métier et/ou fonctionnel et la réexécution des règles de transformation s'il y'a besoin de changement au niveau fonctionnel du changement (cf. Tableau 3 du chapitre I). De plus, l'automatisation de l'orchestration en utilisant des langages d'orchestration qui supportent l'évolution dynamique de la composition permet d'effectuer des changements au niveau structurel du changement (ajout ou suppression de services ou fragments de processus).

Elle répond aussi aux manques des langages de modélisation de processus utilisés dans l'industrie de nous jours (cf. section 3.3 du chapitre I), à savoir ;

- la simplicité du modèle de processus : la nature des acteurs métier du domaine de la conception de produits manufacturés exige l'utilisation de concepts simples et bien définis permettant de faciliter la tâche de définition et de mise en œuvre du processus de conception de produits manufacturés. Ceci est assuré à travers la manipulation de concepts familiers du domaine exposés sous forme d'opérations de *SCPs*. De plus, la méthode que nous proposons permet non uniquement la composition d'opérations de *SCPs* mais aussi de bout de processus exposés sous forme de fragments d'orchestration (sous-processus définis sous forme d'orchestrations métier). Ceci permettra de définir des processus complexes mais non ambigus (la complexité des sous-processus est masquée par les fragments d'orchestration).
- la richesse et expressivité du modèle de processus : la proposition que nous avons faite permet la représentation des concepts nécessaires pour la définition des processus de conception de produits manufacturés. Elle permet aussi de masquer la complexité du processus parent en utilisant des fragments d'orchestrations.

Pour conclure, notre approche propose une démarche de déploiement que nous considérons compréhensible par les personnes métier qui définissent les processus puisqu'ils font recours à des termes familiers aux personnes du domaine. Elle permet aussi un support automatisable et flexible pour les processus de conception de produits manufacturés. Ceci à travers l'exécution des règles de transformation et l'utilisation des langages d'orchestration exécutables).

7. Conclusion

Nous avons présenté dans ce chapitre le cadre que nous proposons pour la définition et la mise en œuvre des processus de conception de produits manufacturés. Ce cadre comprend principalement trois phases : la phase de modélisation du processus de conception sous forme de composition de services, la phase de déploiement métier permettant l'alignement du modèle métier vers le niveau fonctionnel de notre démarche et enfin la phase de déploiement fonctionnel permettant la projection des opérations fonctionnelles PLM vers le niveau technique. Pour chacune de ces phases nous avons proposé les concepts et mécanismes nécessaires pour les mettre en œuvre.

Ainsi nous avons proposé la méthode permettant la modélisation des processus de conception de produits manufacturés par les acteurs métier du domaine. Ceci en utilisant une approche basée sur la composition de services. La méthode de modélisation que nous avons proposée a été décrite grâce à un ensemble de méta-modèles dont l'objectif est de représenter une description uniforme des concepts manipulés (cf. section 3.3 et 3.4). Ces concepts décrivent l'organisation du processus de conception, ses entités principales (les orchestrations métier, les opérations de *SCPs*, les flots de contrôles, etc.).

Afin d'assurer l'alignement entre les différents niveaux de notre approche, nous avons présenté les méta-modèles permettant d'exprimer le processus de conception, défini au niveau métier, en termes des concepts du niveau fonctionnel et technique de notre approche. En outre, le chapitre a permis de présenter une démarche de déploiement des opérations métier des *SCPs* et des opérations fonctionnels PLM sur un système PLM spécifique. Cette démarche de déploiement possède un double avantage. D'une part, elle facilite la tâche de l'utilisateur métier tout au long du déploiement de son processus sur un système PLM. D'autre part, elle permet l'automatisation de la démarche d'alignement entre les différents modèles de notre approche. En effet, les deux catalogues de déploiement métier et fonctionnel constituent une source d'information réutilisable lors de l'exécution des règles de transformation. Ceci évite de faire appel à des personnes ayant des connaissances techniques afin de ré-implémenter l'alignement et le déploiement des opérations métier à chaque fois qu'il y a un besoin de changement. Enfin, notre démarche intègre la flexibilité au cours du choix du déploiement métier et fonctionnel (le déploiement d'une opération peut avoir plusieurs implémentations différentes du fait qu'il peut être implémenté par le patron ou encore par une orchestration définie par l'utilisateur).

Chapitre V :

Illustration de la démarche sur un cas d'application

1. Introduction

Le chapitre précédent nous a permis de présenter le cadre méthodologique (modèles et démarche) permettant d'utiliser les services identifiés. La démarche présentée permet la définition et la mise en œuvre des processus de conception de produits manufacturés sous forme d'orchestrations de services. Cette démarche est composée de trois phases. Une première phase de modélisation des processus de conception sous forme d'orchestrations métier. Cette première phase est suivie par les deux phases de déploiement métier et fonctionnel qui permettent de projeter le processus défini au niveau métier sur les couches de plus bas niveau de l'approche que nous proposons dans cette thèse.

L'objectif de ce chapitre est d'illustrer la démarche proposée à travers un cas d'application afin de valider sa faisabilité. Nous illustrons notre démarche à travers deux itérations. Une première itération permettant d'illustrer l'application de notre démarche lors de la définition et la mise en œuvre d'un processus de conception pour la première fois. La deuxième itération a pour objectif d'illustrer l'application de notre démarche lors d'une modification intervenant sur un processus de conception préalablement défini et mis en œuvre.

Ce chapitre est ainsi divisé en trois sections. La section 2 présente l'étude de cas que nous allons utiliser pour illustrer notre démarche. La Section 3 met en évidence la première itération d'application de la démarche d'usage de services sur l'exemple présenté dans la section 2. La section 4 permet d'illustrer l'application de la démarche d'usage de services en cas de modification du processus de conception pendant son déroulement (itération 2).

2. Présentation de l'étude de cas

L'étude de cas que nous avons choisie pour illustrer notre démarche d'usage de services consiste en une méthode de traitement des modifications utilisée lors du développement d'un produit complexe (avion) proposée par [Riviere 04]. Cette méthode reflète la démarche de résolution de problèmes habituellement mise en œuvre par les concepteurs pour répondre à une demande de modification survenant pendant le processus de développement d'un nouvel avion. Nous introduisons dans cette section les grands principes de la méthode ainsi que les phases qui la définissent.

Comme l'illustre la Figure 46 le traitement d'une modification est déclenché par l'apparition d'un fait technique. Un fait technique est ici considéré comme un problème soulevé par un acteur, impliqué dans le cycle de vie du produit, auquel une réponse (en réalité une solution technique analysée) doit être apportée. C'est pour cette raison qu'on parle de demande de modification. L'implantation physique de cette réponse sera considérée comme la clôture du processus de traitement d'une modification.

La méthode de traitement des modifications proposée par [Riviere 04] est composée de trois phases :

- Une phase d'analyse de problème
- Une phase de résolution de problème
- Une phase de définition de la solution retenue

Nous détaillons, dans ce qui suit, chacune de ces phases.

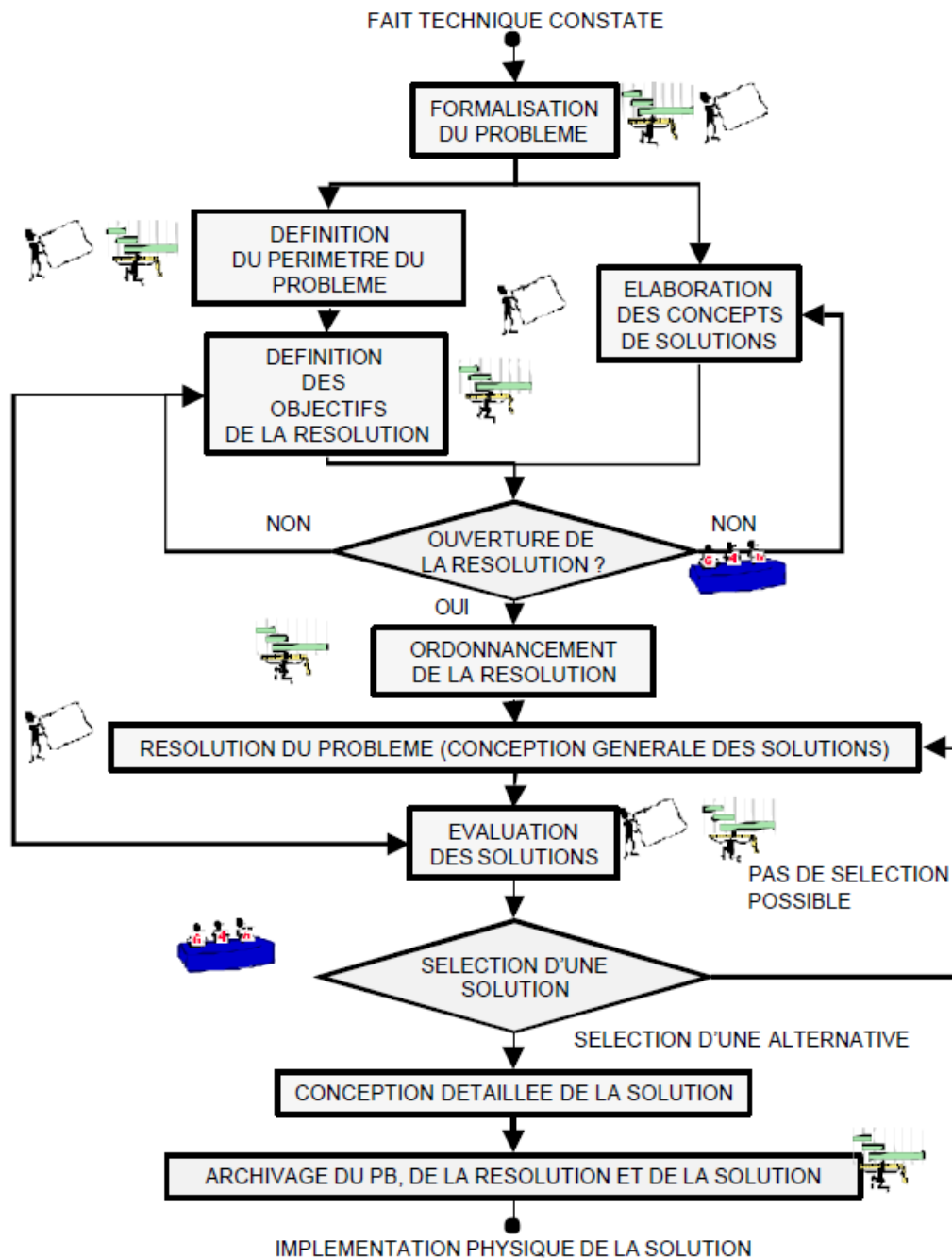


Figure 46 : Méthode de traitement des modifications [Riviere 04]

2.1. Définition et analyse du problème

A partir de la constatation d'un fait technique, le processus de définition et d'analyse du problème permet de formaliser une demande de modification, analyser et définir son périmètre, fixer les objectifs d'une résolution. En effet, un fait technique n'est généralement constaté que par un acteur ou un groupe restreint d'acteurs. La nature des informations issues de ce premier constat est très hétérogène, parfois précise, souvent floue. Il convient donc de formaliser le problème afin que d'autres acteurs puissent se l'approprier lorsqu'ils auront à intervenir dans le traitement. Au-delà de la formalisation du problème, la phase d'analyse permet de déterminer son périmètre, ses enjeux et ses possibles répercussions mais aussi de fixer les objectifs auxquels devra se conformer la résolution (le

processus de résolution mais aussi la solution qui en découlera). Cette phase comprend les étapes suivantes :

- formalisation du problème (de la prise en compte d'un fait technique à la création d'une modification) : cette étape vise à structurer le problème soumis par le découvreur d'un fait technique (un acteur impliqué ou ayant une incidence sur le cycle de vie du produit) afin que le traitement s'appuie sur une demande de modification formalisée. Cette structuration permet de caractériser le fait technique selon l'origine de la demande et de décrire le problème ;
- définition du périmètre du problème (définition de l'étendue de la modification) : cette étape vise à déterminer le périmètre de la modification en prenant en compte le phénomène de propagation, au travers d'une analyse d'impacts ;
- définition des objectifs de la résolution : une fois le périmètre de la demande cerné, il est nécessaire de définir les objectifs du processus de traitement qui vise à y apporter une solution technique. Ces objectifs sont déclinés selon deux axes (1) les objectifs liés au traitement lui-même, exprimés en termes de coût et de délais (objectifs projet), (2) les objectifs liés à la solution technique devant être développée en terme de performances à remplir, de contraintes techniques à respecter mais aussi en terme d'implémentation (objectifs produit/process). A l'issue de cette phase, les coûts et les délais de traitement sont fixés. A ce stade, ce ne sont que des informations prévisionnelles au-delà desquelles le traitement de la modification pourrait affecter le déroulement du projet de développement. A l'issue de cette phase, les objectifs du traitement ainsi que les spécifications de la solution à développer sont connus de l'ensemble des acteurs ;
- génération des premiers concepts de solution : généralement, le problème et la solution se construisent simultanément. Ainsi, lors de la formalisation et pendant l'analyse du problème, les premiers concepts de solutions sont élaborés par les acteurs impliqués. Généralement, la génération des premiers concepts s'appuie sur l'établissement d'analogies entre la demande de modification à traiter et des traitements ayant été réalisés dans le passé. A ce stade, les concepts ne sont que très peu structurés, seuls les plans de principes sont disponibles. Ils sont toutefois suffisants pour comprendre « l'esprit » de la solution et être utilisés comme objet intermédiaire afin que les acteurs confrontent leurs points de vue ;
- décision d'ouverture d'une modification : à ce stade du processus de traitement, il est important de s'interroger sur la suite à donner au traitement, c'est-à-dire si une solution technique doit être apportée comme réponse à la demande de modification initiale. La décision d'ouverture de la modification est prise sur la base des informations issues des phases précédentes, à savoir : la description du problème, l'analyse de ce problème et de ses possibles répercussions, la définition des objectifs et les premiers concepts des solutions techniques. Trois types de décisions peuvent être prises à l'issue de cette phase : une modification est ouverte, donc le traitement se poursuit et une solution technique sera développée puis introduite, aucune modification n'est ouverte et le traitement de la demande est abandonné, la décision d'ouverture est reportée car trop peu d'informations sont disponibles.

2.2. Résolution du problème

La seconde phase du processus de traitement des modifications correspond à la résolution du problème posé par la demande de modification à partir de l'analyse conduite lors des phases préliminaires du

traitement. Dans les faits, la résolution correspond au développement d'une solution technique. Cette phase comprend les étapes suivantes :

- l'ordonnancement du processus de conception des solutions (de la résolution) : permet de définir/ sélectionner à partir d'une bibliothèque de stratégies de processus, le processus de traitement à mettre en œuvre en fonction des caractéristiques de la demande de modification et du contexte dans lequel elle est émise. Une fois le « squelette » de processus adapté à la modification à traiter est disponible, un planning peut être établi et des ressources allouées à chacune des activités en vue de leur réalisation. A l'issue de cette étape, un processus adapté à la demande de modification auquel sont associés des plannings ainsi que des acteurs pour la réalisation de chacune des tâches est défini ;
- la conception générale des solutions (résolution du problème) : cette étape correspond à l'exécution du processus ordonnancé lors de la phase précédente. La résolution proprement dite du problème peut débiter sur la base des premiers concepts élaborés et validés dans la phase de définition et analyse du problème. Le résultat est donc la conception générale des différentes alternatives de solutions sur la base des concepts initialement proposés ainsi que leur documentation ;
- l'évaluation des solutions proposées : afin de répondre à la demande initiale, une solution technique doit être sélectionnée parmi les alternatives étudiées. Pour qu'une décision soit arrêtée, chaque alternative doit être évaluée par rapport aux autres mais aussi par rapport à la solution existante. Le résultat de cette étape est donc les analyses et résultats d'évaluations qui doivent être présentés de manière claire et synthétique afin de faire apparaître les avantages et inconvénients de chaque alternative ;
- la sélection de la solution à définir : à partir de la documentation associée à chacune des solutions alternatives conçues et sur la base des évaluations fournies dans l'étape précédente, une solution doit être sélectionnée. Cette solution sera ensuite définie, lors de la phase de définition de la solution, avant d'être implantée physiquement sur le produit.

2.3. Définition de la solution

Le but de la cette phase est de faire converger la solution sélectionnée, issue de la phase de conception générale vers la définition plus mature du produit. La définition d'une solution lors du traitement d'une modification est donc semblable à la définition initiale des solutions. Cette phase vise donc à produire la documentation de définition associée à la solution (création ou modification de la documentation existante). Cette phase comprend les étapes suivantes :

- Conception détaillée de la solution (aussi appelée phase de définition) : cette étape vise à établir la documentation permettant l'implémentation de la solution sélectionnée. C'est-à-dire sa fabrication et son assemblage. La phase de conception détaillée est intrinsèquement liée à la documentation du produit, qu'il s'agisse de sa création ou de sa mise à jour qui fait alors appel à des mécanismes de versionnement. Enfin, la solution et sa documentation doivent être validées. La documentation de définition est validée lorsqu'elle est jugée conforme aux exigences et aux spécifications établies par les acteurs concernés. C'est donc le processus de validation traditionnellement employé lors des phases de définitions initiales. Mais cette validation ne clôture pas le processus.

- Archivage du problème, du processus de résolution et de la solution : lors du traitement des modifications, les acteurs utilisent souvent des raisonnements par analogies à partir de cas déjà traités. Ces mécanismes permettent d'identifier les similarités entre des problèmes à traiter et ceux pour lesquels une solution a déjà été apportée. Lorsqu'une analogie est découverte, les acteurs peuvent réutiliser certaines informations mais aussi consolider leurs résultats. La sauvegarde d'un traitement est donc primordiale. S'il n'est pas réutilisé dans le futur pour établir des analogies, il permet de conserver le contexte de création d'une solution et sa logique de conception mais aussi d'assurer une conservation de l'historique du système remis en cause (exigence de traçabilité).

L'objectif des sections suivantes est d'appliquer, sur cette étude de cas, la démarche d'usage de services que nous avons proposé dans ce travail de thèse. Comme nous l'avons annoncé précédemment, cette application est menée en 2 itérations. A chacune des itérations, nous illustrerons les différentes phases de notre démarche et nous soulignerons l'intérêt de nos propositions par rapport à nos objectifs.

3. Application de la démarche d'usage de services pour la définition et la mise en oeuvre d'un processus – Itération 1

Nous illustrons dans cette section la phase de définition du processus sous forme d'orchestration de services de conception de produits manufacturés et la phase de déploiement métier. Notons toutefois que nous nous arrêtons à la deuxième phase de notre approche d'usage des services, à savoir la phase de déploiement métier. En effet, la troisième phase de cette approche, phase de déploiement fonctionnel, consiste en la projection de l'orchestration du niveau fonctionnel sur un système PLM particulier. Ainsi, elle nécessite d'avoir en possession le catalogue de services logiciels PLM propre au système PLM sur lequel le processus sera déployé (ce qui n'a pas été traité dans le cadre de la thèse).

3.1. Phase de modélisation du processus de conception

La première étape de la démarche d'usage, proposée dans le chapitre VI de ce manuscrit, consiste à élaborer le modèle du processus (de traitement des modifications) sous la forme d'une orchestration de services du niveau métier.

Rappelons que le concept central du modèle de processus est le service de conception de produits manufacturés (opération de service et fragments d'orchestration métier). La définition du processus de conception de produits manufacturés consiste ainsi à :

- découvrir, à partir du catalogue de services de conception de produits manufacturés, l'ensemble des services métier et les opérations de services qui répondent aux différents besoins du processus à modéliser. Des sous processus déjà définis et exposés sous forme d'orchestration à part entière (i.e fragments d'orchestration) peuvent aussi participer à la définition du processus final.
- composer les fragments d'orchestrations métier et/ou les opérations de services métier choisies, en spécifiant l'ensemble des flux de contrôles qui les relient.

Nous commençons d'abord par le choix des orchestrations métier pré-définies qui peuvent participer à la définition du processus de traitement des modifications (cf. Tableau 29). En effet, tout processus défini sous forme d'orchestration métier est sauvegardé. Ces orchestrations métier peuvent ainsi être réutilisées pour participer à la définition d'autres processus. Elles seront adaptées au cas d'usage au moment du choix du déploiement métier (choix des opérations fonctionnelles PLM nécessaires).

Tableau 29: Fragments d'orchestration métier composant le processus de traitement de modification

Numéro Orchestration Métier	Nom Opération / Orchestration Métier
OM1	DefinitionSolution
OM2	DefinitionPerimetreEtObjectifDuProbleme
OM3	OrdonnancementDuProcessusDeConception
OM4	AnalyseFonctionnelle
OM5	AnalyseDysfonctionnelle
OM6	AnalyseTest
OM7	EvaluationDuneSolution
OM8	ValidationDuneSolution
OM9	ConceptionDuneSolution

L'orchestration métier OM1 est une orchestration permettant la définition d'un principe de solution (cf. Figure 47). Elle permet de répondre aux objectifs de l'étape « génération des premiers concepts de solution » de la première phase du processus de traitement des modification : la « définition et analyse du problème » (cf. Section 2.1).

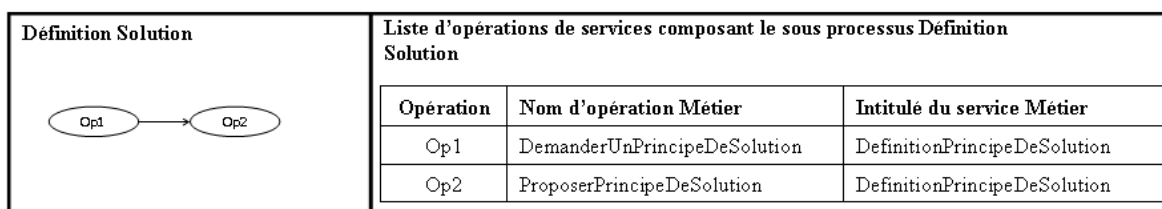


Figure 47 : Fragment d'orchestration métier «DéfinitionSolution»

L'orchestration métier OM2 est un fragment d'orchestration permettant la définition du périmètre et de l'objectif du problème de modification (cf. Figure 48). Elle permet de répondre aux objectifs des deux étapes «définition du périmètre du problème» et «définition des objectifs de la résolution» de la première phase du processus de traitement de modification ; « définition et analyse du problème » (cf. Section 2.1). Cette orchestration métier fait appel à deux sous processus : Définition Devis et Définition Spécification Technique (cf. Figure 49 et Figure 50)

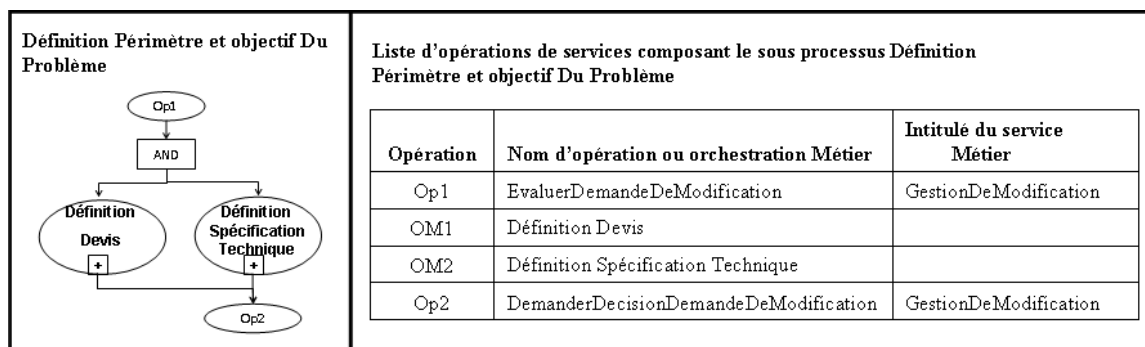


Figure 48 : Fragment d'orchestration métier «DéfinitionPérimètreEtObjectifDuProblème»

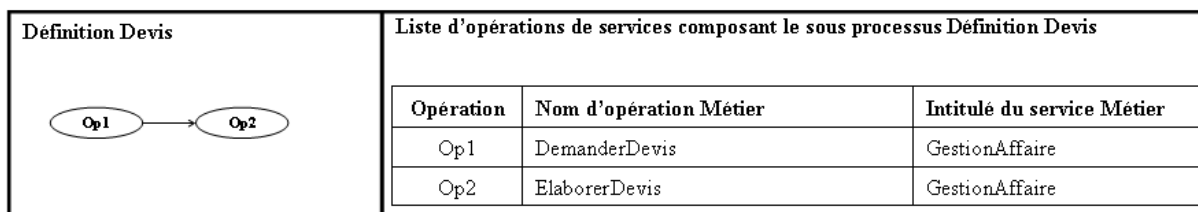


Figure 49 : Fragment d'orchestration «DéfinitionDevis»

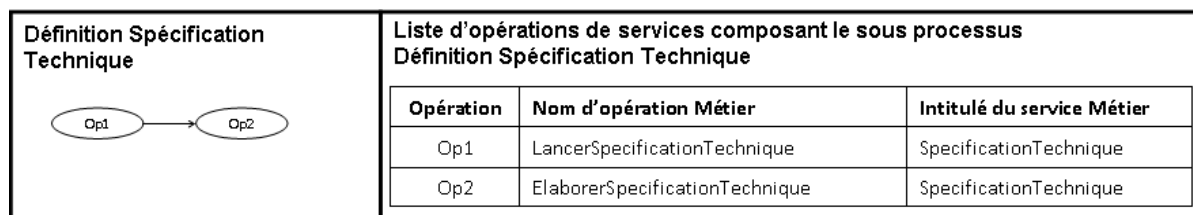


Figure 50 : Fragment d'orchestration «DéfinitionSpécificationTechnique»

Pour mettre en œuvre l'étape «Ordonnancement du processus de conception des solutions» de la deuxième phase du processus de traitement de modification, la « résolution du problème » (cf. section 2.2), nous choisissons le fragment d'orchestration «Ordonnancement du processus de conception» (cf. Figure 51). Cette orchestration métier permet de planifier la conception suite à une validation d'une demande de modification.

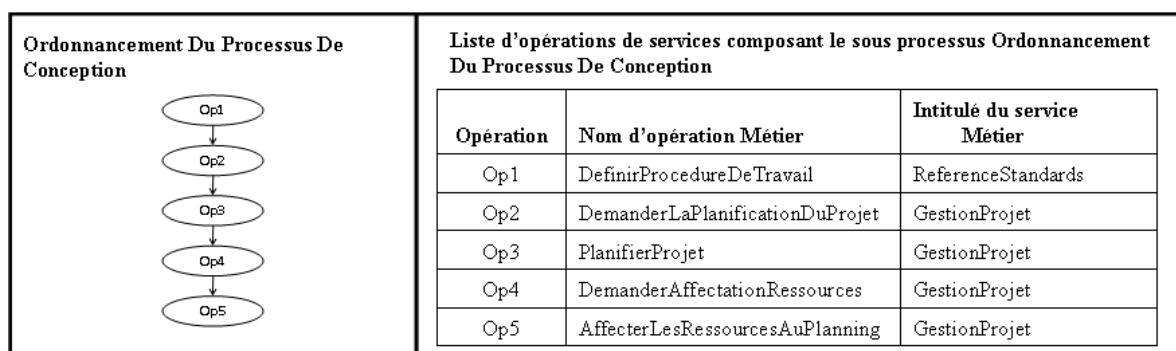


Figure 51: Fragment d'orchestration «Ordonnancement du Processus De conception»

Afin de réaliser la conception générale suite à l'ordonnancement, plusieurs fragments d'orchestrations métier peuvent être réutilisés. Les Figure 52, Figure 53 et Figure 54 présentent les fragments choisis pour répondre aux attentes de l'étape «Conception générale des solutions» de la deuxième phase du processus de traitement de modification ; la « résolution du problème » (cf. section 2.2).

