



HAL
open science

La gouvernance SOA pour une approche de conception de Système d'Information de Médiation : réconciliation non-fonctionnelle de services pour mettre en œuvre les processus métier

Sarra Zribi

► To cite this version:

Sarra Zribi. La gouvernance SOA pour une approche de conception de Système d'Information de Médiation : réconciliation non-fonctionnelle de services pour mettre en œuvre les processus métier. Gestion et management. Ecole des Mines d'Albi-Carmaux, 2014. Français. NNT : 2014EMAC0019 . tel-01233276

HAL Id: tel-01233276

<https://theses.hal.science/tel-01233276>

Submitted on 24 Nov 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

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

École Nationale Supérieure des Mines d'Albi-Carmaux conjointement avec l'INSA de Toulouse

Présentée et soutenue par :

Sarra ZRIBI

le vendredi 24 janvier 2014

Titre :

La gouvernance SOA pour une approche de conception de Système
d'Information de Médiation : réconciliation non-fonctionnelle de
services pour mettre en œuvre les processus métier

École doctorale et discipline ou spécialité :

EDSYS : Génie Industriel 4200046

Unité de recherche :

Centre Génie Industriel, École des Mines d'Albi-Carmaux

Directeur/trice(s) de Thèse :

Hervé PINGAUD

Jury :

Frédérique BIENNIER, Professeur, INSA Lyon, Rapporteur
Olivier PERRIN, Professeur, Université de Lorraine Nancy, Rapporteur
Chihab HANACHI, Professeur, Université de Toulouse 1 Capitole, Président
Hervé PINGAUD, Professeur, ISIS – Université J-F Champollion Albi, Directeur
Jean-Pierre LORRÉ, Directeur de recherche, Linagora, Examineur
Frédéric BÉNABEN, Maître assistant HDR, École des Mines d'Albi-Carmaux, Examineur

La gouvernance SOA pour une approche de conception de Système
d'Information de Médiation : réconciliation non-fonctionnelle de services
pour mettre en œuvre les processus métier



SOA governance in a design approach for Mediation Information System:
Non-functional reconciliation of services to implement business processes

À ma famille, mes amis (ies), qui par leur présence à mes cotés m'ont donné la confiance nécessaire pour accomplir ce travail.

À ma grand-mère « nana », qui me manque trop.

À mon oncle, qui aurait aimé lire cette thèse.

À toi, qui m'aimes.

REMERCIEMENTS

Trois années viennent de s'écouler et le moins que l'on puisse dire c'est que ça n'a pas été un long fleuve tranquille. C'est avec beaucoup de plaisir et un peu de nostalgie que je prends la plume pour remercier tous ceux qui ont rendu possible l'avènement de ce travail. Mais par où commencer ? Par qui commencer ? Tant de personnes ont une place ici qu'il m'est aujourd'hui difficile de n'en oublier aucune. Donc, je tiens à m'excuser par avance des éventuels oublis.

Je commence par le début de ce périple : mes parents sans qui je ne serais pas là aujourd'hui. Ils m'ont été d'un grand soutien moral et ont été pour moi d'immenses rayons de soleil dans les moments durs. Je tiens à leur exprimer ma gratitude infinie et je leur dédie ce travail de thèse de doctorat.

Je tiens à remercier chaleureusement mon directeur de thèse Pr. Hervé Pingaud, mon encadrant à l'école des Mines Frédérick Bénaben et mon encadrant au sein de Linagora Jean-Pierre Lorré. Merci pour la confiance qu'ils m'ont témoignée, pour le soutien qu'ils m'ont accordé et pour l'envie qu'ils m'ont donnée de faire de la recherche. Je leur suis reconnaissante également de m'avoir fait bénéficier de leur compétence, de leur efficacité, de leur assistance précieuse et de leur suivi empressé, parfois exigeant, tout en me laissant une grande indépendance. Je les remercie également pour leur patience, leur côté humain et pour leur aide lorsque les situations rendaient le travail de thèse difficile.

J'adresse mes vifs remerciements aux respectables membres du jury pour bien avoir voulu m'accorder de leur temps précieux pour commenter et juger mon travail. Merci d'avoir fait de cette journée de soutenance un si bon moment. Je remercie très sincèrement les rapporteurs Pr. Olivier Perrin et Pr. Frédérique Biennier pour la lecture détaillée de ce manuscrit et leurs remarques critiques et pertinentes ; merci notamment à Pr. Chihab Hanachi de m'avoir fait l'honneur de présider le jury de thèse.

Ma reconnaissance s'adresse également à Alexandre Zapolsky, Michel-Marie Maudet, Jean-Pierre Lorré et Bertrand Escudé pour m'avoir offert la possibilité de faire cette thèse.

Une personne qui a marqué ma vie et à laquelle je tiens à témoigner mon profond respect et ma sincère reconnaissance : Pr. Mohamed Moalla. Je le remercie pour la confiance qu'il m'a témoignée, pour le temps qu'il m'a réservé malgré ses grandes occupations, et pour l'aide très précieuse qu'il m'a apportée. Soyez assuré de ma gratitude.

Ce travail de recherche n'aurait pas non plus été possible sans les efforts, la magnifique ambiance et les encouragements continus de mes collègues et en particulier ceux du pôle R&D Linagora et Petals Link : Amira, Julien, Nicolas S, Sébastien R, Sébastien G, Nicolas B-D, Laurent,

Nicholas, Graham et Thomas. Merci pour tous les bons moments passés ensemble. Un immense merci à Mira, Juju et Sénior pour leur excellence scientifique, pour leur aide précieuse et pour leur soutien moral tant professionnel que personnel. Merci pour tout deuxième souffle que vous m'avez donné à chaque moment de faiblesse.

Au cours de ces trois années, j'ai pris beaucoup de plaisir à travailler avec l'ensemble des membres du centre de Génie Industriel de l'école des Mines. Merci à Isabelle, Sébastien, Lionel, Nicolas, Anne-Marie, Wenxin, Aurélien, Marie, Michel, Elise, Franck, Didier, François et Paul pour l'accueil chaleureux et l'expertise scientifique. Mes sincères remerciements vont également à Jacques et Matthieu pour leurs conseils et leurs remarques pertinentes lors des répétitions de soutenance. Merci notamment à Elyes et à Olfa qui n'ont cessé de m'apporter leur soutien, qui m'ont gracieusement fourni leurs conseils, et avec qui les discussions m'ont toujours poussée à voir les choses différemment. Je souhaite remercier l'équipe de doctorants Manel, Aurélie, Guillaume, Safae, Tiexin, Andrés, Shadan et Alexandre et je leur souhaite beaucoup de courage.

Mes remerciements s'adressent aussi évidemment à ma sœur Mery. Le mot même de « remerciements » est faible pour traduire ce que je lui dois. Merci pour le soutien quotidien, à la fois affectueux et efficace. Sans toi, rien n'aurait été possible. Merci également à mon beau-frère Samsoum pour son aide et pour tous les bons moments. J'attends avec impatience de voir ma petite crevette Nounou pointer le bout de son nez.

Pour leurs encouragements dans toutes mes initiatives, leur patience, leur écoute, leur aide, leur réconfort, leur implication, leurs éclats de rire, leur amour, leur apport moral lors des moments de doute et de découragement vécus tout au long de mon parcours dans la vie, je tiens à remercier infiniment les personnes qui me sont les plus proches : ma famille et mes amis (ies). Un immense merci également à ma deuxième famille à Toulouse pour tous les bons moments et discussions enrichissantes.

Enfin, je tiens à remercier toutes les personnes croisées lors des projets, conférences et ailleurs, qui m'ont offert un moment de leur temps pour satisfaire ma curiosité et répondre à mes questions.

Résumé

Depuis plusieurs années, la mondialisation du marché, l'augmentation des exigences des clients, et la recherche permanente d'une baisse des coûts induisent un accroissement fort de la complexité du monde industriel. Ainsi, la collaboration inter-organisationnelle devient essentielle pour maintenir et renforcer la compétitivité des entreprises. Dans cette optique, le système d'information peut, selon sa flexibilité et sa robustesse, freiner ou faciliter cette collaboration. Cette problématique est le point de départ du projet MISE (*Mediation Information System Engineering*) qui propose une solution pour la conception et la réalisation d'un système d'information collaboratif. Cette solution se base sur une démarche d'ingénierie dirigée par les modèles, couplée à une approche *Business Process Management* (BPM) et reposant sur les *Architectures Orientées Services* (SOA). Cette démarche se déroule en deux étapes : (i) la génération d'une cartographie de processus répondant à une situation collaborative (niveau métier) et (ii) sa transformation en un système exécutable (niveau technique). Les travaux présentés dans ce manuscrit s'inscrivent dans cette démarche. Ils s'intéressent en particulier au passage du niveau métier vers le niveau technique en se focalisant sur les aspects non-fonctionnels. Nous présentons une approche permettant (i) d'annoter les activités métier à l'aide d'exigences non-fonctionnelles lors de la modélisation des processus collaboratifs, (ii) de rationaliser la gestion des services et leurs propriétés non-fonctionnelles au moyen de la gouvernance SOA et enfin (iii) de faire une réconciliation non-fonctionnelle entre les activités et les services. Nous avons validé les concepts de notre approche au travers de deux prototypes : *Petals BPM-NFR* pour la modélisation et l'annotation non-fonctionnelle des processus métier et *EasierGov-NFR* pour la gouvernance des services et de leurs propriétés non-fonctionnelles ainsi que la réconciliation non-fonctionnelle. Un cas d'étude illustre notre approche pour prouver son applicabilité. Ces travaux s'inscrivent au sein du projet européen FP7 CHOReOS.

Mots clés : Réconciliation non-fonctionnelle, Gouvernance SOA, Interopérabilité, Système d'information, Gestion des processus métier, Services Web.

Abstract

Over the last years, the globalization, the increase of customers requirements, and the constant search for lower costs, have induced a steep complexity of organization's management. Therefore, inter-organizational collaboration becomes essential to maintain and enhance the competitiveness of enterprises. In this context, the information system may, depending on its flexibility and its robustness, hamper or facilitate this collaboration. One approach to resolve this problem has been taken by the MISE (Mediation Information System Engineering) project. It aims to provide an effective implementation of a collaborative information system, based on Model-Driven Engineering, coupled with a Business Process Management approach (BPM) and supported by Service-Oriented Architecture (SOA). This solution consists in two steps: (i) the generation of a business processes map from the description of a collaborative situation (business level), and (ii) the transformation of these process models into an executable system (technical level). The works detailed in this PhD take roots in this solution and improve it by adding non-functional aspects management during the transition from the business level to the technical level. We present an approach that: first (i) annotates the business activities with Non-Functional Requirements (NFR) in the processes modelling stage, second (ii) rationalizes the management of services and their non-functional properties through SOA governance, and third (iii) makes a non-functional reconciliation between activities and services. We validate our approach through two prototypes: *Petals BPM-NFR* for modelling and non-functional annotation of business process, and *EasierGov-NFR* for the governance of services and their non-functional properties, and the non-functional reconciliation. A case study illustrates our approach to demonstrate its applicability. This work is supported by the FP7 European project CHOReOS.

Keywords: Non-functional reconciliation, SOA governance, Interoperability, Information system, Business process management, Web services.

Table des matières

TABLE DES MATIERES	I
LISTE DES FIGURES	V
LISTE DES TABLEAUX	IX
LISTE DES ALGORITHMES	XI
INTRODUCTION GÉNÉRALE	1
1. DESCRIPTION DU CONTEXTE	1
2. PRESENTATION DE LA PROBLEMATIQUE GENERALE.....	2
3. ORGANISATION DU MANUSCRIT	3
CHAPITRE I : CONTEXTE ET VISION D'ENSEMBLE.....	6
I.1. INTRODUCTION	7
I.2. EVOLUTION DU SYSTEME D'INFORMATION: UN CONTEXTE CONCURRENTIEL ET COLLABORATIF	7
I.2.1. Un problème d'agilité	8
I.2.2. Un problème d'interopérabilité	8
I.2.3. Un problème de correspondance métier / technique	9
I.3. SYSTEME D'INFORMATION DE MEDIATION	9
I.4. LIMITES DES TRAVAUX ANTERIEURS ET PROBLEMATIQUES	12
I.4.1. Modélisation et réconciliation métier / technique : la non prise en compte des propriétés non- fonctionnelles.....	12
I.4.2. Manque de rationalisation de la gestion des services et de leurs propriétés non-fonctionnelles	15
I.5. PRESENTATION DES OBJECTIFS DE LA THESE	15
I.6. POSITIONNEMENT DES TRAVAUX DE LA THESE	16
I.6.1. Positionnement par rapport à Linagora Labs	17
I.6.2. Positionnement par rapport à MISE	18
I.7. CONCLUSION	19
CHAPITRE II : ETAT DE L'ART.....	21
II.1. INTRODUCTION	23
II.2. ARCHITECTURE ORIENTEE SERVICES (SOA).....	23
II.2.1. Les services	24
II.2.2. Les services Web	26
II.2.3. Les propriétés non-fonctionnelles des services Web	28

II.2.4. Synthèse sur les propriétés non-fonctionnelles	31
II.3. RATIONALISATION DE LA GESTION DES SERVICES ET DE LEURS PROPRIETES NON-FONCTIONNELLES : GOUVERNANCE SOA	32
II.3.1. Qu'est-ce que la gouvernance SOA ?	32
II.3.2. Les cadres de gouvernance SOA	35
II.3.3. Synthèse sur les cadres existants	37
II.4. GESTION DES PROCESSUS METIER (BPM).....	40
II.4.1. Les processus et les processus métier	40
II.4.2. Gestion des processus métier.....	41
II.4.3. Classification des processus métier	42
II.5. BPM DANS UN CONTEXTE SOA	43
II.5.1. Composition de services Web	43
II.5.2. Modélisation graphique des processus métier orientés SOA	44
II.5.3. Synthèse sur la modélisation des processus métier orientés SOA	48
II.5.4. Annotation du BPMN avec des exigences non-fonctionnelles.....	48
II.5.5. Synthèse sur les annotations du BPMN à l'aide d'exigences non-fonctionnelles	52
II.6. RECONCILIATION NON-FONCTIONNELLE	53
II.6.1. Etude des approches existantes.....	53
II.6.2. Synthèse sur la réconciliation non-fonctionnelle.....	55
II.7. CONCLUSION.....	55
CHAPITRE III : PROPRIETES NON-FONCTIONNELLES : ANNOTATION DES PROCESSUS ET GOUVERNANCE DES SERVICES	57
III.1. INTRODUCTION.....	58
III.2. PETALS BPM-NFR : PLATEFORME DE MODELISATION ET D'ANNOTATION NON-FONCTIONNELLE DES PROCESSUS.....	60
III.2.1. Architecture générale de Petals BPM-NFR	60
III.2.2. Modélisation des processus métier	61
III.2.3. Annotation non-fonctionnelle des processus BPMN 2.0	62
III.3. EASIERGOV-NFR : GOUVERNANCE SOA POUR LA GESTION DES SERVICES ET DES PROPRIETES NON-FONCTIONNELLES	65
III.3.1. Découverte SOA.....	66
III.3.2. Gestion des services	67
III.3.3. Gestion des propriétés non-fonctionnelles	68
III.3.3.1. Annotation non-fonctionnelle des services	68
III.3.3.2. Modèle de contrat de services Web (WS-Agreement Template)	70
III.3.3.3. Classification des propriétés non-fonctionnelles	71

III.3.3.4. Gestion des propriétés non-fonctionnelles dans EasierGov-NFR	74
III.4. CONCLUSION	76
CHAPITRE IV : RECONCILIATION NON-FONCTIONNELLE DE SERVICES	78
IV.1. INTRODUCTION.....	79
IV.2. RECONCILIATION DE SERVICES	79
IV.3. RECONCILIATION NON-FONCTIONNELLE DE SERVICES.....	80
IV.4. RECONCILIATION NON-FONCTIONNELLE DE SERVICES DE TYPE « 1-1 ».....	84
IV.4.1. Présentation générale de l’approche « 1-1 »	84
IV.4.2. Algorithme de réconciliation non-fonctionnelle de type « 1-1 »	84
IV.4.2.1. Étape 1 : Classification des valeurs des exigences non-fonctionnelles.....	86
IV.4.2.2. Étape 2 : Élimination des services candidats ayant des valeurs aberrantes	87
IV.4.2.3. Étape 3 : Normalisation et centrage des services candidats par rapport aux exigences non-fonctionnelles de l’activité métier	89
IV.4.2.4. Étape 4 : Classement et sélection des services candidats.....	95
IV.5. RECONCILIATION NON-FONCTIONNELLE DE SERVICES DE TYPE « 1-N »	98
IV.5.1. Présentation générale de l’approche « 1-N »	98
IV.5.2. Transformation d’une réconciliation non-fonctionnelle de type « 1-N » en une de type « 1-1 »	100
IV.5.2.1. Étape 1 : Décomposition en plusieurs sous-compositions de deux services	100
IV.5.2.2. Étape 2 : Calcul de la valeur de la propriété non-fonctionnelle de l’ensemble de la sous-composition.....	101
IV.6. CONCLUSION	106
CHAPITRE V : IMPLEMENTATION ET ILLUSTRATION.	108
V.1. INTRODUCTION	109
V.2. PLATEFORME A2S.....	109
V.2.1. Plateforme de modélisation : Petals BPM-NFR	110
V.2.1.1. Architecture technique	110
V.2.1.2. Implémentation avec GWT et Java	112
V.2.2. Plateforme de gouvernance SOA et de réconciliation : EasierGov-NFR	115
V.2.2.1. Rationalisation des services et de leurs propriétés non-fonctionnelles	115
V.2.2.2. Moteur de réconciliation non-fonctionnelle	117
V.2.3. Intégration dans une architecture SOA open-source	119
V.2.4. Démarche de qualité constante	120
V.3. ILLUSTRATION DE LA DEMARCHE PROPOSEE	121
V.3.1. Présentation du cas d’étude.....	121
V.3.2. Déroulement de la méthodologie	125

V.3.2.1. Modélisation et annotation non-fonctionnelle du processus	125
V.3.2.2. Réconciliation métier / technique	128
V.4. CONCLUSION.....	134
CONCLUSION GENERALE ET PERSPECTIVES	136
1. RAPPEL DU CONTEXTE ET DES OBJECTIFS DE LA THESE	136
2. SYNTHÈSE SUR LES CONTRIBUTIONS DE CES TRAVAUX	136
3. PERSPECTIVES	138
BIBLIOGRAPHIE.....	142
ANNEXE 1 : PROPRIÉTÉS NON-FONCTIONNELLES DE WSQF	152
ANNEXE 2 : ANALYSE CROISÉE : EXIGENCES NON-FONCTIONNELLES MÉTIER / WSQF.....	156
ANNEXE 3 : CAS D'ÉTUDE SAZRI / ENTREPRISE DE MARKETING PREDICTIF	160

Liste des figures

Figure 1 : Organisation du manuscrit et articulation des chapitres	5
Figure 2 : Principe du Système d'Information de Médiation [Bénaben et al., 2008].....	11
Figure 3 : Réconciliation de services d'un point de vue fonctionnel et sémantique	14
Figure 4 : Problématique générale des propriétés non-fonctionnelles lors de la sélection de services	15
Figure 5 : Méthodologie générale des travaux de la thèse	17
Figure 6 : Résultats de recherche de Linagora Labs	18
Figure 7 : Positionnement des travaux de la thèse dans MISE 2.0	20
Figure 8 : Structuration d'un service	24
Figure 9 : Principe de fonctionnement des services	25
Figure 10 : Principe de découverte et d'utilisation d'un service Web	27
Figure 11 : Propriétés non-fonctionnelles proposées par le standard WSQF. [OASIS, 2010]	30
Figure 12 : Gouvernance d'entreprise, gouvernance informatique et gouvernance SOA	33
Figure 13 : Gouvernance SOA selon la trilogie de Marks [Marks, 2008].....	34
Figure 14 : Niveaux d'abstraction d'un processus BPM [Crusson, 2003]	42
Figure 15 : Modélisation de processus avec les réseaux de Petri [De Backer et al., 2005].....	44
Figure 16 : Modélisation de processus avec IDEFØ [Bia-Figueiredo et al., 2011]	45
Figure 17 : Modélisation de processus avec le profil UML [Koehler, et al., 2007].....	46
Figure 18 : Modélisation de processus avec BPMN 2.0 [Meyer, 2011]	47
Figure 19 : Extension de BPMN : intégration des exigences de sécurité [Rodríguez et al., 2007]	50
Figure 20 : Extension de BPMN : control case et operating condition [Pavlovski et al., 2008]	51
Figure 21 : Extension de BPMN : le méta-modèle BPQRM [Heinrich et al., 2011]	52
Figure 22 : Analyse croisée : besoins / existant et manquant	59
Figure 23 : Modélisation, annotation non-fonctionnelle et gouvernance SOA	60
Figure 24 : Architecture générale de Petals BPM-NFR	61
Figure 25 : Interface graphique de Petals BPM.....	62