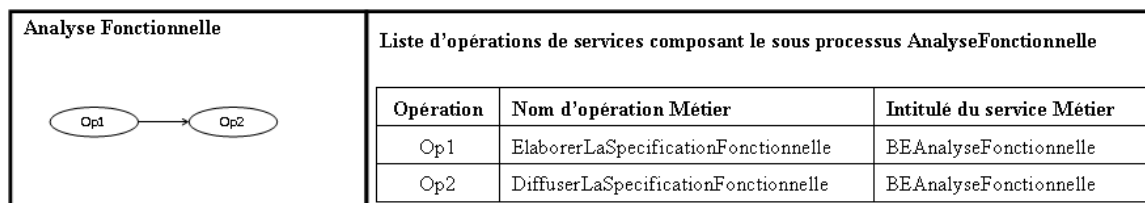


Figure 52 : Fragment d'orchestration «AnalyseFonctionnelle»

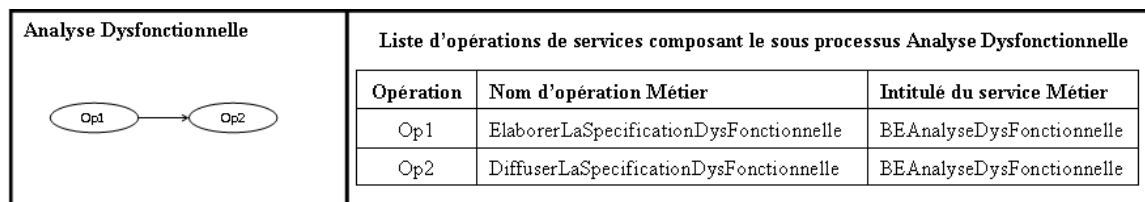


Figure 53 : Fragment d'orchestration «AnalyseDysfonctionnelle»

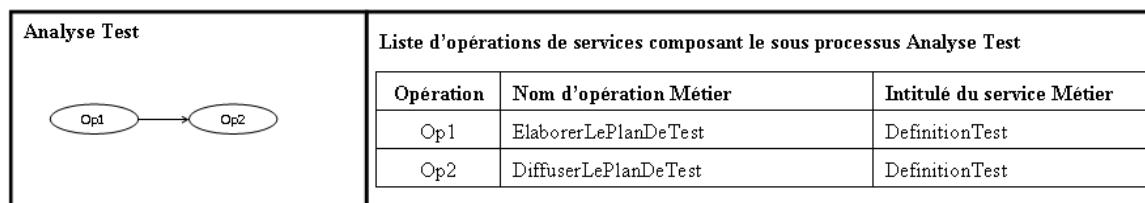


Figure 54 : Fragment d'orchestration «AnalyseTest»

Par la suite et afin qu'une décision puisse être arrêtée, chaque alternative doit être évaluée. Ainsi, nous choisissons de réutiliser le fragment d'orchestration « Évaluation d'une solution » (cf. Figure 55) pour remplir l'étape « évaluation des solutions proposées » de la deuxième phase du processus de traitement de modification ; la « résolution du problème » (cf. section 2.2).

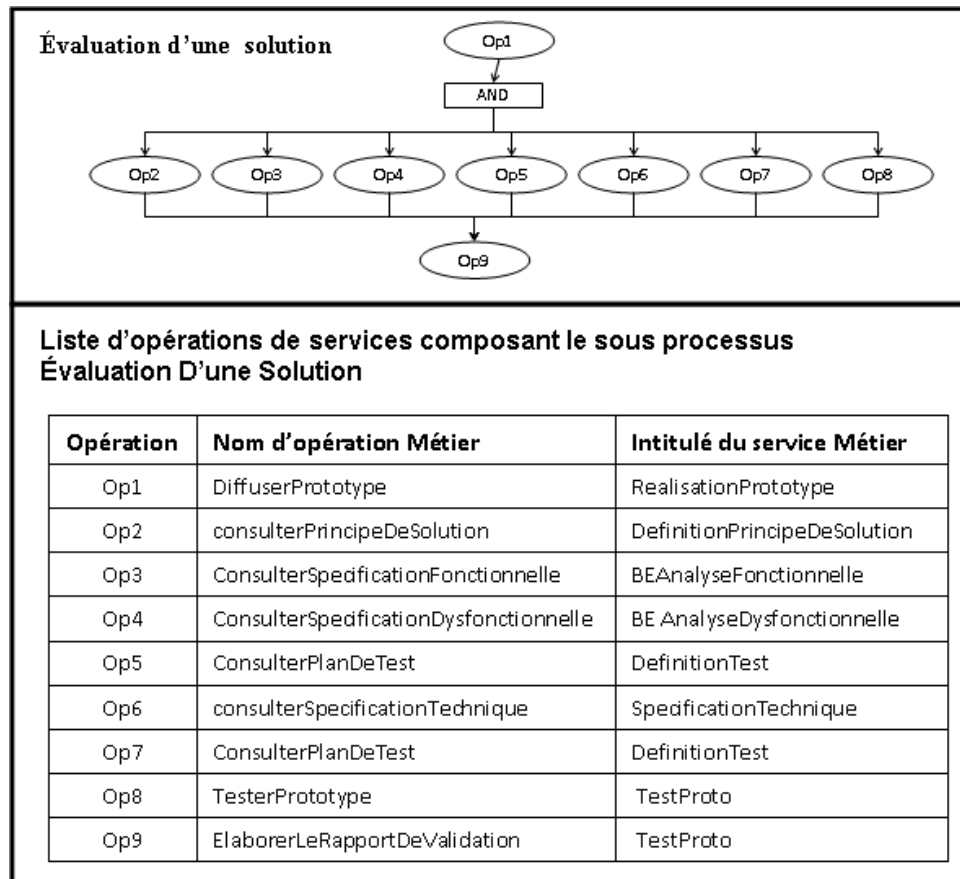


Figure 55 : Fragment d'orchestration «ÉvaluationDuneSolution»

La dernière étape de la deuxième phase du processus de traitement de modification, « résolution du problème » (cf. section 2.2), est celle qui a pour objectif la « sélection de la solution à définir ». Pour remplir cette tâche, il faut d'abord valider les solutions proposées. Pour ce faire, nous utilisons le fragment d'orchestration « Validation d'une solution » (cf. Figure 56).

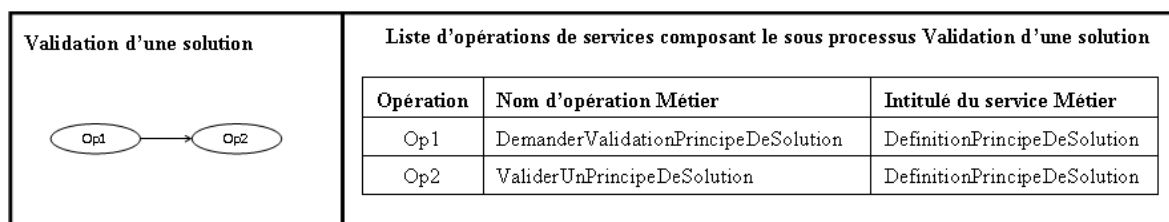


Figure 56 : Fragment d'orchestration «ValidationDuneSolution»

Une fois une solution est choisie, la phase « définition de la solution » (cf. 2.3) du processus de traitement des modification consiste a établir la documentation permettant l'implémentation de la solution sélectionnée. Pour ce faire, nous utilisons le fragment d'orchestration « Conception d'une Solution » (cf. Figure 57).

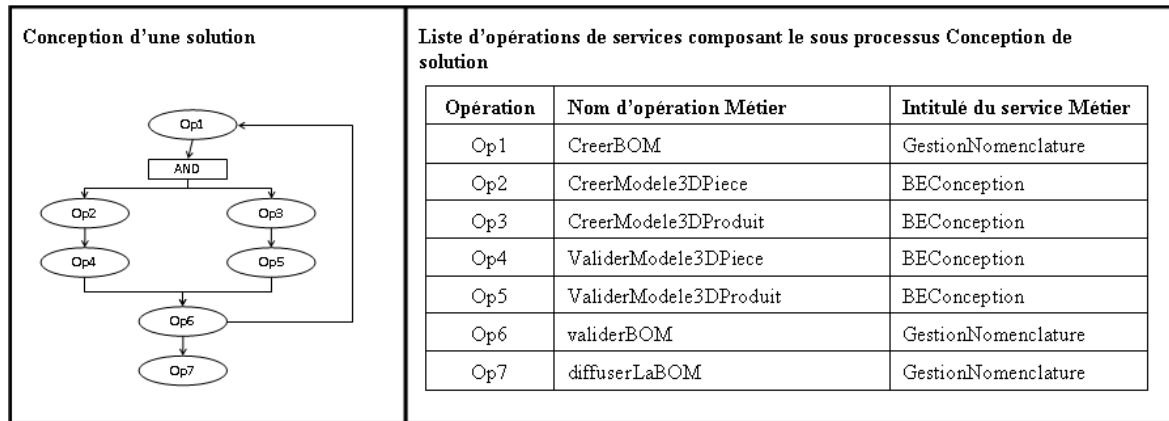


Figure 57 : Fragment d'orchestration «ConceptionDuneSolution»

Pour définir le processus de traitement des modifications, nous avons également besoin d'autres opérations métier. Le Tableau 30 énumère celles-ci.

Tableau 30: Opérations de services de conception de produits manufacturés composant le processus de traitement de modification

Numéro Opération	Nom Opération Métier	Nom du Service
Op1	<i>elaborerLaDemandeDeModification</i>	<i>GestionDeModification</i>
Op2	<i>demanderEvaluationdeLaDemandeDeModification</i>	<i>GestionDeModification</i>
Op3	<i>validerDemandeDeModification</i>	<i>GestionDeModification</i>
Op4	<i>diffuserLePlanningDuProjet</i>	<i>GestionProjet</i>
Op5	<i>creerPrototype</i>	<i>RealisationPrototype</i>
Op6	<i>diffuserPrincipeDeSolution</i>	<i>DefinitionPrincipeDeSolution</i>

Le modèle du processus du traitement de modifications illustré dans la Figure 58 décrit l'orchestration métier finale qui définit l'ensemble des flux de contrôles reliant les fragments d'orchestrations métier et les opérations de services métier choisies.

L'application présentée dans cette section montre que notre proposition respecte les contraintes spécifiques aux processus du domaine de la conception de produits manufacturés (cf. section 2.3 du chapitre I) et répond aux objectifs de dynamisme et d'agilité que nous nous sommes fixés dans ce travail de thèse et d'agilité que nous nous sommes fixés dans ce travail de thèse. Comme nous l'avons précisé dans le premier chapitre, les processus de conception de produits manufacturés sont tendus. La concurrence du marché induit une forte pression pour pouvoir mettre des produits sur le marché dans les plus brefs délais et avec des coûts réduits. L'adaptation des services métiers que nous avons proposés au cas du processus traitement de modification nous laisse dire que les services métiers proposés sont génériques et réutilisables pour la définition de divers types de processus du domaine de la conception de produits. La démarche proposée permet de réduire le temps de définition du processus en réutilisant des traitements répétitifs. Elle permet aussi de définir des processus complexes en masquant la complexité des processus composites. Ceci a travers la réutilisation de fragments d'orchestrations prêts à l'emploi tels que les fragments conception d'une solution, validation d'une solution, etc. que nous avons utilisée pour la définition du processus de traitement des modifications.

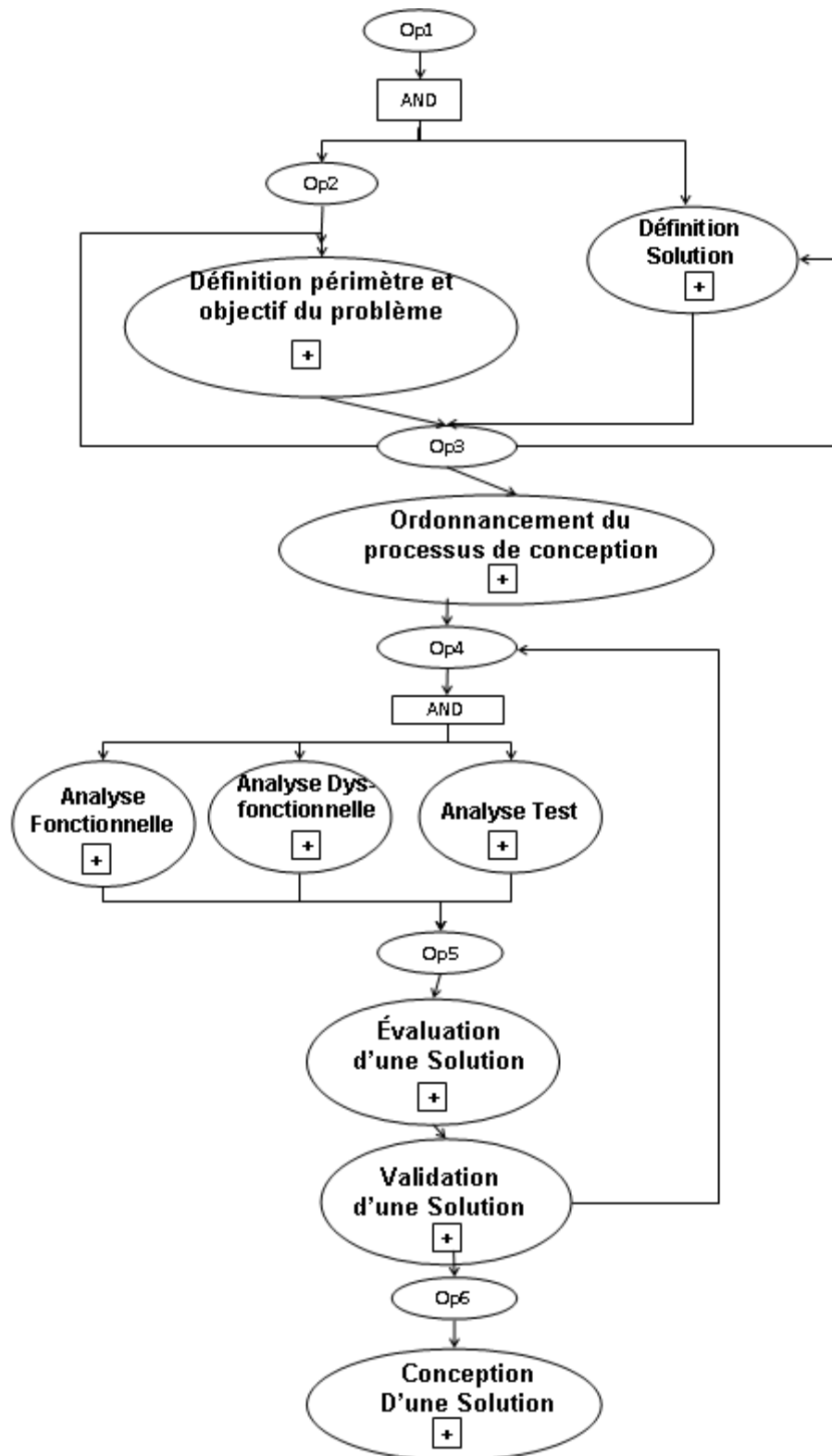


Figure 58 : Processus de traitement des modifications

L'étape suivante consiste à déployer le processus modélisé. Ceci consiste à choisir pour chaque opération métier composant le processus et ses fragments d'orchestrations métier, le patron ou une des orchestrations fonctionnelles prédéfinies dans le déploiement métier de l'opération. Il est possible également de définir une nouvelle orchestration fonctionnelle permettant de mettre en œuvre l'opération métier en question. C'est l'objet de la section suivante de mettre en œuvre ce déploiement.

3.2. Phase de déploiement métier

Comme défini dans le chapitre IV, le déploiement métier correspond au passage du niveau métier au niveau fonctionnel. La première phase de la mise en œuvre du processus défini dans la section 3.1 consiste donc en la transformation du modèle d'orchestration du niveau métier en un modèle d'orchestration du niveau fonctionnel, qui joue le rôle de relais entre le domaine métier et le logiciel PLM utilisé.

Nous proposons de commencer d'abord par appliquer la démarche générique de déploiement que nous avons proposée pour l'utilisateur de notre approche afin de choisir les opérations et orchestrations fonctionnelles PLM permettant la mise en œuvre des opérations métier composant le processus de traitement des modifications. Par la suite, nous réutilisons les règles de transformations, définies dans le chapitre IV, afin d'assurer l'alignement entre le niveau métier et le niveau fonctionnel de notre approche.

3.2.1 Correspondance Métier / Fonctionnel

Afin d'effectuer le déploiement métier, le concepteur du processus doit consulter la fiche de correspondance Métier/Fonctionnel pour chacune des opérations métier composant le processus de traitement des modifications (cf. Tableau 30) et ceux composant les fragments d'orchestrations participant à ce processus. Ainsi, il pourra choisir d'utiliser le patron ou une orchestration fonctionnelle alternative ou bien de définir une nouvelle orchestration fonctionnelle qui répond à son contexte en utilisant la liste des opérations fonctionnelles PLM et les fragments d'orchestrations prédéfinies contenues dans la fiche de déploiement.

Ci-dessous nous montrons un exemple de fiche qui définit les opérations fonctionnelles PLM qui peuvent participer au déploiement de l'opération métier *ElaborerDemandeDeModification* (cf. Figure 59). Comme nous pouvons le voir, le déploiement métier de l'opération métier *ElaborerDemandeDeModification* propose plusieurs orchestrations fonctionnelles alternatives. D'une part, la création de la demande de modification peut être faite à partir d'un modèle de demande ou bien à partir d'un document vierge. Pour cela nous trouvons différents fragments d'orchestrations prédéfinies permettant la création de la demande de modification mais selon une façon donnée. D'autre part, la demande de modification peut être liée à un fait technique qui est lié à un produit ou bien à un article. Nous trouvons ainsi des versions d'orchestrations fonctionnelles permettant de consulter un fait technique lié au produit et de lier la demande au produit une fois qu'elle est faite. De même, nous trouvons des versions d'orchestrations fonctionnelles permettant de consulter un fait technique lié à un article et de lier la demande à l'article en question. Nous considérons que le fait technique, objet de la demande de modification, est lié à un produit et nous choisissons le patron comme orchestration fonctionnelle PLM pour mettre en œuvre l'opération métier *ElaborerDemandeDeModification*.

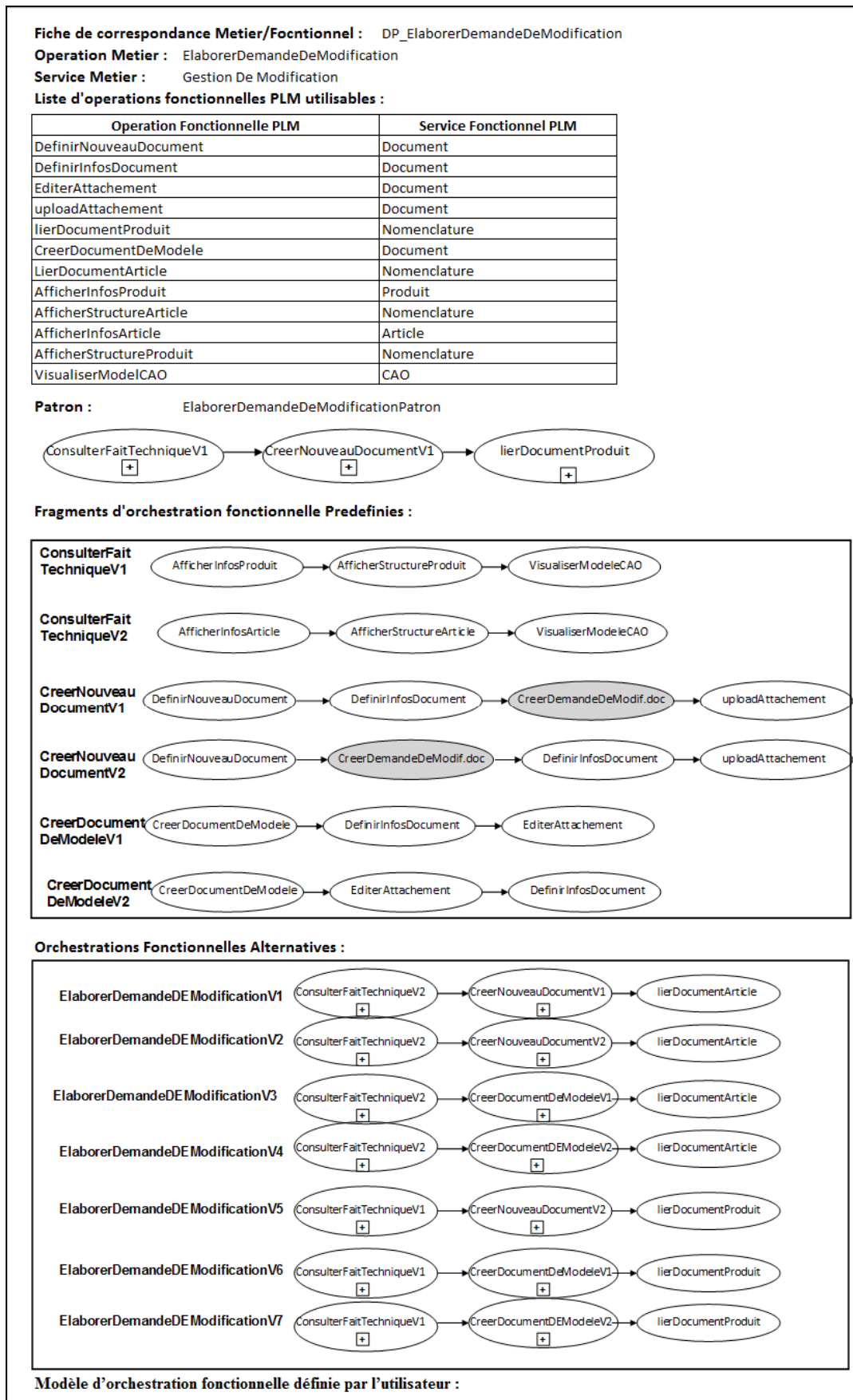


Figure 59 : Fiche de correspondance métier/fonctionnel de l'opération ElaborerDemandeDeModification

La Figure 60 montre un deuxième exemple de fiche qui détaille les différentes possibilités de déploiement de l'opération métier *DemanderEvaluationDemandeDeModification*. Pour cette opération métier nous choisissons d'utiliser l'orchestration fonctionnelle alternative *DemanderEvaluationDemandeDeModificationV1*.

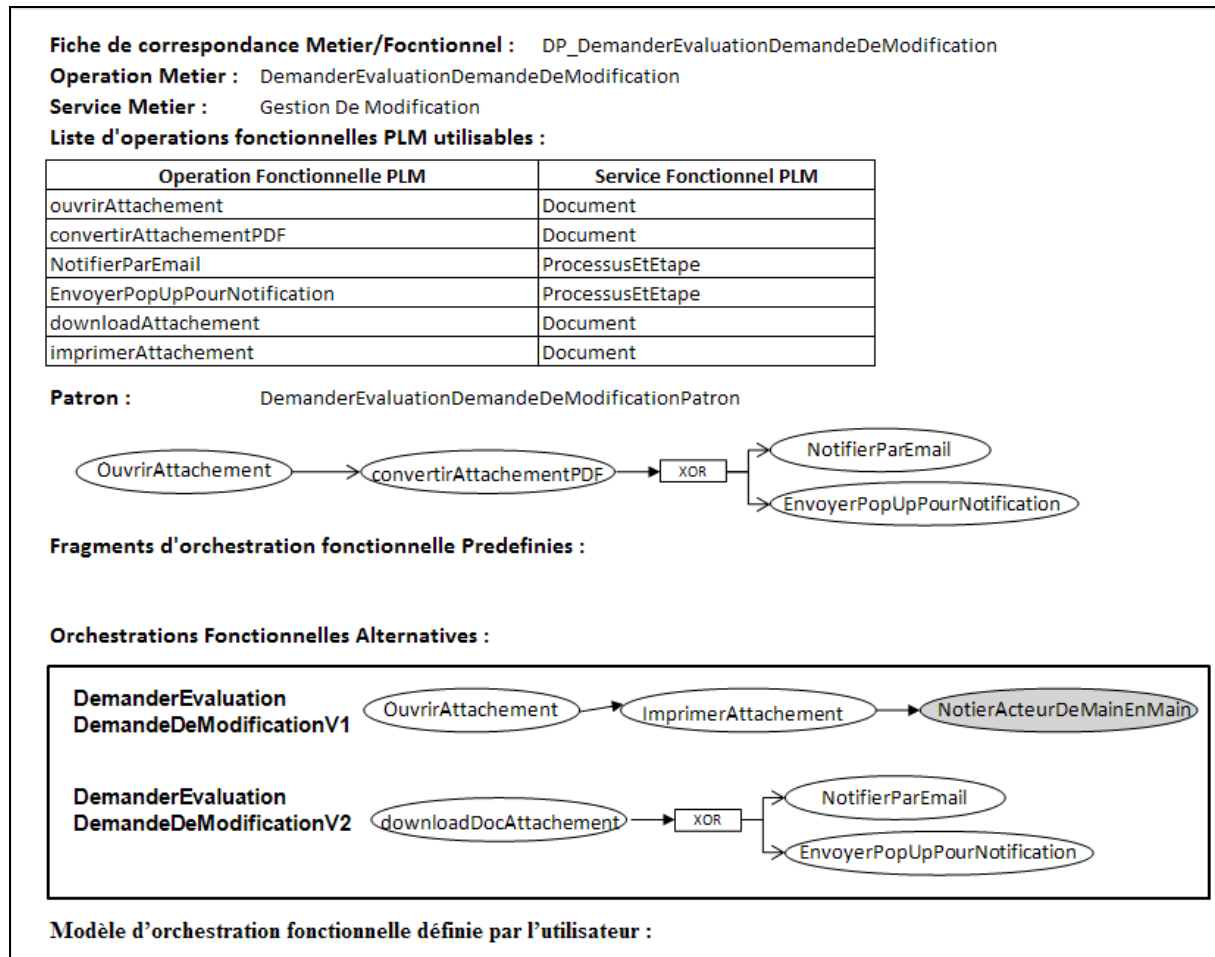


Figure 60 : Fiche de correspondance métier/fonctionnel de l'opération DemanderEvaluationDemandeDeModification

Pour terminer cette étape du déploiement métier nous adoptons la même démarche pour toutes les opérations métier composant le processus de traitements des modifications. Afin de ne pas alourdir cette section, nous présentons les fiches de correspondance des différentes opérations participant au processus dans l'Annexe B.

Nous pouvons dire que cette première étape de la démarche de déploiement répond en partie à un autre objectif de notre travail, à savoir la flexibilité au niveau fonctionnel. En effet, elle intègre la flexibilité au cours du choix du déploiement. Le déploiement d'une opération peut avoir plusieurs implémentations différentes du fait qu'il peut être implémenté par le patron, les orchestrations fonctionnelles alternatives ou encore une orchestration définie par l'utilisateur en combinant les fragments d'orchestrations prédéfinies et/ou les opérations fonctionnelles PLM listées dans la fiche. Cette démarche facilite aussi la tâche de l'utilisateur métier. L'utilisateur n'aura qu'à choisir ou définir visuellement l'orchestration fonctionnelle permettant la mise en œuvre de l'opération métier et l'étape suivante (exécution des règles de transformation) assure l'automatisation de cet alignement.

3.2.2 Alignement entre le niveau métier et le niveau fonctionnel

Une fois toutes les orchestrations fonctionnelles PLM, permettant la mise en œuvre des opérations métier composant le processus de traitement des modifications sont choisies (étape de correspondance Métier/Fonctionnel), l'alignement entre les niveaux Métier et Fonctionnel peut être effectué, à travers la réutilisation des règles de transformations, définies dans le chapitre IV. Nous illustrons dans ce qui suit la transformation du modèle de processus de traitement des modifications, défini sous forme d'orchestration métier, en un modèle d'orchestration fonctionnelle.

Le modèle du processus du traitement de modifications illustré dans la Figure 61 décrit l'orchestration fonctionnelle résultat de l'application des règles de transformations R0, R1 et R2 de la transformation T1 (cf. chapitre IV). Elle définit l'ensemble des flux de contrôles reliant les fragments d'orchestrations fonctionnelles implémentant les opérations de services métier et les fragments d'orchestration métier du processus de traitement des modifications.

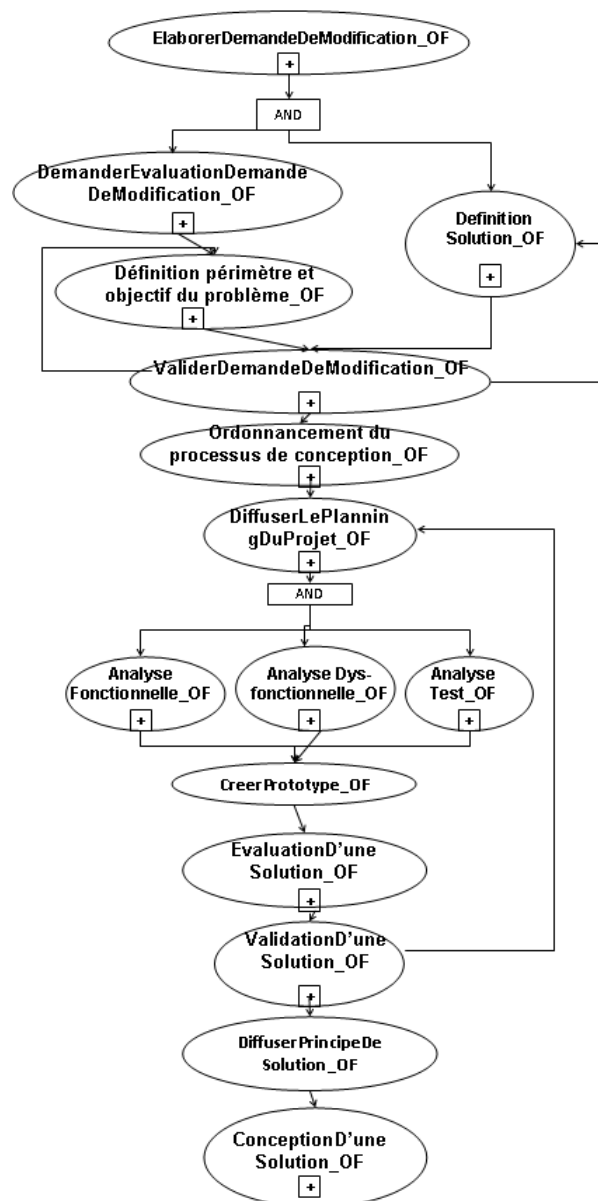


Figure 61 : Orchestration fonctionnelle : déploiement métier du processus de traitement des modifications

L'application de la règle R0 sur l'orchestration métier *processus de traitement des modifications* (cf. Figure 58) crée l'orchestration fonctionnelle *déploiement métier du processus de traitement des modifications* et fait appel à la règle R1 qui s'occupe du déploiement des opérations métier (cf. Tableau 30, règle R1.2) et des fragments d'orchestrations métier (cf. Tableau 29, règle R1.1) composant le processus de traitement des modifications. La règle R1 permet aussi la création des correspondants de l'ensemble des flux de contrôles (R1.3) reliant les fragments d'orchestrations métier et les opérations métier du processus de traitement des modifications. Puis à travers l'exécution de la règle R2.1 (si le flux de contrôle créé est de type séquence) ou bien la règle R2.2 (si le flux de contrôle créé est de type AND, OR ou XOR), les orchestrations fonctionnelles (créées suite à l'exécution des règles R1.1 et R1.2) sont reliées par les flots de contrôles (créés suite à l'exécution de la règle R1.3) et orchestrées conformément au modèle d'orchestration *du processus de traitement des modifications*. Nous illustrons dans ce qui suit quelques exemples d'application des règles R1.1 et R1.2.

La Figure 62 illustre la transformation du sous processus métier (*i.e* orchestration métier) *DefinitionSolution* (cf. Figure 58) en une orchestration fonctionnelle *DefinitionSolution_OF*. Cette orchestration fonctionnelle est issue de l'exécution de la règle R1.1 (présentée dans le tableau 22 du chapitre IV). En effet, cette règle assure l'alignement de toutes les orchestrations métier de l'orchestration « mère » *Traitement des modifications*. Le premier fragment de processus à aligner est l'orchestration métier *DefinitionSolution*, qui orchestre deux opérations métier *DemanderUnPrincipeDESolution* et *ProposerPrincipeDeSolution* (cf. Figure 46). La règle R1.1 transforme ainsi cette orchestration métier en une orchestration fonctionnelle (*i.e* création de l'OF *DefinitionSolution_OF*) et fait de nouveau appel à la règle R1 pour transformer les opérations métier composant cette orchestration. La règle R1.2 est ainsi exécutée pour créer deux nouvelles orchestrations fonctionnelles : *DemanderUnPrincipeDESolution_OF* et *ProposerPrincipeDeSolution_OF*, les associer à leur orchestration mère *DefinitionSolution_OF* et leurs affecter les déploiements respectifs des deux opérations métier *DemanderUnPrincipeDESolution* et *ProposerPrincipeDeSolution* (cf. Annexe B). La règle R1.3 est ensuite exécutée pour créer le flot de contrôle de type séquence qui lie ces deux orchestrations fonctionnelles ainsi créées et fait appel à la règle R2.1 pour affecter la source (*DemanderUnPrincipeDESolution_OF*) et la cible (*ProposerPrincipeDeSolution_OF*) du flot de contrôle de type séquence.

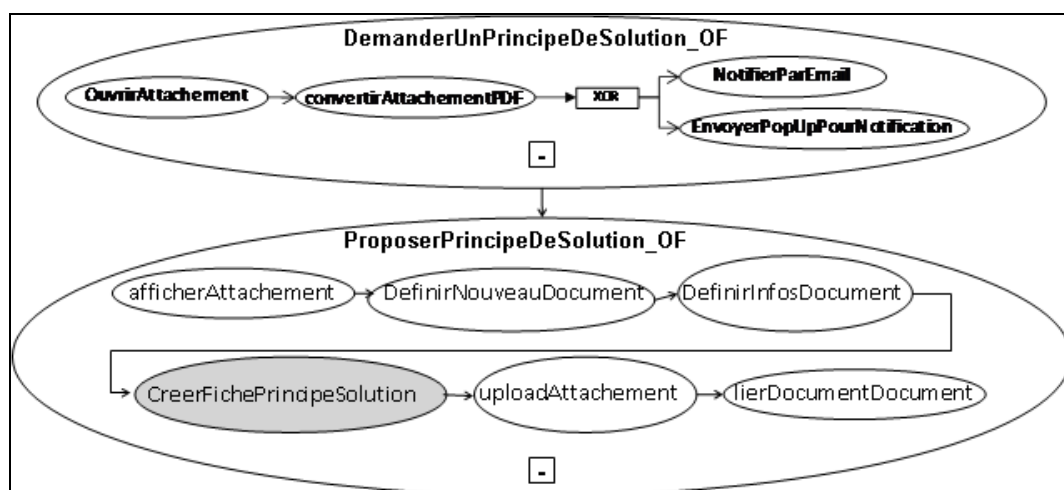


Figure 62 : Orchestration Fonctionnelle DefinitionSolution_OF

A travers cette illustration d'alignement les règles de transformation que nous avons proposées dans le chapitre IV de ce manuscrit semblent être génériques et applicables pour divers processus de conception définis sous forme d'orchestration métier. L'automatisation de ces règles permettra l'automatisation de la démarche d'alignement entre les différents modèles de notre approche. Comme nous avons pu le voir, le catalogue de déploiement métier (contenant les fiches de correspondance Métier/Fonctionnel) constitue aussi une source d'information réutilisable lors de l'exécution des règles de transformation. En effet, comme nous l'avons expliqué plus haut, pour aligner une opération métier au niveau fonctionnel, une orchestration fonctionnelle est créée et le déploiement choisi à la première étape de la phase de déploiement métier est affecté à cette orchestration. Ceci permet de répondre à un autre objectif de notre démarche, à savoir l'agilité de mise en œuvre du processus et son automatisation. Notons aussi que cette démarche masque la complexité d'implémentation des opérations constituant le processus, ce qui permettra d'éviter de faire appel à des personnes ayant des connaissances techniques et de réduire par conséquent les délais de mise en œuvre du processus.

3.3. Phase de déploiement fonctionnel

Pour pouvoir être exécuté sur un système PLM particulier, les opérations fonctionnelles PLM qui composent l'orchestration fonctionnelle définie dans la section 3.2 doivent être supportées, à leur tour, via l'appel aux opérations de services logiciels PLM. Comme nous l'avons mentionné dans le chapitre III, les services logiciels PLM sont ceux qui implémentent totalement ou partiellement les services fonctionnels PLM dans un système PLM particulier. Ainsi, pour une exécution du processus sur un système PLM donné, la transformation du modèle d'orchestration du niveau fonctionnel en un modèle d'orchestration du niveau technique nécessite d'avoir en possession les services logiciels propres au système PLM en question. Nous n'illustrons pas les étapes de cette phase car nous n'avons pas défini un catalogue de services logiciels PLM propre à un système PLM particulier.

4. Application de la démarche d'usage de services lors du changement d'un processus déjà défini et mis en œuvre – Itération 2

Cette section permet d'illustrer la mise en œuvre d'une modification survenant sur le processus défini dans la section précédente. Nous supposons qu'il y a eu un changement dans la façon de traiter les modifications. Ce changement concerne la deuxième phase du processus de traitement de modification, à savoir la résolution du problème (cf. section 2.2). Précisément il concerne l'étape « Conception générale des solutions » de cette phase. Dans un souci d'amélioration de l'étape de la conception générale, le choix de composants standards qui peuvent faire partie de la solution doit être fait à ce stade pour qu'il puisse être validé lors du choix de la solution finale.

Le changement consiste ainsi à rajouter un fragment d'orchestration métier « Choix de composants standards » à l'orchestration métier Traitement des modifications (cf. Figure 58). Ce fragment d'orchestration sera composé parallèlement aux fragments d'orchestrations métier « AnalyseFonctionnelle », « AnalyseDysfonctionnelle » et « AnalyseTest » (cf. Figure 63).

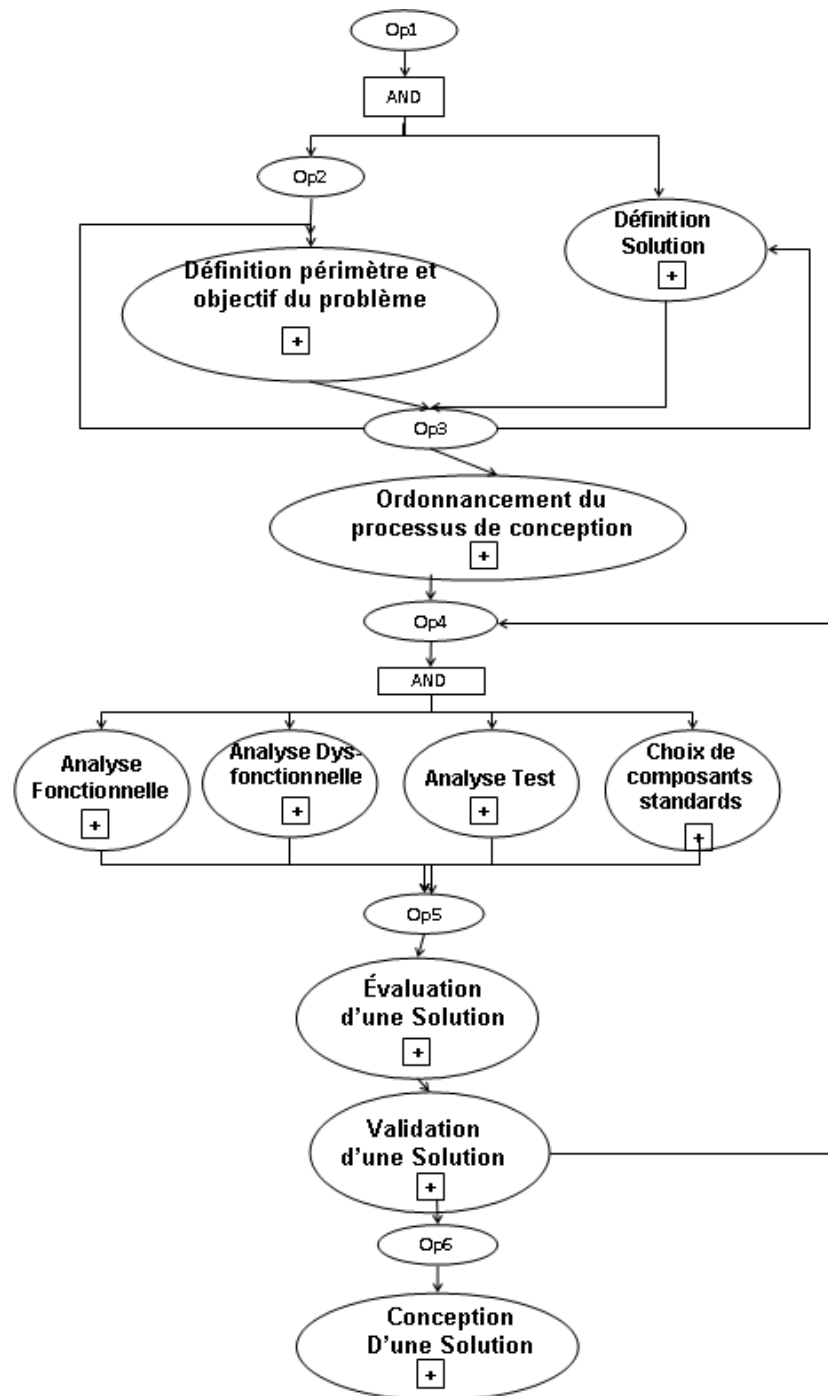


Figure 63 : Changement dans le processus de traitement des modifications

La figure Figure 64 présente le contenu du fragment rajouté « Choix de composants standards ».

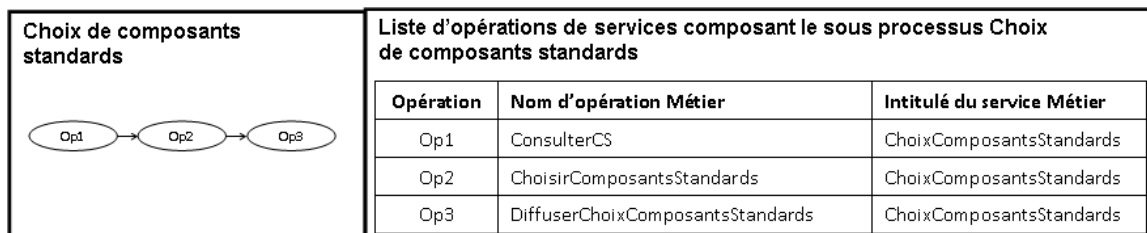


Figure 64 : Fragment d'orchestration « ChoixDeComposantsStandards »

Pour mettre en œuvre ce changement, la première étape consiste à ajouter le nouveau fragment de processus au processus de traitement des modifications (cf. Figure 63). Par la suite, le concepteur devra choisir le déploiement métier des opérations métier constituant ce nouveau fragment (cf. Figure 65, Figure 66 et Figure 67).

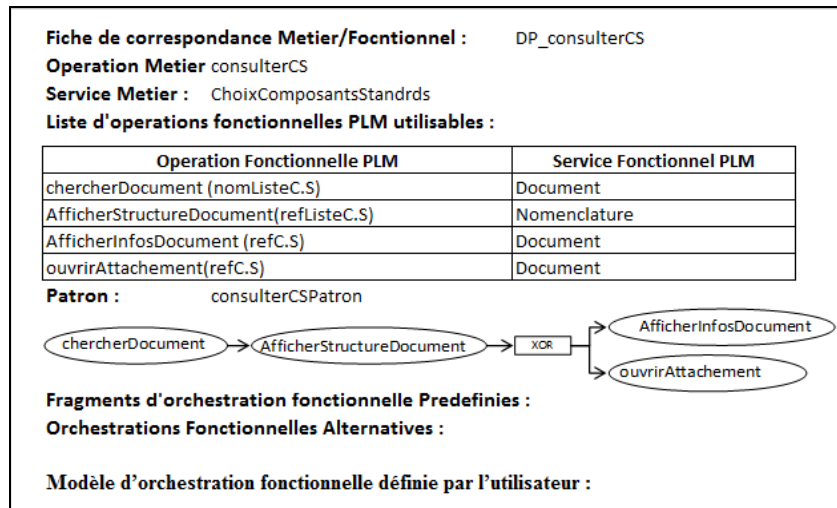


Figure 65 : Fiche de correspondance métier/fonctionnel de l'opération ConsulterCS

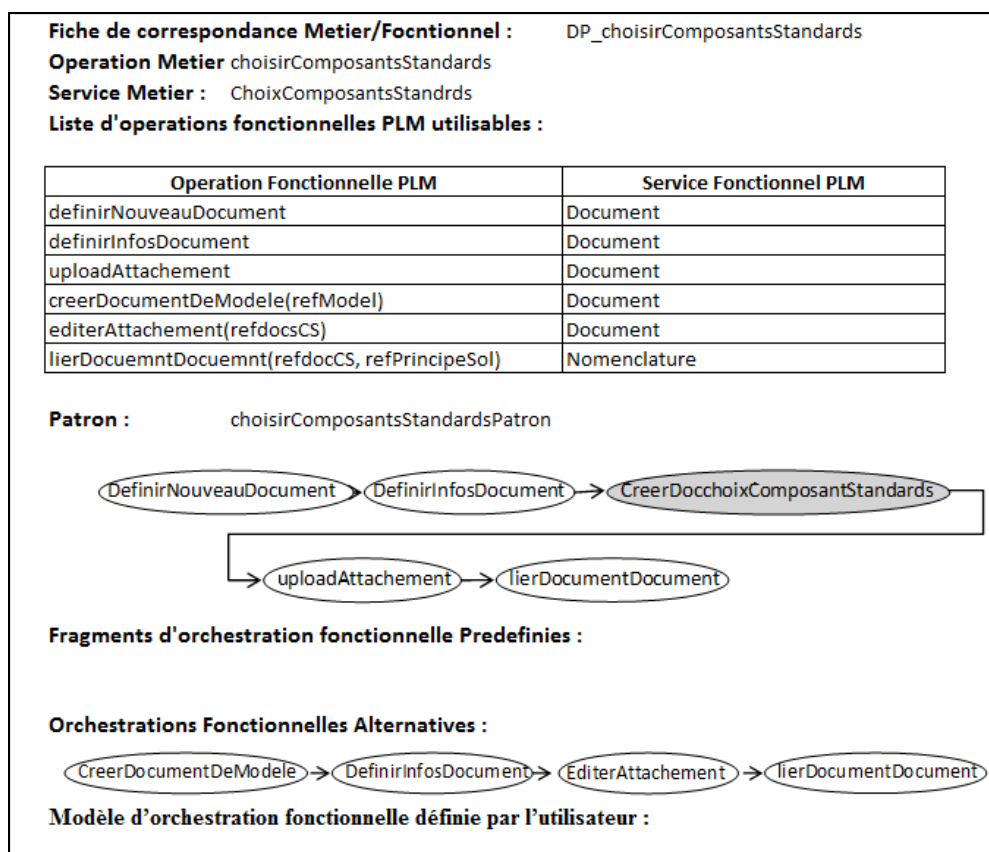


Figure 66 : Fiche correspondance métier/fonctionnel de l'opération choisirComposantsStandards

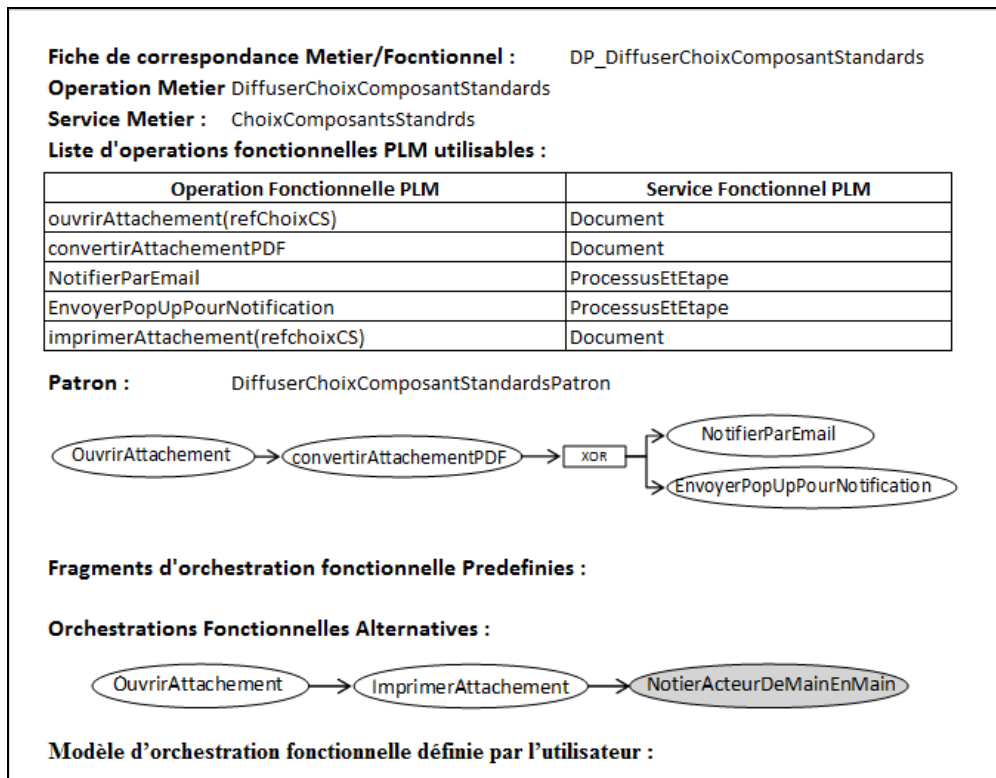


Figure 67 : Fiche correspondance métier/Fonctionnel de l'opération diffuserChoixComposantsStandards

Une fois le choix des orchestrations fonctionnelles permettant la mise en œuvre des opérations métier est fait, l'utilisateur de la démarche (responsable PLM ou concepteur) devra lancer l'exécution de la règle de transformation R1 avec comme entrée le fragment d'orchestration métier « ChoixDeComposantsStandards » (cf. Figure 64). Le résultat de cette transformation crée l'orchestration fonctionnelle « ChoixdeComposantsStandard_OF » (cf. Figure 68).

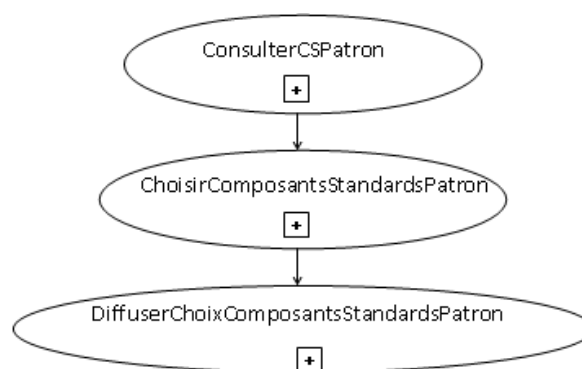


Figure 68 : Orchestration fonctionnelle ChoixdeComposantsStandard_OF

Enfin, les deux nouveaux fragments d'orchestrations métier (ChoixdeComposantsStandard) et fonctionnelle (ChoixdeComposantsStandard_OF) peuvent être rajoutés respectivement aux orchestrations métier (traitement des modifications) et fonctionnelle (déploiement métier du processus de traitement des modifications). En effet, comme nous l'avons dit dans la section 4 du chapitre II, nous optons pour l'utilisation des langages d'orchestration qui supportent l'adaptation dynamique non planifiée des orchestrations de services tels que PXL [Verjus 11] et π -Diapason [Pourraz 08]. En utilisant ce type de langage, le fragment d'orchestration fonctionnelle

« ChoixdeComposantsStandard_OF » peut être substitué au déploiement du processus de traitement des modifications (cf. Figure 69).

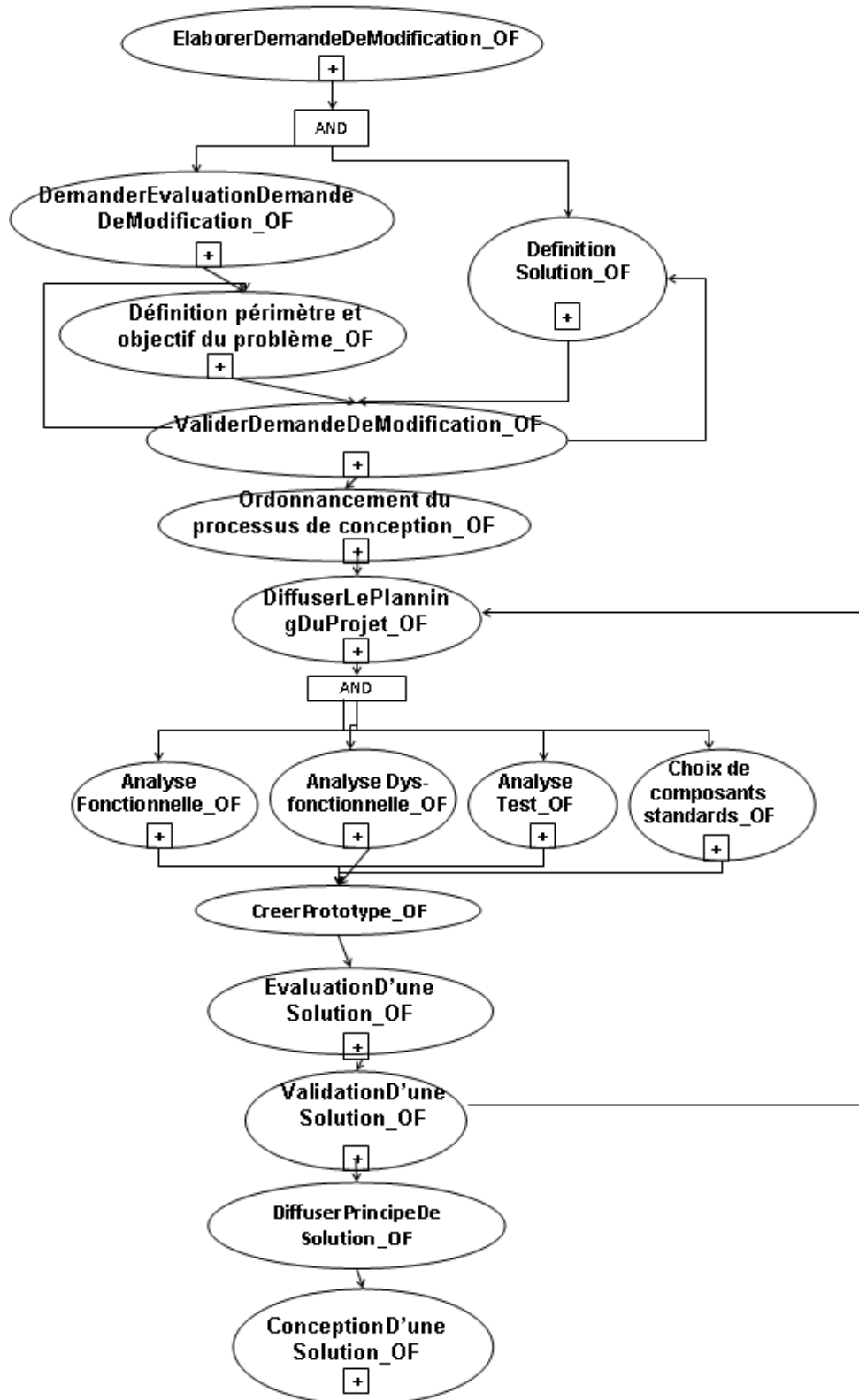


Figure 69 : Changement dans le déploiement métier du processus de traitement des modifications

Ainsi, nous pouvons conclure que la démarche d'usage et de déploiement des services proposée répond aussi à un autre objectif de notre travail, à savoir la flexibilité. Comme nous avons pu le voir à

travers la mise en œuvre du changement du processus de traitement des modifications, la démarche d'usage que nous avons proposé permet de répondre rapidement à un besoin de changement et de projeter la définition métier du changement vers un support flexible. Dans cette section, nous avons illustré un exemple de changement au niveau structurel du processus (ajout d'une étape dans le processus métier) qui a été traduit par l'ajout d'un fragment d'orchestration métier. La prise en compte de ce changement est assurée par le choix du déploiement métier associé au fragment d'orchestration métier rajouté et la ré-exécution des règles de transformation nécessaires pour la génération de l'orchestration fonctionnelle correspondante au changement. Ceci évite de faire appel à des personnes ayant des connaissances techniques afin de ré-implémenter l'alignement et le déploiement des opérations ou fragments métier à chaque fois qu'il y a un besoin de changement. De plus, l'automatisation de l'orchestration en utilisant des langages d'orchestration qui supportent l'évolution dynamique de la composition permet de mettre en œuvre ce changement à travers les mécanismes de substitution proposés dans les travaux de [Pourraz 08]. Notons que l'objet du changement peut être fonctionnel et non pas structurel (cf. Tableau 3 du chapitre I), c'est-à-dire modifier le déploiement d'une opération métier qui fait partie de l'orchestration. Ceci peut être assuré à travers la modification de l'implémentation du déploiement métier et la ré-exécution des règles de transformation nécessaires pour générer la nouvelle orchestration fonctionnelle. Enfin, les perspectives de nos travaux permettront de remplacer l'ancienne orchestration par la nouvelle à travers l'utilisation des langages supportant l'évolution dynamique.

5. Conclusion

Nous avons illustré dans ce chapitre, à travers un cas d'application, comment nos propositions répondent à notre problématique annoncée dans le chapitre I. Nous avons pu voir à travers l'illustration des différentes phases de notre démarche que notre proposition répond aux objectifs suivants :

L'agilité et le dynamisme : comme nous l'avons dit dans le premier chapitre, les processus de conception de produits manufacturés sont tendus. La concurrence du marché induit une forte pression pour pouvoir mettre des produits sous le marché dans les plus brefs délais et avec des coûts réduits. La première phase de notre démarche d'usage a illustré que le résultat de l'approche pour la réutilisation que nous avons proposée dans le chapitre III est bien exploitable. Les services de conception de produits manufacturés que nous avons proposés sont bien génériques et réutilisables pour la définition de n'importe quel processus du domaine de la conception de produits manufacturés. Cela permet de répondre à l'objectif agilité à travers la réutilisation des opérations des services métier du domaine de la conception (*SCP*) ce qui permettra de réduire le temps de définition du processus par réutilisation des traitements répétitifs.

Expressivité et simplicité du processus et de la démarche : comme le montre le cas d'application, notre démarche permet de définir des processus complexes en masquant la complexité des processus composites. Ceci à travers la réutilisation de fragments d'orchestrations prêts à l'emploi tels que ceux que nous avons utilisés pour la définition du processus de traitement des modifications. De plus, la démarche de déploiement que nous avons proposée facilite la tâche de l'utilisateur métier. En effet, l'utilisateur n'aura qu'à choisir où définir graphiquement l'orchestration fonctionnelle permettant la mise en œuvre de l'opération métier et l'étape suivante (exécution des règles de transformation) s'occupe de l'automatisation de cet alignement. Notons aussi que cette démarche masque la complexité d'implémentation des opérations constituant le processus, ce qui permettra d'éviter de faire appel à des personnes ayant des connaissances techniques et de réduire par conséquent les délais de mise en place et du changement du processus.