Figure 26 : Architecture de Petals BPM – NFR : annotation non-fonctionnelle détaillée	64
Figure 27 : Présentation générale d'EasierGov [Zribi et al., 2012].....	65
Figure 28 : Présentation générale d'EasierGov-NFR	66
Figure 29 : Présentation de l'architecture d'EasierGov-NFR	67
Figure 30 : Cas d'utilisation du fournisseur de service	68
Figure 31 : Cas d'utilisation du consommateur	68
Figure 32 : Structure d'un template SLA selon WS-Agreement [Andrieux et al., 2007]	70
Figure 33 : Gestion des variables non mesurables : Variables Client	76
Figure 34 : Réconciliation non-fonctionnelle : sélection et classement des services.....	79
Figure 35 : Réconciliation de services	80
Figure 36 : Réconciliation non-fonctionnelle de services	80
Figure 37 : Exemple d'une construction d'un pont en Lego	81
Figure 38 : Exemple d'une construction d'un pont en Lego : réconciliation « 1-1 »	82
Figure 39 : Exemple d'une construction d'un pont en Lego : réconciliation « 1-N ».....	83
Figure 40 : Réconciliation non-fonctionnelle « 1-1 »	84
Figure 41 : Classement des services candidats dans les zones A et B	95
Figure 42 : Classement des services candidats de la zone B : analyse selon les valeurs négatives .	96
Figure 43 : Réconciliation non-fonctionnelle « 1-N »	98
Figure 44 : Proposition de transformation d'une approche « 1-N » vers une approche « 1-1 ».....	99
Figure 45 : Problématique de passage d'une approche « 1-N » vers une de type «1-1»	100
Figure 46 : Exemple de décomposition en sous-compositions de couple de services	101
Figure 47 : Branchement en parallèle	102
Figure 48 : Branchement séquentiel	102
Figure 49 : Branchement exclusif.....	103
Figure 50 : Calcul du temps de réponse global d'une composition de services Web	105
Figure 51 : Architecture de la plateforme A2S.....	109
Figure 52 : Fonctionnalités de Petals BPM-NFR	110
Figure 53 : Architecture technique de Petals BPM-NFR.....	111

Figure 54 : Modélisation des processus métier annotés avec Petals BPM-NFR (coté client).....	111
Figure 55 : Capture d'écran de la plateforme Petals BPM-NFR	114
Figure 56 : Principales méthodes de la librairie « services API »	115
Figure 57 : Principales méthodes de la librairie « SLA API »	116
Figure 58 : SLA : adresse du service	117
Figure 59 : Moteur de réconciliation non-fonctionnelle	118
Figure 60 : Intégration de Petals BPM-NFR et EasierGov-NFR dans une architecture SOA.....	120
Figure 61 : Résultats des tests d'intégration continue de Petals BPM-NFR	120
Figure 62 : Résultats des tests d'intégration continue d'EasierGov-NFR.....	121
Figure 63 : Processus métier collaboratif pour l'amélioration des ventes de Sazri par le marketing prédictif	123
Figure 64 : Modélisation du cas d'étude Sazri/MarketingPrédictif dans Petals BPM-NFR	126
Figure 65 : Annotation non-fonctionnelle de l'activité « Élargir le fichier clients »	127
Figure 66 : Extrait du fichier BPMN 2.0 : activité « Élargir le fichier clients ».....	128
Figure 67 : Réconciliation fonctionnelle : activité « Élargir le fichier clients ».....	129
Figure 68 : Extraits du template SLA du service ÉFC 1 de l'entreprise 7Marketing	130
Figure 69 : Résultats de la réconciliation non-fonctionnelle de l'activité « Élargir le fichier clients »	134
Figure 70 : Positionnement par rapport aux travaux du projet MISE	139
Figure 71 : Innovation industrielle vs Originalité scientifique	140
Figure 72 : Fichier BPMN 2.0 de l'activité « Élargir le fichier clients » annotée	160
Figure 73 : Template SLA du service ÉFC 1 de l'entreprise 7Marketing.....	162

Liste des tableaux

Tableau 1 : Synthèse sur les cadres de gouvernance SOA.....	39
Tableau 2 : Synthèse sur les approches de réconciliation non-fonctionnelle	55
Tableau 3 : Extrait du tableau « analyse croisée exigences non-fonctionnelles métier / WSQF »	63
Tableau 4 : Classification générale des propriétés non-fonctionnelles de WSQF	73
Tableau 5 : Extension de WS-Agreement par les propriétés non-fonctionnelles de WSQF de type variables Fournisseur	74
Tableau 6 : Classification des valeurs des propriétés non-fonctionnelles selon leur type	86
Tableau 7 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas des branchements en parallèle et séquentiel.....	103
Tableau 8 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas d'un branchement exclusif.....	104
Tableau 9 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas d'un branchement inclusif.....	104
Tableau 10 : Fonctionnalités des activités du processus métier pour l'amélioration des ventes de Sazri par le marketing prédictif	124
Tableau 11 : Exigences non-fonctionnelles de l'entreprise Sazri.....	125
Tableau 12 : Préférences (poids) de l'entreprise Sazri par rapport aux exigences non-fonctionnelles	125
Tableau 13 : Propriétés non-fonctionnelles de la catégorie « Business Quality Value ». [OASIS, 2010].....	152
Tableau 14 : Propriétés non-fonctionnelles de la catégorie « Service Level Measurement Quality ». [OASIS, 2010].....	153
Tableau 15 : Propriétés non-fonctionnelles de la catégorie « Interoperability ». [OASIS, 2010].	153
Tableau 16 : Propriétés non-fonctionnelles de la catégorie « Business Processing ». [OASIS, 2010]	154
Tableau 17 : Propriétés non-fonctionnelles de la catégorie « Manageability ». [OASIS, 2010]...	154
Tableau 18 : Propriétés non-fonctionnelles de la catégorie « Security ». [OASIS, 2010].....	155

Liste des algorithmes

Algorithme 1 : Normalisation et centrage des valeurs des propriétés des services candidats de type numérique	90
Algorithme 2 : Centrage des valeurs des propriétés des services candidats de type booléen	91
Algorithme 3 : Normalisation et centrage des valeurs des propriétés des services candidats dans le cas d'un intervalle de valeurs numériques	94
Algorithme 4 : Classement et sélection des services candidats appartenant à la zone A	97
Algorithme 5 : Classement et sélection des services candidats appartenant à la zone B	97

Introduction générale

« La lecture commence les yeux fermés. »

Yvon Rivard

1. Description du contexte

L'environnement dans lequel se situent les entreprises est devenu de plus en plus complexe, ouvert, mondialisé et dans un contexte concurrentiel en expansion. Les entreprises ont pris conscience du fait qu'elles peuvent garantir leur développement futur, maintenir et renforcer leur compétitivité, en faisant appel aux nouvelles technologies et en ayant recours aux collaborations inter-organisationnelles. Ces collaborations sont devenues essentielles au bon fonctionnement et à l'excellence de l'entreprise. Le système d'information est au cœur de ces collaborations. Il peut, selon sa robustesse et sa flexibilité, les freiner ou au contraire les faciliter.

Une des solutions de conception d'un tel système d'information a été apportée par le projet MISE (*Mediation Information System Engineering*) [Bénaben, 2012]. Ce projet interne pluriannuel du Centre de Génie Industriel de l'École des Mines d'Albi-Carmaux, vise à définir et à outiller une approche de conception de Système d'Information de Médiation (SIM) facilitant la collaboration (quels que soient le domaine ou les partenaires mis en jeu). Ce projet, lancé en 2004, se base sur trois concepts clés pour la conception d'un tel système :

- l'architecture dirigée par les modèles (*Model Driven Architecture* - MDA) [Miller et al., 2003] couplée à une approche de gestion de processus métier (*Business Process Management* - BPM). Cette démarche permet de réduire la complexité du système d'information en séparant les besoins métier et techniques ;
- l'architecture orientée service (connues par l'acronyme anglo-saxon SOA pour *Service-Oriented Architecture*) [Newcomer et al., 2004]. Ces architectures simplifient les interconnexions entre les applications en se basant sur le principe de composants indépendants (les services) et de couplage faible ;
- le médiateur [Wiederhold et al., 1992] en tant que composant logiciel implanté au sein d'une technologie de bus de service d'entreprise et permettant d'implémenter le système à base de médiation.

La deuxième itération du projet MISE vise à automatiser autant que possible la conception de ce système en se basant sur deux étapes majeures. La première se focalise sur la caractérisation de la situation collaborative et de la transformation d'un modèle de cette situation en un modèle de cartographie de processus métier collaboratifs (niveau métier). La deuxième est responsable de la transformation des ces processus métier en des processus techniques exécutables (niveau technique). Le passage de la première étape à la deuxième passe par la réconciliation entre les besoins métier exprimés par les architectes métier et les services techniques disponibles.

2. Présentation de la problématique générale

Une première approche de réalisation de la réconciliation métier / technique a été établie dans le cadre des travaux de thèse de N. Boissel-Dallier [Boissel-Dallier, 2012] où il s'est intéressé à l'apport de la sémantique pour la réconciliation fonctionnelle (basée sur les besoins fonctionnels). Toutefois, le nombre de services disponibles qui ne cesse d'augmenter et la prolifération des services techniques qui répondent exactement à la même fonctionnalité nous ont menés au questionnement suivant : *comment pouvons-nous affiner cette liste de services pour une meilleure réconciliation métier / technique ?*

La réponse à cette question se trouve dans la prise en considération des propriétés non-fonctionnelles. En effet, ces propriétés ne peuvent être écartées ou même négligées lors de la phase de réconciliation. Elles jouent un rôle essentiel lors de la sélection de services pour les activités métier notamment pour trancher entre plusieurs services qui répondent aux mêmes attentes fonctionnelles. La nécessité d'établir une telle réconciliation non-fonctionnelle (en considérant les critères non-fonctionnels), nous a donc menés aux questionnements suivants : *quelles exigences non-fonctionnelles sont nécessaires aux architectes métier pour exprimer leurs besoins non-fonctionnels ? Comment peuvent-ils annoter les processus métier collaboratifs à l'aide d'exigences non-fonctionnelles ? Comment pouvons-nous gérer et manipuler les propriétés non-fonctionnelles des services techniques ? Comment établir la réconciliation entre les exigences non-fonctionnelles des activités métier et les propriétés non-fonctionnelles des services techniques (parmi ceux qui répondent aux besoins fonctionnels) ?*

Telles sont les quatre questions auxquelles nous essayons de répondre dans les travaux décrits dans ce manuscrit de thèse. En effet, les réponses aux deux premières questions passent tout d'abord par une étude des critères non-fonctionnels nécessaires à l'annotation des activités du processus. Cette annotation se fait lors de la phase de modélisation du processus. Afin de faciliter l'expression de ces besoins métier non-fonctionnels, il peut s'avérer judicieux que cette annotation soit également graphique. Quant aux troisième et quatrième questions, il est nécessaire d'établir un cadre de gouvernance SOA permettant à la fois la rationalisation de la gestion des services (ainsi que leurs propriétés non-fonctionnelles) et l'établissement de la réconciliation non-fonctionnelle en

sélectionnant pour chacune des activités métier, le ou les services qui répondent (ou le cas échéant qui se rapprochent) aux besoins non-fonctionnels.

3. Organisation du manuscrit

Le travail de cette thèse s'intitule « *la gouvernance SOA pour une approche de conception de Système d'Information de Médiation : réconciliation non-fonctionnelle de services pour mettre en œuvre les processus métier* » et propose de répondre aux enjeux décrits précédemment. Pour y parvenir, nous avons choisi de l'organiser en cinq chapitres :

Le chapitre I dresse la problématique et donne une vision d'ensemble des travaux réalisés dans le cadre cette thèse. Ce chapitre s'intéresse tout d'abord au contexte de l'étude, à savoir la collaboration entre les entreprises et la nécessité d'avoir des technologies qui promeuvent cette collaboration. Par la suite, il présente les travaux précédents des deux équipes (MISE et Linagora Labs) afin de définir leurs limites. Enfin, nous détaillons les objectifs auxquels répond cette thèse à savoir : (i) la prise en compte des exigences non-fonctionnelles lors de la modélisation des processus métier collaboratifs et lors de l'automatisation de la transformation entre les activités métier et les services techniques et (ii) la rationalisation de la gestion des services dans un registre permettant de gérer les services et leurs métadonnées (e.g. : les descriptions de services, les propriétés non-fonctionnelles et les contrats de services).

Le chapitre II présente une étude bibliographique autour des concepts, des approches et des technologies liés aux objectifs abordés précédemment. Cet état de l'art détaille, en premier lieu, l'architecture orientée service et sa composante de base : les services (en particulier les services Web qui sont des services orientés pour être utilisés à travers le Web), ainsi que les travaux existants autour de leurs propriétés non-fonctionnelles. Par ailleurs, il présente la gestion des processus métier d'une manière générale, pour la spécifier par la suite dans un contexte SOA en présentant les travaux existants concernant la modélisation et l'annotation non-fonctionnelle des processus métier. Enfin, ce chapitre s'intéresse à définir la gouvernance SOA et à présenter les travaux existants traitant de cette méthodologie ainsi qu'à la réconciliation non-fonctionnelle des services Web.

Le chapitre III présente deux prototypes établis au cours de nos travaux. Dans le premier, nous proposons des solutions pour la modélisation des processus métier collaboratifs dans un contexte SOA avec la possibilité d'annoter graphiquement les activités métier à l'aide d'exigences non-fonctionnelles. Dans le deuxième, nous proposons un registre de gouvernance SOA permettant, d'une part, de rationaliser la gestion des services, leurs propriétés non-fonctionnelles et leurs modèles de contrats de services et d'autre part de fournir un moteur de réconciliation non-fonctionnelle.

Le chapitre IV détaille ce moteur de réconciliation non-fonctionnelle entre les activités métier et les services techniques disponibles. En effet, cette réconciliation consiste à épurer les résultats fournis

par la réconciliation fonctionnelle en se basant sur les exigences non-fonctionnelles de l'activité métier. Deux mécanismes sont présentés en détail dans ce chapitre : (i) une réconciliation non-fonctionnelle unitaire « 1-1 » qui consiste à la sélection d'un service technique pour une activité métier et (ii) une réconciliation non-fonctionnelle de composition de services « 1-N » où pour une activité métier, une composition de services est sélectionnée.

Le chapitre V présente l'implémentation réalisée au cours de ces travaux. Elle couvre, d'une part, la modélisation du processus métier collaboratif et son annotation non-fonctionnelle. D'autre part, le registre de gouvernance SOA permettant la gestion des services et de leurs métadonnées, et la réconciliation non-fonctionnelle. Par la suite, afin de montrer l'applicabilité de nos approches, nous illustrons notre méthodologie sur la base d'un cas d'étude fictif présentant la collaboration entre deux entreprises.

Une conclusion générale et la présentation de perspectives clôturent les travaux de ce manuscrit. Nous commençons par rappeler le contexte et les objectifs de cette thèse. Par la suite, nous présentons une synthèse sur les diverses contributions de nos travaux. Enfin, nous proposons les perspectives et les travaux futurs qui peuvent être envisagés.

Dans la Figure 1, nous présentons l'organisation du manuscrit et l'articulation des chapitres.

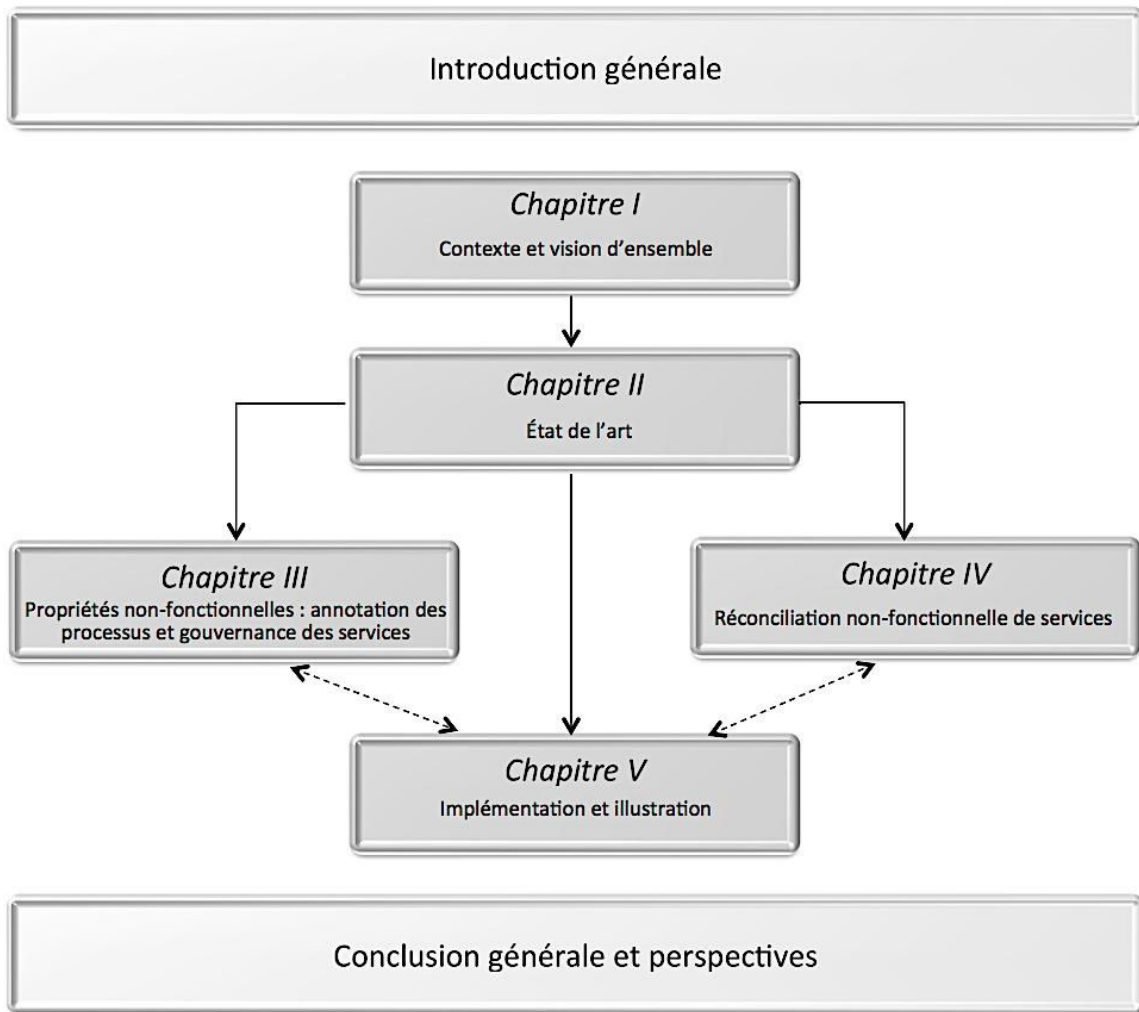


Figure 1 : Organisation du manuscrit et articulation des chapitres.

Chapitre I

Contexte et vision d'ensemble

« La vraie connaissance est de connaître l'étendue de son ignorance. »

Confucius

I.1. INTRODUCTION	7
I.2. EVOLUTION DU SYSTEME D'INFORMATION: UN CONTEXTE CONCURRENTIEL ET COLLABORATIF	7
I.2.1. Un problème d'agilité	8
I.2.2. Un problème d'interopérabilité	8
I.2.3. Un problème de correspondance métier / technique	9
I.3. SYSTEME D'INFORMATION DE MEDIATION	9
I.4. LIMITES DES TRAVAUX ANTERIEURS ET PROBLEMATIQUES	12
I.4.1. Modélisation et réconciliation métier / technique : la non prise en compte des propriétés non-fonctionnelles	12
I.4.2. Manque de rationalisation de la gestion des services et de leurs propriétés non-fonctionnelles	15
I.5. PRESENTATION DES OBJECTIFS DE LA THESE	15
I.6. POSITIONNEMENT DES TRAVAUX DE LA THESE	16
I.6.1. Positionnement par rapport à Linagora Labs	17
I.6.2. Positionnement par rapport à MISE	18
I.7. CONCLUSION	19

I.1. Introduction

Dans une ère marquée par une révolution technologique de l'information et de la communication, le système d'information est devenu un élément indispensable et central dans toute entreprise. Les entreprises ont pris conscience qu'il est essentiel d'avoir un système d'information capable de suivre les évolutions de l'environnement et de supporter les collaborations inter-organisationnelles afin de garantir leur maintien et leur croissance. Cependant, dans ce contexte de collaborations et avec l'augmentation des applications au sein du système d'information, il devient de plus en plus difficile à ce dernier de s'y adapter : manque d'agilité, d'interopérabilité et de correspondance entre les processus métier et techniques.

Dans ce chapitre, nous décrivons le contexte évolutif et collaboratif du système d'information et les problèmes qui en découlent (section I.2). Afin de répondre à ces problèmes, nous présentons dans la section I.3 une solution proposée par le projet MISE qui est le *Système d'Information de Médiation* (SIM). La section I.4 détaille et analyse les limites des travaux antérieurs autour de la conception et de la réalisation de ce SIM et adresse la problématique. Par la suite, nous présentons une série d'objectifs pour nos travaux de thèse (section I.5). Enfin, nous clôturons ce chapitre en positionnant ces objectifs par rapport à l'équipe en charge du projet MISE du centre Génie Industriel de l'école des Mines d'Albi et à par rapport Linagora Labs où ont été réalisés les travaux de cette thèse.

I.2. Evolution du Système d'Information : un contexte concurrentiel et collaboratif

Le Système d'Information (SI) est constitué de l'ensemble des ressources humaines et matérielles nécessaires pour collecter, enregistrer, véhiculer et traiter l'information au sein de l'entreprise [De Courcy, 1992]. Initialement basé sur une architecture centralisée où les applications étaient monolithiques et les données sont traitées d'une manière redondante, le SI a évolué, durant les années 80, vers une architecture Client / Serveur puis vers une architecture distribuée. Depuis le début des années 90 et avec l'accélération de la mondialisation, l'environnement devient complexe, instable et marqué par une concurrence accrue.

L'innovation est devenue un point essentiel et stratégique pour toute entreprise. Afin d'assurer leur maintien et de se procurer les avantages concurrentiels indispensables à leur croissance et à leur rentabilité, les entreprises se retrouvent dans la nécessité de collaborer davantage. Dans un tel contexte, le SI se trouve concerné étant donné que la collaboration des entreprises passe par la collaboration des SIs. Les architectures citées ci-dessus bien qu'efficaces ne permettent pas de s'adapter à ce contexte concurrentiel, exacerbé et collaboratif en faisant évoluer leurs processus, car le

SI ne peut suivre cette évolution. De ce fait, plusieurs problèmes se posent : un problème d'agilité, un problème d'interopérabilité et un problème de correspondance métier / technique.

I.2.1. Un problème d'agilité

Face aux évolutions et aux changements imprévus de l'environnement interne et/ou externe, il devient essentiel de concevoir le système d'information d'une manière agile.

La notion d'agilité est devenue une propriété notoire à la fin des années 90. Elle est surtout corrélée à une dynamique d'évolution de la collaboration inter-organisationnelle. Cette évolution est décrite dans [Bénaben et al., 2007] comme « *la transformation de la structure figée vers un environnement fluide* ». Elle est également décrite par « *la transformation d'une construction statique de lego vers un organisme vivant* » [Luzeaux et al., 2008].

Dans la littérature, l'ensemble des définitions du concept d'agilité s'accorde sur le fait qu'elle représente la capacité de l'entreprise à :

- répondre rapidement et efficacement aux changements du marché ;
- s'adapter à l'évolution du contexte dans lequel elle évolue ;
- réagir avec flexibilité face à des changements imprévisibles.

Face aux besoins d'adaptabilité permanente, le besoin d'agilité apparaît évident et indispensable afin de permettre, entre autres, la qualité, l'intégration, l'interopérabilité et la cohérence entre le monde métier et le monde technique. Par conséquent, l'agilité du SI devient un objectif majeur et doit être une qualité dont toute entreprise doit disposer pour faire face aux exigences de ses clients, à la concurrence et à l'évolution rapide des technologies [Goranson, 1999], [Desouza, 2006] et [Rouse, 2007].

I.2.2. Un problème d'interopérabilité

Pour atteindre leurs objectifs, les entreprises ont besoin également de faire interagir plusieurs composants de leur système d'information. Il est nécessaire que ces derniers puissent communiquer d'une manière transparente et à moindre effort. Ce mécanisme est généralement appelé *interopérabilité*.

Les auteurs dans [Konstantas et al., 2006] définissent l'interopérabilité comme « *l'aptitude de systèmes à pouvoir travailler ensemble sans effort particulier pour les utilisateurs de ces systèmes* ». H. Pingaud [Pingaud, 2009] précise cette définition en ajoutant que « *l'interopérabilité désigne une capacité de systèmes, nativement étrangers les uns par rapport aux autres, à interagir afin d'établir des comportements collectifs harmonieux et finalisés, sans avoir à modifier en profondeur leur structure ou leur comportement individuel* ».

L'interopérabilité peut être donc vue comme étant la compatibilité que possèdent deux ou plusieurs composants à agir ensemble. Mais, ceci peut s'avérer très complexe quand le nombre de composants mis en jeu et le degré d'hétérogénéité augmentent.

I.2.3. Un problème de correspondance métier / technique

L'agilité et l'interopérabilité sont nécessaires pour la gestion de la collaboration au sein du SI. Toutefois, ceci n'est pas suffisant. Il est également indispensable d'assurer une correspondance (appelée aussi réconciliation) efficace entre les mondes métier et technique, ce qui constitue un objectif majeur pour toute entreprise. En effet, une incompréhension entre ces deux mondes peut constituer un frein à son évolution. La réconciliation consiste à apporter des solutions agiles en trouvant les bons services qui répondent aux besoins des activités métier du processus collaboratif.

Nous distinguons trois problèmes de correspondance qui, une fois résolus, peuvent permettre une connexion pertinente entre les processus métier et les systèmes partenaires : un problème informationnel, un problème fonctionnel et un problème comportemental [Bénaben et al., 2010]. Ces problèmes peuvent être exprimés différemment : *comment assurer la communication entre les différents composants ? Comment assurer la réconciliation entre les activités métier et les services techniques ? Comment obtenir des processus exécutables depuis des processus métier ?* [Bénaben et al., 2013].

La réconciliation entre les activités métier d'un processus et les services techniques n'est pas aussi simple que nous pouvons l'imaginer étant donné la différenciation du niveau de granularité entre ces deux niveaux. En effet, les activités métier définies par les décideurs et qui correspondent aux besoins métier (fonctionnels et non-fonctionnels) sont donc de faible granularité. Inversement, les services techniques qui sont définis et implémentés par les équipes techniques sont de forte granularité.

I.3. Système d'Information de Médiation

Un système d'information agile, interopérable et permettant de supporter tous les processus de l'entreprise (collaboratifs ou non) et la correspondance métier / technique est appelé un Système d'Information de Médiation (*SIM ou MIS pour Mediation Information System*). Ceci est le point de départ du projet MISE (*Mediation Information System Engineering*) qui propose une solution pour la conception et la réalisation d'un SIM. Ce projet ambitionne de développer une approche et des méthodes pour la conception de systèmes d'information collaboratifs à partir de systèmes d'information interopérables suivant un principe de médiation entre ces systèmes.

Le concept de médiation tel que présenté par Wiederhold dans [Wiederhold et al., 1992] est à l'origine de cette idée de système à comportement fédéré, il y a plus de 20 ans, par un courant de travaux qui s'intéressaient à la recherche d'informations distribuées dans des bases de données indépendantes. Le projet MISE a adopté et étendu ces idées à la problématique d'entreprises souhaitant collaborer.

De multiples formes de connaissances sur ce que les partenaires veulent échanger et sur le comment ils veulent ou peuvent le faire, sont reconnues aujourd'hui comme des facteurs clés pour l'interopérabilité entre entreprises. À cette fin, des approches d'ingénierie à base de connaissances sont étudiées comme un faisceau de pistes pertinentes tant dans l'espace de formulation du problème d'interopérabilité que dans celui de proposition de solutions. En effet, puisque le principe même de l'interopérabilité en exploitation cherche à faire interagir et connecter les partenaires sans qu'ils aient à faire de concession sur leur structure propre ou sur leur comportement interne, il est nécessaire de traiter des formes d'incompatibilité à bâtir des échanges, des résistances dues à une certaine hétérogénéité entre contributeurs.

Dans l'optique du projet MISE, d'après F. Bénaben [Bénaben et al., 2008], c'est au composant médiateur du système d'information que nous demandons de traiter ces résistances. Il joue le rôle de l'intermédiaire et sera capable de gérer les caractéristiques de la collaboration, de convertir les données et de connecter les systèmes d'information des différents partenaires mis en jeu. La Figure 2 illustre ce concept de Système d'Information de Médiation.

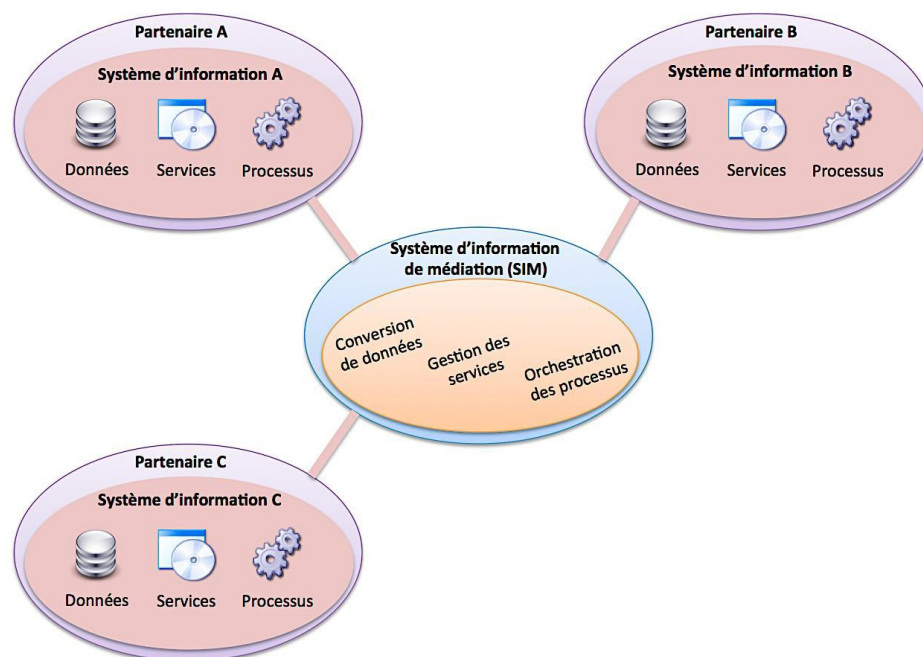


Figure 2 : Principe du Système d'Information de Médiation [Bénaben et al., 2008].

Une *première définition* d'un SIM collaboratif à base de médiation et du médiateur (en tant que composant logiciel permettant d'implanter le système à base de médiation) est un résultat acquis dans le cadre du projet MISE.

Une démarche d'ingénierie dirigée par les modèles du médiateur couplée par une approche de gestion de processus métier en se basant sur l'architecture SOA, a été développée dans le cadre du projet. Les thèses soutenues par J. Touzi [Touzi, 2007], V. Rajsiri [Rajsiri, 2009] et, S. Truptil [Truptil, 2011] ont structuré les premières étapes de ce projet.

Les travaux de J. Touzi portaient sur la conception de ce SI collaboratif. Il s'est intéressé au passage d'un modèle du niveau CIM (*Computer Independent Model*) - où les partenaires fournissent leur processus de collaboration - vers un modèle du niveau PIM (*Platform Independent Model*) de SI de médiation basé sur l'architecture SOA et qui décrit une réponse aux spécifications définies dans le CIM.

V. Rajsiri s'est intéressée à un niveau plus abstrait du projet MISE : le niveau métier. Elle a défini un système à base de connaissances afin de générer automatiquement le modèle CIM et ceci en offrant aux différents partenaires la possibilité de décrire la collaboration souhaitée.

La thèse soutenue par S. Truptil a poursuivi ce travail pour être capable, non seulement de concevoir un système d'information collaboratif à base de médiation, mais aussi d'en faire une maintenance évolutive dans un double mouvement de rétro-ingénierie et d'ingénierie adapté à l'évolution récurrente du besoin et qui fournit de l'agilité en exploitation.

La *deuxième itération* du projet MISE vise à automatiser autant que possible la conception du SIM. Cette itération se compose de deux étapes majeures :

- la caractérisation de la situation collaborative et la transformation d'un modèle de situation collaborative en un modèle de cartographie de processus métier (*niveau métier*) ;
- la transformation de ces processus métier collaboratifs en des processus techniques exécutables (*niveau technique*).

La première étape est au cœur des travaux de thèse de W. Mu [Mu, 2012]. Elle a enrichi les travaux de V. Rajsiri en automatisant la génération de la caractérisation et la transformation d'un modèle de situation collaborative en un modèle de cartographie de processus métier collaboratifs.

La thèse de N. Boissel-Dallier [Boissel-Dallier, 2012], complète ces travaux en faisant la transformation des processus métier en des processus techniques exécutables (la deuxième étape). Ce passage consiste à sélectionner parmi les services disponibles ceux qui couvrent les fonctionnalités des activités métier modélisées dans les différents processus et annotées sémantiquement. Toutefois, la prolifération des services techniques qui répondent exactement à la même fonctionnalité a mené au

questionnement suivant : *comment pouvons-nous gérer ces services ? Comment pouvons-nous perfectionner cette liste de services pour une meilleure réconciliation métier / technique ?*

I.4. Limites des travaux antérieurs et problématiques

Dans le passage à l'acte lors de la transformation des processus métier collaboratifs (depuis une cartographie de processus) en des processus techniques, nous avons constaté des verrous qui peuvent constituer des freins à l'évolution de cette transformation. Ces verrous consistent en :

- l'absence de la prise en considération des exigences non-fonctionnelles : ces dernières sont aussi importantes que les aspects fonctionnels et sémantiques. Leur prise en compte est essentielle lors de la phase de modélisation des processus métier et au cours de la phase de réconciliation de services ;
- l'absence d'un registre (ou annuaire) de services rationalisé : ce dernier promeut la rationalisation de la gestion des services et de leurs propriétés non-fonctionnelles et s'attache à aller au delà d'un simple catalogue de services.

Ces constatations nous ont menés à poser ces quatre questions : *quelles exigences non-fonctionnelles sont nécessaires aux architectes métier pour exprimer leurs besoins non-fonctionnels ? Comment peuvent-ils annoter les processus métier collaboratifs à l'aide d'exigences non-fonctionnelles ? Comment pouvons-nous gérer et manipuler les propriétés non-fonctionnelles des services techniques ? Comment établir la réconciliation entre les exigences non-fonctionnelles des activités métier et les propriétés non-fonctionnelles des services techniques (parmi ceux qui répondent aux besoins fonctionnels) ?*

Telles sont les questions auxquelles nous essayons de répondre au cours des travaux décrits dans cette thèse. Mais avant de porter nos réflexions sur la recherche des réponses à ces questions, nous détaillons dans les sous-sections qui suivent ces verrous.

I.4.1. Modélisation et réconciliation métier / technique : la non prise en compte des propriétés non-fonctionnelles

Dans le cadre du projet MISE, les travaux de N. Boissel-Dallier [Boissel-Dallier, 2012] s'intéressent à la modélisation des processus métier enrichis par la sémantique et à la réconciliation entre les activités de ce processus et les services techniques. Ses travaux ont visé à étudier l'apport de la modélisation sémantique des services et à définir un moteur de réconciliation basé sur un filtre fonctionnel qui s'appuie sur l'utilisation d'ontologies. Ce filtre, détaillé dans la Figure 3, consiste à automatiser le passage de la cartographie de processus collaboratifs vers un système d'information fonctionnel en utilisant une base de connaissances (ou format d'ontologies).

Ceci est fait dans le but de tirer profit des apports de la sémantique pour les services. À l'issue de l'application de ce filtre, nous obtenons une liste de services (ou de compositions de services) qui répondent aux besoins fonctionnels de chacune des activités métier préalablement annotées sémantiquement. Ces services sont par la suite classés selon l'ordre croissant de leur réponse aux besoins fonctionnels de l'activité.

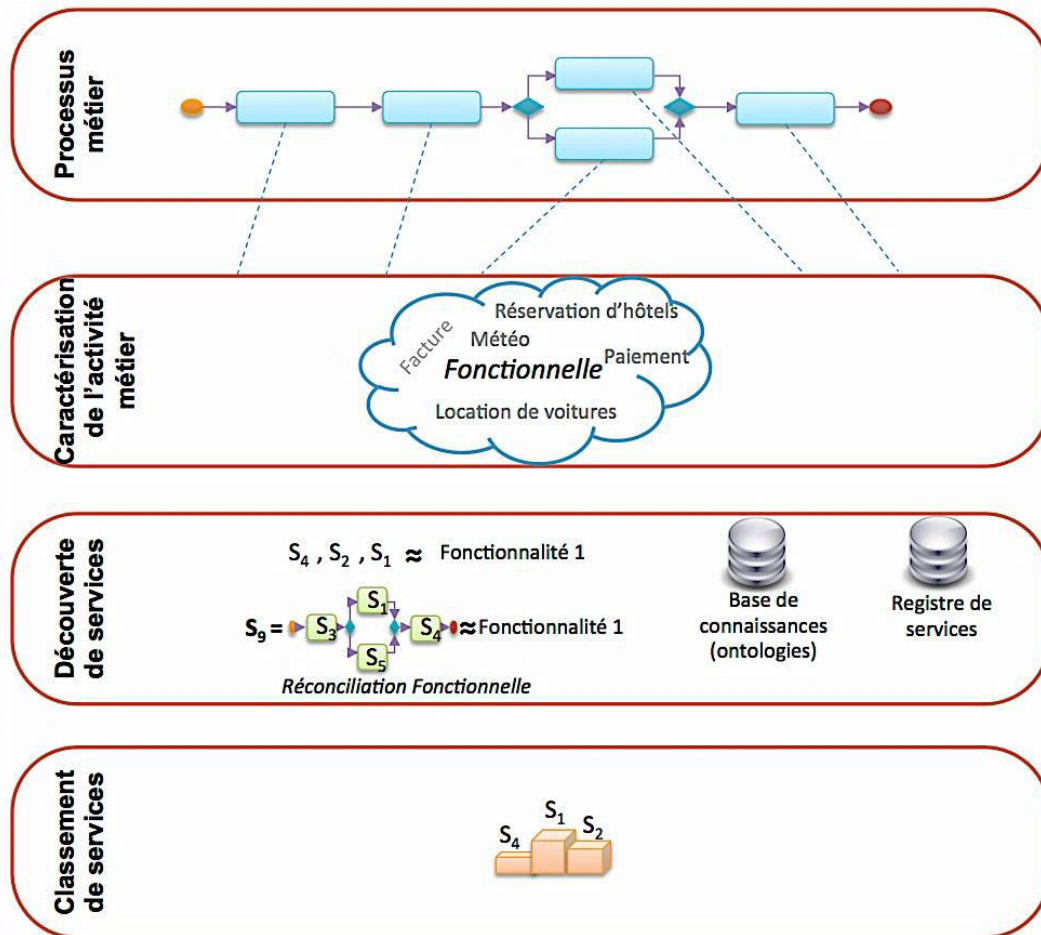


Figure 3 : Réconciliation de services d'un point de vue fonctionnel et sémantique.

Toutefois, cette liste peut contenir plusieurs services qui couvrent quasiment les mêmes besoins fonctionnels de l'activité métier du processus collaboratif. Le besoin d'affiner cette liste de services cibles est donc devenu majeur. Par conséquent, il est également devenu nécessaire d'améliorer les mécanismes de réconciliation de services en intégrant, en plus des exigences fonctionnelles, les exigences non-fonctionnelles (e.g. : la performance, le prix, le temps de réponse, la sécurité, etc.). Cette intégration permet, d'une part, d'optimiser le temps de la recherche et de calcul pour toute sélection de services et d'autre part d'améliorer la réactivité des entreprises. Pour cela et tel que la Figure 4 l'illustre, nous sommes face aux deux problématiques suivantes :

- *Affinement de la sélection unitaire de services* : il est essentiel que l'utilisateur renseigne sur les activités métier du processus collaboratif, en plus des aspects sémantiques, les exigences non-fonctionnelles. Ces propriétés non-fonctionnelles couvrent non seulement le niveau métier (telles que le prix, la note attribuée au service ou à son fournisseur, etc.) mais aussi les attentes d'un point de vue technique (à savoir : le temps de réponse, la disponibilité, etc.) ;
- *Affinement de la sélection de composition de services* : il arrive parfois de ne pas trouver un service qui répond aux besoins fonctionnels du client parmi ceux déjà publiés dans le registre. Cependant, il est nécessaire d'étendre la sélection unitaire de services par la composition d'un ensemble de services qui concordent fonctionnellement avec les exigences fonctionnelles de l'activité métier. Dans ce contexte dynamique, l'utilisation des informations non-fonctionnelles s'avère essentielle pour améliorer le choix des services cibles, à titre d'exemple nous citons la qualité de service (*Quality of Service - QoS*), la disponibilité, etc.

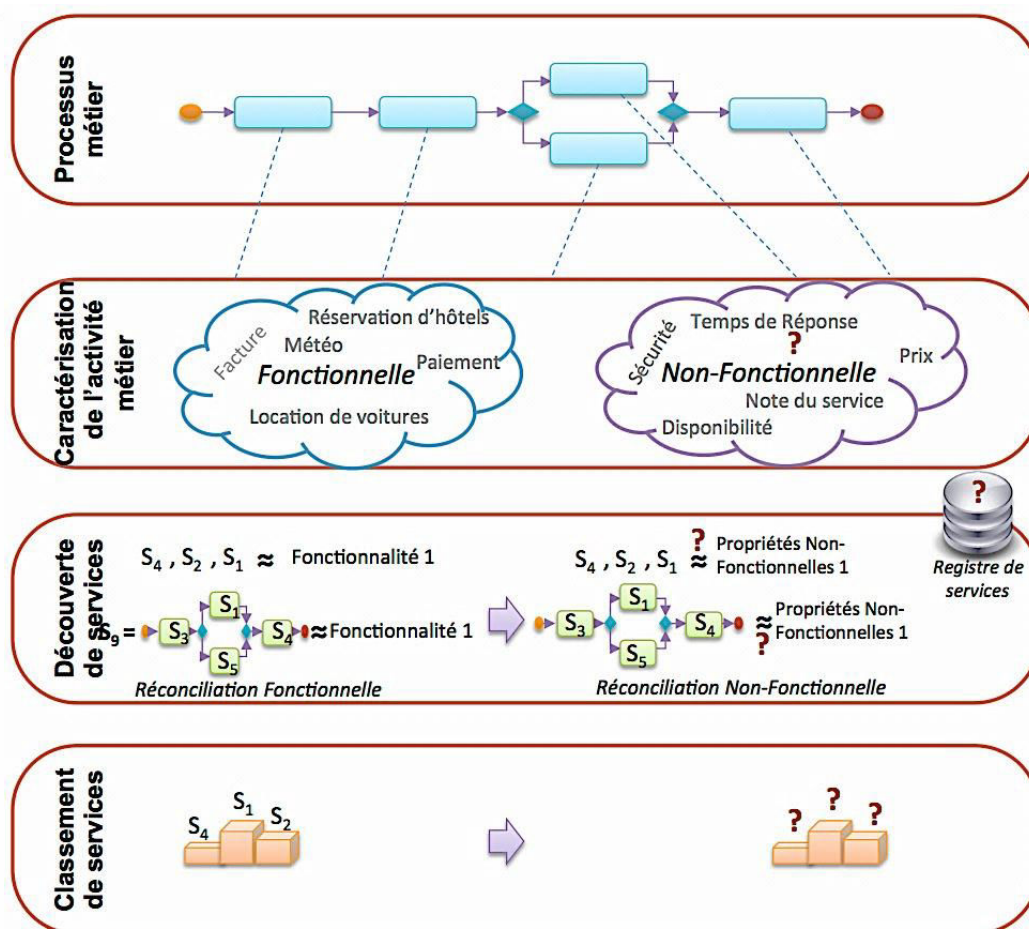


Figure 4 : Problématique générale des propriétés non-fonctionnelles lors de la réconciliation de services.