Automatisation et réutilisation : l'exemple d'alignement que nous avons illustré lors de la deuxième étape de la phase de déploiement métier montre que les règles de transformation que nous avons proposées dans le chapitre IV de ce manuscrit sont bien génériques et réutilisables pour tous les processus de conception définis sous forme d'orchestration métier. L'automatisation de ces règles en utilisant un formalisme tel que ATL ou QVT (en perspective de ce travail) permettra l'automatisation de la démarche d'alignement entre les différents modèles de notre approche. Le catalogue de déploiement métier constitue aussi une source d'information réutilisable et exploitable lors de l'exécution des règles de transformation. Ceci permet de répondre à l'objectif agilité de mise en œuvre du processus et son automatisation à travers l'exécution des règles de transformation génériques.

Flexibilité : notre démarche répond au besoin de flexibilité à deux niveaux de changement. La flexibilité au niveau fonctionnel est assurée au cours de la première étape de la phase de déploiement métier (choix du déploiement). Le déploiement d'une opération peut avoir plusieurs implémentations différentes du fait qu'il peut être implémenté par le patron, par les orchestrations fonctionnelles alternatives ou encore par une orchestration définie par l'utilisateur. La flexibilité au niveau fonctionnelle peut être aussi faite au moment de l'exécution à travers la modification de l'implémentation du déploiement métier et la ré-exécution des règles de transformation nécessaires pour générer la nouvelle orchestration fonctionnelle et enfin par le remplacement de l'ancienne orchestration par la nouvelle en utilisant les mécanismes des langages d'orchestration supportant l'évolution dynamique (ce dernier point n'a pas été implémenté dans le cadre de nos travaux). La flexibilité peut être aussi assurée au niveau structurel du processus (ajout ou suppression d'un fragment de processus). Comme nous avons pu le voir à travers la deuxième itération (section 4), notre démarche permet de répondre rapidement à un besoin de changement et de projeter la définition métier du changement vers un support flexible. Ceci est assuré à travers le choix du déploiement métier du fragment métier objet du changement et la ré-exécution des règles de transformation nécessaires pour la génération de l'orchestration fonctionnelle correspondante au changement.

Conclusion Générale

Bilan

Dans ce travail de thèse nous avons présenté une démarche permettant la gestion flexible des processus de conception de produits manufacturés, adaptée aux objectifs de recherche initiaux.

Les constats formulés au début du travail et sur lesquels s'est basée la définition de la problématique étaient :

- les processus de conception possèdent une structure complexe. Il faut pouvoir les représenter entièrement. De plus, ils sont réalisés par des acteurs métier. Ils doivent donc être bien définis et clairement compris par ces acteurs. Cela nécessite d'avoir, à la fois, des **formalismes de représentations simples et expressifs** et **une facilité de prise en main** de la démarche pour un expert métier, non informaticien.
- Les processus de conception sont tendus : la concurrence du marché induit une forte pression pour pouvoir mettre des produits sur le marché dans les plus brefs délais et avec des coûts réduits. **La capacité de réutilisation et d'automatisation des tâches répétitives et de l'enchaînement des activités** est une caractéristique essentielle des outils de coordination de ces processus (cela permet de réduire les délais et de décharger la ressource responsable de la coordination) ;
- Les processus de conception sont évolutifs : le contenu et l'ordre d'exécution des activités qui les composent peuvent être initialement inconnus ou partiellement connus. Ils impliquent la nécessité de prendre en compte les aléas et les changements qui peuvent survenir. La capacité d'intégration de **la flexibilité est donc une caractéristique essentielle de ces processus et de leurs outils de support**.

Partant de ces constats, les travaux présentés dans le cadre de cette thèse ont eu pour objectif de mettre en place une démarche pour la gestion flexible des processus de conception en abordant quatre aspects :

- se concentrer sur la partie informatisée du SI support aux processus de conception de produits manufacturés afin de la rendre flexible et de favoriser la réutilisation de l'existant. Dans ce cadre, le PLM, support à l'exécution des activités des processus de conception, joue un rôle majeur. Il doit être doté de l'agilité nécessaire qui lui permettra de définir et d'adapter les processus de conception de produits manufacturés aux différents besoins de changement. Cela revient à ré-exploiter l'existant. L'objectif est de **favoriser la capitalisation et la réutilisation des savoirs (fonctionnalités du système) et des savoir-faires (méthodes de travail)**.
- **masquer la complexité des sous processus** et fournir au concepteur un cadre méthodologique avec des concepts simples, bien définis et non ambigus qui lui permettent de définir facilement les processus de conception de produits manufacturés. Cela revient à **réutiliser des fragments de processus prêts à l'emploi** et les composer **en utilisant des concepts simples et compréhensibles** par des acteurs métier. Ainsi, les activités et besoins métier du domaine de la conception de produits manufacturés doivent être capitalisés et présentés sous forme de traitements réutilisables qui peuvent être compris par les analystes métiers,
- la mise en place d'une démarche permettant de définir et mettre en œuvre le processus. Cette démarche doit proposer des modèles permettant la définition et la réalisation du processus tout en **assurant un continuum de transformations dans les spécifications du processus**. Cette démarche doit être compréhensible, flexible et générique pour pouvoir être utilisée dans plusieurs organisations,

- ***L'accélération du cycle de développement*** des processus de conception de produits manufacturés en proposant une démarche qui permet d'implémenter des processus de conception de produits manufacturés flexibles et dynamiques en s'appuyant sur des techniques réutilisables et automatisables. Ceci favorisera l'agilité de l'entreprise et la prise en compte des changements imprévus.

L'investigation dans la littérature des techniques de modélisation, outils d'automatisation et des approches de gestion flexible des processus, susceptibles de réaliser le cahier de charges ainsi posé a permis de fixer les éléments clés sur lesquels se base notre approche. Il s'agit alors de proposer une démarche basée sur les approches orientées services favorisant la réutilisation des traitements (sous forme de services) et l'utilisation de langages d'orchestration qui permettent la composition de services et supportent l'évolution dynamique.

La démarche que nous proposons suit une démarche MDE et est basée sur trois couches : métier, fonctionnelle et technique. L'architecture en trois couches de services permet d'accroître l'agilité et renforcer la réutilisation des spécifications du processus. En effet, le fait d'adopter une orientation service au niveau métier permet d'être plus agile dans la structuration des processus métiers ; les processus peuvent être modélisés de façon beaucoup plus dynamique et flexible puisque les services sont génériques et peuvent être utilisés sans savoir comment sont-ils réalisés dans l'entreprise. La couche de services techniques permet d'implémenter les services métiers dans les systèmes PLM et de bénéficier de l'agilité au niveau de l'exécution également. L'introduction d'une couche intermédiaire entre la modélisation et l'exécution (couche fonctionnelle) permet de réutiliser les spécifications du niveau métier dans plusieurs systèmes PLM. Les services de la couche fonctionnelle étant génériques. Cela permet la réutilisation de la spécification métier projetée sur le niveau fonctionnel dans plusieurs systèmes PLM.

La démarche proposée peut être vue comme la succession de deux démarches complémentaires:

- une démarche *pour la réutilisation* : elle consiste à identifier, spécifier et organiser les services à utiliser durant la définition et mise en œuvre des processus de conception de produits manufacturés.
- une démarche *par la réutilisation* qui facilite l'utilisation des services identifiés. Elle doit permettre la manipulation des spécifications du processus à tous les niveaux de notre approche. Elle facilite la définition du processus, la sélection des services permettant sa mise en œuvre, ainsi que l'alignement et l'intégration des services choisis pour implémenter l'orchestration du processus. Cette démarche doit être assez générique et facilement exploitable par les acteurs métier du domaine de la conception de produits manufacturés.

Une démarche d'identification de services de conception de produits manufacturés et de services fonctionnels PLM est alors définie. Cette démarche propose un enrichissement des techniques d'identification proposées dans la littérature. Des techniques propres aux acteurs du domaine métier tels que l'analyse des modèles de données pour l'identification des services fonctionnels PLM et l'analyse sémantique des modèles de processus pour l'identification des services de conception de produit ont été proposés. De même, un enrichissement des critères de groupement a été proposé. Des critères propres au domaine métier tel que l'utilisation des règles métier et le raffinement du groupement par décomposition du domaine en corps de métier ont été pris en compte. Ceci facilitera la recherche de services par des acteurs métier et permettra ainsi de simplifier la phase de définition de processus. Enfin, cette démarche prend en compte l'aspect humain et propose une description des services découverts, qui soit compréhensible par les acteurs métier. En effet, les opérations du niveau métier ont été définies en termes de concepts du domaine, à savoir les objets métier et les actions métier du domaine de la conception de produits manufacturés. Cette démarche d'identification a été

pensée dans un souci de facilité de prise en main par les acteurs métier. Elle peut être réutilisée pour enrichir les catalogues proposés.

Le référentiel de services obtenu à l'issue de cette démarche d'identification organise les services en deux catalogues de services. Un catalogue de services de conception de produits manufacturés et un catalogue de services fonctionnels PLM. Ces deux catalogues capitalisent les savoirs et savoir-faire du domaine de la conception et des systèmes PLM.

Un cadre pour l'usage des services proposés a été aussi défini. Ce cadre permet la définition et la mise en œuvre des processus de conception de produits manufacturés. Il comprend principalement trois phases : la phase de modélisation du processus de conception sous forme de composition de services (métiers), la phase de déploiement métier permettant l'alignement du modèle métier sur le niveau fonctionnel et enfin la phase de déploiement fonctionnel permettant la projection des opérations fonctionnelles PLM sur le niveau technique (logiciel).

La méthode de définition de processus que nous avons proposée permet de définir des processus tout en masquant la complexité des processus composites. Ceci à travers la réutilisation de fragments d'orchestrations prêts à l'emploi en tant que sous-processus. La réutilisation des services de conception de produits et des fragments d'orchestrations métier permettra de réduire le temps de définition du processus.

Concernant l'automatisation et la mise en œuvre du processus, nous avons défini une démarche générique de déploiement des opérations de services de conception de produit et des services fonctionnels PLM. Cette démarche de déploiement permet la réutilisation des savoir-faire. Les projections entre les opérations des couches métiers et fonctionnelles sont capitalisées et présentées dans des fiches de correspondance. Ceci favorise la réutilisation et facilite la tâche de l'utilisateur tout au long du déploiement de son processus sur un système PLM spécifique. La proposition de règles de transformation génériques permet par ailleurs l'automatisation de la démarche d'alignement entre les différents modèles de notre approche. Le savoir-faire permettant le passage d'une couche à une autre est capitalisé dans les règles de transformation. Ceci évite de faire appel à des personnes ayant des connaissances techniques afin de ré-implémenter l'alignement et le déploiement des opérations métier à chaque fois qu'il y a un besoin de changement et de réduire par conséquent le temps de mise en œuvre du processus.

Enfin, le fait d'adopter un langage d'orchestration permettant de supporter les mécanismes d'évolution dynamiques afin de rendre le processus plus flexible en permettant de rajouter, modifier, supprimer des fragments d'orchestration ou des services à l'orchestration globale. La flexibilité est ainsi assurée, en cas d'évolution dans la structure du processus ou les fonctionnalités du système PLM, à travers la modification de l'implémentation du déploiement, la réexécution des règles de transformation et le remplacement de l'ancienne orchestration par la nouvelle en utilisant les mécanismes des langages d'orchestration supportant l'évolution dynamique.

Perspectives

Ce travail ouvre la voie à notre sens vers diverses perspectives de recherche qui se situent sur deux plans : un plan d'approfondissement de la recherche réalisée et un plan d'élargissement du domaine de la recherche.

Pour ce qui est de l'approfondissement du travail proposé, il serait intéressant dans un premier temps de proposer une bibliothèque de fragments d'orchestration métier complètement formalisés et prêts à l'emploi. Ces fragments d'orchestration doivent être couplés avec une description permettant leur réutilisation. La description d'un fragment d'orchestration peut être définie entre autres par le but du

processus qu'il réalise, la description du processus en langage naturel, un modèle illustrant l'enchaînement des opérations de services de conception participants au fragment d'orchestration et la description des objets métier nécessaires en entrée et fournis en sortie du fragment.

L'application des catalogues de services et de la démarche d'alignement sur le cas d'application, présenté au chapitre V, a en particulier mis en évidence la difficulté de spécifier d'une façon automatique le modèle résultat de la transformation en utilisant les règles de transformations telles qu'elles sont formalisées. Par ailleurs, l'application des règles de transformation associées au passage du niveau métier au niveau fonctionnel constitue un atout pour le cadre méthodologique proposé du fait qu'elles sont génériques et réutilisables pour tout exemple de processus. Il serait intéressant de développer un atelier de modélisation couplé avec un moteur de transformation qui automatise la transformation du processus modélisé. Ceci nécessite la formalisation des règles de transformation dans un langage automatisable tel qu'ATL ou QVT. Cet atelier doit reposer sur la sémantique des langages d'orchestration supportant l'évolution dynamique tels que PXL, de la modélisation à l'exécution du processus.

Dans un second temps, et en ce qui concerne l'expérimentation de la démarche proposée, il serait intéressant d'appliquer les catalogues de services et la démarche d'alignement proposés sur d'autres scénarios industriels avec des processus non déterministes et en présence d'utilisateurs métier (gestionnaires de processus, acteurs métier du domaine de la conception, etc.). Une expérimentation de l'atelier de modélisation et d'exécution des règles de transformation et des processus, que nous avons cité en perspectives, permettra de le valider et de tester ses capacités et ses limites à supporter la démarche proposée.

Pour ce qui est de l'élargissement du domaine de la recherche, il serait intéressant dans un premier temps d'élargir le périmètre fonctionnel en allant au-delà des systèmes PLM pour couvrir tout le SI (i.e. les autres outils de calcul/simulation, de conception CAO, applications patrimoniales, etc.). Par ailleurs, l'approche orientée services, pour et par la réutilisation, que nous avons proposée peut être appliquée pour d'autres types de SI. Nous avons proposé une démarche pour identifier les services fonctionnels PLM, qu'il serait intéressant d'appliquer pour l'identification d'autres catalogues de services fonctionnels (services fonctionnels CAO, services fonctionnels Calcul, services fonctionnels ERP, etc.). La formalisation des processus de conception par réutilisation de services est également à étendre. Pour cela, il faudra définir la correspondance entre les services du niveau métier et les services fonctionnels d'autres logiciels.

Enfin, il serait intéressant d'étudier l'intérêt de l'approche proposée dans le contexte des approches d'ingénierie de produit et de services : on parle de lean manufacturing, réduction des coûts, délais, etc. En effet, dans le domaine du logiciel, on parle de méthodes agiles, de lean software development qui cible la maîtrise des coûts et des délais. Ces méthodes agiles partent du principe qu'il est impossible d'avoir des exigences fixes et stables, même constat objet de notre problématique. De plus, les méthodes agiles ont pour objectif de mettre en place une organisation permettant aux exigences et besoins d'évoluer au cours du temps, de prendre en compte les modifications de périmètre, de fonctionnalités, etc. Cela nécessite que les processus doivent pouvoir être flexibles pour intégrer les inévitables modifications et changements qui interviennent. Notre démarche s'inscrit dans ce sens de méthodes agiles. Il est donc intéressant d'étudier son impact et son intérêt en proposant par exemples des indicateurs.

Références Bibliographiques

[Allilaire 04] Allilaire, F., Idrissi T., ADT: Eclipse development tools for ATL, In: Proceedings of the Second European Workshop on Model Driven Architecture (MDA) with an emphasis on Methodologies and Transformations (EWMDA-2). Computing Laboratory, University of Kent, canterbury, UK. England 2004, p. 171-178.

[Alonso 03] Alonso, G., Casati, F., Kuno, H. and Machiraju, H. Web Services -Concepts, Architectures and Applications. Springer Verlag, 2003.

[Amhed 02] Amhed S., Hansen C.T., A decision-making model for engineering designers, Shahin T.M.M. (eds.), Computer based design, pp. 217- 227, EDC2002, Cambridge, 2002.

[Arkin 03] Arkin, I. (2003). Business process modeling language. <http://www.bpmn.org/Documents/BPML-2003.pdf>.

[Arsanjani 04] Arsanjani, A., 2004, Service-oriented modeling and architecture. How to Identify, Specify, and Realize Services for Your SOA. Accessible: <http://www.ibm.com/developerworks/library/ws-soa-design1/>.

[Arsanjani 08] Arsanjani, A.; Ghosh, S. ; Allam, A. ; Abdollah, T. ; Ganapathy, S. ; Holley, K. SOMA: A method for developing service-oriented solutions. IBM Systems Journal. V.47 , Issue: 3. 2008. pp 377 – 396

[Audros 13] Solution PLM Audros. Accessible : <http://www.audros.fr/cms/easysite/audros/solutions-plm>. [consulte le 30/01/13]

[Benatallah 02] Benatallah, B. Dumas, M. Sheng Q-Z. et H. Ngu. A H.: Declarative composition and peer-to-peer provisioning of dynamic web services. Rakesh Agrawal, Klaus Dittrich et Anne H; H. Ngu, éditeurs, 18th international Conference on ata Engineering (ICDE 2002), pages 297-308, San Jose, California USA, 2002. IEEE Computer Society.

[Benatallah 03] Benatallah, B., Sheng, Q. Z. and Dumas, M., 2003, The Self-Serv environment for Web services composition, IEEE Internet Computing, 7(1), pp. 40-84.

[Ben Halima 08] Ben Halima, R., Guennoun, K., Drira, K., and Jmaiel, M. (2008). Providing predictive self-healing for web services : A qos monitoring and analysis-based approach. Journal of Information Assurance and Security, 3 :175–184.

[Berthier 06] Berthier, D. Enrichissement de la modélisation des processus métiers par le paradigme des systèmes multi agents. Technical report, INT/GET, 2006.

[Bézivin 04] Bézivin, J. 2004, "Sur les principes de base de l'ingénierie des modèles," In L'Objet, vol. 10 pp. 147-157.

[Bitlog 12] ViFlow - Agilium, la solution de workflow complète (BPM - EAI - BAM). Accessible : <http://www.bitlog-gmbh.com/index.php?id=bitlog-agiliumworkflowsung&L=2> [consulte le 7/12/12]

[Boukadi 09a] Boukadi, K. Coopération interentreprises à la demande : Une approche flexible à base de services adaptables. PhD thesis, École Nationale Supérieure des Mines de Saint-Étienne, 2009.

- [Bourdon 07] Bourdon, J., 2007, Multi-agent systems for the automatic composition of semantic web services in dynamic environments, In Rapport de master, École des Mines de Saint Etienne - G2I & Université Joseph Fourier.
- [Carlsen 97] S. Carlsen, J. Krogstie, A. Solvberg, and O.I. Lindland. Evaluating Flexible Workflow Systems. In Proceedings of the Thirtieth Hawaii International Conference on System Sciences (HICSS-30), Maui, Hawaii, 1997. IEEE Computer Society Press.
- [Casati 00] Casati, F. and Discenza, A., 2000, Supporting Workflow Cooperation Within and Across Organisations, In 15th ACM Symposium on Applied Computing (SAC'00) Como, Italy, pp. 9-21.
- [Casati 01] Casati, F. and Shan, M.-C., 2001, Dynamic and adaptive composition of e-services, Information Systems, 26(3), pp. 143-163.
- [Castro 07] Castro, V. d., Vara, J. M. and Marcos, E., 2007, Model Transformation for Service-Oriented Web Applications Development, In the 3th International Workshop on Model-Driven Web Engineering (MDWE'07) Como, Italy.
- [Cauvet 08] Cauvet, C. and Guzelian, G., 2008, Business Process Modeling: a Service-Oriented Approach, In the 41st Hawaii International Conference on System Sciences Hawaii
- [Chaâbane 09] Chaâbane, M., Andoloff, E., Bouzguenda, L., Bouaziz, R., Versions to adress Business Process Flexibility Issue, Advances Databases and information Systems, 2009, pp. 2-14
- [Chang 07] Chang, S. H. and Kim, S. D., 2007, A Service-Oriented Analysis and Design Approach to Developing Adaptable Services, In the IEEE International Conference on Services Computing (SCC 2007), pp. 204-211
- [Chelli 03] Chelli, H. (Ed.) (2003) Urbaniser l'entreprise et son système d'Information. Guide des entreprises agiles.
- [CIMdata 09] CIMdata (2009). PLM Market Growth in 2008, Mid-Year.
- [Cîmpan 05] Cîmpan, S. and Verjus, H. (2005). Challenges in architecture centred software evolution. In CHASE : Challenges in Software Evolution, pages 1–4, Bern, Switzerland.
- [Collet 06] Collet, P. Etat de l'art sur la contractualisation et la composition. RNTL FAROS - Livrable F-1.1 - <http://www2.lifl.fr/faros/pub/uploads/Main/RNTL-FAROS-F1-1.pdf>, 2006.
- [Combemale 09] Combemale, B. Ingénierie Dirigée par les Modèles (IDM) - État de l'art. Accessible : hal.archives-ouvertes.fr/hal-00371565/en/. 2009.
- [Credle 08] Credle, R. Bader, M. Brikler, K. Harris, M. Holt, M. Hayakuna, Y. SOA Approach to entreprise integration for product lifecycle management, Oct. 2008, pp. 66-80.
- [CXP 06] Workflow and Business Process Management Agilium: a novel approach to process management. 2006. Accessible : http://www.inst-informatica.pt/servicos/informacao-e-documentacao/biblioteca-digital/gestao-de-si-ti-1/bpm/01_Agilium_CXP_EN_060324.pdf [consulte 7/12/12]
- [Daoudi 06] Daoudi F., Nurcan S. A framework to evaluate methods' capacity to design flexible business processes. Dans Business Processes and Support Systems: Design for Flexibility - 6th International Workshop on Business Process Modeling, Porto : Portugal (2006).

- [Daoudi 07] Daoudi F. and Nurcan. S. A Benchmarking Framework for Methods to Design Flexible Business Processes. *Software Process Improvement and Practice*, 12:51-63, 2007.
- [Débauche 04] Débauche, B. & Megard, P. 2004. BPM Business Process Management : pilotage métier de l'entreprise Lavoisier.
- [Dolques 09] Dolques, X. Huchard, M. and Nebut, C. Génération de transformation de modèles par application de l'ARC sur des exemples," in LMO'09: Langages et Modèles à Objets, 2009, pp. 61-75.
- [Erl 05] Erl, T. *Service-Oriented Architecture. Concepts, Technology, and Design*, fourth ed. Prentice Hall, 2005.
- [Esper 10] Esper, A. *Integration des approches SOA et orientée objet pour modéliser une orchestration cohérente de services*. Phd thesis. Institut National des sciences Appliquées de Lyon. 1-09-2010.
- [Estrella 03] Estrella, F. Kovacs, Z. McClatchey, R. Toth, N. Solomonides, T. A Component Framework for Description-Driven Systems. 2003. In: *Proceedings of the EMSISE'03 international workshop - Engineering Methods to support information Systems Evolution*. Geneva : Suisse (2003)
- [Fareghzadeh 08] Fareghzadeh, N. Service identification approach to SOA development. *Proceedings of World Academy of Science, Engineering and Technology*, 35, ISSN 2070-3740.2008.
- [FDX50-127 02] FDX50-127 (2002). Outils de management - Maitrise du processus de conception et développement. Accesible : <http://www.boutique.afnor.org/norme/fd-x50-127/outils-de-management-maitrise-du-processus-de-conception-et-developpement/article/622831/fa118828> [20 juin 2012]
- [Fowler 97] Fowler M., *Analysis Patterns, reusable objects models*, Addison Wesley 1997
- [Foster 03] Foster, H., Uchitel, S., Magee, J. and Kramer, J., 2003, Model-based Verification of Web Service compositions In the 18th IEEE International Conference on Automated Software Engineering (ASE 2003) Montreal, Canada, pp. 152-161.
- [Fremantle 02] Fremantle, P. Weerawarana, S. and Khalaf, R. Enterprise services. *Communications of the ACM*, 45(10):77-82, 2002.
- [Gaspard 03] Gaspard, S. Estrella, F. McClatchey, R. Dindeleux, R. 2003. Managing Evolving Business Workflows through the Capture of Descriptive Information. Presented at the eCOMO'2003 4th Int. Workshop on Conceptual Modeling Approaches for e-Business. 11/2003;
- [Gilbert 04] Gilbert, P. et Leclair, P., 2004 « Les systèmes de gestion intégrés. Une modernité en trompe l'oeil ? », dans *Le mythe de l'organisation intégrée, les progiciels de gestion*. *Revue sciences de la société*, n°61.
- [Glatard 09] Glatard, T., Montagnat, J., Bolze, R., Desprez, F. (2009) On scientific workflow representation languages, Report of Laboratoire d'Informatique de Signaux et Systemes de Sophia Antipolis, Accesible : <http://www.i3s.unice.fr/~mh/RR/2009/RR-09.10-J.MONTAGNAT.pdf> [7 December 2012].
- [Hachani 11a] Hachani, S. Gzara, L. and Verjus, H. An SOA Based Approach to Improve Business Processes Flexibility in PLM, *Proceedings of Technological Innovation for Sustainability - Second IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2011*, 349, 67-74, Springer Berlin Heidelberg, 2011

- [Hachani 11b] Hachani, S., Gzara, L. and Verjus, H. Support of Business Processes Flexibility in PLM Systems Using a Services-based Approach, Proceedings of the International Conference on Industrial Engineering and Systems Management (IESM'2011), 547-555, 2011,
- [Hachani 11c] Hachani, S. Gzara, L. and Verjus, H. Business Process Flexibility in Service Composition: An Experiment Using a PLM-Based Scenario, Exploring Services Science - Second International Conference, IESS 2011, 82, 158-172, Springer, 2011.
- [Hachani 12a] Hachani, S., Verjus, H. and Gzara, L. Service-Oriented Approach for Agile Support of Product Design Processes, 7th International Workshop on Enterprise Integration, Interoperability and Networking (EI2N'2012). Lecture Notes in Computer Science Volume 7567, 2012, pp 103-112
- [Hachani 12b] Hachani S., Verjus H., and Gzara L., Support of Product Design Processes Flexibility in PLM Systems using a Service-based Approach. International Journal of Services operations and Informatics (IJSOI). Volume 7, Number 4, 313-329, 2012. ISSN 1741-5403. Ed. Indersciences.
- [Hachani 13] Hachani, S., Gzara, L. and Verjus, H. A Service-Oriented Approach for Flexible Process Support within Enterprises: Application on PLM Systems. Enterprise Information Systems. Volume 7, Number 1, 79-99, 2013. ISSN 1751-7575. Ed. Taylor & Francis.
- [Hachani 13a] Hachani S., Verjus H., and Gzara L. Business Agility and Flexibility in Enterprise Service-based Information Systems: Application on PLM Systems. Dans "Information Systems and Small and Medium-sized Enterprises (SMEs) - State of Art of IS Research in SMEs". Ed.Springer. Accepté pour publication, à paraître.
- [Han 09] Han, Y., Wang, J. and Zhang, P., 2009, Business-oriented service modeling: A case study, Simulation Modelling Practice and Theory, 17(8) 1413-1429.
- [Hao 03] Hao, H. What is service-oriented architecture. O'Reilly Media, Accessible : <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>, 2003.
- [Heinl 99] Heinl, P. Horn, S. Jablonski, S. Neeb, J. Stein, K. and Teschke, M. A Comprehensive Approach to Flexibility in Workflow Management Systems. In G. Georgakopoulos, W. Prinz, and A.L. Wolf, editors, Work Activities Coordination and Collaboration (WACC'99), pages 79-88, San Francisco, February 1999. ACM press.
- [Hepner 06] Hepner, M. Gamble, R. Kelkar, M. Davis, L. and Flagg, D. Patterns of conflict among software components. Journal of Systems and Software, 79(4) : 537-551, 2006.
- [ISO 05] ISO10303-2392005.Industrial automation systems and integration–Product data representation and exchange – Part 239: Application protocol: Product life cycle support. Geneva (Switzerland): International Organization for Standardization; 2005.
- [Jomaa 07] Jomaa, H., « Les déterminants de la performance des projets ERP dans les grandes entreprises », dans Cahier de recherche du CIGREF, octobre 2007.
- [Jouault 05] Jouault, F. et Kurtev, I. (2005). Transforming models with atl. In J.-M. Bruel (Ed.), MoDELS Satellite Events, pp. 128–138. Springer.
- [Kaabi 07a] Kaabi, R. S., Rolland, C. 2007, An Intentional Perspective to Service Modeling and Discovery, Computer Software and Applications Conference, COMPSAC 2007, 2(24-27), pp. 455-460.
- [Kaabi 07b] Kaabi, R. 2007, Une Approche Méthodologique pour la Modélisation Intentionnelle des Services et leur Opérationnalisation. PhD thesis, UNIVERSITE PARIS I - SORBONNE, 2007.