I.4.2. Manque de rationalisation de la gestion des services et de leurs propriétés non-fonctionnelles

La réutilisation de composants applicatifs est portée à son point d'orgue par cette volonté de bâtir toute application à partir de services existants, qu'il faut savoir identifier et invoquer. Ces services techniques répondent à un besoin fonctionnel et peuvent être assemblés dans des processus collaboratifs pour aboutir à des traitements applicatifs plus globaux.

La prolifération des services au sein du système d'information collaboratif des entreprises rend nécessaire la rationalisation de la gestion de leurs descriptions ainsi que la gestion des règles d'utilisation de ces services. La problématique de la gouvernance SOA adresse ces besoins en proposant un registre dédié à la gestion des services et de leurs métadonnées (description des services, les propriétés non-fonctionnelles, les contrats de services, etc.).

I.5. Présentation des objectifs de la thèse

La problématique porte principalement sur l'amélioration de la définition, le pilotage et la correspondance métier / technique. Cette section propose une méthodologie globale permettant : (i) de prendre en considération les exigences non-fonctionnelles lors de la définition des processus collaboratifs métier et (ii) d'affiner et enrichir, grâce à la gouvernance SOA, la problématique de la réconciliation entre le niveau métier et le niveau technique. L'objectif, est de prendre en compte les attentes non-fonctionnelles afin de construire un processus technique correspondant au processus métier qui pourra par la suite s'exécuter dans une infrastructure de services de type bus de services (*Enterprise Service Bus - ESB*) [Chappell, 2004].

Dans un premier temps, nous proposons un schéma d'amélioration du modèle de dynamique collaborative répondant de manière pertinente à la situation collaborative. Ce schéma s'appuie donc sur les résultats de la thèse de W. Mu [Mu, 2012] et envisage en particulier l'ajout d'un principe d'annotation non-fonctionnelle dans les modèles de cartographie collaborative de processus métier. Ceci dans le but d'associer aux processus déduits les éléments d'informations relatifs à certaines exigences attendues lors de l'exécution des processus collaboratifs (telles que la disponibilité, le temps de réponse, le niveau de sécurité, etc.). En effet, au moment de la définition des processus décrivant la dynamique collaborative, il sera possible d'annoter les activités métier à l'aide des propriétés qui expriment les attentes non-fonctionnelles, que ce soit au niveau métier ou bien lors de l'exécution (niveau technique).

Dans un deuxième temps, ces annotations non-fonctionnelles ajoutées au niveau du modèle comportemental collaboratif seront utilisées lors de la réconciliation afin d'améliorer la sélection de services. En effet, cette réconciliation selon ces aspects non-fonctionnels se base sur les résultats des

travaux de thèse de N. Boissel-Dallier [Boissel-Dallier, 2012]. Cependant, les règles de réconciliation (et en particulier les règles d'évaluation de la pertinence des services techniques vis-à-vis des attentes fonctionnelles des activités métier) seront enrichies de considérations non-fonctionnelles. Nous proposons donc un premier environnement de gouvernance SOA permettant de rationaliser la gestion des connaissances (en particulier, les exigences non-fonctionnelles) sur les services et d'affiner les évaluations et améliorer ainsi la sélection et le classement des services techniques en prenant en compte les aspects non-fonctionnels.

La Figure 5 résume la méthodologie proposée pour l'amélioration du passage du processus collaboratif à la sélection des services techniques qui répondent au besoin du métier que ce soit d'un point de vue fonctionnel (en tenant compte des aspects sémantiques) mais aussi des aspects non-fonctionnels.

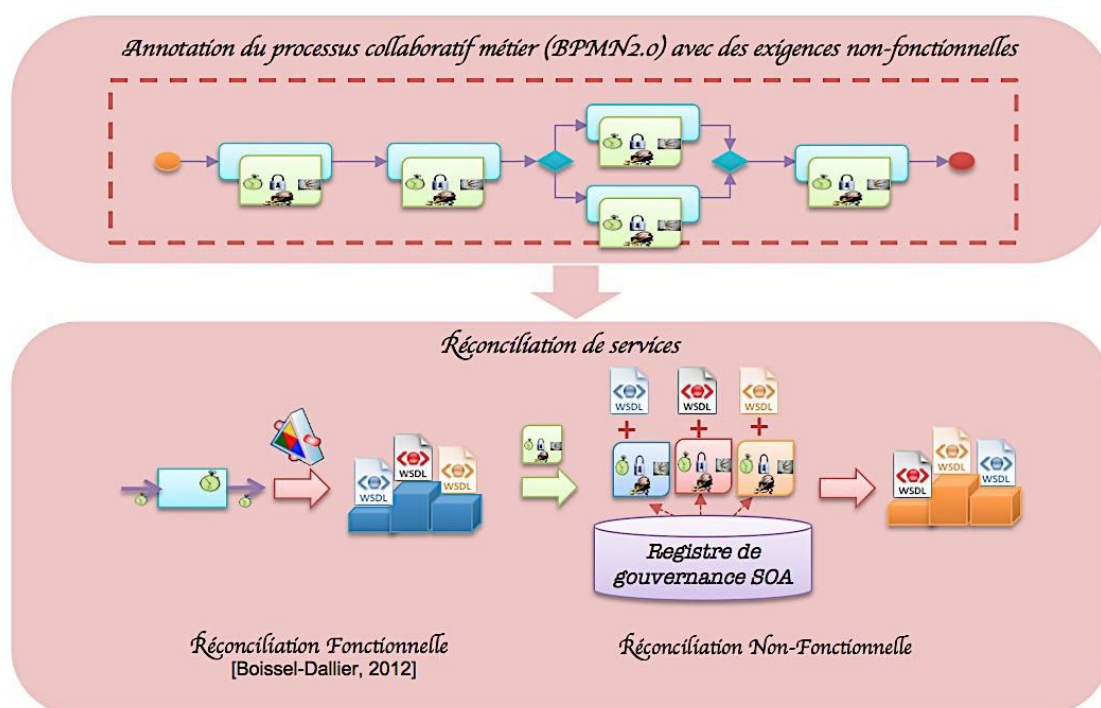


Figure 5 : Méthodologie générale des travaux de la thèse.

I.6. Positionnement des travaux de la thèse

Les travaux de cette thèse de doctorat s'inscrivent dans une dynamique de collaboration entre l'équipe MISE du Centre de Génie Industriel de l'École des Mines d'Albi et le pôle de recherche Linagora Labs de la société Linagora. Ces deux partenaires ont eu l'occasion de collaborer à maintes

reprises dans le cadre de projets de recherche et de thèses de doctorat. Dans ce qui suit, nous présentons le positionnement de nos travaux par rapport à chacun d'eux.

I.6.1. Positionnement par rapport à Linagora Labs

Créée il y a 13 ans, Linagora est une entreprise qui s'inscrit dans le cadre de l'industrie du logiciel libre (*Open Source*). Elle est membre du consortium OW2¹ et comprend un pôle de recherche, représenté par Linagora Labs (anciennement Petals Labs). Linagora Labs investigate des opportunités d'innovation autour des produits Linagora. Une équipe de chercheurs, docteurs et ingénieurs en informatique assurent la gestion et la participation de Linagora aux projets de recherche auprès de partenaires européens, académiques et industriels.

Les principaux domaines de recherche de Linagora Labs concernent : (i) les architectures SOA (la description de services, la composition de services, l'orchestration, le déploiement dans une infrastructure de services ESB et la chorégraphie de services) et (ii) les architectures intergicielles (*middleware*) dans le nuage (*cloud*). Comme l'illustre la Figure 6, cette équipe dispose d'une pile complète d'outils et de bibliothèques libres couvrant les principaux besoins de la SOA. Cela comprend une infrastructure de services (un bus de services) « EasyESB » et ses principaux composants de communication « EasyBpel », des couches applicatives dédiées à la gestion de services, à la sémantique « EasierSBS » et à la surveillance (*monitoring*) « EasierBSM & EasierCos ».

À cette pile s'ajoutent deux outils graphiques : « Petals BPM » qui permet de modéliser des processus BPMN 2.0 (*Business Process Model and Notation*) et « EasiestDemo » qui permet de produire des démonstrations impliquant l'ensemble des composants cités précédemment.



Figure 6 : Résultats de recherche de Linagora Labs.

¹ OW2 (*The open source community for infrastructure software*) : est une communauté internationale indépendante dédiée au développement d'intergiciels industriels libres (<http://www.ow2.org>).

Par ailleurs, Petals Labs (nouvellement Linagora Labs) a conduit, en collaboration avec le centre de Génie Industriel de l'École des Mines, deux thèses portant sur la qualification des situations collaboratives et sur leur implémentation. Ces travaux ont produit des prototypes permettant : (i) d'automatiser le passage de l'expression du besoin collaboratif à la modélisation du processus collaboratif métier, (ii) d'enrichir la modélisation de ces processus collaboratifs en les annotant à l'aide des aspects sémantiques et enfin (iii) d'automatiser la transformation vers un processus collaboratif technique.

C'est dans ce même contexte collaboratif que les travaux de cette thèse se déroulent afin de : (i) enrichir davantage l'outil de modélisation des processus BPMN : Petals BPM par la prise en considération des aspects non-fonctionnels et (ii) rationaliser la gestion des services et de leurs propriétés non-fonctionnelles par la gouvernance SOA.

En effet, Petals Labs a développé un premier logiciel de gouvernance SOA appelé : Petals Master [Master, 2009]. Ce logiciel libre est conçu pour gérer les services : les cataloguer, les indexer, les documenter, les retrouver à partir des informations syntaxiques et sémantiques. Toutefois, Petals Master manquait de certaines fonctionnalités nécessaires surtout dans un contexte de collaboration par lequel les entreprises sont de plus en plus concernées. Parmi ces fonctionnalités nous citons : le stockage et l'utilisation des propriétés non-fonctionnelles pour la gestion des services, et la gestion des contrats de services. Ce besoin de bâtir un environnement complet et évolué pour la gouvernance SOA à partir de la panoplie existante de l'équipe Petals Labs a conduit à sa participation au projet de recherche européen FP7 CHOReOS (*Large Scale CHOReOgraphieS for the Future Internet*) [CHOReOS, 2010] et au sein duquel s'inscrivent les travaux décrits dans ce manuscrit de thèse.

I.6.2. Positionnement par rapport à MISE

Comme nous l'avons détaillé précédemment (cf. section I.3), le projet MISE vise à concevoir, à réaliser et à outiller une approche et des méthodes pour la conception de SIM collaboratif. Les travaux présentés dans ce manuscrit font partie de la deuxième itération du projet et particulièrement au passage depuis la génération d'une cartographie de processus répondant à une situation collaborative vers le système exécutable. Autrement dit, depuis le niveau métier vers un niveau technique (cf. Figure 7).

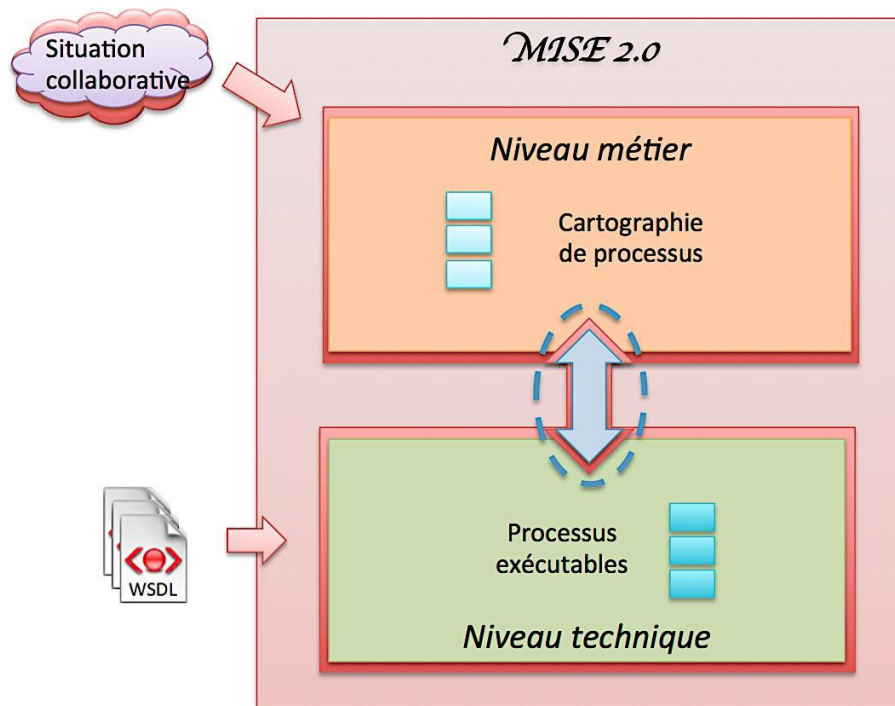


Figure 7 : Positionnement des travaux de la thèse dans MISE 2.0.

I.7. Conclusion

Au cours de ce chapitre, nous avons présenté le contexte général des travaux de la thèse, les problématiques à traiter et les solutions que nous proposons pour en faire face. Avec l'évolution et la complexité de l'environnement dans lequel se situent les entreprises, ces dernières se retrouvent dans le besoin d'innover et de collaborer entre elles pour répondre à des objectifs plus globaux. Ceci mène les entreprises à faire évoluer leur SI vers un Système d'Information de Médiation plus flexible, rapide, et dynamique qui se repose sur l'Architecture Orientée Services pour son apport en termes de couplage lâche et d'agilité, et qui supporte ces processus collaboratifs de l'entreprise. Dans ce contexte collaboratif, les travaux de cette thèse se situent et concernent à la fois le centre Génie Industriel de l'école des Mines d'Albi-Carmaux et l'équipe de recherche Linagora Labs (anciennement Petals Labs) de l'entreprise Linagora. La conception d'un tel système repose sur plusieurs phases. Des travaux de thèse ont été déjà réalisés dans le cadre du projet MISE qui permettent : (i) à partir d'une situation de collaboration, de définir une cartographie de processus qui répond aux besoins métier de l'entreprise ; (ii) d'effectuer la réconciliation de ces processus métier en sélectionnant les services techniques correspondant et de construire les processus exécutables. Néanmoins, certaines fonctionnalités essentielles à l'évolution de tout SI d'une entreprise n'ont pas été abordées et il est donc nécessaire d'améliorer ce passage depuis la définition de la cartographie des processus vers les processus

exécutables. Les travaux de thèse décrits dans ce manuscrit relèvent de cette amélioration et portent sur trois points essentiels :

- la prise en compte des exigences non-fonctionnelles lors de la modélisation des processus collaboratifs par l'annotation non-fonctionnelle des activités métier ;
- la rationalisation de la gestion des services dans un registre robuste. Ce registre doit également aller au delà du stockage et du catalogage des services, mais aussi indexer les métadonnées autour des services telles que les propriétés non-fonctionnelles.
- la prise en considération des exigences non-fonctionnelles lors de la réconciliation métier / technique.

Chapitre II

Etat de l'art

« Rien n'est complet ; à tout il manque quelque chose. »

Victor Hugo

II.1. INTRODUCTION	23
II.2. ARCHITECTURE ORIENTEE SERVICES (SOA)	23
II.2.1. Les services	24
II.2.2. Les services Web	26
II.2.3. Les propriétés non-fonctionnelles des services Web	28
II.2.4. Synthèse sur les propriétés non-fonctionnelles	31
II.3. RATIONALISATION DE LA GESTION DES SERVICES ET DE LEURS PROPRIETES NON-FONCTIONNELLES : GOUVERNANCE SOA	32
II.3.1. Qu'est-ce que la gouvernance SOA ?	32
II.3.2. Les cadres de gouvernance SOA	36
II.3.3. Synthèse sur les cadres existants	37
II.4. GESTION DES PROCESSUS METIER (BPM)	40
II.4.1. Les processus et les processus métier	40
II.4.2. Gestion des processus métier	41
II.4.3. Classification des processus métier	42
II.5. BPM DANS UN CONTEXTE SOA	43
II.5.1. Composition de services Web	43
II.5.2. Modélisation graphique des processus métier orientés SOA	44
II.5.3. Synthèse sur la modélisation des processus métier orientés SOA	48
II.5.4. Annotation du BPMN avec des exigences non-fonctionnelles	48
II.5.5. Synthèse sur les annotations du BPMN à l'aide d'exigences non-fonctionnelles	52

II.6. RECONCILIATION NON-FONCTIONNELLE	53
II.6.1. Etude des approches existantes.....	53
II.6.2. Synthèse sur la réconciliation non-fonctionnelle.....	55
II.7. CONCLUSION.....	55

II.1. Introduction

Ce chapitre présente une étude bibliographique autour des concepts, des approches et des technologies liés aux objectifs abordés dans le chapitre précédent : la gestion des propriétés non-fonctionnelles des services, la prise en compte de ces propriétés lors de l'étape de modélisation des processus métier et lors de la réconciliation entre les activités métier et les services techniques, et enfin la rationalisation de la gestion des services et de leurs métadonnées en utilisant la gouvernance SOA. Cet état de l'art sert de base au travail de recherche visant à atteindre ces objectifs. En nous basant sur la littérature, nous commençons par détailler l'architecture orientée service, sa brique principale : les services, les services Web ainsi que leurs propriétés non-fonctionnelles. Par la suite, nous introduisons les processus métier d'une façon générale avant de nous intéresser aux processus métier appliqués dans un contexte SOA. Nous étudions les standards existants pour la modélisation de ces processus et pour l'annotation avec des exigences non-fonctionnelles. Enfin, nous étudions la gouvernance SOA et les approches de sélection de services selon les exigences non-fonctionnelles.

II.2. Architecture orientée services (SOA)

En quelques années, l'architecture orientée services est devenue un thème majeur pour le système d'information des entreprises. Elle représente une nouvelle approche de conception des applications qui fait gagner en flexibilité, en agilité et en réactivité au niveau du système d'information en reposant sur le principe de couplage lâche et tout en garantissant l'interopérabilité entre ces applications [Erl, 2004], [Krafzig et al., 2005]. Ceci a encouragé les entreprises à baser leur système d'information sur cette nouvelle philosophie. Bieberstein définit l'architecture SOA comme suit :

« *Service-Oriented Architecture enables organizations to be agile and flexible enough to adopt new business strategies and produce new services to overcome the challenges created by business dynamism today.* » [Bieberstein, 2006]

La SOA représente une nouvelle forme architecturale pour les systèmes d'information. Elle fournit une vision où les applications (internes et externes) sont exposées sous forme de services et peuvent communiquer entre elles d'une manière simple [Raymond, 2007]. Dans cette vision, chaque application fournit une partie (ou l'ensemble) de ses fonctionnalités sous forme de services invocables par d'autres applications. Ce concept d'architecture utilise donc les services comme des composants principaux pour la construction des applications distribuées d'une manière facile et peu coûteuse. Ceci

permet au système d'information de prendre en compte les quatre enjeux suivants [Jardim-Goncalves et al., 2006], [Dan et al., 2008] et [Papazoglou et al., 2008] :

- *Rationalisation* : diminution de la duplication des modules et du code, en réduisant les dépendances entre les applications et en urbanisant les liens inter-applicatifs. D'où une réduction des coûts ;
- *Interopérabilité* : possibilité pour les blocs du système d'information (composants, applications, services, etc.) d'échanger des données et des informations tout en assurant la cohérence entre eux ;
- *Réutilisation* : réduction des risques et de la complexité du système d'information par la réutilisation des composants (ou une partie de leurs codes testés) avant de décider de créer des nouveaux ;
- *Agilité* : capacité du système d'information à appréhender et à prendre en compte les évolutions du métier de l'entreprise en y apportant des réponses simples, efficaces et rapides.