- [Kellert 06] Kellert, P. and Toumani, F., 2006, Les services web sémantiques, Revue I3 (Information-Interaction-Intelligence).
- [Khalaf 03] Khalaf, R., Leymann, F. On web services aggregation. In Springer Berlin /Heidelberg, editor, Technologies for E-Services, volume 2819/2003 of Technologies for EServices, pages 1–13. Springer–Verlag, Septembre 2003.
- [Khalgui 12] Khalgui M. Application de Contrôle Industriel à base de Composants. Validation Temporelle et Déploiement. Ed. Paf. ISBN-13: 978-3-8381-7480-8. (2012)
- [Khoshafian 98] Khoshafian, S., Buckiewicz, M. Groupware et workflow. InterÉditions, 1998
- [Kohlborn 09] Kohlborn, T., Korthaus, A., Chan, T., Rosemann, M. Service analysis : A critical assessment of the state of the art. In Newell, S., Whitley, E., Pouloudi, N., Wareham, J., & Mathiassen, L. (Eds.) Proceedings of 17th European Conference on Information Systems. Information Systems in a Globalising World : Challenges, Ethics and Practices, 8-10 June 2009, Verona, Italy. (In Press)
- [Kohlborn 09b] Kohlborn, T., Korthaus, A., Chan, T., Rosemann, M. Identification and Analysis of Business and Software Services—A Consolidated Approach. In. Services Computing, IEEE Transactions on. Vol2. Issue 1. pg 50-64. 2009
- [Lamouri 06] Lamouri, S. (2006). Synchronisation des prises de décisions dans une chaîne logistique : robustesse et stabilité. Mémoire d'Habilitation à Diriger des Recherches, Ecole Doctorale d'Informatique et Electronique de Paris.
- [Larsson 05] Larsson, A., « Engineering Know-Who: Why social connectedness matters to global design teams », Thèse de doctorat de l'Université Technologique de Luleå, Suède, 2005.
- [Leymann 01] Leymann. F. Web services flow language (wsfl 1.0). Accesible : www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf, 2001.
- [Lezoche 08] Lezoche, M., Missikof, M., Tinini, L., Business Process Evolution: a rule-based Approach, at Int. Conference on advanced Information Systems, Montpellier, France: 2008, pp. 407-414.
- [Lopistéguy 03] Lopistéguy P., Etcheverry P., Dagorret P. "Étude de la coordination dans un processus. Une expérience à base de patrons." Congrès INFORSID No21, Nancy , FRANCE (03/06/2003)
- [Margerie 05] Margerie. F. Approcher soa en douceur (ou le pragmatisme de rigueur). <http://www.dotnetguru.org/articles/dossiers/soadouceur/soaendouceur.htm>, 2005.
- [Martin 04] Martin, R. A., Robertson, E. L., & Springer, J. A. 2004, Architectural Principles for Enterprise Frameworks, Computer Science Department, Indiana University, Bloomington, Indiana.
- [Mayer 95] Mayer, J. Benjamin, P.C. Caraway, B.E. and Painter. M.K. A framework and a suite of methods for business process reengineering. <http://www.idef.com>, 1995.
- [McCoy 06] McCoy, D. W. and Plummer, D. C., 2006, Defining, Cultivating and Measuring Enterprise Agility, Gartner Research.
- [Medjahed 05] Medjahed, B. and Bouguettaya, A., 2005, A Dynamic Foundational Architecture for Semantic Web Services, Distributed and Parallel Databases, 17(2), pp. 179–206.

- [Mendling 03] Mendling, J., Muller, M., and Wirtschaftsinformatik, L. (2003). A comparison of bpm and bpm4ws. In In : Proc. 1st Conf. Berliner XML-Tage, pages 305–316.
- [Micaelli 02] Micaelli J-P., 2002. Institutionnalisme, évolutionnisme : le défi de la conception. Actes des Journées d'étude Institutionnalisations et Évolutionnismes-Confrontations autour de perspectives empiriques, Lyon, 2002.
- [Michael 07] Michael, A., Arthur H.M. ter Hofstede, Wil M.P. van der Aalst, and Edmond, D. Dynamic, Extensible and Context-Aware Exception Handling for Workflow. In: Proceedings of the 15th International Conference on Cooperative Information Systems (CoopIS 2007), Vilamoura, Algarve, Portugal, November 2007.
- [Milanovic 04] Milanovic, N. and Malek, M., 2004, Current Solutions for Web Service Composition, IEEE Internet Computing, 8(6), pp. 51-59.
- [Miller 03] Miller, J. and Mukerji, J., 2003, Model Driven Architecture (MDA) 1.0.1 Guide. Object Management Group, Inc., Accessible : <http://www.bibsonomy.org/bibtex/2523ccb6d3fb557fe961d1c0418d4a396/neilernst>
- [Morley 05-b] Morley, C., Berthier, D., Maurice-Demourieux, M. Apport du paradigme agent à la maîtrise des systèmes d'information, 10ème Colloque AIM, Toulouse: 2005
- [Morley 07] Morley, C., Hugues, J., Leblanc, B., & Hugues, O. 2007. Processus métiers et S.I. – Evaluation, modélisation, mise en oeuvre, DUNOD.
- [Nurcan 09] Selmin, N., Rainer S. Service oriented entreprise Architecture for entreprise engineering introduction. In Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009. pg; 247 -253.
- [OASIS 06] OASIS, 2006, Reference Model for Service Oriented Architecture 1.0, available at: <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>.
- [OGSA 05] OGSA, Glossary of Terms, 2005. Accessible: <http://www.ggf.org/documents/GWD-I-E/GFD-I.044.pdf>.
- [OMG 08] Object Management Group. PLM Services 2.0, 2008
- [OMG 08a] Object Management Group. BPMN - Business Process Modeling Notation Specification - OMG Final Adopted Specification (patent). 2008.
- [OMG 08b] Object Management Group. Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification. 4/2008. Accesible : <http://www.omg.org/spec/QVT/1.0/>
- [Overby 05] Overby, E. Bharadwaj, A. and Sambamurthy, V. A Framework for Entreprise Agility and the Enabling Role of Digital Options, Business Agility and Information Technology Diffusion, vol. 180, Springer, Boston, 2005, pp. 295-312
- [Pahl et Beitz 96] PAHL, G., BEITZ, W. 1996. Engineering Design: A systematic approach. Springer-Verlag, ISBN 3-540-19917-9
- [Pahl 07] Pahl G., Beitz W. Engineering Design: a Systematic Approach, 2007 (2nd Ed., Springer-Verlag, London).
- [Pallos 01] Pallos, M., 2001, Service Oriented Architecture: A Primer, EAI Journal, December 2001.

- [Papazoglou 03] Papazoglou. M.P. Service-oriented computing: concepts, characteristics and directions. *Web Information Systems Engineering*, 2003. WISE 2003. Proceedings of the Fourth International Conference on, pages 3–12, Dec. 2003.
- [Papazoglou 06] Papazoglou M.P. and van den Heuvel, W.-J. Service-Oriented Design and Development Methodology, *Int'l J. Web Eng. And Technology (IJWET)*, vol. 2, no. 4, pp. 412-442, 2006.
- [Papazoglou 06a] Papazoglou, M.P. Traverso, P., Dustdar, S Leymann, F. and Krämer B. J. Service-oriented computing : A research roadmap. *Service Oriented Computing (SOC)*, number 05462, 2006.
- [Papazoglou 07] Papazoglou M. and van den Heuvel, W. Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, 16(3):389–415, July 2007.
- [Pegoraro 08] Pegoraro, R., Ben Halima, R., Drira, K., Guennoun, K., and Rosario, J. M. (2008). A framework for monitoring and runtime recovery of web service-based applications. 10th International Conference on Enterprise Information Systems (ICEIS 2008).
- [Peltz 03] Peltz. C. Web services orchestration : A review of emerging technologies, tools, and standards. 2003. Accesible: <http://devresource.hp.com/drc/technicalwhitepapers/WSOrch/WSOrchestration.pdf>,
- [Pfister 96] Pfister C. Szyperski. C. Why objects are not enough. In *Proceedings, International Component Users Conference*, Munich, Germany, 1996. SIGS.
- [PLCS 12] PLCS ISO STEP standard (ISO 10303-239) <http://www.plcs-resources.org/ap239/index.html> (juin 2012)
- [Pourraz 07] Pourraz, F. Diapason : une approche formelle et centree architecture pour la composition evolutive de services Web. PhD thesis, L'UNIVERSITE DE SAVOIE, 2007.
- [Pourraz 07a] Pourraz, F. and Verjus, H. Diapason : An engineering environment for designing, implementing and evolving service orchestrations. *ERCIM News Magazine - Special : Service-Oriented Computing*, No. 70, pp. 41-43, 2007.
- [Pourraz 07b] Pourraz, F. and Verjus, H. Diapason : An engineering environment for designing, executing and evolving service-oriented architectures. *Second International Conference on Software Engineering Advances (ICSEA 2007)*, 2007.
- [Pourraz 08] Pourraz, F. and Verjus, H. (2008). Managing Service-Based EAI Architectures Evolution Using a Formal Architecture-Centric Approach, volume 3 of *Lecture Notes in Business Information Processing*, pages 269–280. Springer Berlin Heidelberg.
- [Qiao 09] Qiao, M., Khendek, F., Serhani, A., Dssouli, R., and Glitho, R. (2009). An architecture for automatic qos adaptation for composite web services. *International Journal of Web Services Practices*, 4(1) : 18–27.
- [Ramollari 07] Ramollari, E., Dranidis, D. and Simons, A. J. H., 2007, A Survey of Service Oriented Development Methodologies, In *The 2nd European Young Researchers Workshop on Service Oriented Computing* University of Leicester, UK
- [Ramus 03] Ramus V., Gallardo E., Zaeh F., Loynes C., Mathieu V. Orientation service des entreprises industrielles. 25/06/2003. <http://www.industrie.gouv.fr/enjeux/servindus.htm>

- [Regev 06] Regev, G. Soffer, P. and Schmidt. R. Taxonomy of Flexibility in Business Processes. In Proceedings of the 7th Workshop on Business Process Modelling, Development and Support (BPMDS'06), 2006. <http://lamswww.epfl.ch/conference/bpmds06/taxbpflex>.
- [Regev 07] Regev, G. Bider, I. and Wegmann, A.. Defining Business Process Flexibility with the Help of Invariants. Software Process Improvement and Practice, 12:65-79, 2007.
- [Riviere 04] RIVIERE, A. Gestion de configuration et des modifications lors du développement de grands produits complexes en ingénierie concourante – Cas d’application Aéronautique. PhD thesis, INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE, 2004.
- [Robin 04] Robin V., Girard Ph., and Barandiaran D., 2004. A model of design environments to support collaborative design management. In: 5th International Conference on Integrated Design and Manufacturing in Mechanical Engineering, IDMME 2004, 5-7 April, Bath.
- [Rockwell 06] Rockwell. [Http://www.holobloc.com](http://www.holobloc.com). Rockwell automation, 2006.
- [Ross 77] Ross. D. Structured analysis : a language for communicating ideas. IEEE Transactions on Software Engineering, Vol.3, N_1, janvier 1977.
- [Russell 06] Russell, N. ter Hofstede, A.H.M. van der Aalst, W.M.P. and Mulyar. N. Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-22, BPMcenter.org, 2006. Available at : <http://www.workflowpatterns.com/patterns/control/>
- [Saaksvuori 05] Sääksvuori, A., Immonen, A. Product lifecycle management, Birkhäuser, 2005.
- [Scaravetti 04] Scaravetti. D., Nadeau, J-P., Pailhes, J., Sebastian, P., 2004. Structuring of embodiment design problem based on the product lifecycle. In: International journal of Product Development, Special Issue on Product Lifecycle Management. Vol 2 (1/2), July ,47-70.
- [Schonenberg 07] Schonenberg, M.H. Mans, R.S. Russell, N.C. Mulyar, N.A. and van der Aalst W.M.P.. Towards a Taxonomy of Process Flexibility (Extended Version). BPM Center Report BPM-07-11, BPMcenter.org, 2007.
- [Simon 73] Simon, H., « The structure of ill-structured problems », Artificial intelligence, 4, 1973, pp 181-20.
- [Simon 89] Simon, H., « The sciences of the artificial », 2nd Edition, The MIT Press, Cambridge, MA, 1989.
- [Smith 02] Smith, H., Neal, D., Ferrara, L., & Hayden, F. 2002. The Emergence of Business Process Management CSC's Research Services.
- [Snowdon 07] Snowdon, R.A. Warboys, B.C. Greenwood, R.M. Holland, C.P. Kawalek, P.J. and Shaw D.R.. On the Architecture and Form of Flexible Process Support. Software Process Improvement and Practice, 12:21{34, 2007.
- [Stohr 97] Stohr, E.A., Zhao, J.L. 1997. A technology adaptation model for business process automation. Proceedings of the Thirtieth Hawaii International Conference on System Sciences, 1997. vol4, 405 -414.
- [Szyperski 02] Szyperski, C. Gruntz, D. and Murer. S. Component Software: Beyond Object-Oriented Programming. Addison-Wesley Professional. 2nd Edition, England, 2002.

- [TC 07] TC, O. W. (2007). Web services business execution language version 2.0. Accessible: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
- [Théroutde 02] Théroutde, F. Formalisme et système pour la représentation et la mise en œuvre des processus de pilotage des relations entre donneurs d'ordre et fournisseurs, thèse de Doctorat de l'Institut National Polytechnique de Grenoble, 2002.
- [Thatte 01] Thatte. S. Xlang - web services for business process design. <http://www.gotdotnet.com/team/xmlwsspecs/xlang-c>, 2001.
- [Tollenaere 98] Tollenaere K., 1998. Conception de produits mécaniques : méthodes, modèles et outils. Paris:Hermès Science.
- [Tomas 00] Tomas, J-L. ERP et Progiciels intégrés : la mutation des systèmes d'information, 2e édition. Dunod, Janvier 2000.
- [Tudor 06] Tudor. H. Modélisation des processus métiers : État de l'art et conseils pratiques. Technical report, CITI, 2006.
- [Ulmer 11] Ulmer, J.-S., Approche générique pour la modélisation et l'implémentation des processus. PhD thesis, Institut National Polytechnique de Toulouse, 2011.
- [Ulrich 00] Ulrich, K.T., Eppinger, S.D., « Product design and development », McGraw-Hill International, Second edition, USA, 2000.
- [van der Aalst 00] Van der Aalst, W.M.P. and Jablonski S.. Dealing with Workflow Change: Identification of Issues and Solutions. International Journal of Computer Systems, Science, and Engineering, 15(5):267-276, 2000.
- [VanDerAalst 03] Van der Aalst, W.M.P. ter Hofstede, Arthur H.M, Weske, M,. Business Process Management: A survey. Dans BPM 2003, LNCS 2678, pp. 1-12, 2003. Springer-Verlag Heidelberg 2003
- [VanDerAalst 03a] VanDerAalst, W M.P 2003. Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language (2003)
- [Van der Aalst 03b] Van der Aalst, W.M.P ter Hofstede, A.H.M. Kiepuszewski, B. and Barros. A.P. Workflow Patterns. Distributed and Parallel Databases, 14(3), pages 5-51, July 2003. Accesible: <http://www.workflowpatterns.com/patterns/control/>
- [VanDerAalst 04] Van der Aalst, W. M. P. Business process management demystified: A tutorial on models, systems and standards for workflow management. In J. Desel, W. Reisig, and G. Rozenberg, editors, Lectures on Concurrency and Petri Nets, volume 3098 of Lecture Notes in Computer Science, pages 1–65. Springer-Verlag, Berlin, 2004.
- [Van der Aalst 04a] van der Aalst, W.M.P. Aldred, L. Dumas, M. and ter Hofstede. A.H.M. 2004. Design and Implementation of the YAWL system. To appear in Proc. of The 16th Intel. Conf. on Advanced Information Systems Engineering (CAiSE 04), Riga, Latvia, June 2004.
- [Verjus 08] Verjus, H. and Pourraz, F. (2008). Diapason : A formal approach for supporting agile and evolvable information system service-based architectures. In Proceedings of the 10th International Conference on Enterprise Information Systems (ICEIS'08), volume ISAS-2, pages 76–81, Barcelona, Spain. INSTICC-WfMC-AAAI.

[Verjus 11] Verjus, H., Pourraz, F., and Fakhfakh, N. (2011). Cadre conceptuel pour la modélisation et la supervision d'architectures à base de services. Une approche pour l'ingénierie des SI à base de services. Développement des SI à base de modèles : exigences, traçabilité et co-conception - Ingénierie des Systèmes d'Information - Revue des sciences et technologies de l'information, 16(5) :43–72.

[Vernadat 07] Vernadat, F. Interoperable enterprise systems: Architectures and methods, In. Information Control Problems in Manufacturing, Volume 12, Part 1, 2007.

[Wendt 05] Wendt, T. Brigl, B. and Winter. A. Assessing the integration of information system components. 1st international workshop on Interoperability of heterogeneous information systems (IHIS'05), 2005.

[WFMC 12] WfMC 2012, Workflow Management Coalition, Workflow Standard. Process Definition Interface, XML Process Definition Language. WFMC-TC-1025. V 2.2 . 24 Fevrier 2012.

[WFMC 99] WfMC 1999, Workflow Management Coalition - Terminology and Glossary - WFMC-TC-1011 (patent).

[White 04] White, S. 2004. Process Modeling Notations and Workflow Patterns.

[Windchill 13] Solution PLM PTC Windchill. Accessible: <http://fr.ptc.com/product/windchill/>. [31/01/13].

[Zimmermann 04] Zimmermann, O. Krogdahl, P. and Gee, C. Elements of Service-Oriented Analysis and Design. An Interdisciplinary Modeling Approach for SOA Projects. Accesible: <http://www.ibm.com/developerworks/library/ws-soad1/>, 2004.

[zur Muehlen 08] zur Muehlen, M. Business Process Management Standards - An Overview. 2008.

[zur Muehlen 08a] Zur Muehlen, M. & Recker, J. 2008, How Much Language is Enough? Theoretical and Practical Use of the Business Process Modeling Notation, In 20th International Conference on Advanced Information Systems engineering (CAISE 2008).

Annexe A : Exemple de fiches descriptives des services de conception de produit et des services fonctionnels PLM – Version en anglais des deux catalogues de services

1. Exemples de fiches descriptives des services de conception de produit

- Fiche descriptive du service GestionAffaire

<u>Service GestionAffaire</u>		
<u>Operation</u>	<u>Entrée</u>	<u>Sortie</u>
Diffuser le cahier des charges client	adresseDuParticipant	referenceCdC
Evaluer le cahier des charges client	referenceCdC	referenceCdC
Valider le cahier de charges client	referenceCdC	referenceCdC
Consulter le cahier des charges	referenceCdC	
Demander une évaluation du cahier de charges client	adresseDuParticipant, referenceCdC	
Demander une validation cahier des charges	adresseDuParticipant, referenceCdC	
Diffuser Plan client	adresseDuParticipant	refPlanClient
Evaluer Plan Client	refPlanClient	refPlanClient
Valider le Plan Client	refPlanClient	refPlanClient
Consulter Plan Client	refPlanClient	
Demander une évaluation du Plan Client	adresseDuParticipant, refPlanClient	
Demander une validation du Plan Client	adresseDuParticipant, refPlanClient	
Elaborer devis	referenceCdC	refDevis
Elaborer devis	refDemandeModification	refDevis
Diffuser devis	adresseDuParticipant	refDevis
Evaluer Devis	refDevis	refDevis
Valider Devis	refDevis	refDevis
Demander devis	adresseDuParticipant, refCdC	
Demander devis	adresseDuParticipant, refDemandeModification	
Consulter devis	refDevis	
Demander l'évaluation d'un devis	adresseDuParticipant, refDevis	
Demander la validation d'un devis	adresseDuParticipant, refDevis	

- Fiche descriptive du service DefinitionTest

<u>Service DefinitionTest</u>		
Operation	Entrée	Sortie
Elaborer plan de test	refDemandeModification	refPlanDeTest
Elaborer plan de test	refSpecFonctionnelle	refPlanDeTest
Diffuser le plan de test	adresseDuParticipant	refPlanDeTest
Evaluer Plan de test	refPlanDeTest	refPlanDeTest
Valider plan de test	refPlanDeTest	refPlanDeTest
Demander plan de test	adresseDuParticipant, refSpecFonctionnelle	
Demander plan de test	adresseDuParticipant, refDemandeModification	
Consulter plan de test	refPlanDeTest	
Demander une évaluation du plan de test	adresseDuParticipant, refPlanDeTest	
Demander une validation du plan de test	adresseDuParticipant, refPlanDeTest	

- Fiche descriptive du service BEAnalyseFonctionnelle

<u>Service BEAnalyseFonctionnelle</u>		
Operation	Entrée	Sortie
Elaborer la specification fonctionnelle	refDemandeModification	refSpecFonctionnelle
Elaborer la specification fonctionnelle	refCdC	refSpecFonctionnelle
Demander une évaluation de la specification fonctionnelle	adresseDuParticipant, refSpecFonctionnelle	
Evaluer la specification fonctionnelle	refSpecFonctionnelle	refSpecFonctionnelle
Valider la specification fonctionnelle	refSpecFonctionnelle	refSpecFonctionnelle
Lancer l'etude fonctionnelle	adresseDuParticipant, refDemandeModification	
Lancer l'etude fonctionnelle	adresseDuParticipant, refCdC	
Consulter la specification fonctionnelle	refSpecFonctionnelle	
Diffuser la specification fonctionnelle	adresseDuParticipant	refSpecFonctionnelle
Demander une validation de la specification fonctionnelle	adresseDuParticipant, refSpecFonctionnelle	

- Fiche descriptive du service SpecificationTechnique

<u>Service SpecificationTechnique</u>		
Operation	Entrée	Sortie
Elaborer la specificationTechnique	refDemandeModification	refSpecTechnique
Elaborer la specificationTechnique	refCdC	refSpecTechnique
Diffuser la specification technique	adresseDuParticipant	refSpecTechnique
Evaluer la specification technique	refSpecTechnique	refSpecTechnique
Valider la specification technique	refSpecTechnique	refSpecTechnique
Lancer l'etude technique	adresseDuParticipant, refCdC	
Lancer l'etude technique	adresseDuParticipant, refDemandeModification	
Consulter la specification technique	refSpecTechnique	
Demander une évaluation de la specification technique	adresseDuParticipant, refSpecTechnique	
Demander une validation de la specification technique	adresseDuParticipant, refSpecTechnique	

2. Exemples de fiches descriptives des services fonctionnels PLM

- Fiche descriptive du service Participant

<u>Service Participant</u>		
Opération	Entrée	Sortie
CreerParticipant	nomParticipant	refParticipant
LierParticipantAuRole	refParticipant, refRole	
RajouterInfosParticipant	refParticipant, tel, adresseMail	
EnleverRoleDeParticipant	refParticipant, refRole	
EnleverUnParticipant	refParticipant	
ModifierInformationParticipant	refParticipant, tel, adresseMail	
AfficherInformationParticipant	refParticipant	tel, adresseMail, nom, prenom

- Fiche descriptive du service Document

<u>Service Document</u>		
Opération	Entrée	Sortie
DefinirNouveauDocument	nomDocument	refDocument
DefinirInfosDocument	refDocument,nomEditeur, nomDocument,	
AjouterAttachement	refDocument, URLAttachement	refAttachement
CreerDocumentDeModele	refModeleDocument, nomDocument	refDocument
EditerAttachement	refAttachement	
ModifierInfosDocument	refDocument, refEditeur, nomDocument,	
ExporterDocumentCommeModele	refDocument	refModeleDocument
AfficherInfosDocument	refModeleDocumen	refEditeur, nomDocument, dateCreation
NouvelleVersionDuDocument	refDocument	numVersion
ChangerStatutDocument	refDocument, nouvelStatut	
AfficherVersionDocument	refDocument	numVersion
AfficherHistoriqueDocument	refDocument	dateCreation, dateVersion
AfficherInfosAttachement	refAttachement	nomAttachement, dateCreation
OuvrirAttachement	refAttachement	
SupprimerAttachement	refAttachement	
DownloadAttachement	refAttachement	
UploadAttachement	URL	refAttachement
AjouterSignatureElectronique	refAttachement	
CheckOutDocument	refDocument	
CheckInDocument	refDocument	
ChercherDocument	refDocument	URL
ImprimerAttachement	refAttachement	
ConvertirAttachmentPDF	refAttachement	URLPdf
AfficherDocumentEditeur	refDocument	refEditeur
DetruireDocument	refDocument	

- Fiche descriptive du service Produit

<u>Service Produit</u>		
Opération	Entrée	Sortie
DefinirNouveauProduit	libelleProduit	refProduit
DefinirInfosProduit	refProduit, dateCreation, refEditeur	
CreerProduitDeModele	refModeleProduit, libelleProduit	refProduit
ModifierInfosProduit	refProduit, dateCreation, refEditeur,	
NouvelleVersionProduit	refProduit	numVersion
ChangerstatutProduit	refProduit, statutProduit	
NouvelleRevisionProduit	refProduit	numRevision
GenererDocumentationProduit	refProduit	refDocument
ExporterProduitCommeModele	refProduit	refModeleProduit
ChercherProduct	refProduit	URLProduit
AfficherInfosProduit	refProduit	versionProduit, statutProduit,
AfficherAttachmentProduit	refProduit	refAttachement
VoirEditeurProduit	refProduit	refEditeur
VoirHistoriqueProduit	refProduit	dateCreation, dateVersion
DetruireProduit	refProduit	
AfficherVersionProduit	refProduit	numVersion
CheckInProduit	refProduit	
CheckOutProduit	refProduit	

- Fiche descriptive du service Nomenclature

<u>Service Nomenclature</u>		
Opération	Entrée	Sortie
LierArticleProduit	refProduit, refArticle, typeLien	refLien
LierDocumentProduit	refDocument, refProduit, typeLien	refLien
LierDocumentArticle	refDocument, refArticle, typeLien	refLien
ComparerStructuresProduits	refProduit, refProduit	
DefinirQuantiteLien	refLien, quantite	
ModifierQuantiteLien	refLien, quantite	
AfficherStructureArticle	refArticle	
AfficherStructureProduit	refProduit	
AfficherCasD'emploiArticle	refArticle	refProduit
AfficherCasD'emploiDocument	refDocument	refArticle, refProduit
DetruireLienArticleProduit	refLien	
DetruireLienDocumentProduit	refLien	
DetruireLienDocumentArticle	refLien	
CopierStructureArticleCommeModele	refArticle	refModeleStructureArticle
CopierStructureProduitCommeModele	refProduit	refModeleStructureProd

3. Version en anglais des services de conception de produits

- Version en anglais des services de conception de produits (PDS en anglais)

BusinessManagement	MarketAnalysis	ProjetManagement
DistributeSpecificationSheet	ElaborateMarketAnalysisSheet	ElaborateProjectPlannification
EvaluateSpecificationSheet	DistributeMarketAnalysisSheet	DistributeProjectSchedule
ValidateSpecificationSheet	EvaluateMarketAnalysisSheet	EvaluateProjectSchedule
ConsultSpecificationSheet	ValidateMarketAnalysisSheet	ValidateProjectSchedule
SpecificationSheetEvaluationRequest	InitiateMarketAnalysisSheetElaboration	ProjectPlanificationRequest
SpecificationSheetValidationRequest	ConsultMarketAnalysisSheet	ConsultProjectSchedule
DistributeCustomerPlan	MarketAnalysisSheetEvaluationRequest	ProjectScheduleEvaluationRequest
EvaluateCustomerPlan	MarketAnalysisSheetValidationRequest	ProjectScheduleValidationRequest
ValidateCustomerPlan		RessourceAllocationToProject
ConsultCustomerPlan		DistributeProjectRessourceAllocation
CustomerPlanEvaluationRequest		EvaluateProjectRessourceAllocation
CustomerPlanValidationRequest		ValidateProjectRessourceAllocation
ElaborateQuotation		ProjectRessourcesAllocationRequest
DistributeQuotation		ConsultProjectRessourceAllocation
EvaluateQuotation		ProjectRessourceEvaluationRequest
ValidateQuotation		ProjectRessourceValidationRequest
QuotationElaborationRequest		CreateProject
ConsultQuotation		NotifyProjectStartup
QuotationEvaluationRequest		ProjectEvaluation
QuotationValidationRequest		ProjectValidation
BomManagement	DesignDepartementFunctionalAnalysis	DesignDepartementDysfonctionnalAnalysis
CreateBOM	ElaborateFunctionalSpecificationSheet	ElaborateDysfonctionnalAnalysis
DistributeBOM	DistributeFunctionalSpecificationSheet	DistributeDysfonctionnalAnalysis
EvaluateBOM	EvaluateFunctionalSpecificationSheet	EvaluateDysfonctionnalAnalysis
ValidateBOM	ValidateFunctionalSpecificationSheet	ValidateDysfonctionnalAnalysis
BOMElaborationRequest	FunctionalSpecificationSheetElaborationRequest	DysfonctionnalAnalysisElaborationRequest
ConsultBOM	ConsultFunctionalSpecificationSheet	ConsultDysfonctionnalAnalysis
BOMEvaluationRequest	FunctionalSpecificationEvaluationRequest	DysfonctionnalAnalysisEvaluationRequest
BOMValidationRequest	FunctionalSpecificationValidationSheet	DysfonctionnalAnalysisValidationRequest
DesignManagement	PrototypeRealisation	Material
ElaborateProductPlan	CreatePrototype	chooseMaterials
DistributeProductPlan	LaunchPrototypeCreation	distributeMaterialChoice
EvaluateProductPlan	DistributePrototype	MaterialChoiceEvaluationRequest
ValidateProductPlan	AskForThePrototype	ValidateMaterialChoice
ProductPlanElaborationRequest		MaterialChoiceRequest
ConsultProductPlan		ConsultMaterial
ProductPlanEvaluationRequest		MaterialChoiceEvaluationRequest
ProductPlanValidationRequest		MaterialChoiceValidationRequest
ElaborateDetailPlan		
DistributeDetailPlan		
EvaluateDetailPlan		
ValidateDetailPlan		
DetailPlanElaborationRequest		
ConsultDetailPlan		
DetailPlanEvaluationRequest		
DetailPlanValidationRequest		
Elaborate3DProductModel		
Distribute3DProductModel		
Evaluate3DProductModel		
Validate3DProductModel		
3DProductModelElaborationRequest		
Consult3DProductModel		
3DProductModelEvaluationRequest		
3DProductModelValidationRequest		
Elaborate3DPartModel		
Distribute3DPartModel		
Evaluate3DPartModel		
Validate3DPartModel		
3DPartModelElaborationRequest		
Consult3DPartModel		
3DPartModelEvaluationRequest		
3DPartModelValidationRequest		
ElaborateSimulation		
distributeSimulation		
evaluateSimulation		
validateSimulation		
SimulationElaborationRequest		
ConsultSimulation		
SimulationEvaluationRequest		
SimulationElaborationRequest		
StandardReference	TestDefinition	TechnicalSpecification
DistributeNorm	ElaborateTestPlan	ElaborateTechniqueSpecification
EvaluateNorm	DistributeTestPlan	DistributeTechniqueSpecification
selectNorm	EvaluateTestPlan	EvaluateTechniqueSpecification
ConsultNorm	ValidateTestPlan	ValidateTechniqueSpecification
NormEvaluationRequest	TestPlanElaborationRequest	TechniqueSpecificationElaborationRequest
NormDecisionRequest	ConsultTestPlan	ConsultTechniqueSpecification
ElaborateWorkingProcess	TestPlanEvaluationRequest	TechniqueSpecificationEvaluationRequest
DistributeWorkingProcess	TestPlanValidationRequest	TechniqueSpecificationValidationRequest
EvaluateWorkingProcess		
validateWorkingProcess		
WorkingProcessElaborationRequest		
ConsultWorkingProcess		
WorkingProcessEvaluationRequest		
WorkingProcessValidationRequest		
TestProto	StandardComponent	ChangeManagement
ElaborateValidationReport	ChooseStandardComponent	ElaborateChangeRequest
DistributeValidationReport	DistributeStandardComponentChoice	DistributechangeRequest
EvaluateValidationReport	EvaluateStandardComponentChoice	EvaluerChangeRequest
ValidateValidationReport	ValidateStandardComponentChoice	ValidateChangeRequest
ValidationReportElaborationRequest	StandardComponentChoiceRequest	ConsultChangeRequest
consultValidationReport	ConsultStandardComponentList	ChangeRequestEvaluationRequest
ValidationReportEvaluationReport	StandardComponentChoiceValidationRequest	ChangeRequestDecisionRequest
ValidationReportValidationRequest	StandardComponentChoiceEvaluationRequest	
TestPrototype		
ValidatePrototype		
PrototypeEvaluationRequest		
PrototypeValidationRequest		
SolutionIdea		
ProposeSolutionIdea		
DistributeSolutionIdea		
Evaluer un principe de solution		
ValidateSolutionIdea		
SolutionIdeaRequest		
ConsultSolutionIdea		
SolutionIdeaEvaluationRequest		
SolutionIdeaValidationRequest		

4. Version en anglais des services fonctionnels PLM

- Version en anglais des services fonctionnels PLM

Document	Product	Compenent	Nomenclature
NewDocument (type)	DefineNewProduct	DefineComponent	LinkComponentToProduct
DefineDocumentData	DefinePartData	DefineComponentData	LinkDocumentToProduct
AddDocumentAttachment	CreateFromProduct	CreateFromComponent	LinkDocumentToComponent
NewDocumentFromModel	UpdateProductData	UpdateComponentData	UpdateLinkCPQuantity
EditDocumentAttachement	NewProductVersion	NewComponentVersion	UpdateLinkDPQuantity
CompleteDocumentData	ChangeProductStatus	NewComponentRevision	UpdateLinkDCQuantity
ExportDocumentToModel	NewProductRevision	ChangeComponentStatus	DisplayComponentStructure
DisplayDocumentData	EditProductToDocumentTemplate	EditComponentToDocumentTemplate	DisplayProductStructure
NewDocumentVersion	ExportProductToModel	ExportComponentToModel	LinkProductToProject
ChangeDocumentStatus	SearchProduct	SearchComponent	BrowsingUpComponent
NewDocumentRevision	DisplayProductData	DisplayComponentData	BrowsingUpDocument
UpdateDocumentData	DisplayProductAttachement	DisplayComponentAttachement	MoveComponentFromProduct
DisplayDocumentHistoric	DisplayProductEditor	DisplayComponentEditor	MoveComponentFromComponent
DisplayAttachementData	DisplayProductHistoric	DisplayComponentHistoric	MoveDocumentFromProduct
OpenDocAttachment	DeleteProduct	DeleteComponent	MoveDocumentFromComponent
DeleteDocAttachment	GetProductVersion	GetComponentVersion	CompareProductsStructure
DownloadAttachment	LockProduct	LockComponent	CompareComponentsStructure
UploadAttachment	UnlockProduct	UnlockComponent	CompareProductStructure
AddElectronicSignature			CompareComponentStructure
CheckOutDocument	Processes	Indicator	CopyProductStructure
CheckInDocument	DisplayAchievedStep	getProcessExecutionTime	CopyComponentstructure
SearchDocument	GetCurrentStep	getStepExecutionTime	GetComponentTopParent
PrintAttachment	GetCurrentStepStatus	compareExecutionInstanceEtape	
PrintAttachmentToPDF	GetActorStepRole	displayNumberProcessParticipant	CAO
ExportDocumentToModel	AddUserActorToStep		RetrieveASMStructure
DisplayDocumentEditor	AddUserInformToStep	Organization	RetrievePartQuantity
DeleteDocument	GetStepDecisionActor	getParticipantIDfromUserLogin	VisualizeCAO
GetDocumentVersion	SendPopUpForNotification	chegeParticipantInformation	
	AddDocumentToProcess	addRole	
Right	GetCandidateActorForStepStatuts	deleteParticipant	
AffectRole	RemoveUserStepActor	addParticipant	
deleteRole	GetStepDefaultActor	deleteRoleFromParticipant	
defineRoleAction	PutNotificationOnTheDesktop	displayParticipantInformation	
mooveActionFromRole	GetDelegationList/replacingActors		
	NotifyActorByE-Mail		
	ActorConfrentation		

Annexe B : Fiches de correspondance Métier/Fonctionnel et les orchestrations fonctionnelles résultat de la transformation des opérations métier – cas d'application

1. Fiches de correspondance métier/fonctionnel des opérations composant le processus de traitement des modifications.

- Fiche correspondance Métier/Fonctionnel de l'opération métier demanderPrincipeDeSolution

Fiche de correspondance Metier/Fonctionnel : DP_DemanderUnPrincipeDeSolution

Operation Metier : DemanderUnPrincipeDeSolution

Service Metier : DefinitionPrincipeDeSolution

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrir attachement(demandeDeModif)	Document
convertirAttachementPDF	Document
ImprimerAttachement	Document
NotifierParEmail	Processus et Etape
EnvoyerPopUpPourNotification	Processus et Etape

Patron : DemanderUnPrincipeDeSolutionPatron

```

    graph LR
      A([OuvrirAttachement]) --> B([convertirAttachementPDF])
      B --> C{XOR}
      C --> D([NotifierParEmail])
      C --> E([EnvoyerPopUpPourNotification])
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

```

    graph LR
      A([OuvrirAttachement]) --> B([ImprimerAttachement])
      B --> C([NotierActeurDeMainEnMain])
  
```

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier ProposerPrincipeDeSolution

Fiche de correspondance Métier/Fonctionnel : DP_ProposerPrincipeDeSolution

Operation Metier : ProposerPrincipeDeSolution

Service Metier : DefinitionPrincipeDeSolution

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement(demandeDeModif)	Document
DefinirNouveauDocument	Document
DefinirInfosDocument	Document
uploadAttachement	Document
creerDocumentDeModele	Document
editerAttachement	Document
LierDocumentDocument(principeDeSol,demandedeModif)	Nomenclature

Patron : ProposerPrincipeDeSolutionPatron

```

graph LR
    A(afficherAttachement) --> B(DefinirNouveauDocument)
    B --> C(DefinirInfosDocument)
    C --> D(CreerFichePrincipeSolution)
    D --> E(uploadAttachement)
    E --> F(LierDocumentDocument)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

```

graph LR
    A(CreerDocumentDeModele) --> B(DefinirInfosDocument)
    B --> C(EditerAttachement)
    C --> D(LierDocumentDocument)
  
```

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier evaluerDemandeDeModification

Fiche de correspondance Metier/Fonctionnel : DP_EvaluerDemandeDeModification
Operation Metier EvaluerDemandeDeModification
Service Metier : GestionDeModification
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrir attachement(demandeDeModif)	Document
chercherProduit(refProd)	Produit
afficherStructureProduit(refProd)	Nomenclature
afficherCasD'emploiProduit(refProd)	Nomenclature
definirNouveauDocument(EvaluationDoc)	Document
definirInfosDocument(EvaluationDoc, date)	Document
uploadAttachement	Document
lierDocumentDocument(demandeModif, EvaluationDoc)	Document

Patron : EvaluerDemandeDeModificationPatron

Fragments d'orchestration fonctionnelle Predefines :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier demanderDecisionDemandeDeModification

Fiche de correspondance Metier/Fonctionnel : DP_DemanderDecisionDemandeDeModification
Operation Metier DemanderDecisionDemandeDeModification
Service Metier : GestionDeModification
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrir attachement(demandeDeModif)	Document
convertirAttachementPDF	Document
ImprimerAttachement	Document
NotifierParEmail	Processus et Etape
EnvoyerPopUpPourNotification	Processus et Etape

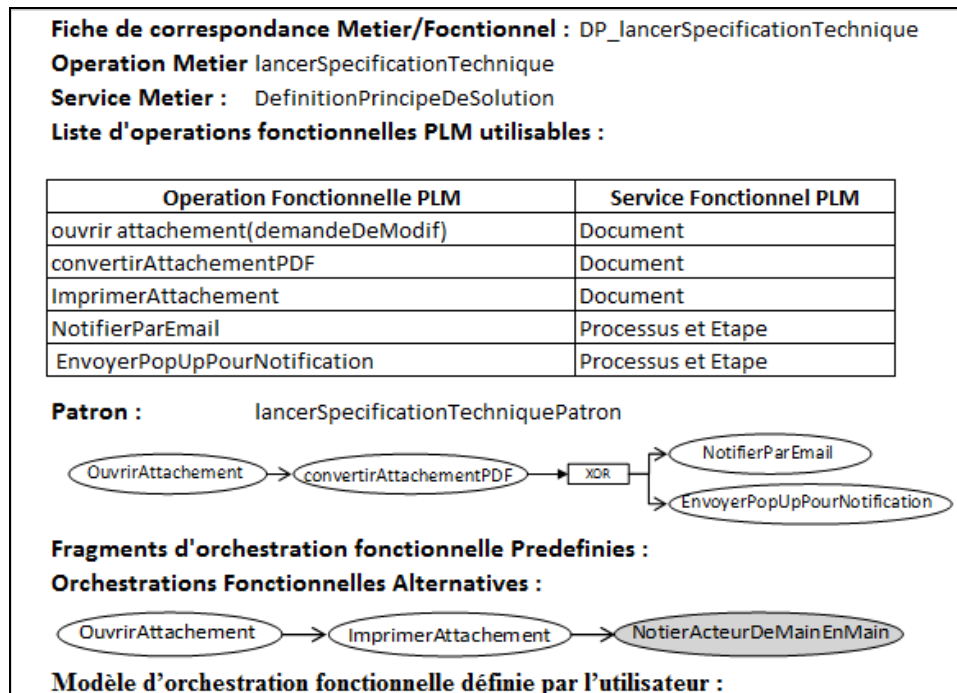
Patron : DemanderDecisionDemandeDeModificationPatron

Fragments d'orchestration fonctionnelle Predefines :

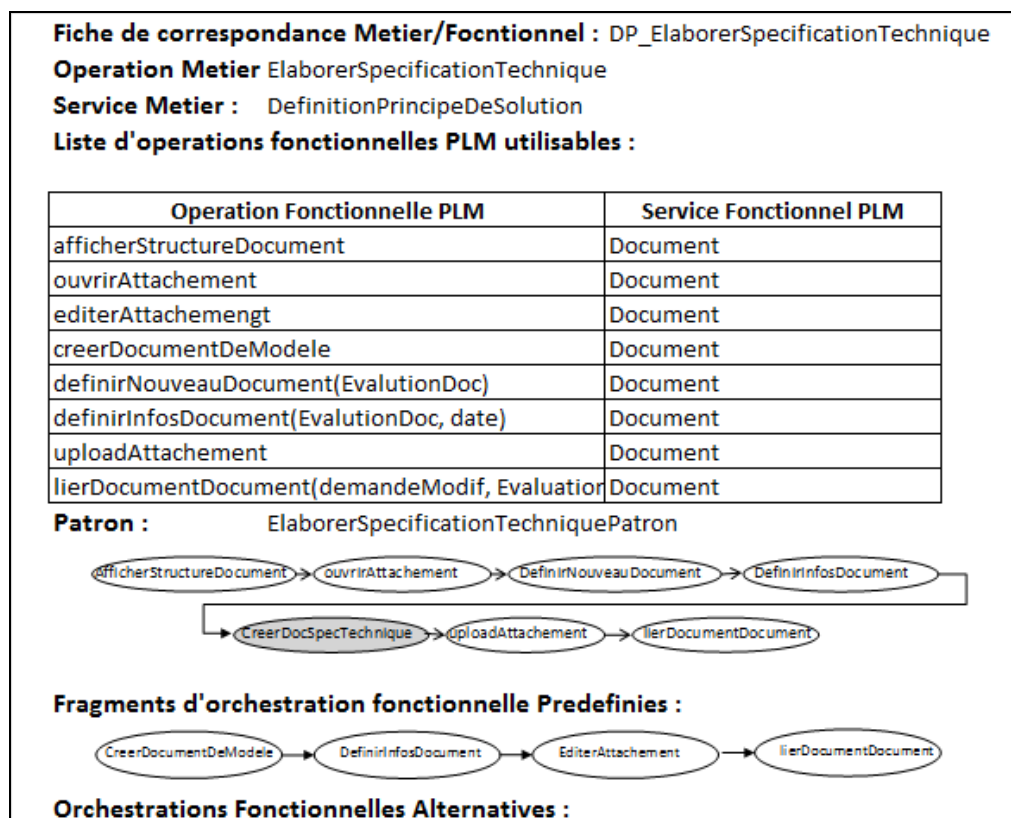
Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

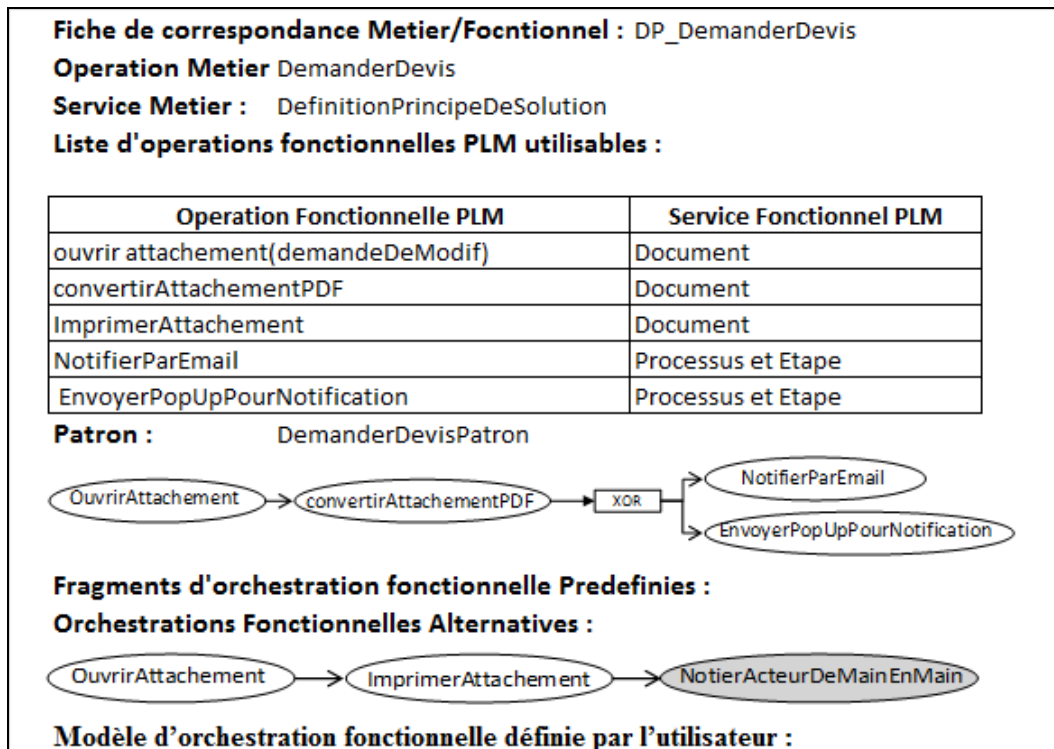
- Fiche correspondance Métier/Fonctionnel de l'opération métier lancerSpecificationTechnique



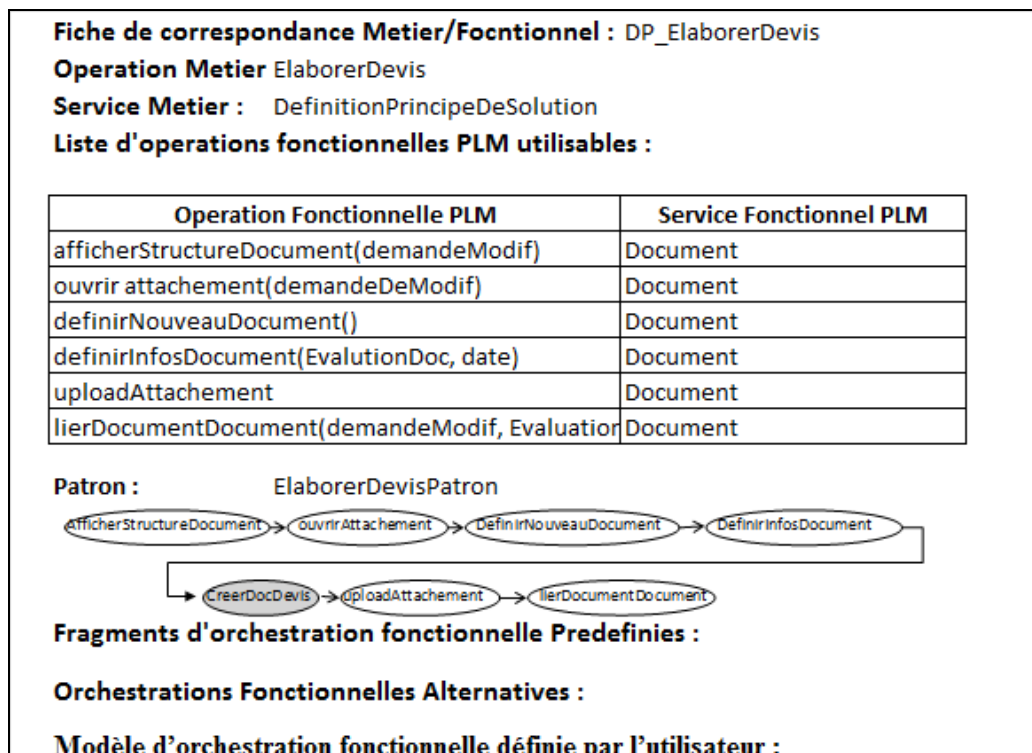
- Fiche correspondance Métier/Fonctionnel de l'opération métier elaborerSpecificationTechnique



- Fiche correspondance Métier/Fonctionnel de l'opération métier demanderDevis



- Fiche correspondance Métier/Fonctionnel de l'opération métier elaborerDevis



- Fiche correspondance Métier/Fonctionnel de l'opération métier validerDemandeDeModification

Fiche de correspondance Métier/Fonctionnel : DP_ValiderDemandeDeModification
Operation Metier : ValiderDemandeDeModification
Service Metier : Gestion De Modification
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement (demandeModif)	Document
checkInDocument((demandeModif)	Document
changerStatutDocument(demandeModif,statut)	ProcessusEtEtape
checkoutDocument(demandeModif)	Document
AjouterSignatureElectronique(demandeModif, sig)	Document

Patron : ValiderDemandeDeModificationPatron

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

- Fiche correspondance Métier/Fonctionnel de l'opération métier definirProcedureDeTravail

Fiche de correspondance Métier/Fonctionnel : DP_DefinirProcedureDeTravail
Operation Metier : DefinirProcedureDeTravail
Service Metier : ReferenceStandards
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrir attachement(demandeDeModif)	Document
chercherDocument(demandeDeModif)	Document
afficherStructureProduit(refProd)	Nomenclature
definirNouveauDocument(EvaluationDoc)	Document
definirInfosDocument(EvaluationDoc, date)	Document
uploadAttachement	Document
lierDocumentDocument(demandeModif, EvaluationDoc)	Document

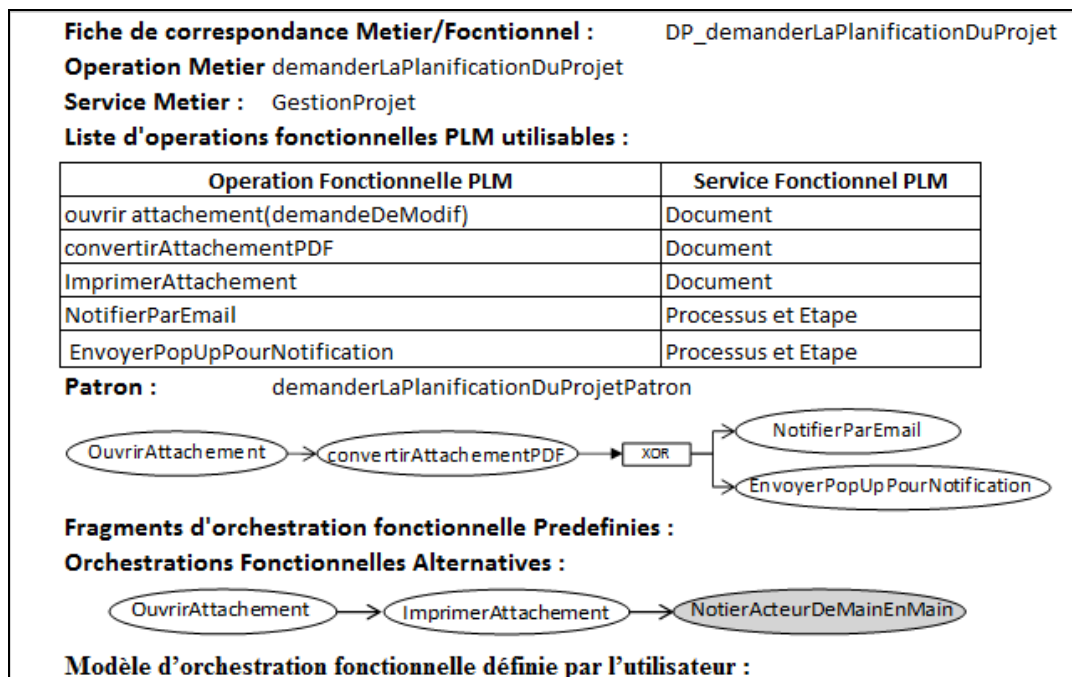
Patron : DefinirProcedureDeTravailPatron

Fragments d'orchestration fonctionnelle Predefinies :

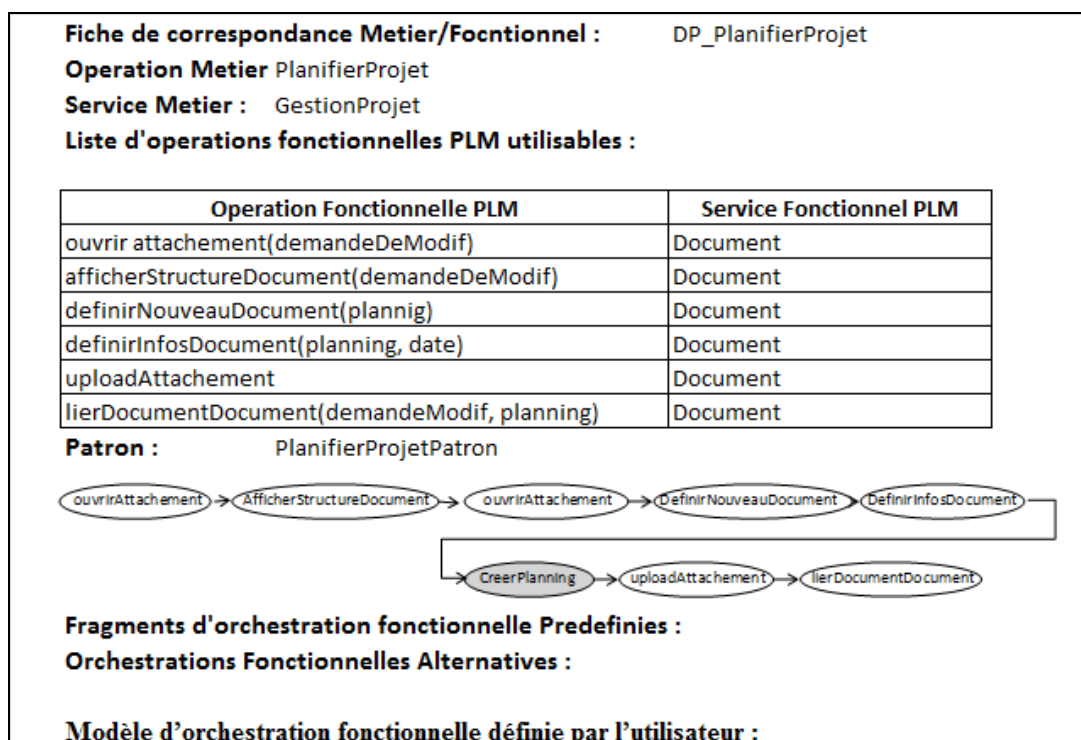
Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier demanderLaPlanificationDuProjet



- Fiche correspondance Métier/Fonctionnel de l'opération métier planifierProjet



- Fiche correspondance Métier/Fonctionnel de l'opération métier demanderAffectationRessources

Fiche de correspondance Metier/Focntionnel : DP_DemanderAffectationRessources
Operation Metier DemanderAffectationRessources
Service Metier : GestionProjet
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrir attachement(planning)	Document
convertirAttachementPDF	Document
ImprimerAttachement	Document
NotifierParEmail	Processus et Etape
EnvoyerPopUpPourNotification	Processus et Etape

Patron : DemanderAffectationRessourcesPatron

```

graph LR
    A([OuvrirAttachement]) --> B([convertirAttachementPDF])
    B --> C{XOR}
    C --> D([NotifierParEmail])
    C --> E([EnvoyerPopUpPourNotification])
  
```

Fragments d'orchestration fonctionnelle Predefinies :
Orchestrations Fonctionnelles Alternatives :
Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier affecterLesRessourcesAuPlanning

Fiche de correspondance Metier/Focntionnel : DP_AffecterLesRessourcesAuPlanning
Operation Metier AffecterLesRessourcesAuPlanning
Service Metier : GestionProjet
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
checkInDocument(Planning)	Document
downloadAttachement	Document
afficherRole	RoleEtAction
afficherCasD'emploiRole	RoleEtAction
afficherInfosParticipant	Participant
uploadAttachement	Document
checkInDocument(Planning)	Document