II.2.1. Les services

Les services représentent la brique de base d'une architecture SOA et peuvent être définis ainsi :

« Un service est une unité logicielle autonome et indépendante de toute plateforme, qui peut être décrite, publiée, découverte, orchestrée et programmée en utilisant des protocoles standards dans le but de créer des réseaux d'applications collaboratives distribuées à très grande échelle. » [Baligand, 2008]

Un service est composé de trois parties principales, tel que illustré dans la Figure 8 : (i) l'interface qui constitue la partie « *publique* » et contient entre autres la liste des opérations, les entrées et les sorties, (ii) l'implémentation qui constitue la partie « *privée* » et correspond au développement des fonctionnalités du service. Même si l'implémentation dépend de la plateforme utilisée, l'approche SOA permet d'abstraire l'hétérogénéité technique grâce à la publication d'une interface commune appellable. Enfin (iii) le point de déploiement (*end-point*) qui représente le point par lequel le service peut être invoqué, généralement il correspond à une adresse URL (*Uniform Resource Locator*).

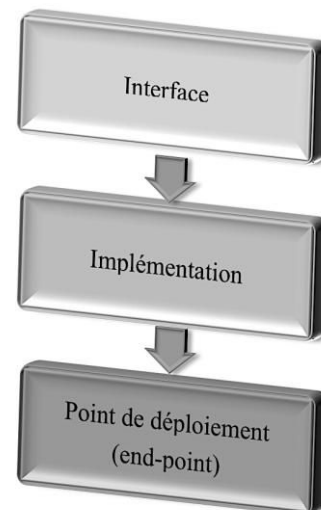


Figure 8 : Structuration d'un service.

Ce qui distingue l'architecture SOA des autres architectures, c'est le couplage lâche et la granularité des services. Ces propriétés permettent de faciliter la reconfiguration des processus quand les fonctions métier évoluent ou changent. Le concept de service, qui est véhiculé par le modèle de l'architecture orientée services, se veut indépendant de la mise en œuvre des applications constituantes. D'autre part, la granularité des services rend possible et plus flexible l'adaptation, l'évolution des applications et leurs collaborations avec d'autres.

Le fournisseur du service et les utilisateurs (appelés aussi consommateurs) interagissent entre eux sans qu'ils n'aient à faire de concession sur la structure propre, le comportement interne et l'implantation des services. Concrètement, cela se déroule en quatre phases réunificatrices comme l'illustre la Figure 9 ci-après.

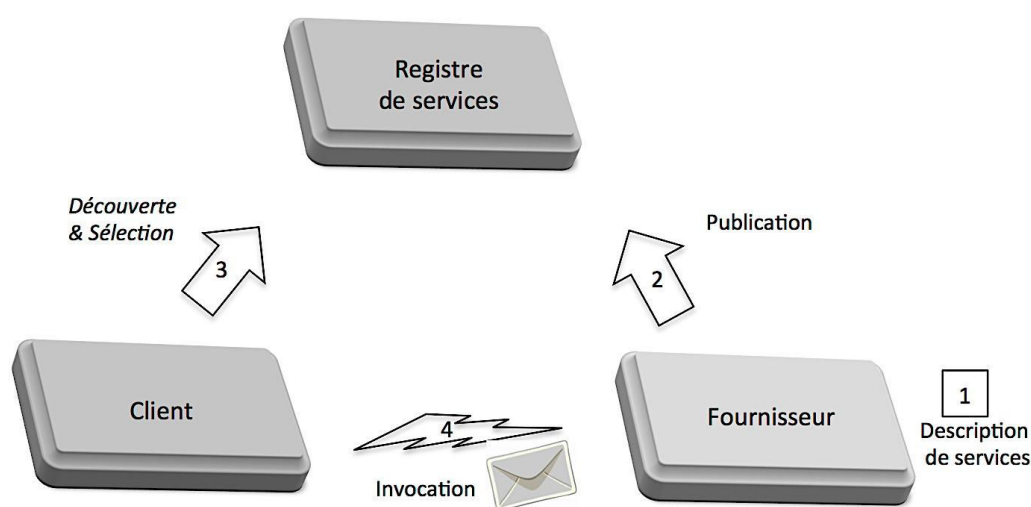


Figure 9 : Principe de fonctionnement des services.

- 1) La *description de services* : les fournisseurs décrivent tous les détails concernant les fonctionnalités auxquelles leurs services répondent, à savoir : les paramètres d'entrée, le format et le type de données retournées ainsi que le protocole à utiliser lors des requêtes dans le but de faciliter la tâche aux fournisseurs ;
- 2) La *publication de services* : les fournisseurs, après avoir décrit leurs services, vont les publier à l'extérieur pour qu'ils soient disponibles pour les utilisateurs. La publication consiste donc à enregistrer les descriptions de services dans le registre ;
- 3) La *découverte et la sélection de services* : cette phase recouvre la possibilité donnée aux utilisateurs de rechercher un service parmi tous ceux qui ont été déjà publiés dans le registre de services. Par la suite, l'utilisateur peut choisir un ou plusieurs services disponibles ;

- 4) *L'invocation de services* : cette dernière phase consiste à invoquer le service choisi pour l'utiliser. L'invocation se fait via la transmission de messages.

II.2.2. Les services Web

Les services Web sont des services qui peuvent être publiés, découverts et invoqués par le biais de l'internet. Cette technologie de services constitue une solution pour implémenter une architecture SOA [Newcomer et al., 2004]. Cependant, un service Web peut être considéré comme étant une application autonome, mise à disposition d'autres applications et qui peut être publiée, découverte et invoquée via une infrastructure du Web [Tidwell, 2000] et [Alonso et al., 2004]. Le consortium UDDI (*Universal Description, Discovery and Integration*) fournit une définition plus spécifique d'un service Web et le caractérise comme suit :

« *Un service Web est une application métier autonome et modulaire qui possède des interfaces basées sur des standards et orientées Internet.* » [OASIS, 2000]

Selon cette définition, les services Web fonctionnent d'une manière qui n'est pas intégrée. C'est-à-dire qu'au lieu d'intégrer toutes les fonctionnalités dans une seule application globale, chacune des fonctionnalités est développée en tant qu'un service Web, ce qui facilitera sa réutilisation par d'autres applications. Le consortium W3C propose une définition encore plus détaillée autour des services Web :

« *A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages.* » [W3C, 2004]

Le service Web est une application logicielle, autonome, modulaire et appelable via les protocoles du Web par d'autres applications existantes sur le Web permettant l'échange de données. Les requêtes de demandes et de réponses doivent être conformes aux protocoles et technologies standards. Les trois standards les plus connus et piliers des services Web sont les suivants :

- **WSDL** : (*Web Service Description Language*) [Christensen et al., 2001] est un format XML qui décrit les informations nécessaires des services (comment se connecter et utiliser un service Web, etc.). Il décrit : (i) l'interface publique : les différentes opérations qu'elle propose et la structure des messages d'entrée/sortie pour chacune et (ii) le protocole de

communication : le format des messages que le service utilise, les méthodes que le client peut invoquer et la localisation du service ;

- **UDDI** : (*Universal Description, Discovery and Integration*) [OASIS, 2000] normalise une solution d'annuaire distribué de services Web, permettant à la fois la publication et l'exploration (recherche) de services ;
- **SOAP** : (*Simple Object Access Protocol*) [W3C, 2000] est un protocole d'échange inter-applications indépendant de toute plateforme et basé sur le langage XML.

Ces standards permettent, dès lors qu'ils sont implémentés sur les plateformes et les produits, aux clients et aux services de communiquer sur un très large spectre de plateformes et d'environnements. Dans la Figure 10, nous décrivons le principe de la découverte et de l'utilisation d'un service Web. Le client souhaitant rechercher un service Web effectue une requête de découverte sur le registre de services UDDI. Le registre, à son tour, trie parmi les services dont il dispose en se basant sur les critères demandés par le client avant de lui retourner une liste d'adresses des fichiers WSDL qui répondent à ses besoins. Le client pourra par la suite accéder au service via le document WSDL et l'invoquer pour l'utiliser grâce au protocole SOAP.

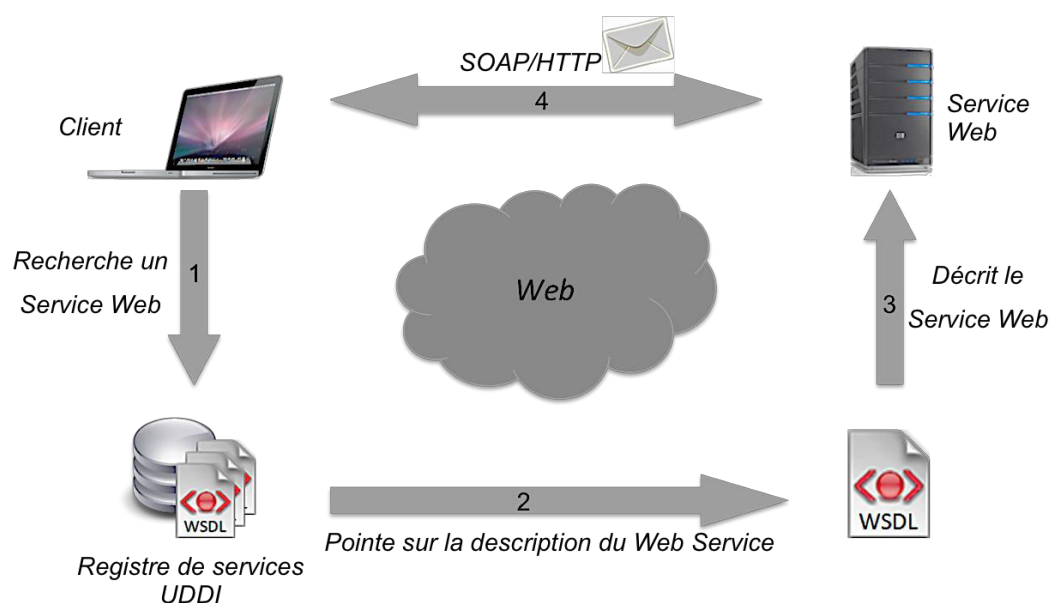


Figure 10 : Principe de découverte et d'utilisation d'un service Web.

Les approches basées sur ces trois technologies, à savoir WSDL, UDDI et SOAP, permettent une découverte fonctionnelle des services Web. C'est-à-dire une découverte selon les propriétés fonctionnelles qui décrivent les opérations accomplies par le service. D'autres approches ont été développées afin d'enrichir cette découverte. Parmi ces approches nous citons : les *approches non-fonctionnelles* qui proposent des propriétés non-fonctionnelles pour définir les services Web d'un

point de vue qui va au delà de ses fonctionnalités intrinsèques telles que : la performance, la qualité, la sécurité, l'interopérabilité, la fiabilité, etc. Au cours de ces travaux, notre intérêt se porte sur les approches non-fonctionnelles afin d'enrichir davantage la description des services.

II.2.3. Les propriétés non-fonctionnelles des services Web

Les propriétés non-fonctionnelles jouent un rôle très important et représentent un critère essentiel pour la pertinence d'un service [Papazoglou et al., 2007] lors de la sélection [Liu et al., 2004] ou de la composition [Zeng et al., 2004]. Cette section permet de décrire les propriétés non-fonctionnelles des services et de présenter quelques approches courantes de leur modélisation.

Les propriétés non-fonctionnelles, aussi appelées les propriétés extra-fonctionnelles, des services concernent tout ce qui ne découle pas directement des aspects fonctionnels comme la performance, le prix, la sécurité [Chung et al., 1995]. À titre d'exemple, dans le projet CHOReOS, pour pouvoir faire un déploiement auto-adaptatif du bus de services, un service Web Amazon offrant cette fonctionnalité est utilisé. Toutefois, l'utilisation est payante. Le prix du service constitue une propriété non-fonctionnelle.

Dans la littérature et dans une architecture SOA, les propriétés non-fonctionnelles d'un service sont souvent liées au concept de la qualité de service (*Quality of Service* - QoS) et sont même parfois utilisées sans aucune distinction. Toutefois, les auteurs dans [Cardoso et al., 2004], ne limitent pas les propriétés non-fonctionnelles qu'à ce seul concept de QoS et proposent une classification en deux catégories : les propriétés quantitatives et les propriétés qualitatives. Les propriétés quantitatives sont exprimées par des valeurs et dépendent fortement de l'infrastructure réseau tels que le temps de réponse, la disponibilité, la fiabilité, etc. Les propriétés qualitatives ne dépendent pas de l'infrastructure réseau et sont parfois plus simples à énoncer, par exemple la sécurité et l'interopérabilité qui sont exprimées en valeur booléenne, etc. Dans la suite de ce manuscrit, nous considérons cette définition contextuelle des propriétés non-fonctionnelles :

« Les propriétés non-fonctionnelles d'un service Web regroupent l'ensemble des paramètres qui correspondent à la réalisation de ses fonctionnalités. Ces paramètres peuvent être quantitatifs ou qualitatifs. C'est-à-dire l'ensemble de ses caractéristiques qui relèvent de l'infrastructure réseau (*Qualité de Service*) et celles qui ne le sont pas (*sécurité, interopérabilité, etc.*). » [Cardoso et al., 2004]

Plusieurs approches et modèles se sont intéressés à enrichir la description des services Web en rajoutant des propriétés non-fonctionnelles [O'Brien et al., 2007]. Dans ce qui suit, nous présentons quelques modèles et langages pour l'annotation non-fonctionnelle des services.

QoS-based WS Description : *Quality of Service Web Service Description* est une ontologie développée sous le nom OWL-Q (*Web Ontology Language for Quality*). Elle représente une extension du standard OWL-S (*Web Ontology Language for Services*) [Kritikos et al., 2006]. Elle est basée sur un ensemble de facettes (onze facettes) qui sont développées séparément et qui peuvent être étendues. La facette principale concerne la définition d'un système complet pour associer des métriques aux propriétés non-fonctionnelles. Si ce langage permet la distinction entre les propriétés statiques et celles qui ne le sont pas (c'est-à-dire les propriétés qui ont des valeurs qui peuvent changer lors de l'exécution du service telles que le temps de réponse, la disponibilité, etc.), il ne permet pas de définir une hiérarchie entre ces attributs et ne fournit pas de liaisons avec les standards de qualité existants.

WS-QoS Ont : *Web Service-Quality of Service Ontology* est une ontologie proposée par Tran dans [Tran, 2008] et fournit une description de la QoS pour les services Web. Cette ontologie fournit une taxonomie de propriétés qui couvre la performance, la disponibilité, la fiabilité et les aspects de sécurité. Néanmoins, cette ontologie ne prend pas en considération les différents niveaux des propriétés non-fonctionnelles. C'est-à-dire les propriétés non-fonctionnelles qui sont appliquées au niveau métier et celles qui sont plutôt techniques.

WS-Policy : *Web Service-Policy*, est un standard développé par le consortium W3C en 2006 [W3C, 2006]. Il est composé de deux spécifications XML distinctes : *WS-Policy* et *WS-Policy Attachment* (ce dernier est défini pour permettre la compatibilité entre les différentes versions des standards WSDL et WS-Policy). Ce standard a pour objet d'améliorer l'interaction entre le fournisseur du service et les utilisateurs en rajoutant des informations supplémentaires (ou métadonnées) concernant les services (au-delà de celles qui sont déjà prévues dans le document WSDL). Le concept de base de WS Policy est l'*assertion*. Une assertion est une métadonnée qui exprime une exigence spécifique pour un service donné. Par exemple, une assertion pourrait indiquer des propriétés non-fonctionnelles telles que : un niveau requis de sécurité, l'utilisation d'un certain protocole, ou un temps de réponse requis lors de l'invocation du service, etc. Une assertion dans WS-Policy peut prendre la forme suivante :

<ServiceQuality ResponseTime_seconde = "1">

WSQF : la spécification de *Web Service Quality Factors* a été publiée en 2010 par le comité technique du *Web Service Quality Model* en tant que spécification d'OASIS [OASIS, 2010]. Elle regroupe les caractéristiques de base d'un service Web en proposant un ensemble de propriétés (ou facteurs) non-fonctionnelles. Comme l'illustre la Figure 11, ces propriétés sont classées en six groupes, à savoir *Business Value Quality*, *Service Level Measurement*, *Interoperability*, *Business Processing*, *Manageability* et enfin *Security*.

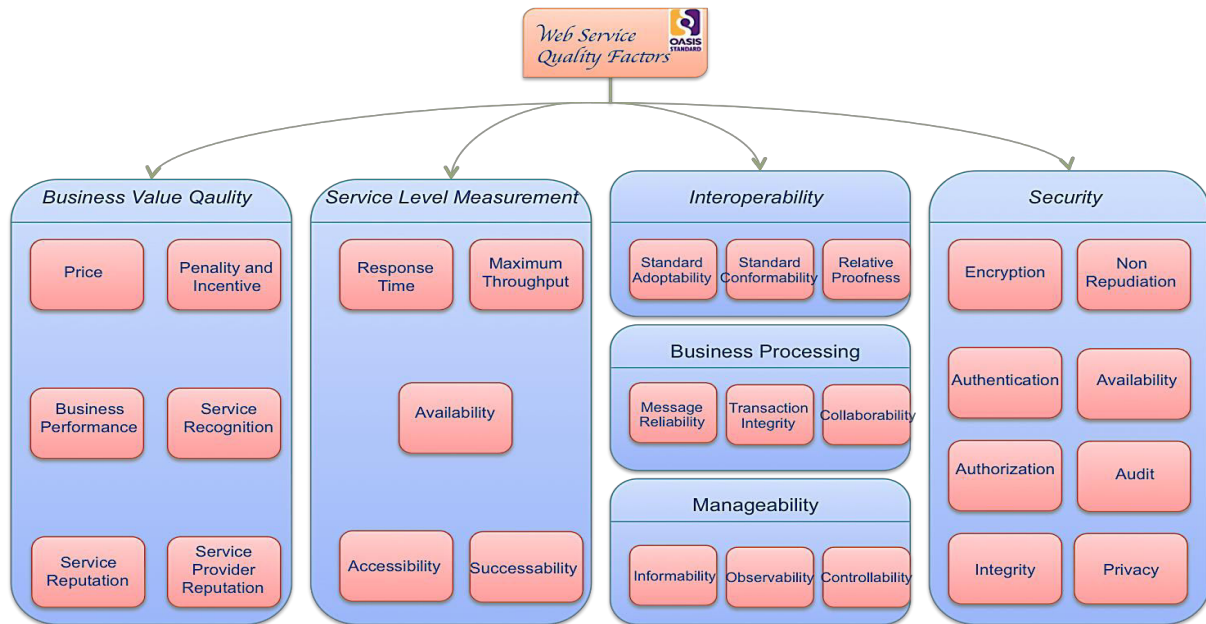


Figure 11 : Propriétés non-fonctionnelles proposées par le standard WSQF. [OASIS, 2010].

- Valeur de la qualité métier (*Business Value Quality*) : cette catégorie de propriétés se réfère à l'évaluation des services Web d'un point de vue métier. Parmi les propriétés non-fonctionnelles qui le composent, nous citons : le prix (*Price*), la popularité du service (*Service Recognition*), la réputation du service (*Service Reputation*), la réputation du fournisseur du service (*Service Provider Reputation*), etc.
- Mesure du niveau de la qualité de service (*Service Level Measurement Quality*) : cette catégorie caractérise la qualité du service qui doit accompagner l'utilisation du service. Cette catégorie regroupe le temps de réponse (*Response Time*), le débit maximum (*Maximum Throughput*), la disponibilité (*Availability*, représente le pourcentage du temps au cours duquel le service Web, en état de fonctionnement, est disponible), l'accessibilité (*Accessibility*, représente le pourcentage du temps au cours duquel le service Web, en état de fonctionnement et disponible sur le réseau, est accessible) et enfin la successabilité (*Successability*, représente le pourcentage de succès lors d'une exécution complète du service Web) ;
- Interopérabilité (*Interoperability*) : afin de garantir l'interopérabilité entre les services Web, l'utilisation conforme des standards lors de l'implémentation est fortement conseillée. Parmi les propriétés non-fonctionnelles d'interopérabilité, nous citons l'adoptabilité des normes (*Standard Adoptability*), la conformité aux standards (*Standard Conformability*) et la pérennité relative (*Relative Proofness*) ;
- Traitement métier (*Business Processing*) : les services Web peuvent être menés à être utilisés dans des échanges critiques entre les partenaires métier, et dans ce cas, la fiabilité et la stabilité des services Web sont très importantes. *Business Processing* évalue ces

éléments intégrant aussi la fiabilité des messages (*Message Reliability*), l'intégrité des transactions (*Transaction Integrity*) et la capacité de collaboration (*Collaborability*, représente la capacité d'une plateforme de services à définir, contrôler et gérer les flux de services entre les participants afin de permettre aux services Web de participer dans une collaboration telle qu'une orchestration ou une chorégraphie) ;

- *Gérabilité (Manageability)* : les fonctions de gérabilité permettent d'évaluer le fonctionnement du service par rapport à ce qui a été prévu. Par ailleurs, en cas de défaillance, ces fonctions permettent d'évaluer l'existence des informations primitives et nécessaires au maintien du fonctionnement du service ;
- *Sécurité (Security)* : les services Web sont fréquemment exposés à des réseaux publics ce qui les rend vulnérables aux attaques et aux fraudes. La sécurité identifie le degré de protection du service par rapport aux diverses menaces. Nous retrouvons dans cette catégorie : le cryptage (*Encryption*), la non-répudiation (*Non-Repudiation*), l'authentification (*Authentication*), la disponibilité (*Availability*), l'autorisation (*Authorization*), le contrôle (*Audit*), l'intégrité (*Integrity*), et la confidentialité (*Privacy*).