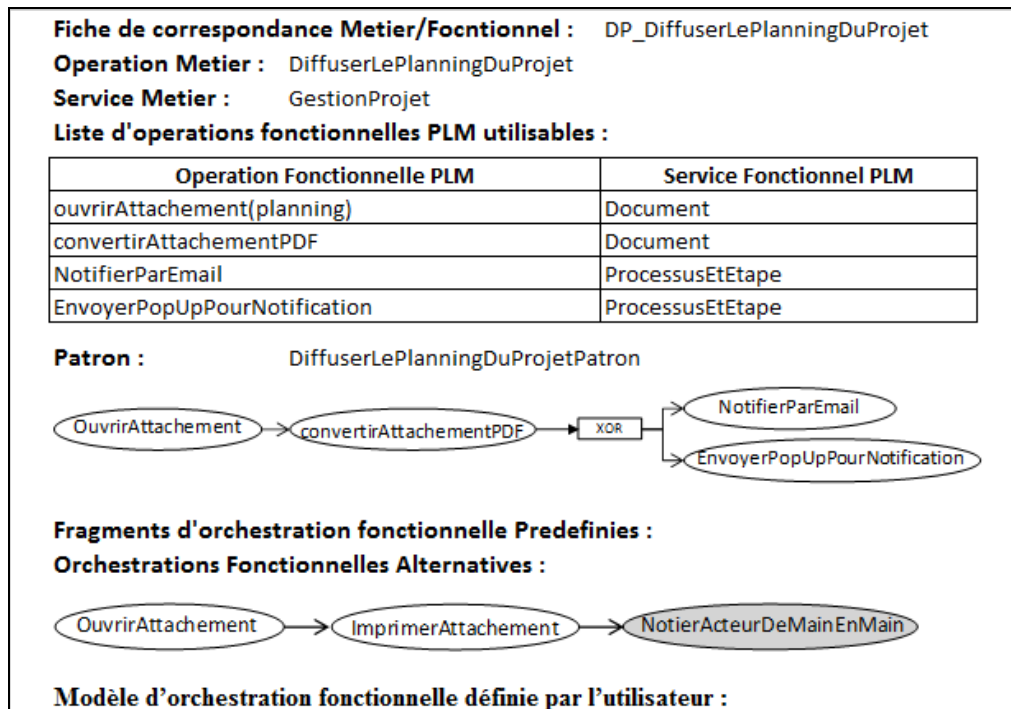
Patron : AffecterLesRessourcesAuPlanningPatron

```

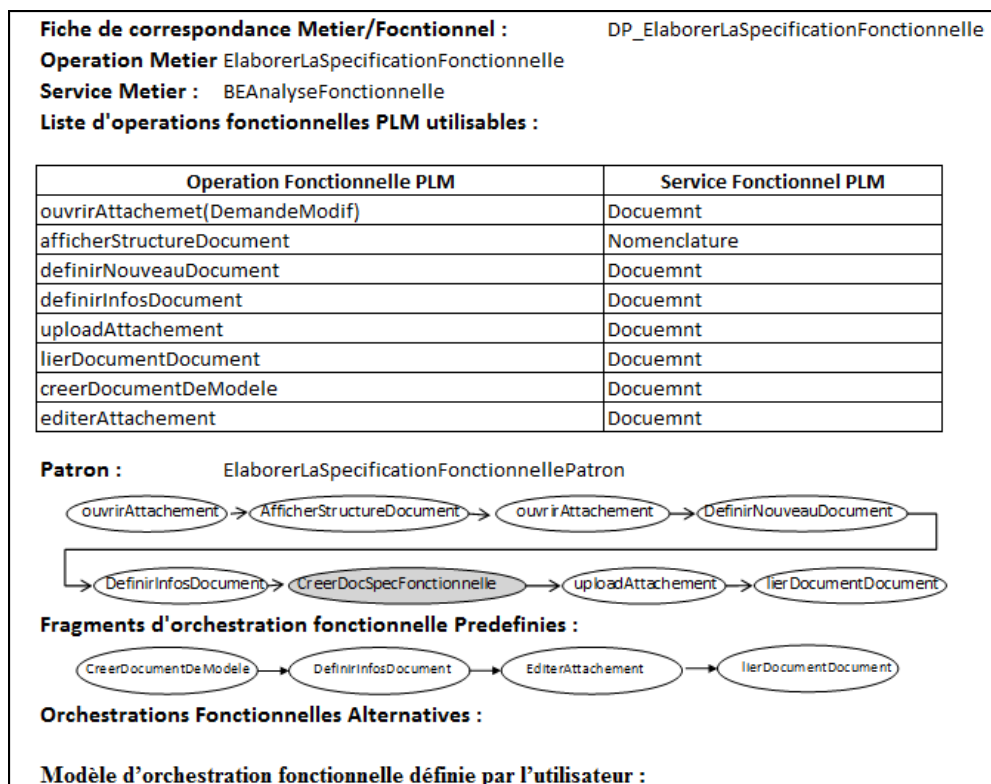
graph LR
    A([checkInDocument]) --> B([DownloadAttachement])
    B --> C([AfficherRole])
    C --> D{AND}
    D --> E([AfficherCasD'emploiRole])
    D --> F([AfficherInfosParticipant])
    E --> G([AffecterRessourcesPlanning])
    F --> G
    H([UploadAttachement]) --> I([checkOutDocument])
  
```

Fragments d'orchestration fonctionnelle Predefinies :
Orchestrations Fonctionnelles Alternatives :
Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier diffuserLePlanningDuProjet



- Fiche correspondance Métier/Fonctionnel de l'opération métier elaborerLaSpecificationFonctionnelle



- Fiche correspondance Métier/Fonctionnel de l'opération métier diffuserLaSpecificationFonctionnelle

Fiche de correspondance Metier/Fonctionnel : DP_DiffuserLaSpecificationFonctionnelle

Operation Metier DiffuserLaSpecificationFonctionnelle

Service Metier : BEAnalyseFonctionnelle

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement	Document
convertirAttachementPDF	Document
NotifierParEmail	ProcessusEtEtape
envoyerPopUpPourNotification	ProcessusEtEtape
inprimerAttachement	Document

Patron : DiffuserLaSpecificationFonctionnellePatron

```

graph LR
    A([OuvrirAttachement]) --> B([convertirAttachementPDF])
    B --> XOR{XOR}
    XOR --> C([NotifierParEmail])
    XOR --> D([EnvoyerPopUpPourNotification])
  
```

Fragments d'orchestration fonctionnelle Predefinies :

```

graph LR
    A([OuvrirAttachement]) --> B([ImprimerAttachement])
    B --> C([NotierActeurDeMainEnMain])
  
```

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier elaborerLaSpecificationDysfonctionnelle

Fiche de correspondance Metier/Fonctionnel : DP_ElaborerLaSpecificationDysFonctionnelle

Operation Metier ElaborerLaSpecificationDysFonctionnelle

Service Metier : BEAnalyseDysFonctionnelle

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachemet(DemandeModif)	Docuemnt
afficherStructureDocument	Nomenclature
definirNouveauDocument	Docuemnt
definirInfosDocument	Docuemnt
uploadAttachement	Docuemnt
lierDocumentDocument	Docuemnt
creerDocumentDeModele	Docuemnt
editerAttachement	Docuemnt

Patron : ElaborerLaSpecificationDysFonctionnellePatron

```

graph LR
    A([ouvrirAttachement]) --> B([AfficherStructureDocument])
    B --> C([ouvrirAttachement])
    C --> D([DefinirNouveauDocument])
    D --> E([DefinirInfosDocument])
    E --> F([CreerDocSpecDysFonctionnelle])
    F --> G([uploadAttachement])
    G --> H([lierDocumentDocument])
    H --> A
  
```

Fragments d'orchestration fonctionnelle Predefinies :

```

graph LR
    A([CreerDocumentDeModele]) --> B([DefinirInfosDocument])
    B --> C([EditerAttachement])
    C --> D([lierDocumentDocument])
  
```

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier diffuserLaSpecificationSysfonctionnelle

Fiche de correspondance Metier/Fonctionnel : DP_DiffuserLaSpecificationDysFonctionnelle

Operation Metier DiffuserLaSpecificationDysFonctionnelle

Service Metier : BEAnalyseDysFonctionnelle

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement	Document
convertirAttachementPDF	Document
NotifierParEmail	ProcessusEtEtape
envoyerPopUpPourNotification	ProcessusEtEtape
inprimerAttachement	Document

Patron : DiffuserLaSpecificationDysFonctionnellePatron

```

    graph LR
      A([ouvrirAttachement]) --> B([convertirAttachementPDF])
      B --> XOR{XOR}
      XOR --> C([NotifierParEmail])
      XOR --> D([EnvoyerPopUpPourNotification])
  
```

Fragments d'orchestration fonctionnelle Predefinies :

```

    graph LR
      A([ouvrirAttachement]) --> B([ImprimerAttachement])
      B --> C([NotierActeurDeMainEnMain])
  
```

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier elaborerLePlanDeTest

Fiche de correspondance Metier/Fonctionnel : DP_ElaborerLePlanDeTest

Operation Metier ElaborerLePlanDeTest

Service Metier : DefinitionTest

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement(DemandeModif)	Docuemnt
afficherStructureDocument	Nomenclature
definirNouveauDocument	Docuemnt
definirInfosDocument	Docuemnt
uploadAttachement	Docuemnt
lierDocumentDocument	Docuemnt
creerDocumentDeModele	Docuemnt
editerAttachement	Docuemnt

Patron : ElaborerLePlanDeTestPatron

```

    graph LR
      A([ouvrirAttachement]) --> B([AfficherStructureDocument])
      B --> C([ouvrirAttachement])
      C --> D([DefinirNouveau Document])
      D --> E([DefinirInfosDocument])
      E --> F([CreerDocPlanDeTest])
      F --> G([uploadAttachement])
      G --> H([lierDocumentDocument])
  
```

Fragments d'orchestration fonctionnelle Predefinies :

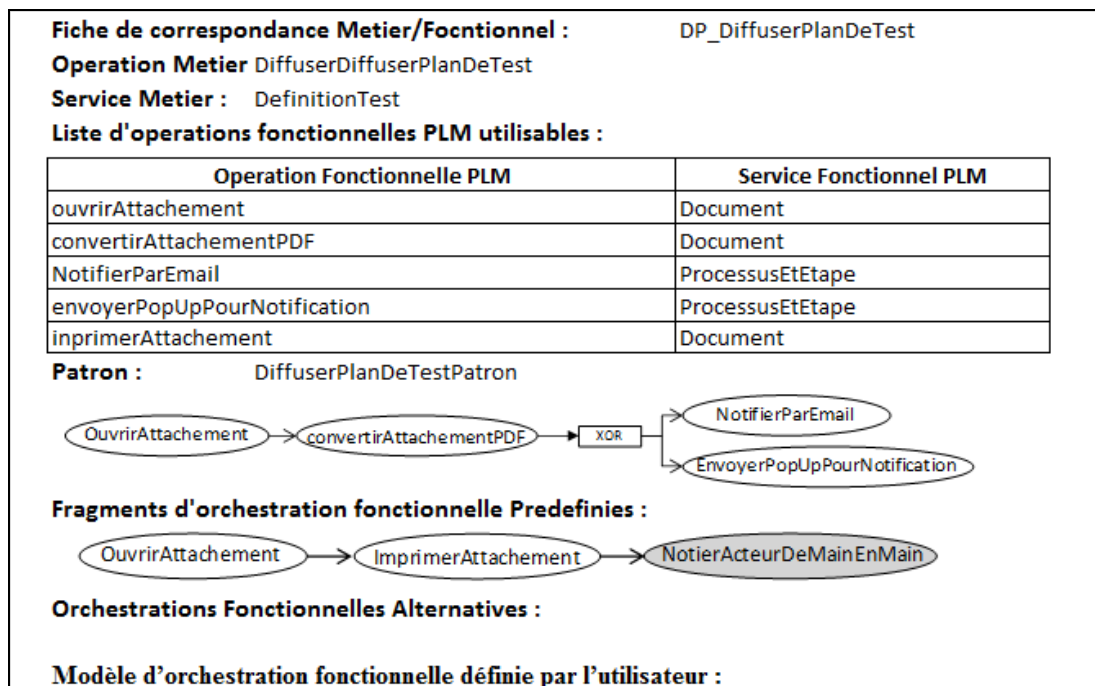
```

    graph LR
      A([CreerDocumentDe Modele]) --> B([DefinirInfosDocument])
      B --> C([EditerAttachement])
      C --> D([lierDocumentDocument])
  
```

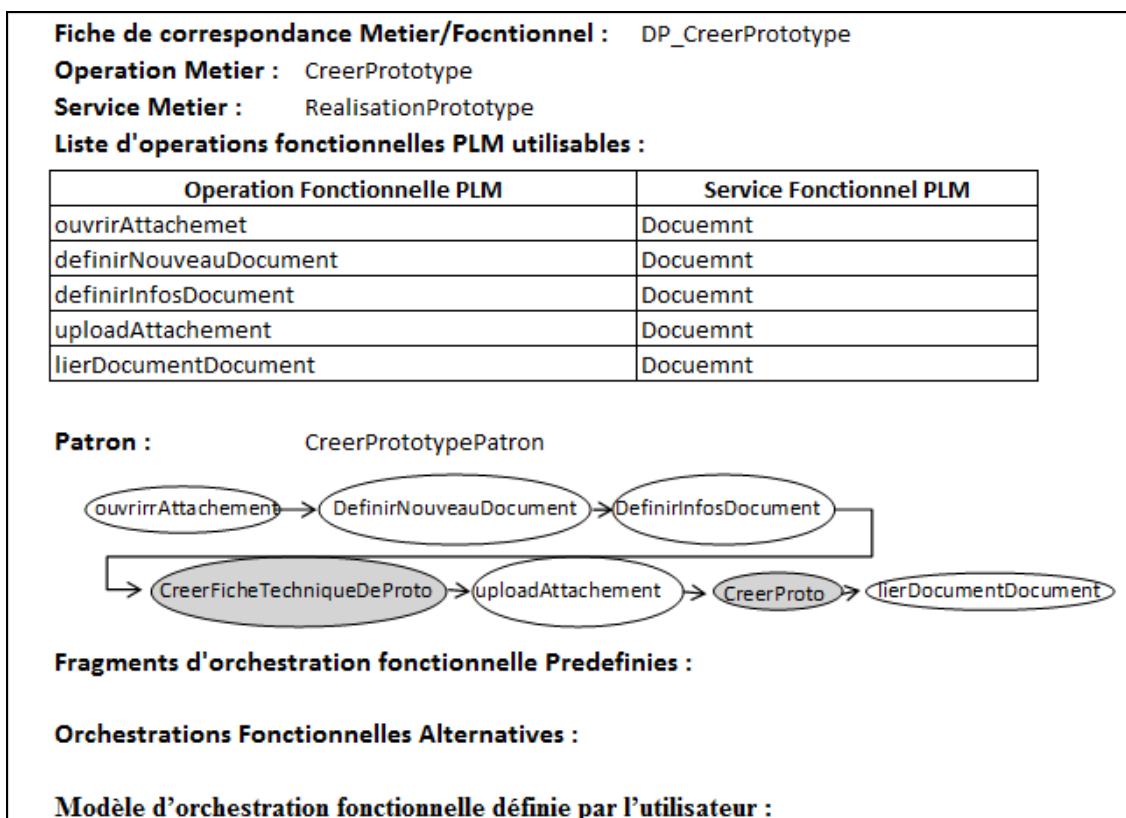
Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier diffuserPlanDeTest



- Fiche correspondance Métier/Fonctionnel de l'opération métier creerPrototype



- Fiche correspondance Métier/Fonctionnel de l'opération métier diffuserPrototype

Fiche de correspondance Metier/Focntionnel : DP_DiffuserPrototype

Operation Metier : DiffuserPrototype

Service Metier : RealisationPrototype

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement	Document
convertirAttachementPDF	Document
NotifierParEmail	ProcessusEtEtape
envoyerPopUpPourNotification	ProcessusEtEtape
inprimerAttachement	Document

Patron : DiffuserPrototypePatron

```

graph LR
    A([ouvrirAttachement]) --> B([convertirAttachementPDF])
    B --> XOR{XOR}
    XOR --> C([NotifierParEmail])
    XOR --> D([EnvoyerPopUpPourNotification])
  
```

Fragments d'orchestration fonctionnelle Predefinies :

```

graph LR
    A([ouvrirAttachement]) --> B([ImprimerAttachement])
    B --> C([NotierActeurDeMainEnMain])
  
```

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier consulterPrincipeDeSolution

Fiche de correspondance Metier/Focntionnel : DP_consulterPrincipeDeSolution

Operation Metier : consulterPrincipeDeSolution

Service Metier : DefinitionPrincipeDeSolution

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement(demandeModif)	Document
chercherDocument(PrincipeDESolution)	Document
AficcherStructureDocument	Document

Patron : consulterPrincipeDeSolutionPatron

```

graph LR
    A([ouvrirAttachement]) --> B([ChercherDocument])
    B --> C([AfficherStructure])
    C --> D([OuvrirAttachement])
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier consulterSpecificationFonctionnelle

Fiche de correspondance Métier/Fonctionnel : DP_ConsulterSpecificationFonctionnelle

Operation Metier : ConsulterSpecificationFonctionnelle

Service Metier : DefinitionTestBEAnalyseFonctionnelle

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
OuvrirAttachement(FicheTechnique)	Document
ChercherDocument(DemandeDeModif)	Document
AfficherStructure(DemandeDEModif)	Nomenclature
AfficherStructure(PrincipeDeSolution)	Nomenclature
Ouvrirattachement (SpecFonctionnelle)	Document

Patron : ConsulterSpecificationFonctionnellePatron

```

    graph LR
      A(ouvrirAttachement) --> B(ChercherDocument)
      B --> C(AfficherStructure)
      C --> D(AfficherStructure)
      D --> E(OuvrirAttachement)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier consulterSpecificationDysfonctionnelle

Fiche de correspondance Métier/Fonctionnel : DP_ConsulterSpecificationDysfonctionnelle

Operation Metier : ConsulterSpecificationDysfonctionnelle

Service Metier : BEAnalyseDysFonctionnelle

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
OuvrirAttachement(FicheTechnique)	Document
ChercherDocument(DemandeDeModif)	Document
AfficherStructure(DemandeDEModif)	Nomenclature
AfficherStructure(PrincipeDeSolution)	Nomenclature
Ouvrirattachement (SpecDysFonctionnelle)	Document

Patron : ConsulterSpecificationDysfonctionnellePatron

```

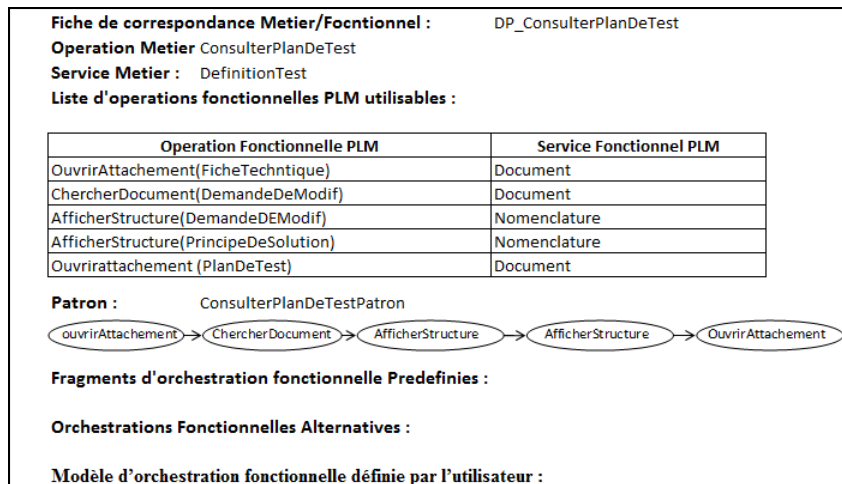
    graph LR
      A(ouvrirAttachement) --> B(ChercherDocument)
      B --> C(AfficherStructure)
      C --> D(AfficherStructure)
      D --> E(OuvrirAttachement)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

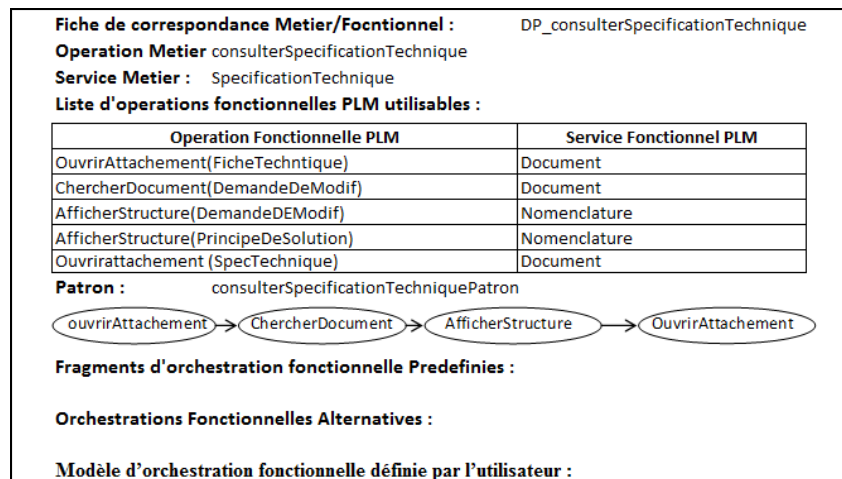
Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

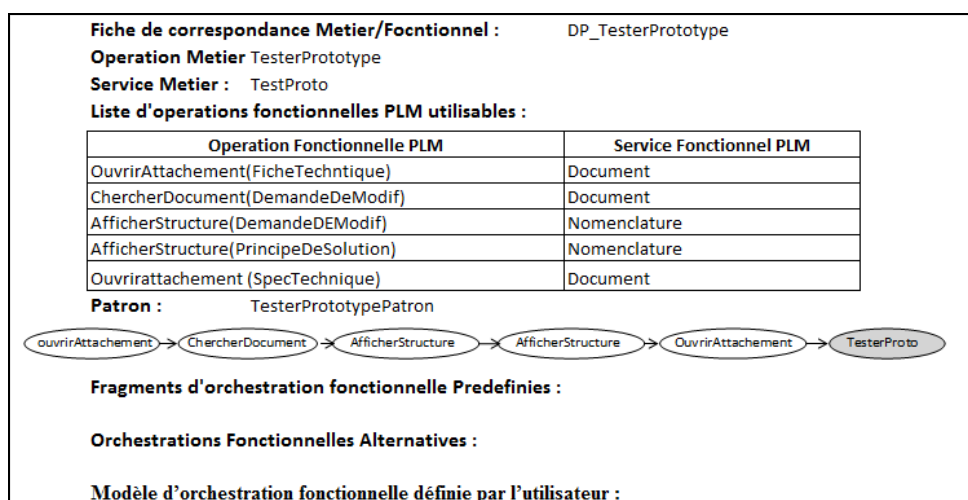
- Fiche correspondance Métier/Fonctionnel de l'opération métier consulterPlanDeTest



- Fiche correspondance Métier/Fonctionnel de l'opération métier consulterSpecificationTechnique



- Fiche correspondance Métier/Fonctionnel de l'opération métier testerProto



- Fiche correspondance Métier/Fonctionnel de l'opération métier élaborerLeRapportDeValidation

Fiche de correspondance Metier/Fonctionnel : DP_ElaborerLeRapportDeValidation

Operation Metier : ElaborerLeRapportDeValidation

Service Metier : TestProto

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
definirNouveauDocument	Docuemnt
definirInfosDocument	Docuemnt
uploadAttachement	Docuemnt
lierDocumentDocument	Nomenclature

Patron : ElaborerLeRapportDeValidationPatron

```

graph LR
    A(Definir Nouveau Document) --> B(Definir Infos Document)
    B --> C(Créer Rapport De Validation)
    C --> D(upload Attachement)
    D --> E(lier Document Document)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier demanderValidationPrincipeDeSolution

Fiche de correspondance Metier/Fonctionnel : DP_DemanderValidationPrincipeDeSolution

Operation Metier : DemanderValidationPrincipeDeSolution

Service Metier : DefinitionPrincipeDeSolution

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement	Document
convertirAttachementPDF	Document
NotifierParEmail	ProcessusEtEtape
envoyerPopUpPourNotification	ProcessusEtEtape
inprimerAttachement	Document

Patron : DemanderValidationPrincipeDeSolutionPatron

```

graph LR
    A(Ouvrir Attachement) --> B(convertir Attachement PDF)
    B --> XOR{XOR}
    XOR --> C(Notifier Par Email)
    XOR --> D(Envoyer Pop Up Pour Notification)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

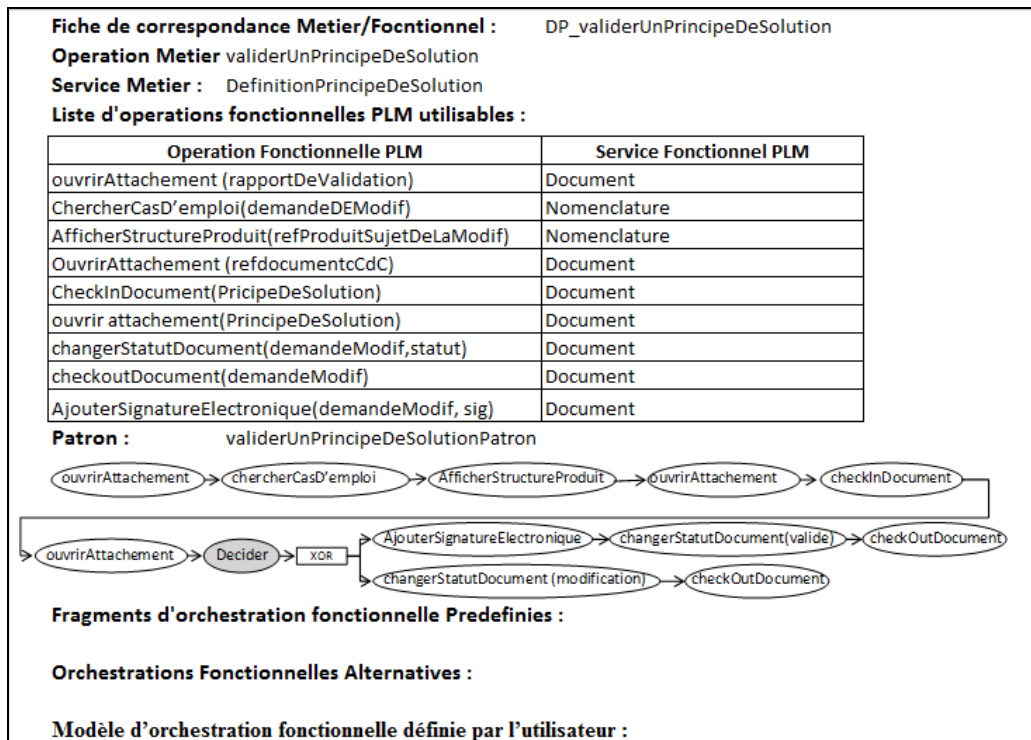
```

graph LR
    A(Ouvrir Attachement) --> B(Imprimer Attachement)
    B --> C(Notifier Acteur De Main En Main)
  
```

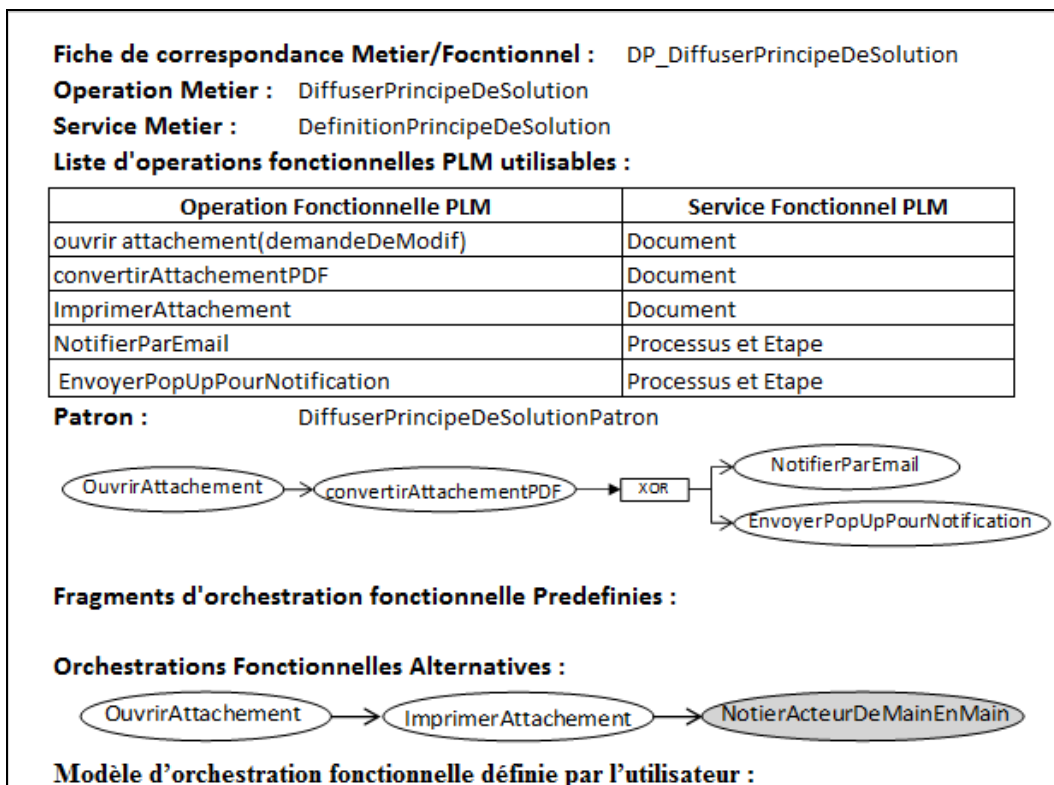
Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier validerPrincipeDeSolution



- Fiche correspondance Métier/Fonctionnel de l'opération métier diffuserPrincipeDeSolution



- Fiche correspondance Métier/Fonctionnel de l'opération métier creerBOM

Fiche de correspondance Metier/Focntionnel : DP_CreerBOM
Operation Metier CreerBOM
Service Metier : GestionNomenclature
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement(PrincipeDeSolution)	Document
NouvelleVersionProduit(refProduit)	Produit
DefinirInformationProduit	Produit
DefinirArticle	Article
DefinirInformationArticle	Article
LierArticleProduit	Nomenclature

Patron : CreerBOMPatron

```

    graph LR
      A(ouvrirAttachement) --> B(NouvelleVersionProduit)
      B --> C(definirInformationsProduit)
      C --> D(DefinirArticle)
      D --> E(DefinirInformationArticle)
      E --> F(lierArticleProduit)
    
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier creerModele3DPiece

Fiche de correspondance Metier/Focntionnel : DP_CreerModele3DPiece
Operation Metier CreerModele3DPiece
Service Metier : BEConception
Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement(PrincipeDeSolution)	Document
DefinirNouveauDocument	Document
DefinirInfosDocument	Document
uploadAttachement	Document
LierDocumentAArticle	Nomenclature
creerDocumentDeModele	Document
editerAttachement	Document

Patron : CreerModele3DPiecePatron

```

    graph LR
      A(ouvrirAttachement) --> B(DefinirNouveauDocument)
      B --> C(DefinirInfosDocument)
      C --> D(CreerModeleCADPiece)
      D --> E(uploadAttachement)
      E --> F(LierDocumentDocument)
    
```

Fragments d'orchestration fonctionnelle Predefinies :

```

    graph LR
      A(CreerDocumentDeModele) --> B(DefinirInfosDocument)
      B --> C(EditerAttachement)
      C --> D(LierDocumentDocument)
    
```

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier creerModele3DProduit

Fiche de correspondance Metier/Fonctionnel : DP_CreerModele3DProduit

Operation Metier CreerModele3DProduit

Service Metier : BEConception

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
ouvrirAttachement(PrincipeDeSolution)	Document
afficherStructureProduit	Nomenclature
DefinirNouveauDocument	Document
DefinirInfosDocument	Document
uploadAttachement	Document
LierDocumentAArticle	Nomenclature
creerDocumentDeModele	Document
editerAttachement	Document

Patron : CreerModele3DProduitPatron

```

    graph LR
      A(ouvrirAttachement) --> B(afficherStructureProduit)
      B --> C(DefinirNouveauDocument)
      C --> D(DefinirInfosDocument)
      D --> E(creerModeleCAOProduit)
      E --> F(uploadAttachement)
      F --> G(lierDocumentDocument)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

```

    graph LR
      A(CreerDocumentDeModele) --> B(DefinirInfosDocument)
      B --> C(EditerAttachement)
      C --> D(lierDocumentDocument)
  
```

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier validerModele3DPiece

Fiche de correspondance Metier/Fonctionnel : DP_ValiderModele3DPiece

Operation Metier ValiderModele3DPiece

Service Metier : BEConception

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
AffichernfosAttachement(refModele3DPiece)	Document
VisualiserIModeleCAO(refmodele3DPiece)	CAO
OuvrirAttachement(principeDeSolution)	Document
ModifierStatutDocument	Document

Patron : ValiderModele3DPiecePatron

```

    graph LR
      A(afficherInfosAttachement) --> B(visualiserModeleCAO)
      B --> C(ouvrirAttachement)
      C --> D(modifierStatutDocument)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier validerModele3DProduit

Fiche de correspondance Métier/Fonctionnel : DP_ValiderModele3DProduit

Operation Metier : ValiderModele3DProduit

Service Metier : BEConception

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
AffichernfosAttachement(refModele3DProduit)	Document
VisualiserIModeleCAO(refmodele3DProduit)	CAO
OuvrirAttachement(principeDeSolution)	Document
ModifierStatutDocuemnt	Document

Patron : ValiderModele3DProduitPatron

```

graph LR
    A(afficherInfosAttachement) --> B(visualiserModeleCAO)
    B --> C(ouvrirAttachement)
    C --> D(modifierStatutDocument)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier validerBOM

Fiche de correspondance Métier/Fonctionnel : DP_validerBOM

Operation Metier : validerBOM

Service Metier : GestionNomenclature

Liste d'operations fonctionnelles PLM utilisables :

Operation Fonctionnelle PLM	Service Fonctionnel PLM
AffichernfosAttachement(refProduit)	Document
VisualiserIModeleCAO(refmodele3DProduit)	CAO
OuvrirAttachement(principeDeSolution)	Document
ModifierStatutDocuemnt	Document

Patron : validerBOMPatron

```

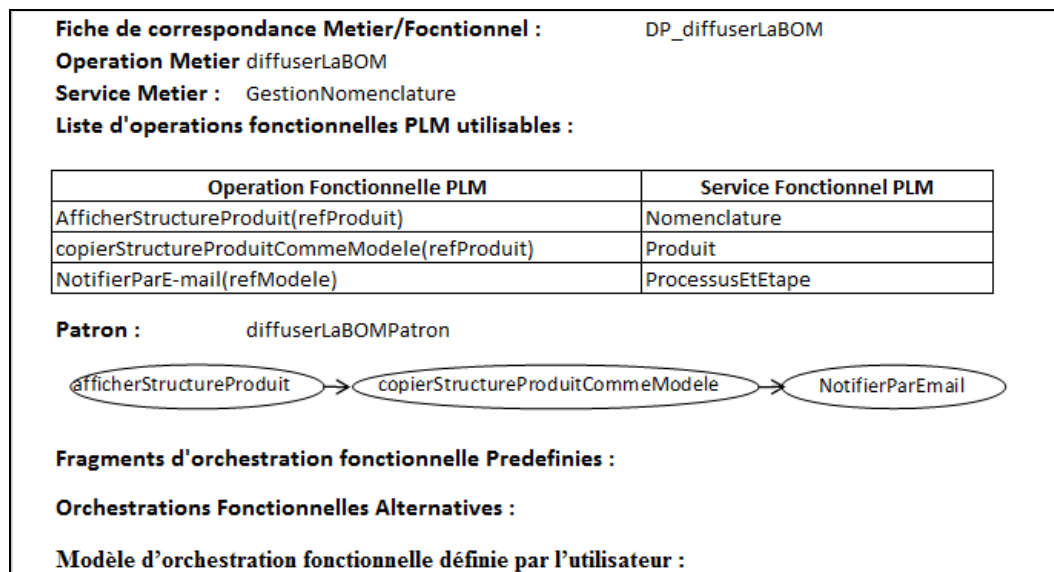
graph LR
    A(afficherStructureProduit) --> B(visualiserModeleCAOProduit)
    B --> C(ouvrirAttachement)
    C --> D(changerStatutProduit)
  
```

Fragments d'orchestration fonctionnelle Predefinies :

Orchestrations Fonctionnelles Alternatives :

Modèle d'orchestration fonctionnelle définie par l'utilisateur :

- Fiche correspondance Métier/Fonctionnel de l'opération métier diffuserLaBOM

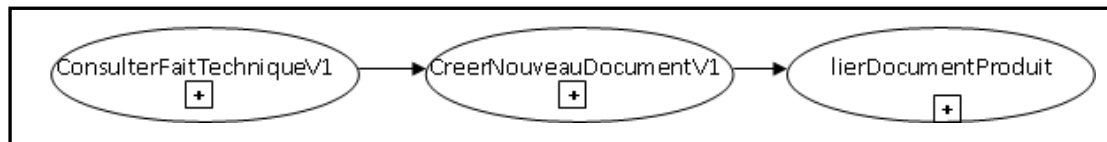


2. Transformation des opérations métier et orchestrations métier composants le processus de traitements des modifications

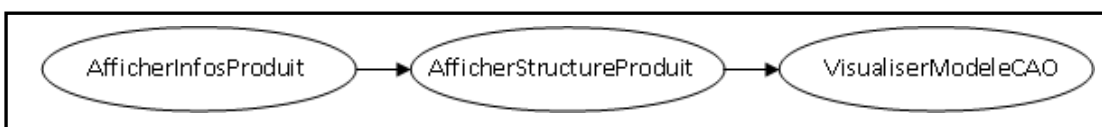
Dans cette section nous illustrons les différentes orchestrations fonctionnelles issues de la transformation des opérations métier et fragments d'orchestrations métier composant le processus de traitement des modifications.

2.1 Transformation des opérations métier composants le processus de traitements des modifications en orchestration fonctionnelle

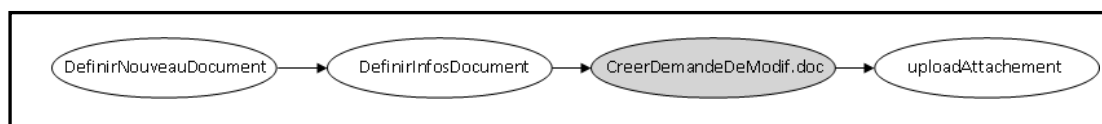
- Transformation de l'opération métier elaborerDemandeDeModification



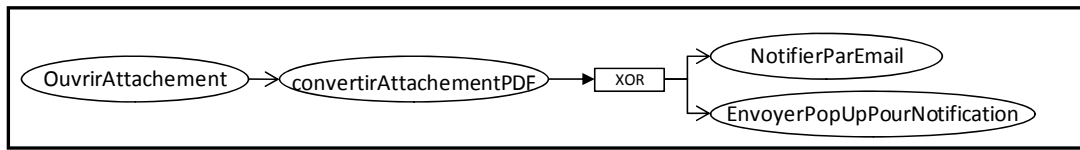
- Consulter fait techniqueV1



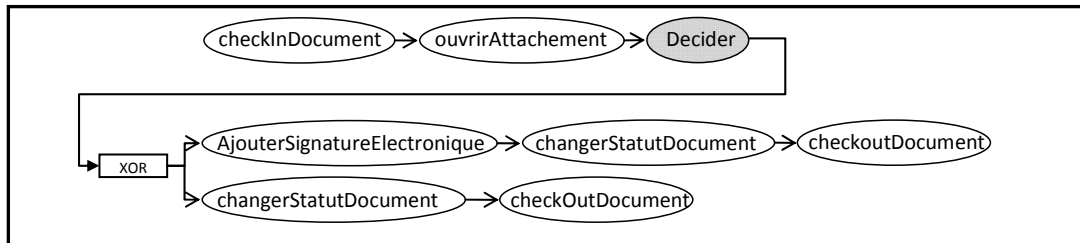
- CreerNouveauDocumentV1



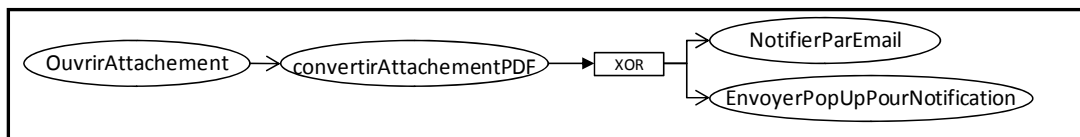
- Transformation de l'opération métier demanderEvaluationDemandeDeModification



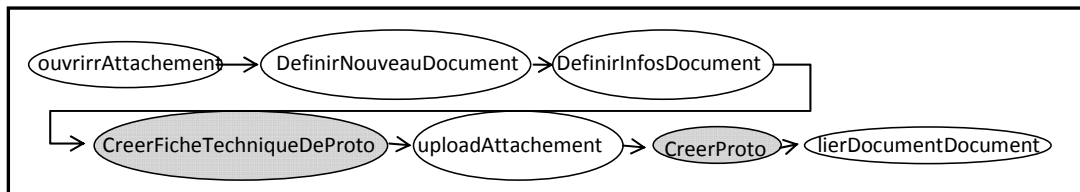
- Transformation de l'opération métier validerDemandeDeModification



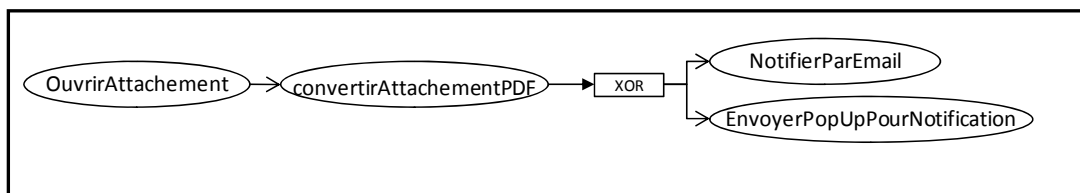
- Transformation de l'opération métier diffuserLePlanningDuProjet



- Transformation de l'opération métier creerPrototype

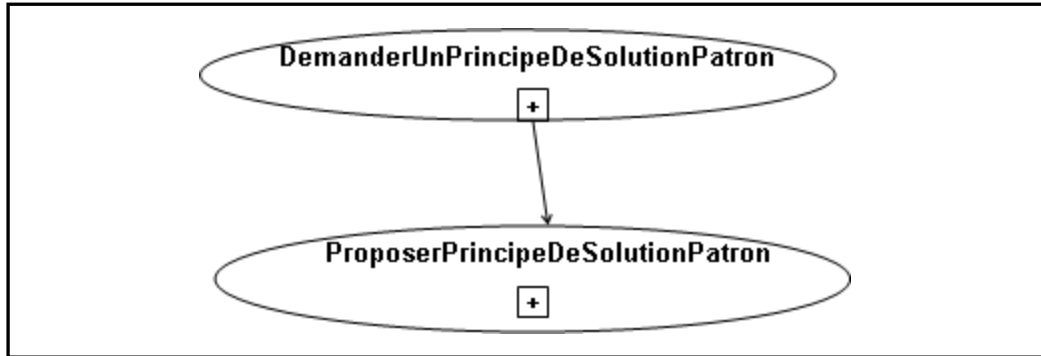


- Transformation de l'opération métier diffuserPrincipeDeSolution

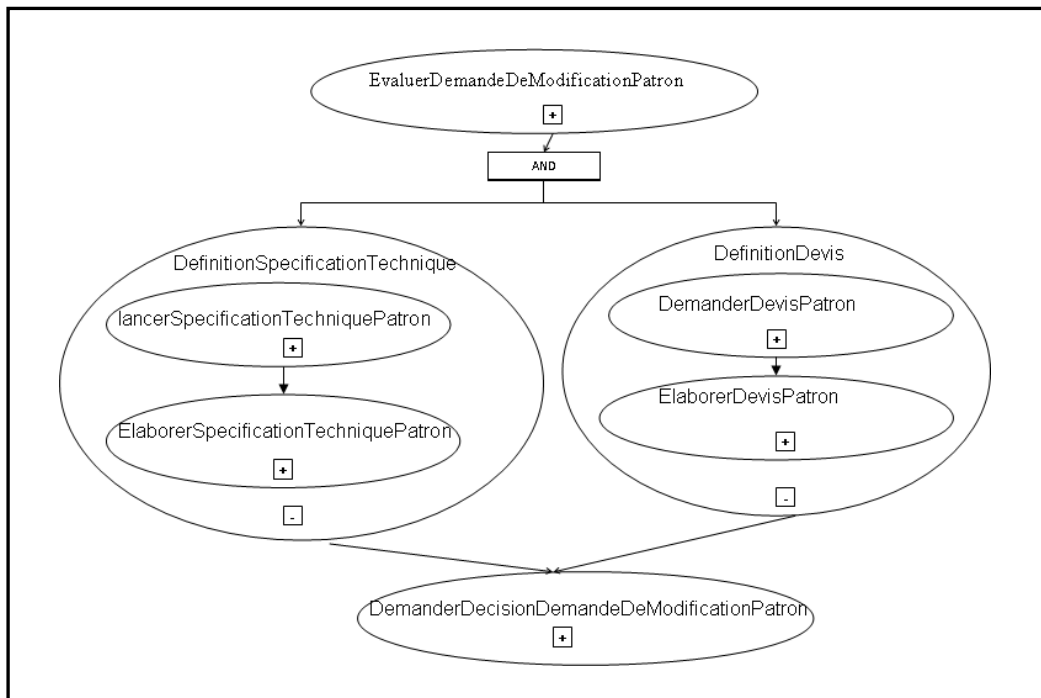


2.2 Transformation des orchestrations métier

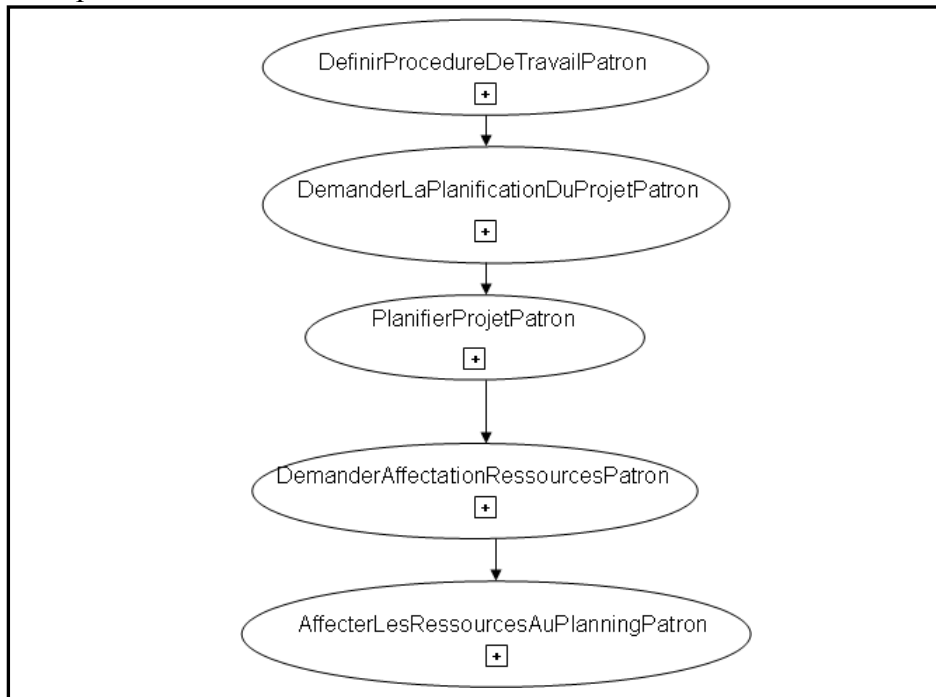
- Transformation de l'orchestration métier Definition Solution



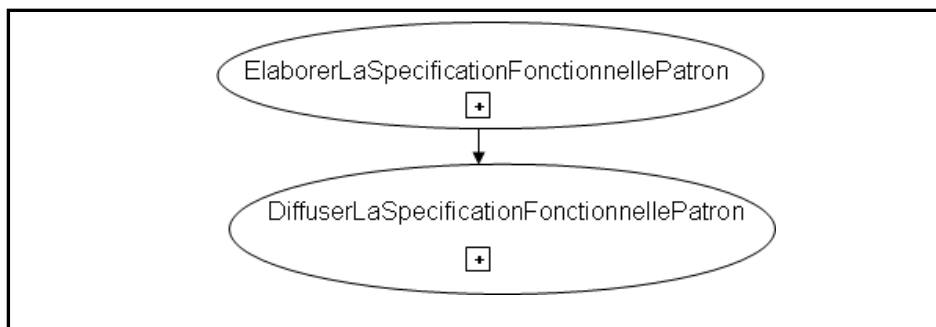
- Transformation de l'orchestration métier Definition Perimetre Et Objectif Du Probleme



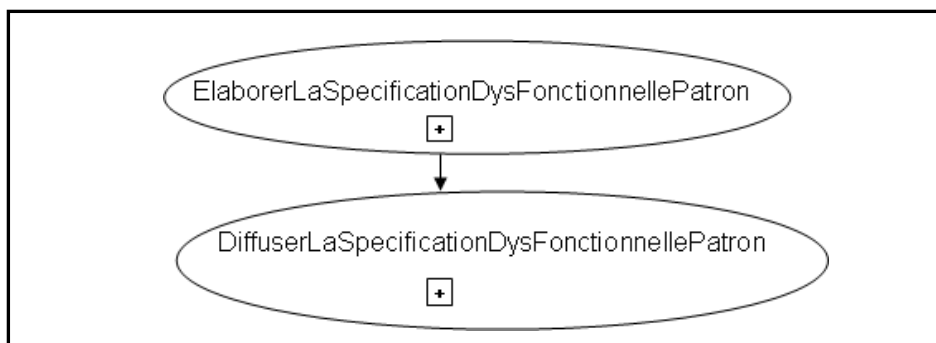
- Transformation de l'orchestration métier Ordonnancement Du Processus De Conception



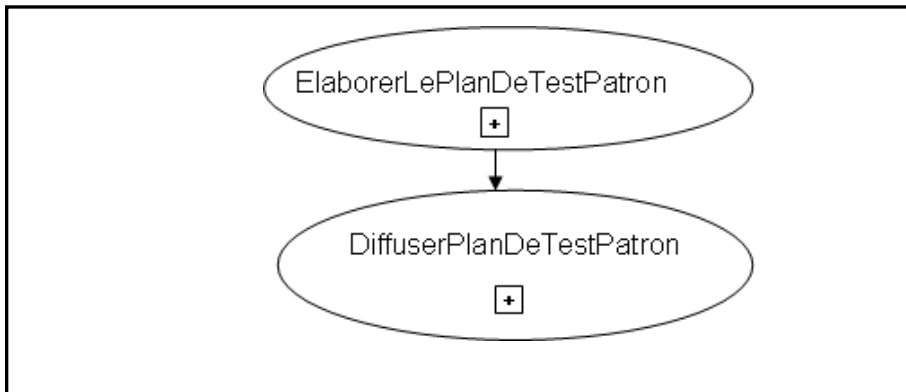
- Transformation de l'orchestration métier Analyse Fonctionnelle



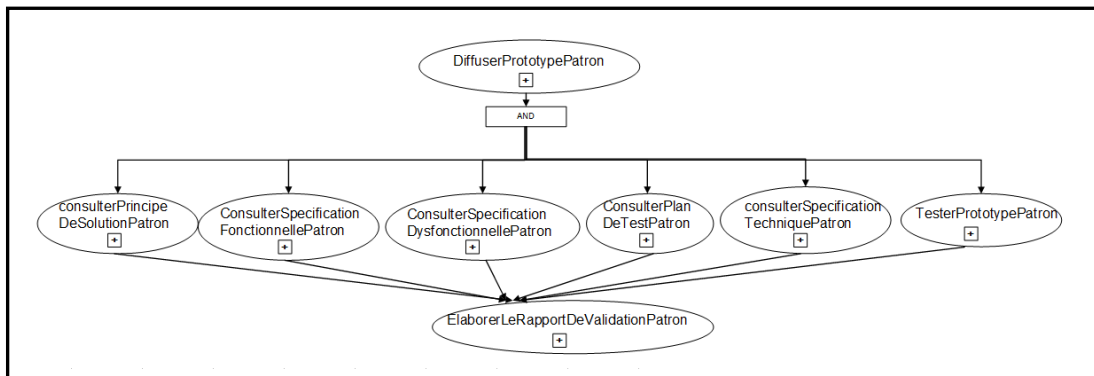
- Transformation orchestration métier Analyse Dysfonctionnelle



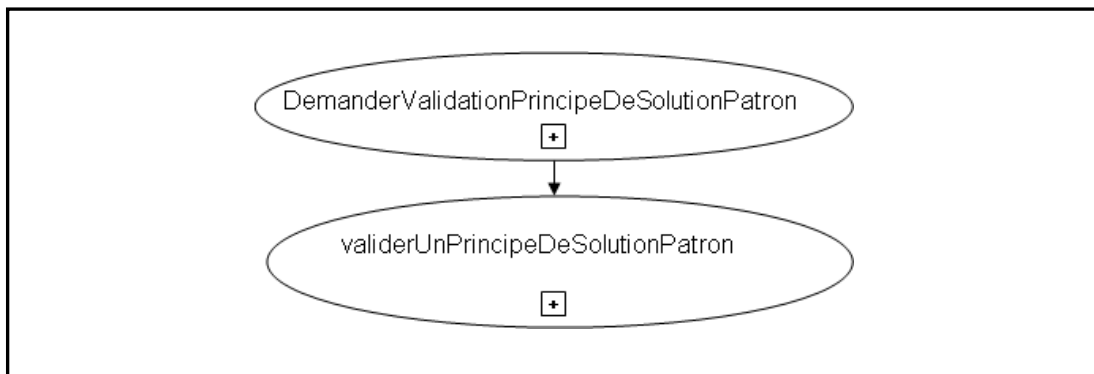
- Transformation de l'orchestration métier Analyse Test



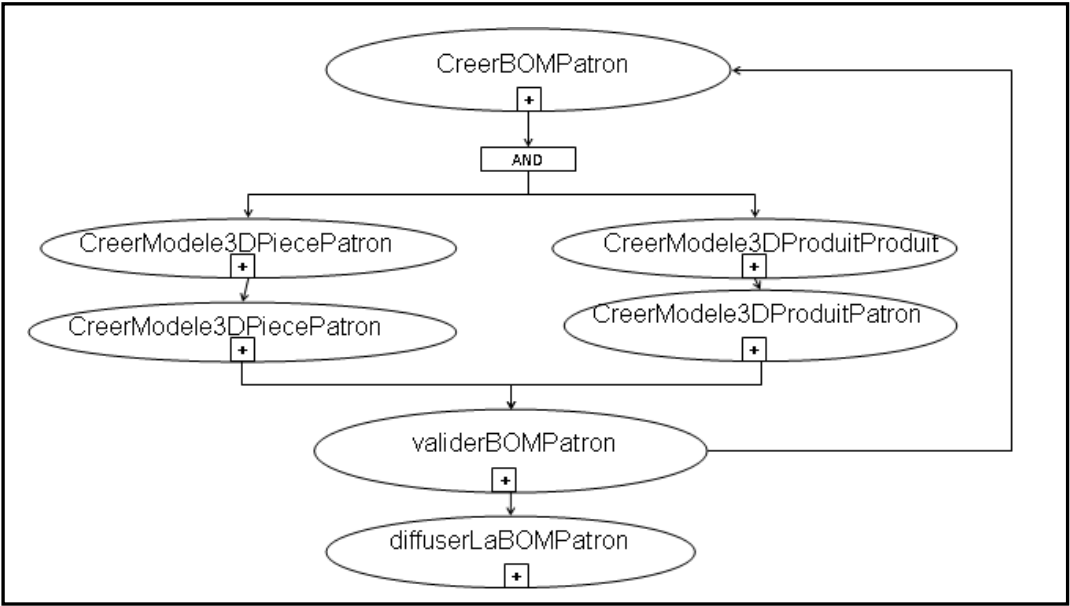
- Transformation de l'orchestration métier Evaluation D'une Solution



- Transformation de l'orchestration métier Validation D'une Solution



- Transformation de l'orchestration métier Conception D'une Solution



Résumé

La dynamique de l'offre et de la demande des produits manufacturiers ainsi que le raccourcissement de leurs cycles de vie obligent les entreprises industrielles à se doter de processus de développement produit dynamiques et agiles. Nos travaux se positionnent sur le support informatisé de ces processus de développement qui sont actuellement gérés par les systèmes PLM. L'objectif d'un support informatisé est d'accélérer le processus en automatisant la notification et la diffusion des informations. Il permet également de garder trace des opérations et décisions effectuées et d'accroître la standardisation des processus. Face à la rigidité des solutions actuellement proposées pour gérer les processus vis-à-vis des modifications survenant dans le processus, notre objectif est de proposer une approche permettant de modifier un processus en cours d'exécution sans devoir le redéfinir et le relancer dans son ensemble. Pour y parvenir, nous avons proposé, une approche qui décline une orientation services inspirée des architectures orientées services (SOA). Ces architectures permettent de définir des applications modulaires, en utilisant des services faiblement couplés. Notre objectif est de décliner une telle architecture exploitée essentiellement pour les systèmes logiciels et le Web, au niveau métier de l'entreprise afin de modéliser et d'exécuter de manière flexible des processus de conception de produits par composition de services réutilisables. Nous proposons une démarche d'identification des services du domaine métier des processus de conception produit et du domaine fonctionnel du PLM. Ces services sont organisés dans deux catalogues de services métiers et fonctionnels. Notre approche s'inscrit dans le cadre de l'Ingénierie Dirigée par les Modèles (IDM) avec une architecture de référence à trois niveaux et des mécanismes d'alignement entre les niveaux métier, fonctionnel et logiciel. Ces mécanismes d'alignement entre les niveaux permettent d'intégrer l'évolution et d'automatiser le déploiement d'un processus de conception du niveau métier aux niveaux fonctionnel et logiciel.

Mots-clés : Processus de conception, PLM, Approches Orientées Services, MDE, Flexibilité et Agilité

Abstract

To cope with market dynamic and shortened time to market, industrial companies need to implement an effective management of their design processes (DPs) and product information. Unfortunately, Product Lifecycle Management (PLM) systems which are dedicated to support design activities are not efficient as it might be expected. Indeed, DPs are changing, emergent and non deterministic, due to the business environment under which they are carried out. The aim of this work is to propose an alternative approach for flexible process support within PLM systems to facilitate the coupling with the environment reality. The purpose of a support system is to accelerate the process by automating the notification and dispatching of information and activities between actors. It also allows to keep track of transactions and decisions made and to increase processes standardization. Our goal is to propose a solution which allows process change at run-time without having to redefine and restart the whole of process activities. To achieve this, we proposed an approach based on service-oriented architectures (SOA). These architectures allow defining modular applications, using loosely coupled services. They are mainly exploited for software systems and Web development. Our goal is to decline such architectures at the business level of a company in order to perform flexible DPs deployment based on services reuse and composition. We propose an identification approach for business level services (product design services) and functional PLM services. These services are organized in two catalogs of business and functional services. Our approach is based on Model Driven Approach with three levels which propose alignment mechanisms between business, functional and technical levels. These alignment mechanisms between levels allow integrating change and automating design process deployment.

Keywords : Design process, PLM, SOA, MDA, Agility and Flexibility.