Nous invitons le lecteur à se référer à l'Annexe 1 pour retrouver la description de toutes les propriétés non-fonctionnelles proposées par la spécification WSQF.

II.2.4. Synthèse sur les propriétés non-fonctionnelles

De nombreuses approches basées sur des ontologies ont été proposées pour l'amélioration de la description des services Web en rajoutant des propriétés non-fonctionnelles comme QoS-based WS Description et WS-QoSOnt. Tous ces travaux ne permettent pas d'identifier des niveaux d'abstractions des propriétés non-fonctionnelles, c'est-à-dire les propriétés qui concernent le niveau métier et celles qui sont plutôt d'un niveau technique. De même, ces approches basées sur les ontologies ne permettent pas de distinguer les propriétés non-fonctionnelles qui sont mesurables (qui doivent être contrôlées et suivies lors de l'exécution du service Web) de celles qui ne le sont pas.

Ainsi, les propriétés non-fonctionnelles telles que le temps de réponse ou la disponibilité (qui sont des propriétés mesurables) se retrouvent au même niveau que la sécurité ou l'interopérabilité par exemple. De plus, certaines de ces approches ne se sont pas intéressées à établir des liaisons avec les différents standards qui existent pour la qualité de services (par exemple WS-Security pour les propriétés de sécurité). Cependant, il n'existe pas de lien concret entre les propriétés non-fonctionnelles définies par ces approches et les propriétés des différents standards de qualité. Pourtant, ce type de lien s'avère crucial notamment pour l'interopérabilité et pour éviter les problèmes de terminologie (en utilisant des standards communément admis).

Ces problèmes ont été résolus par l'apparition de la spécification WSQF de OASIS [OASIS, 2010]. Elle présente une solution plutôt complète pour la description des services Web en prenant en compte l'aspect non-fonctionnel. WSQF définit un ensemble assez large de propriétés non-fonctionnelles et offre la possibilité d'être lié à des standards tels que le WS-Interoperability (WS-I) pour la gestion de l'interopérabilité, le standard WS-Security pour la gestion de la sécurité, ou le standard WS-Policy pour la gestion des règles. Pour ses multiples avantages, nous avons décidé d'utiliser la spécification WSQF d'OASIS pour l'amélioration de la définition des services par la prise en considération des propriétés non-fonctionnelles.

II.3. Rationalisation de la gestion des services et de leurs propriétés non-fonctionnelles : gouvernance SOA

Avec la prolifération des services au sein du système d'information, il est devenu essentiel aux entreprises de rationaliser leur gestion ainsi que la gestion de leurs propriétés non-fonctionnelles. La problématique de la gouvernance permet d'adresser ces besoins. Nous présentons dans cette section un des composants clés de l'architecture SOA : *la gouvernance SOA* ainsi qu'une étude sur les cadres de gouvernance existants.

II.3.1. Qu'est-ce que la gouvernance SOA?

Le terme *gouvernance* vient du grec ancien *kubernân* qui signifie piloter un navire ou un char [De Oliveira Barata, 2003]. Dans le monde des entreprises, la gouvernance d'entreprise (*Corporate Governance*) représente une fonction essentielle pour les systèmes d'information et peut être définie comme étant un processus que l'entreprise met en place pour veiller à ce que les choses soient faites conformément aux règles préalablement définies. Dans [Boutillier et al., 2002], les auteurs définissent la gouvernance d'entreprise par *l'ensemble des processus, réglementations, lois, et institutions influant la manière dont l'entreprise est dirigée, administrée et contrôlée*.

S'inspirant de la gouvernance d'entreprise, la gouvernance informatique (*IT Governance*), appelée aussi gouvernance des systèmes d'information, garantit le pilotage du SI afin d'assurer l'accomplissement des objectifs de l'entreprise et d'accroître ses intérêts [Marks, 2008]. À ce titre, la gouvernance informatique fait partie intégrante de la gouvernance d'entreprise. Les responsables du système d'information sont alors concernés par la prise des décisions et la spécification des droits, des règles et des responsabilités concernant l'ensemble du SI afin d'accroître son efficacité [Afshar et al., 2007].

La problématique de la gouvernance SOA est une particularisation de la gouvernance informatique en se focalisant sur les éléments spécifiques à la mise en œuvre de l'architecture SOA

[Bennett et al., 2011]. La Figure 12 permet de positionner la gouvernance SOA par rapport à la gouvernance informatique et à la gouvernance d'entreprise.

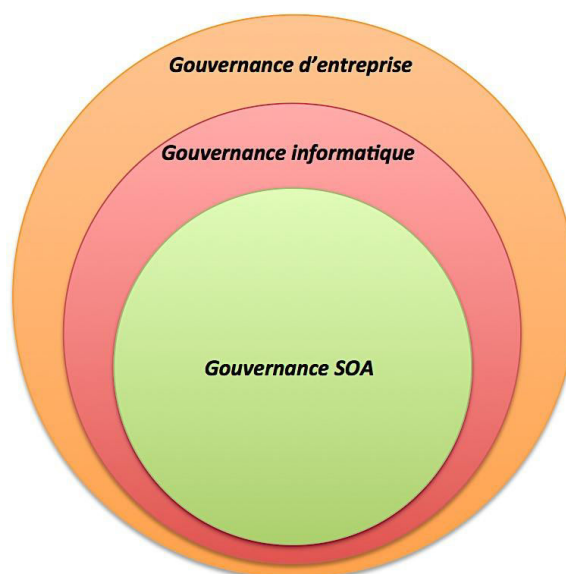


Figure 12 : Gouvernance d'entreprise, gouvernance informatique et gouvernance SOA.

La gouvernance SOA joue un rôle fondamental pour la réussite de toute architecture SOA. En effet, le manque de gouvernance peut être un sérieux obstacle et constitue la raison la plus commune de l'échec de cette architecture [Afshar et al., 2007]. Elle intervient sur tous les niveaux du cycle de vie depuis la phase de la conception jusqu'à la phase d'exécution afin de permettre aux entreprises de bénéficier pleinement de l'architecture SOA [Derler et al., 2007] et [Matsumura, 2007].

Les auteurs, dans [Papazoglou et al., 2007], définissent la gouvernance SOA comme étant un domaine de recherche majeur pour la conception et l'exécution d'une SOA. Néanmoins, elle n'est pas clairement définie dans la littérature. Nous essayons donc de s'appuyer sur quelques définitions afin de cerner les principes de la gouvernance SOA :

- Dans [Brown et al., 2006], les chercheurs d'IBM définissent la gouvernance SOA comme un sous-ensemble de la gouvernance informatique en attribuant des droits, des politiques et des mesures autour des services, des processus et du cycle de vie d'une SOA. Ceci est dans le but de répondre aux différentes préoccupations telles que la publication des services, la gestion des versions, le financement, la surveillance, la découverte, etc.
- Dans [Afshar et al., 2007], les chercheurs d'Oracle définissent la gouvernance SOA comme une interaction entre les politiques et règles (Quoi?), les responsables (Qui ?) et les processus (Comment ?) afin d'assurer la réussite de l'architecture SOA. La gouvernance SOA permet de veiller au maintien de l'alignement stratégique et de démontrer la valeur

des investissements SOA réalisés. Elle couvre tous les niveaux du cycle de vie : la conception, le développement, le déploiement, et l'exécution des services. Dans le même principe, les chercheurs de SUN [Wheaton, 2007] présentent la gouvernance SOA comme la capacité d'organiser, d'exécuter et de reconfigurer les interactions de services dans une architecture SOA.

- Dans [Marks, 2008], l'auteur regroupe plusieurs définitions de la gouvernance SOA afin de présenter la sienne : la gouvernance SOA comporte la définition, la mise en œuvre et l'exécution d'une approche SOA afin de l'aligner en permanence avec les objectifs métier de l'entreprise. L'exécution de cette stratégie se fait conformément aux directives et aux contraintes définies par un ensemble de principes et de règles.

En se basant sur les définitions précédentes, la gouvernance SOA peut être révélée, comme l'illustre la Figure 13, dans la trilogie suivante :

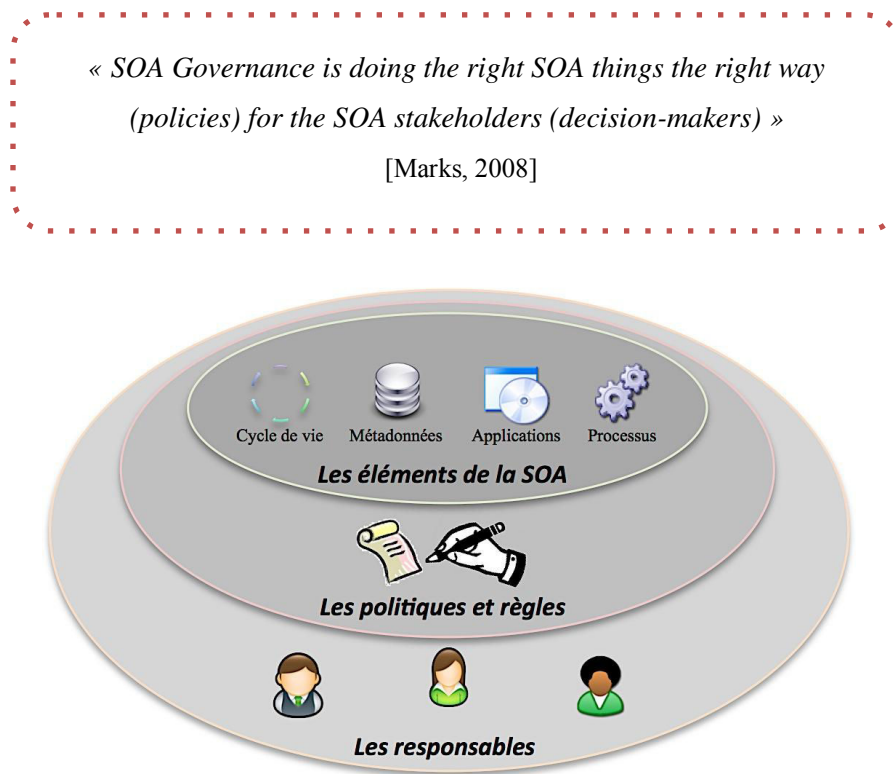


Figure 13 : Gouvernance SOA selon la trilogie de Marks [Marks, 2008].

La gouvernance SOA, alors, est la définition, la mise en œuvre, et le contrôle d'un modèle de décisions pour les responsables SOA. Ce modèle s'assure du bon suivi d'une stratégie SOA appropriée et alignée avec les objectifs métier de l'entreprise, et que l'exécution de cette stratégie est conforme aux directives et contraintes définies par un ensemble de politiques et règles pour chacun des éléments de la SOA.

II.3.2. Les cadres de gouvernance SOA

Les entreprises souhaitant migrer leur SI vers une approche SOA, ont besoin de définir et de mettre en place leur propre cadre de gouvernance SOA qui traduit les exigences métier et les transpose dans un ensemble de règles et de normes guidant l'évolution de la SOA. Cette section propose une étude sur des approches et cadres de gouvernance SOA.

Dans [Derler et al., 2007], les auteurs proposent un modèle générique et deux outils pour la gouvernance. Les services sont décrits à partir de leur cycle de vie tout en prenant en compte à chaque étape les rôles des acteurs. Les auteurs identifient trois rôles majeurs : (i) le développeur du service : celui qui se charge de l'implémentation du service, sa structure, son architecture technique, etc. (ii) le chef de produit : détermine les exigences du client, définit les spécifications métier du service, etc. et (iii) l'administrateur : attribue les rôles et les droits à chacun des acteurs. Les deux outils que les auteurs proposent sont : *Service Repository Console* et *Service Browser*. Le premier représente un annuaire contenant les fonctionnalités du service, sa description et ses dépendances. Quant au Service Browser, il est utilisé pour la recherche des services préalablement publiés dans le Service Repository. Cette approche propose une méthode standard et un modèle assez solide pour établir une gouvernance SOA. Néanmoins, les auteurs n'abordent pas la gouvernance dans le cas où les services ont besoin de collaborer ensemble.

Une *approche méthodologique* est présentée dans [Schepers et al., 2008] où les auteurs affirment que la gouvernance SOA est plus qu'un processus que met en place l'entreprise mais permet aussi de veiller en permanence à son alignement sur les objectifs stratégiques. Cette approche repose sur six phases : (i) définition d'une stratégie SOA : cette phase vise à aligner le comportement SOA sur les objectifs métier de l'entreprise, (ii) répartition des pouvoirs décisionnels et attribution des responsabilités aux principaux acteurs impliqués dans la gouvernance (les actionnaires, la direction, conseil d'administration, etc.) afin de faire prendre conscience à chacun de ses droits et de ses obligations, (iii) gestion de portefeuilles de services d'une manière à optimiser la réutilisation des services existants avant de développer des nouveaux, (iv) gestion et contrôle complets du cycle de vie de services (de la conception à la production des services), (v) intégration de procédures de suivi et vérification : cette phase permet de s'assurer de la conformité du déroulement des services par rapport aux règles et politiques préalablement définies et enfin (vi) gestion du contrat de service : elle spécifie les accords établis entre le fournisseur et celui qui va l'utiliser (le temps de réponse, le prix, les pénalités en cas de non conformité, etc.). Dans cette approche, si les auteurs présentent des directives assez claires sur la mise en place d'une gouvernance SOA, aucun détail n'est donné sur sa réalisation.

L'approche d'IBM est présentée dans [Brown et al., 2006] comme une approche de cycle de vie de gouvernance en quatre phases. Il se compose d'(i) une phase de planification au cours de laquelle les besoins sont établis, et les procédures et mécanismes existants sont évalués afin de voir comment

les améliorer, (ii) une phase de définition, pendant laquelle le cadre de la gouvernance SOA est défini en incluant les structures organisationnelles, les rôles et les règles à mettre en place durant tout le cycle de vie des services, (iii) une phase de validation durant laquelle le cadre de gouvernance retenu est mis en place au sein de l'entreprise et enfin (iv) une phase de déploiement, de suivi et de vérification du processus de la gouvernance où les politiques et les mécanismes mis en place sont contrôlés afin de vérifier leur respect par rapport aux accords engagés dans les contrats de services et d'analyser l'efficacité des mesures établies, etc. S'appuyant sur ce cadre, IBM propose une solution de gouvernance SOA appelée **IBM Web Sphere Service Registry and Repository** [WebSphere, 2010]. Cet outil de gouvernance offre, entre autres, les fonctionnalités de découverte et de publication de services, la gestion du cycle de vie, l'interfaçage avec leur infrastructure de déploiement, etc.

WSO2 Governance Registry [WSO2, 2010] est un outil de gouvernance SOA open-source de l'entreprise WSO2. Il permet l'intégration des fonctionnalités avancées de la gouvernance telles que la publication et la recherche des services, la gestion du cycle de vie, la gestion des dépendances et des relations entre les services, la gestion des utilisateurs et du contrôle d'accès aux services, la gestion du cycle de vie et la gestion de la gouvernance à la phase de l'exécution. Le registre de gouvernance WSO2 permet le stockage, la gestion des métadonnées liées aux services et la gestion des processus métier. Il offre également les fonctionnalités de gestion du cycle de vie, de gestion des politiques et règles, de gestion des relations et de gestion dépendances. Enfin, WSO2 fournit des fonctionnalités avancées comme la possibilité de gérer le registre de gouvernance à distance grâce à l'utilisation du protocole APP² (*Atom Publishing Protocol*).

CentraSiteTM [CentraSite, 2010] est une solution de gouvernance SOA co-développée par l'entreprise Software AG et Fujitsu pour soutenir l'architecture SOA. Elle se base sur des standards, garantit une qualité constante, maintient les services en ligne avec les besoins stratégiques de l'entreprise et optimise la réutilisation des ressources existantes. Ce registre de gouvernance offre d'une part un contrôle et une visibilité métier pour la SOA et d'autre part la gestion de la découverte et de la publication de services et des processus, la gestion des versions, la gestion du cycle de vie des services, la gestion des règles et des processus, et l'analyse des dépendances. Cette offre de gouvernance SOA apporte une plus grande agilité métier et une flexibilité technique dans la mesure où elle permet l'intégration avec d'autres infrastructures techniques. Toutefois, elle ne couvre pas certaines fonctionnalités essentielles pour la gouvernance telles que la gestion des contrats de services.

HP SOA Systinet [Systinet, 2009] est un cadre de gouvernance SOA implémenté par l'entreprise Hewlett-Packard (HP). Il fournit un registre permettant de cataloguer et de rechercher les données concernant les services et les processus métier. Il permet aussi la gestion des dépendances entre les

² Atom Publishing Protocol (APP) est un protocole de publication basé sur http et permet de gérer les ressources Web.

services individuels et entre les compositions de services. De même que la majorité des cadres de gouvernance SOA, Systinet couvre la fonctionnalité de la gestion du cycle de vie.

Petals Master [Master, 2009] est un outil de gouvernance SOA du consortium OW2³ développé par l'entreprise EBM WebSourcing (Petals Link). C'est un cadre open source conçu pour gérer les services, les indexer, les documenter, les retrouver, gérer leur cycle de vie, les dépendances entre eux, leurs évolutions et leurs versions. Petals Master peut être déployé en tant qu'un outil autonome ou peut être intégré dans une infrastructure de services. Toutefois, Petals Master ne couvre pas les fonctionnalités de gestion des contrats de services et des propriétés non-fonctionnelles.

Mule Galaxy [Galaxy, 2009] est une plateforme de gouvernance SOA développée par l'entreprise Mule Soft. Elle fournit un registre de gouvernance qui facilite la gestion de la SOA en proposant des fonctionnalités telles que la gestion du cycle de vie, la recherche de services, le stockage des documents décrivant les services, la gestion des changements de version, la gestion de la dépendance, la découverte de services, la gestion des artefacts et le déploiement des services. Bien que Mule Galaxy soit un produit prometteur, il reste similaire à d'autres cadres de gouvernance open source et peut être amélioré par la gestion des propriétés non-fonctionnelles et la par la réconciliation des services.

Il existe d'autres approches et cadres de gouvernance SOA sur le marché, nous pouvons citer : Cosminexus (l'outil de Hitachi, [Cosminexus, 2010]), et Governance Interoperability Framework (GIF, [GIF, 2010]). Ils fournissent généralement les fonctionnalités de découverte de services, de publication de services et de gestion du cycle de vie.

II.3.3. Synthèse sur les cadres existants

Plusieurs cadres de gouvernance SOA existent et particulièrement dans la communauté industrielle et offrent majoritairement les fonctionnalités de gouvernance, telles que : la publication de services (enregistrement de la description du service dans le registre de gouvernance), la découverte de services (la recherche d'un service dans le registre parmi ceux publiés), la gestion des contrats de services (le contrat est un document réalisé par le fournisseur et l'utilisateur et spécifie les accords établis entre eux pour l'utilisation du service), la gestion de cycle de vie (la gestion du service durant toutes les phases de son cycle de vie), la gestion des utilisateurs et des droits d'accès (la gestion du qui fait quoi) et le déploiement (la synchronisation du registre de gouvernance directement avec une infrastructure de déploiement de services). Dans le Tableau 1, nous présentons une analyse comparative de ces cadres. Chaque colonne représente un cadre et chaque ligne une fonctionnalité.

³ OW2 est une association internationale à but non lucratif dédiée au développement d'intégrés Open Source de qualité industrielle. Elle regroupe des entreprises et des organismes de recherche de premier plan tels que l'INRIA, Bull, France Télécom, Linagora, Thales Group ou Red Hat.

Rappelons notre objectif initial qui est la rationalisation de la gestion des services et de leurs propriétés non-fonctionnelles. Ceci dans le but d'assurer et d'optimiser la réconciliation non-fonctionnelle entre les activités métier et les services techniques à partir d'un processus modélisé graphiquement. Les deux dernières lignes montrent qu'aucun cadre, parmi ceux étudiés, ne couvre ni la gestion des propriétés non-fonctionnelles (d'une façon assez complète et en utilisant un standard) ni la réconciliation non-fonctionnelle. La dernière colonne présente, notre vision d'un cadre de gouvernance, appelé EasierGov-NFR permettant à la fois de gérer :

- la publication de services ;
- la découverte de services ;
- la gestion des contrats de services ;
- la synchronisation avec un bus de services pour assurer le déploiement ;
- la gestion des propriétés non-fonctionnelles en se basant sur le standard WSQF ;
- la liaison avec un outil de modélisation graphique de processus métier collaboratifs (comprenant l'annotation non-fonctionnelle) afin d'assurer la réconciliation métier / technique.

Tableau 1 : Synthèse sur les cadres de gouvernance SOA.

	Web Sphere Service Registry and Repository	WO2 Governance Registry	CentraSite	Systinet	Petals Master	Mule Galaxy	EasierGov-NFR
Publication de services	X	X	X	X	X	X	X
Découverte de services	X	X	X	X	X	X	X
Gestion des versions			X	X	X	X	
Gestion des dépendances		X	X				
Gestion du cycle de vie	X	X	X	X	X	X	
Gestion des contrats de services	X						X
Gestion des utilisateurs et des droits d'accès	X					X	
Déploiement	X				X		X
Gestion des propriétés non fonctionnelles (en se basant sur un standard)							X
Réconciliation non-fonctionnelle activités métier / services							X

II.4. Gestion des processus métier (BPM)

Dans cette section, nous présentons les différents concepts de base liés aux processus métier, à leur gestion ainsi qu'à leur modélisation graphique.

II.4.1. Les processus et les processus métier

Processus : le terme processus peut se traduire comme « avancer » et « améliorer ». Ce concept est utilisé dans plusieurs domaines. Dans les domaines industriels, Morley dans [Morley et al., 2005] définit un processus comme étant « *l'organisation d'un ensemble finalisé d'activités effectuées par des acteurs et mettant en jeu des entités* ». Théroude [Théroude, 2002] enrichit cette définition en mettant en évidence, d'une part, un aspect objectif lié à une finalité industrielle et, d'autre part, un aspect structurel des processus :

« *Un processus est 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.* »

[Théroude, 2002]

L'approche processus est recommandée aux entreprises par la norme ISO9001 (*International Organization for Standardization*) afin de réaliser une démarche de qualité [Brandenburg et al., 2006]. Elle consiste donc à décrire et à concevoir d'une façon plus pragmatique l'organisation de l'entreprise dans le but de satisfaire les besoins de ses clients. Il existe plusieurs types de processus, mais généralement ils sont classés dans trois grandes familles [Cattan, 2000], [Debauch et al., 2004] :

- Processus de pilotage : appelés aussi processus de management ou processus décisionnels et ont pour fonction d'organiser les objectifs de l'entreprise ;
- Processus opérationnels : ont pour but de contribuer directement à la réalisation du produit ou de la prestation : de la phase de détection du besoin client jusqu'à sa satisfaction ;
- Processus de support : appelés aussi processus de soutien et ont pour mission de fournir les périphériques et les ressources (humaines, matérielles, informationnelles, etc.) nécessaires à tous les processus pour accomplir les objectifs métier.

Processus métier : les processus métier constituent de plus en plus la colonne vertébrale des SIs modernes. Ils sont au cœur de la réorganisation des entreprises pour une meilleure intégration de

leurs départements et de leurs personnels [Godart et al., 2009]. Un processus métier consiste à orchestrer des activités qui visent, grâce à leur déroulement, à atteindre l'objectif souhaité. Les activités sont des éléments d'action atomiques qui transforment une ressource d'entrée en une ressource de sortie. Ces activités peuvent (i) être exécutées manuellement, partiellement ou complètement automatisées sans aucune intervention humaine, (ii) être exécutées en parallèle ou de façon séquentielle et (iii) être déclenchées par des événements ou produire des événements [Weske, 2012]. Un processus métier peut être mesuré par des contrôles de conformité mis en œuvre selon deux types de propriétés :

- Les propriétés fonctionnelles mises en place sur les entrées et les sorties du processus ;
- Les propriétés non-fonctionnelles qui sont relatives à la performance du processus..

Un processus métier peut être affecté entièrement à une seule organisation ou peut interagir avec d'autres processus métier appartenant à diverses organisations afin de répondre à un objectif plus global. Dans ce cas, il est appelé : *processus métier collaboratif* (ou aussi processus métier inter-organisationnel).

II.4.2. Gestion des processus métier

La gestion des processus métier (*Business Process Management - BPM*) est une discipline collaborative qui consiste à gérer les processus métier d'une entreprise. La gestion des processus métier est définie par le Pr. Van der Aalst comme suit :

« La gestion des processus métier (BPM) inclut les méthodes, les techniques et les outils pour modéliser, exécuter, contrôler et analyser les processus opérationnels métier en s'appuyant sur des acteurs tels que les personnes, les organisations, les applications, les documents et toute autre source d'information. » [Van der Aalst et al., 2003]

Les deux objectifs majeurs de cette approche sont : (i) avoir une meilleure vue globale de l'ensemble des processus métier de l'entreprise et optimiser les interactions entre ses différentes activités voire les automatiser (dans le cas où ceci est possible) et (ii) gérer les processus métier en s'intéressant aux différentes étapes du cycle de vie des processus depuis la modélisation jusqu'à l'établissement du produit final.

Le BPM tend à unifier toutes les activités de l'entreprise. Crusson, dans [Crusson, 2003], définit trois niveaux pour modéliser les processus métier : le niveau métier, le niveau fonctionnel, et le niveau technique. Ceci permet de découpler la modélisation métier et fonctionnelle de la modélisation

technique d'un processus afin de limiter l'impact de toute modification. Toutefois, pour faire évoluer les processus il est nécessaire de suivre l'une de ces deux approches : l'approche *top-down* et l'approche *bottom-up* (cf. Figure 14) [Crusson, 2003].

- L'approche *top-down* est utilisée lorsque le processus est modifié par le niveau métier ou fonctionnel.
- L'approche *bottom-up* est utilisée lorsqu'une modification intervient dans l'implémentation des services techniques. Dans la majorité des cas, une modification à ce niveau n'a pas vraiment d'impacts sur les niveaux métier et fonctionnels.

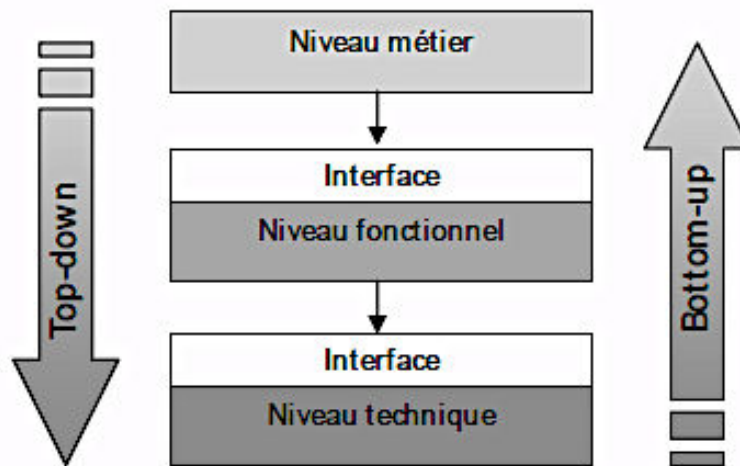


Figure 14 : Niveaux d'abstraction d'un processus BPM [Crusson, 2003].

II.4.3. Classification des processus métier

Il existe deux types de classements des processus métier : les processus abstraits et les processus exécutables (connus aussi sous le nom des processus concrets).

- Les processus abstraits constituent une représentation haut niveau des processus métier [Crusson, 2003]. La définition des processus abstraits est indépendante des aspects techniques. En d'autres termes, nous ne précisons pas quelles actions sont effectivement réalisées par les activités (e.g. Comment une demande de devis est reçue ? Quelle application est utilisée pour vérifier la disponibilité des produits ?, etc.). Le processus abstrait est la base utilisée par les équipes techniques pour créer un processus exécutable ;
- Les processus exécutables sont obtenus par la spécialisation du processus métier abstrait et se focalisent sur l'implémentation privée et propre à chacun des participants. Ils spécifient un certain nombre d'informations techniques (e.g. les services impliqués dans le processus, la transformation des données effectuées, l'intégration des utilisateurs comme

participants dans le processus, le format des messages échangés, les protocoles de transport utilisés, etc.).

II.5. BPM dans un contexte SOA

Pour rester compétitive, une organisation doit pouvoir analyser l'efficacité de ses processus métier et les améliorer sans être contrainte par le système d'information sous-jacent. Le BPM ne peut suffire à lui seul pour créer un système d'information agile. Il ne se préoccupe pas de lier les processus métier avec le système d'information (certains processus métier sont d'ailleurs intrinsèquement décorélés). C'est là qu'intervient l'architecture SOA. En faisant évoluer le système d'information d'un paradigme d'applications à un paradigme de services, l'architecture SOA offre une interface parfaite pour le BPM. Elle permet d'associer les étapes des processus métier à des appels de services, sans se soucier de l'implémentation effective de ces derniers. En effet, les processus BPM assurent l'interopérabilité au niveau métier, tandis que l'utilisation des standards et des services Web réutilisables sont préconisés par la SOA afin d'assurer l'interopérabilité au niveau technique. Cette universalité facilite la composition de plusieurs services hétérogènes permettant de répondre à un objectif commun. Dans cette section, nous présentons les approches pour la description de la composition de services ainsi que les modèles et langages pour leur représentation graphique au niveau métier.

II.5.1. Composition de services Web

La composition de services Web est l'un des défis majeurs du paradigme émergent SOA. Elle permet de créer de nouvelles fonctionnalités en faisant collaborer les fonctionnalités des services Web existants afin de répondre à une finalité commune et d'apporter une valeur ajoutée. La modélisation d'une telle technique dans un flot de processus métier peut se faire principalement selon deux approches : (i) une approche statique et centralisée : l'*orchestration*, ou (ii) une approche dynamique et complètement décentralisée : la *chorégraphie*.

Orchestration : l'orchestration de services Web définit un schéma de services abstraits qu'un moteur d'orchestration (appelé aussi moteur de composition) instancie et exécute en fonction des services concrets disponibles. Ce moteur d'orchestration coordonne et contrôle l'exécution des différentes opérations sur les services participants. Par conséquent, l'orchestration est centralisée et définit d'une façon explicite les opérations, l'ordre d'invocation des services Web, ainsi que les interactions [Juric et al., 2006]. Ces interactions peuvent être inter-applications et/ou inter-organisations, et peuvent résulter en un modèle d'exécution transactionnel de longue durée. BPEL (*Business Process Execution Language*) [OASIS, 2007] est reconnu comme un langage standard de l'orchestration de services.

Chorégraphie : contrairement à l'orchestration, la chorégraphie ne dépend pas d'un orchestrateur central. Elle décrit la collaboration entre une collection de services afin d'atteindre un objectif commun. La chorégraphie est basée sur la collaboration et est principalement utilisée pour l'échange de messages entre plusieurs services (potentiellement appartenant à des partenaires hétérogènes) au sein des processus métier publics [Peltz, 2003]. Tous les services Web mêlés dans la chorégraphie, doivent connaître le processus métier, les opérations à exécuter, les messages à échanger, ainsi que le moment de diffusion des messages. WS-CDL (*Web Services Choreography Description Language*) [W3C, 2005] est l'un des langages recommandés pour la description de chorégraphie de services.

II.5.2. Modélisation graphique des processus métier orientés SOA

Une modélisation précise du processus est primordiale car c'est d'elle que dépend tout le déroulement du processus tel qu'il est voulu par l'auteur du modèle. Il existe de nombreux types de modélisation des processus métier mais dans ces travaux nous nous intéressons particulièrement à la modélisation graphique des processus métier. Dans ce qui suit, nous présentons quelques modèles et langages de ce type de modélisation.

Les réseaux de Petri (*Petri nets* - PN) ont été présentés par Carl Adam Petri [Petri, 1962]. Ils représentent une notation graphique plus formelle et plus abstraite que toutes les autres notations [Murata, 1989]. Elle est définie sous la forme d'un flux d'informations qui est décrit, analysé, et contrôlé dans les systèmes où les activités sont asynchrones et concurrentielles, distribuées, parallèles, non déterministes et/ou stochastiques. Cette notation graphique permet également de mettre en place des équations d'état, des équations algébriques et d'autres modèles mathématiques qui gèrent le comportement d'un système. L'exemple de la Figure 15 illustre une modélisation graphique d'un processus à l'aide des réseaux de Petri [De Backer et al., 2005]. La transition A renvoie le déclenchement en parallèle des transitions B ou NOT B, et C ou NOT C (selon la vérification de l'équation). La fin du déroulement de ces transitions constitue le début de la transition D.

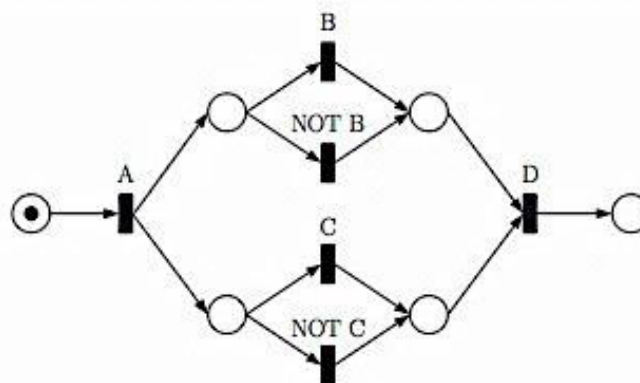


Figure 15 : Modélisation de processus avec les réseaux de Petri [De Backer et al., 2005].

Toutefois, la modélisation de processus métier avec les réseaux de Petri n'inclut pas une représentation organisationnelle (les acteurs) ni une représentation de la prise en considération des événements qui peuvent être déclenchés au cours de l'exécution du processus.

IDEFØ - *Icam (Integrated computer-aided manufacturing) DEFinition Ø* connu aussi sous le nom commercial SADT (*Structured Analysis and Design Technique*) est une méthode orientée flux et conçue pour modéliser graphiquement les processus métier. Cette méthode a connu une diffusion importante dans les milieux industriels et est centrée sur la représentation des activités d'une organisation ou d'un système [Greeff et al., 2004]. Elle consiste en la décomposition du processus d'une façon hiérarchique. La représentation du diagramme global, appelé aussi diagramme de contexte, se traduit par une boîte (l'activité) et les quatre flèches situées à ses points cardinaux : les flux entrants (à gauche), les flux sortants (à droite), les contraintes de contrôle (en haut) et les mécanismes (en bas). Ce diagramme peut être décomposé en plusieurs diagrammes « enfants », chacun correspond aux détails d'une boîte [Bia-Figueiredo et al., 2011]. Dans la Figure 16, nous présentons un exemple de diagramme de modélisation avec IDEFØ. Le diagramme principal, nommé A-0, représente le processus global. Le diagramme du premier niveau de décomposition (A0) est composé d'une suite d'activités. Seule l'activité 3 a fait l'objet d'une deuxième décomposition (d'où la mention du numéro A3 sous la boîte 3). Cette méthodologie a le mérite d'être bien structurée mais elle fournit des graphiques à l'aspect un peu technique, visuellement pas assez faciles de compréhension et très clairement orientés sur les flux de données.

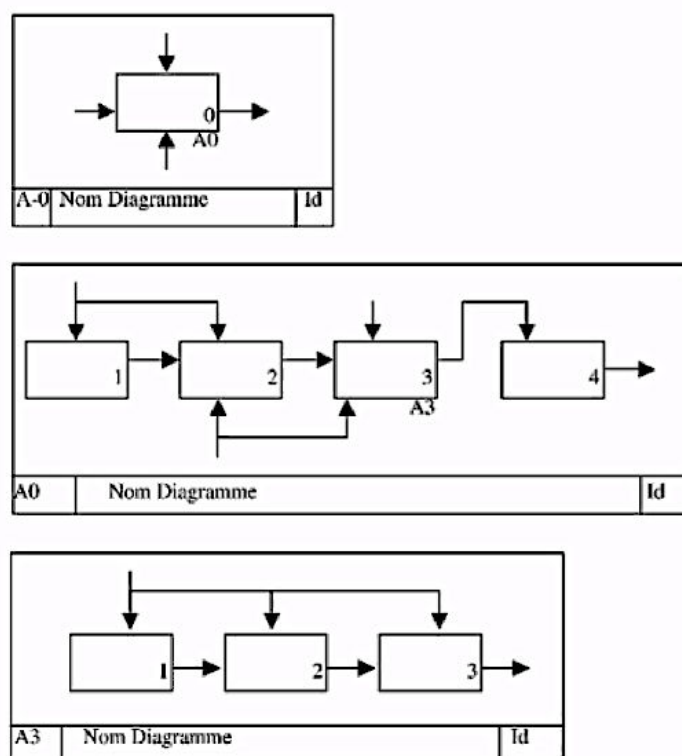


Figure 16 : Modélisation de processus avec IDEFØ [Bia-Figueiredo et al., 2011].

UML *Unified Modeling Language* [Booch et al., 1998] propose un ensemble de diagrammes spécifiques, tels que les diagrammes d'activités, les diagrammes de séquences, les diagrammes de classes, etc. Pour la modélisation des processus métier, UML propose un mécanisme d'extensibilité permettant de spécialiser les diagrammes d'activités. IBM propose un profil UML appelé « *UML2 Activity Diagram* » permettant de prendre en compte une telle extension [Koehler, et al., 2007]. La Figure 17 illustre un exemple de processus modélisé à l'aide de ce profil. Le processus commence par un flux de début (nœud gris) et un flux de fin (nœud blanc contenant une croix). L'élément qui représente une activité est appelé *Call Behavior* et est représenté par un rectangle. La *Decision* représente un état exclusif (seule l'activité qui vérifie la condition s'exécute). Les *Fork* et *Join* indiquent respectivement le début et la fin d'une exécution parallèle des activités (Subprocess 1 et Task2). Le petit rectangle gris au début du branchement des activités 3 et 4 définit le branchement inclusif (Task 3, Task 4, ou les deux qui s'exécutent).

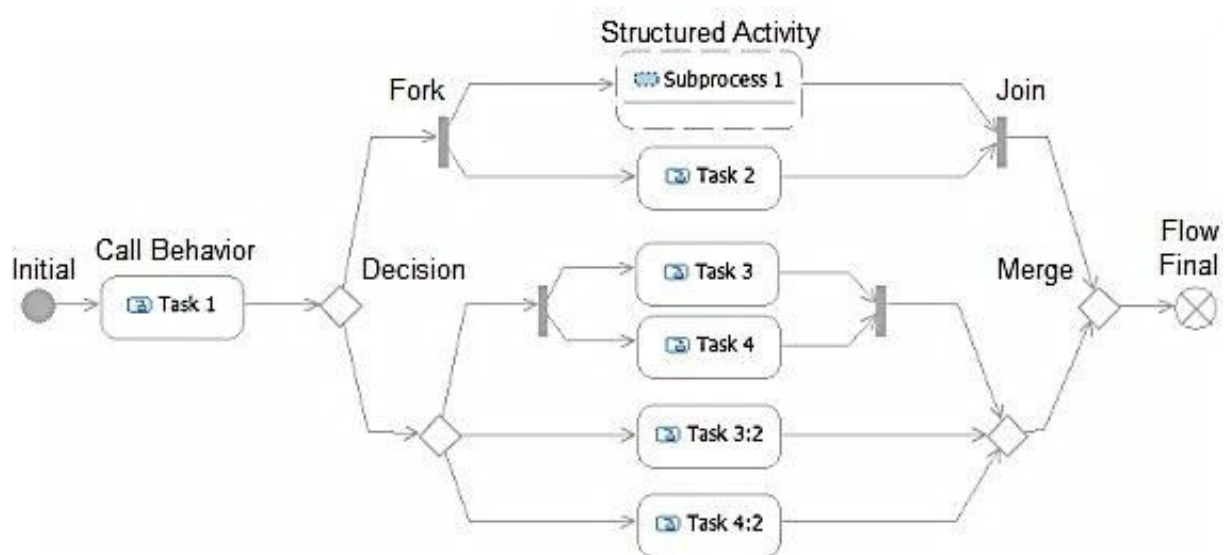


Figure 17 : Modélisation de processus avec le profil UML [Koehler, et al., 2007].

Cette approche est propriétaire à l'équipe IBM et consiste à définir un profil UML pour chaque outil. Toutefois, cela se fait en dépit de l'interopérabilité puisqu'il devient impossible dans un outil de travailler sur un processus modélisé avec un autre outil.

BPMN *Business Process Model and Notation* [Stephen, 2004] est un standard développé par le groupe BPMI (*Business Process Management Initiative*) et intégré par la suite par OMG (*Object Management Group*). Il vise à fournir une représentation graphique normalisée et complète des processus métier. La dernière version de BPMN a été publiée en Février 2011 sous le nom de BPMN 2.0 [OMG, 2011]. Cette nouvelle version est devenue le standard principal pour la modélisation des processus métier comme elle permet non seulement de modéliser une très grande variété de processus métier, mais aussi de modéliser les diagrammes qui sont à la fois faciles à comprendre et peuvent être

exécutables. Ainsi, il comble le passage entre la modélisation des processus métier et leur exécution automatique. La modélisation graphique est au cœur de la spécification BPMN 2.0 qui propose un ensemble d'éléments assez intuitif et permet de gérer les événements qui peuvent être déclenchés au cours de l'exécution du processus. BPMN 2.0 permet également de modéliser deux types d'activités : (i) les activités dites interactives (qui nécessitent une intervention humaine pour leur réalisation) et les activités automatiques (qui leur exécution n'a besoin d'aucune intervention). Dans la Figure 18, nous présentons un exemple de modélisation avec BPMN 2.0 d'un processus métier de contrôle de tweets avant leur publication. La soumission d'un tweet déclenche le début du processus, si le responsable de communication décide que le contenu du tweet est conforme aux politiques de l'entreprise (tâche humaine), il est donc publié par le système (tâche automatique), sinon des modifications sont demandées. Le premier nœud (à gauche) constitue l'événement du début du processus (déclenché pour chaque soumission de tweet), le nœud de droite constitue la fin du processus. Les rectangles avec des coins arrondis symbolisent les activités et l'indicateur en haut à gauche définit le type de l'activité : humaine (un être humain) ou automatique (un rouage). Enfin, le losange représente le branchement, dans cet exemple il s'agit d'un branchement exclusif.

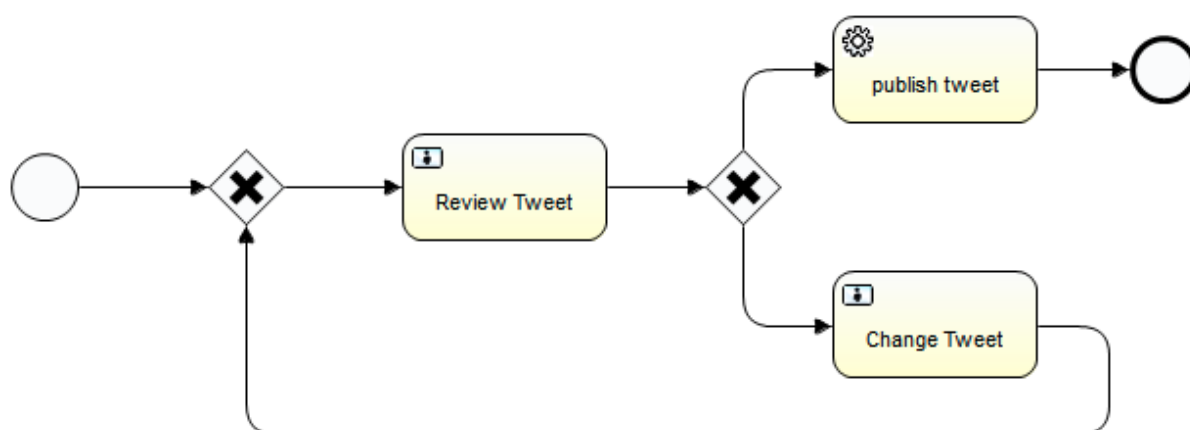


Figure 18 : Modélisation de processus avec BPMN 2.0 [Meyer, 2011].

Par ailleurs, pour assurer l'interopérabilité, chaque diagramme BPMN 2.0 correspond à une sérialisation XML unique contenant toutes les informations sémantiques liées au processus métier lui-même. Une telle sérialisation unique pour la modélisation du processus métier permet son utilisation dans d'autres outils de modélisation. BPMN 2.0 fournit également un standard appelé BPMN DI (*BPMN Diagram Interchange*) qui est basé aussi sur une sérialisation XML [Bray et al., 1997]. Ce standard est utilisé pour compléter la notation textuelle de BPMN 2.0 en ajoutant des informations graphiques. De plus, il permet d'importer des diagrammes de type BPMN 2.0 directement dans un outil de modélisation graphique s'ils respectent les règles définies dans la spécification. De plus, le standard BPMN 2.0 est compatible avec des langages basés sur XML pour l'exécution des processus

et fournit une connexion directe avec le langage standard BPEL (*Business Process Execution Language*) pour une exécution automatique des processus.

II.5.3. Synthèse sur la modélisation des processus métier orientés SOA

Les processus métier occupent une place centrale dans la mise en œuvre et l'évolution des SIs. Ils se présentent comme un outil clé à la compréhension du fonctionnement de l'entreprise d'une part, et d'autre part à l'organisation des activités et l'amélioration des interactions entre elles.

La modélisation des processus métier représente une étape essentielle à travers laquelle les concepteurs définissent les nouveaux processus métier d'une façon abstraite ou détaillée, ou améliorent des processus déjà existants.

Nous avons défini dans cette section, les langages de modélisation graphique du BPM les plus utilisés : les réseaux de Petri, IDEF0, UML et BPMN 2.0.

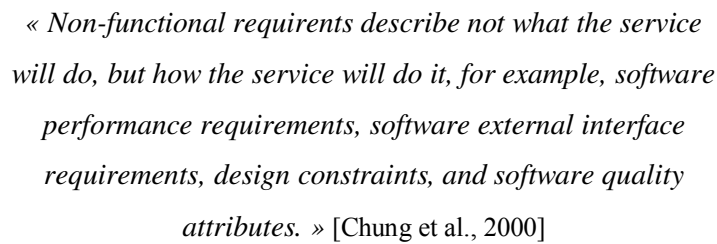
Les deux premiers prennent en compte seulement une représentation de la gestion comportementale du processus. La modélisation des processus métier est donc réduite à une simple séquence d'activités.

Les deux derniers (à savoir, UML et BPMN 2.0) représentent des langages plus avancés qui incluent une représentation informationnelle (les données) et organisationnelle (les acteurs). Cette catégorie fournit une typologie riche pour modéliser les activités, les événements, les flux, etc. L'utilisation de ces formalismes avancés lors de la modélisation des processus métier permet de couvrir les différents aspects et concepts des processus. La principale différence entre ces deux langages est que BPMN 2.0 offre la possibilité d'être lié directement à un environnement d'exécution (et ce par exemple grâce au standard BPEL). Tandis que pour UML, il est nécessaire de créer une correspondance entre la représentation graphique des processus et un langage d'exécution afin de permettre un déploiement direct des processus modélisés.

En conséquence, BPMN dans sa version la plus récente (2.0) semble être le meilleur choix pour la modélisation graphique des processus métier.

II.5.4. Annotation du BPMN avec des exigences non-fonctionnelles

L'importance des exigences non-fonctionnelles (appelées aussi contraintes) pour les processus métier a été fortement reconnue durant ces dernières années. Plusieurs définitions d'exigences non-fonctionnelles existent, mais nous retenons celle présentée par Chung :



« *Non-functional requirements describe not what the service will do, but how the service will do it, for example, software performance requirements, software external interface requirements, design constraints, and software quality attributes.* » [Chung et al., 2000]

Les exigences non-fonctionnelles (*Non-Functional Requirements* - NFR) ne décrivent pas ce que le service fait (les fonctionnalités), mais plutôt comment il les remplit. Dans la littérature, il existe de multiples travaux autour de la modélisation des exigences non-fonctionnelles dans les processus métier ([Mylopoulos et al., 1992], [Chung et al., 1995], [Bresciani et al., 2002] et [Lu et al., 2006]). Toutefois, ceux qui traitent particulièrement les processus métier de type BPMN 2.0 sont beaucoup plus limités même s'il existe plusieurs ouvrages abordant ce sujet ([Demirors et al., 2003], [Cysneiros et al., 2004], [Gorton et al., 2006] et [Hepp et al., 2007]).

Bien qu'il y ait une couverture quasi complète des caractéristiques fonctionnelles lors de la modélisation des processus métier, les exigences non-fonctionnelles ne sont généralement pas prises en considération. Cela peut conduire à la non prise en compte (voire à l'exclusion) d'informations importantes et particulièrement dans un contexte des architectures SOA où les processus sont amenés à collaborer. Les exigences non-fonctionnelles ne sont pas explicitement traitées par le processus BPMN. L'approche envisagée est plutôt d'étendre le formalisme en associant une approche ou un modèle pour annoter les activités du processus métier.

Les auteurs dans [Rodriguez et al., 2007] présentent une solution d'annotation du BPMN à l'aide d'exigences non-fonctionnelles. Ils proposent dans leur approche deux méta-modèles : (i) un méta-modèle appelé *Business Process Diagram* (BPD) qui reprend les différents éléments du processus BPMN et détaille les relations entre eux et (ii) un méta-modèle de sécurité (*Security Requirement*) qui étend le BPD en intégrant les aspects de sécurité suivants : la non-répudiation (*non-repudiation*), la détection des risques et des attaques (*attack harm detection*), l'intégrité (*integrity*), la confidentialité (*privacy*) et le contrôle d'accès (*access control*). La Figure 19 permet d'illustrer ces deux méta-modèles et montre la relation entre eux.

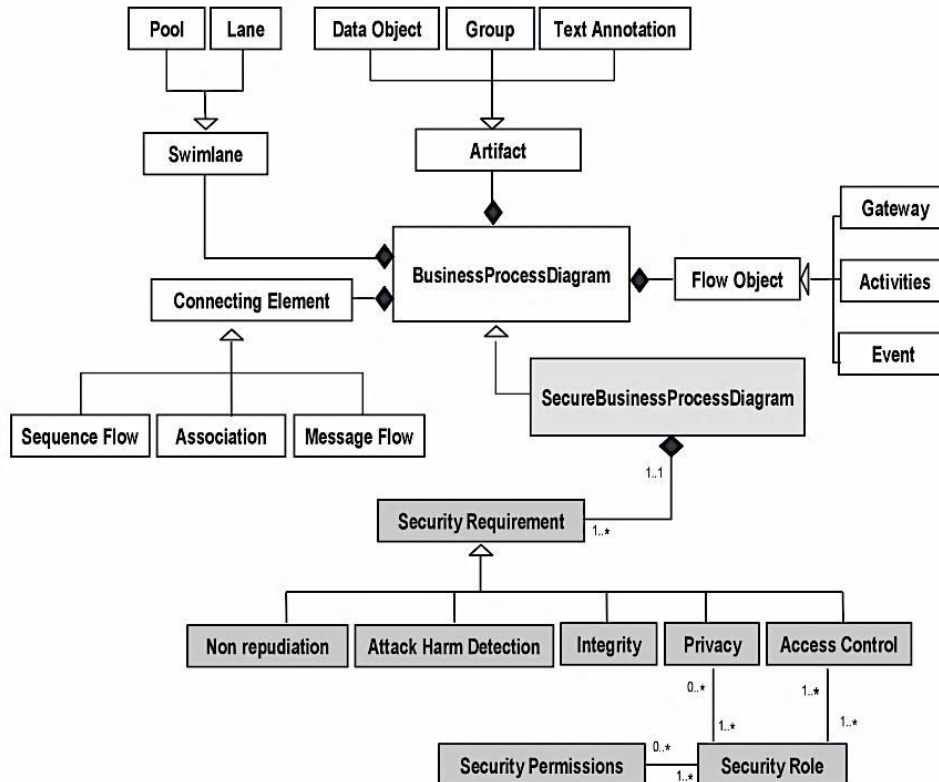


Figure 19 : Extension de BPMN : intégration des exigences de sécurité [Rodriguez et al., 2007].

Cette approche permet aux analystes métier d'exprimer les contraintes de sécurité à partir de leur perspective. Cependant, elle se limite qu'à la modélisation de la sécurité. Certes, ces propriétés sont assez importantes et particulièrement dans un contexte collaboratif. Mais, ceci n'est pas suffisant pour couvrir les autres exigences des utilisateurs telles que le temps de réponse, le prix et la disponibilité qui sont également du même ordre d'importance.

Les auteurs dans [Pavlovski et al., 2008] présentent une approche qui permet d'étendre la modélisation d'une activité d'un processus BPMN en lui associant des contraintes non-fonctionnelles. Pour réaliser ceci et comme l'illustre la Figure 20, ils proposent deux nouveaux artefacts « *operating condition* » et « *control case* » :

- état de fonctionnement (*operating condition*) : cet artefact est associé directement à une activité du processus métier. Il permet d'indiquer les exigences non-fonctionnelles qui sont associées à l'activité en question. Ces exigences peuvent être : la performance, les politiques de sécurité, la disponibilité et la fiabilité. Cette extension représente une vue de haut niveau sur les contraintes potentielles à définir ;
- cas de contrôle (*control case*) : cet artefact est en outre optionnel et est associé à l'*operating condition*. Il consiste en un ensemble d'informations détaillées concernant les

contraintes non-fonctionnelles. Il permet également de définir les contrôles à mettre en place afin d'atténuer les risques métier associés aux contraintes non-fonctionnelles.

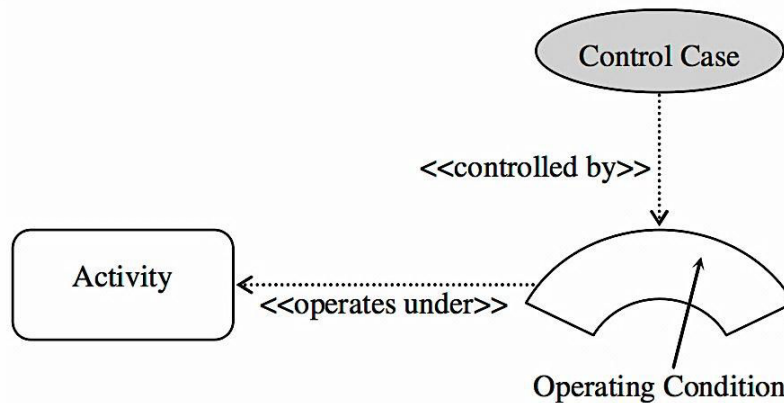


Figure 20 : Extension de BPMN : control case et operating condition [Pavlovski et al., 2008].

Toutefois, la liste des contraintes non-fonctionnelles que les auteurs proposent dans *operating condition* est de très haut niveau. Ils ne spécifient pas, par exemple, quelles propriétés non-fonctionnelles sont couvertes par les politiques de sécurité (le cryptage, l'authentification, la non répudiation, etc.). De plus, si cette approche présente une solution d'extension du processus BPMN en permettant l'annotation des activités métier par des contraintes non-fonctionnelles, elle ne présente pas de méta-modèle permettant de modéliser ces contraintes.

Les travaux d'Heinrich [Heinrich et al., 2011] optent également pour l'extension du standard BPMN en lui considérant la gestion des contraintes non-fonctionnelles. Ils présentent dans leur approche un méta-modèle intitulé BPQRM (*Business Process Quality Reference-Model*). Ce méta-modèle permet d'associer un large ensemble de propriétés non-fonctionnelles à chacun des composants du processus métier (contexte (*context*), activité (*activity*), ressource (*resource*), acteur (*actor*), informations sur les objets (*information object*), données des objets (*data object*)) et les relations entre eux. La Figure 21 présente le méta-modèle BPQRM.

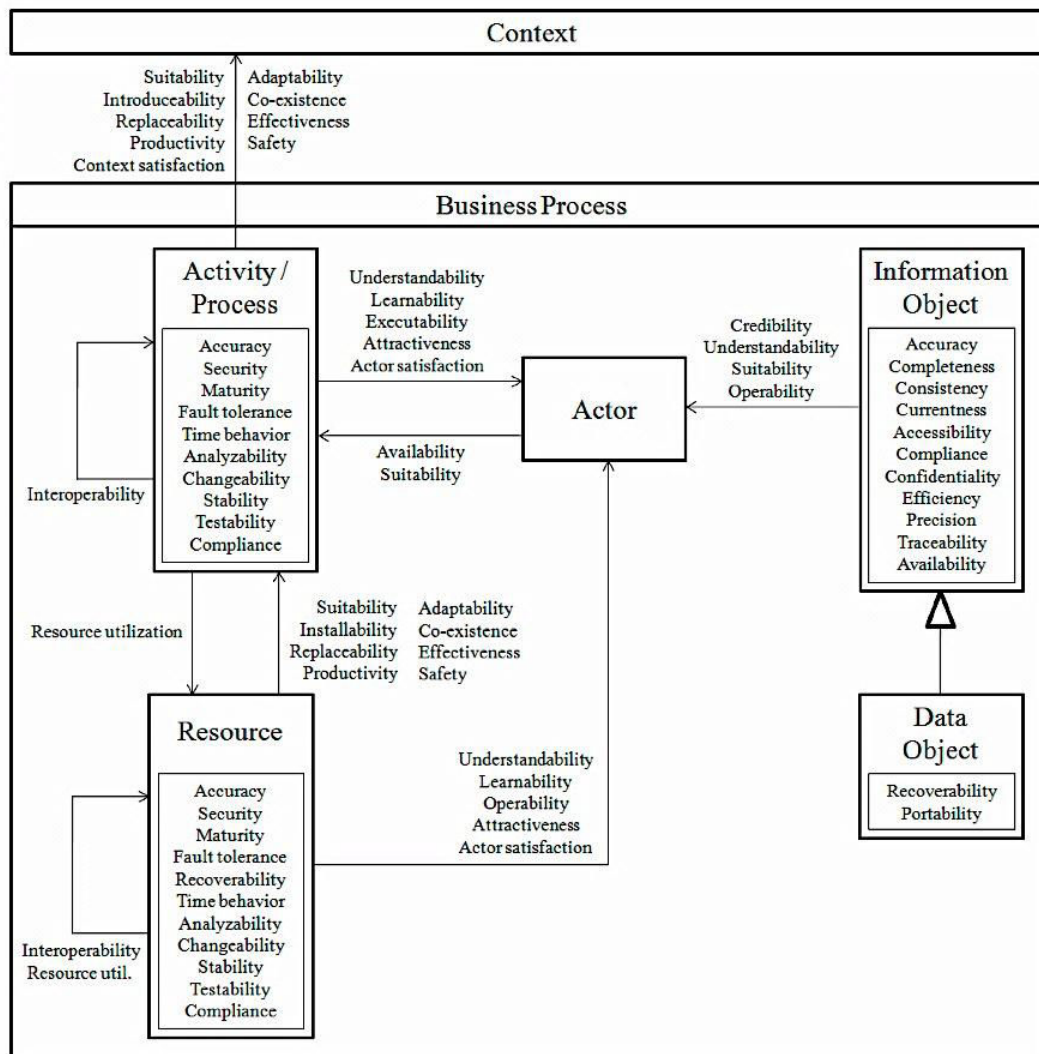


Figure 21 : Extension de BPMN : le méta-modèle BPQRM [Heinrich et al., 2011].

Bien que BPQRM identifie plusieurs propriétés non-fonctionnelles, ce méta-modèle ne représente pas un standard et ne fait pas référence également à l'utilisation des standards (entre autres pour une question de terminologie). Or, implémenter des standards est particulièrement recommandé dans le sens où les processus sont amenés à collaborer avec d'autres provenant de différentes entreprises et réalisés à partir des technologies hétérogènes.

II.5.5. Synthèse sur les annotations du BPMN à l'aide d'exigences non-fonctionnelles

La prise en compte des exigences non-fonctionnelles au sein de la modélisation des processus métier est essentielle pour satisfaire au mieux les attentes des utilisateurs. Il est donc utile d'avoir une notation graphique simple, facile et complète qui permettra de gérer cet ensemble de concepts lors de la modélisation graphique du processus. Le standard BPMN 2.0 nous offre la possibilité de gérer quelques uns parmi ces concepts (par exemple la sécurité) mais à un niveau d'abstraction qui, à notre

avis, n'est pas suffisant. En effet, l'approche BPMN 2.0 ne considère pas explicitement l'annotation d'une activité métier par des exigences non-fonctionnelles.

Les travaux de [Rodriguez et al., 2007], [Pavlovski et al., 2008], et [Heinrich et al., 2011] présentés précédemment décrivent des approches qui étendent la modélisation des processus métier BPMN en prenant en considération les exigences non-fonctionnelles. Si ces travaux varient considérablement en complexité et en exhaustivité, ils n'implémentent pas d'une manière complète les standards d'annotation dédiés. Or, ceci est essentiel pour assurer l'interopérabilité. Cette dimension prend plus d'importance lorsqu'elle est appliquée dans un contexte collaboratif où les processus sont amenés à communiquer avec d'autres provenant de différentes entreprises, pour accomplir une finalité commune globale.

II.6. Réconciliation non-fonctionnelle

Avec la prolifération des services Web, il est devenu d'une nécessité patente de différencier les services Web qui répondent aux mêmes fonctionnalités afin de satisfaire au mieux les exigences métier des utilisateurs. Dans cette section, nous présentons quelques approches de la littérature permettant la réconciliation non-fonctionnelle entre les activités métier et les services. Par la suite, nous présentons une synthèse sur ces approches.

II.6.1. Etude des approches existantes

Dans le cadre des architectures SOA, la prise en considération des contraintes non-fonctionnelles devient particulièrement cruciale. En effet, ces contraintes constituent un critère décisif dans le choix de services parmi des offres de plus en plus compétitives et ayant des fonctionnalités quasi similaires. Il existe plusieurs approches dans la littérature qui prennent en compte les exigences non-fonctionnelles lors de la découverte et de la sélection des services.

Guo et al., dans [Guo et al., 2011], présentent une méthode pour la sélection des services selon les propriétés fonctionnelles et les propriétés de qualité de service. Leur approche pour la partie non-fonctionnelle consiste à choisir le meilleur service parmi les services candidats (qui résultent de la réconciliation fonctionnelle et sémantique). Ils proposent un algorithme de sélection en 3 étapes :

- la première consiste à quantifier les attributs de qualité de service selon leurs types : valeurs numériques, valeurs textuelles, valeurs booléennes, intervalles, etc. ;
- la deuxième étape consiste à normaliser les valeurs de chaque service candidat selon le type de la valeur de la propriété non-fonctionnelle. Les auteurs proposent une normalisation par rapport à la valeur maximale (ou à la valeur minimale selon le sens de variation de la variable) de chacune des propriétés de l'ensemble des services candidats ;

- la troisième et dernière étape consiste à calculer le degré pondéré de la QoS pour chacun des services candidats afin de les classer selon un ordre décroissant.

Les auteurs, dans la méthode qu'ils proposent, ont traité la majorité des types des propriétés non-fonctionnelles. Cependant, ils ne prennent pas du tout en considération les valeurs des propriétés non-fonctionnelles souhaitées par l'utilisateur. Leur réconciliation non-fonctionnelle consiste, en effet, à un filtre destiné à classer les services candidats entre eux selon seulement les valeurs de poids attribuées par l'utilisateur. En aucun cas les valeurs cibles pour les critères non-fonctionnels ne sont prises en compte.

Dans [Badr et al., 2008], les auteurs présentent une approche de sélection des services Web basée sur les diverses exigences non-fonctionnelles de l'utilisateur. Les préférences de l'utilisateur sont considérées en tant qu'une entrée supplémentaire vers le système de sélection de services. Ce système, à son tour, classe les services disponibles sur la base des préférences de l'utilisateur. Cette approche se base sur le calcul de la somme pondérée qui modifie les pondérations entre les fonctions objectives pour obtenir le front de Pareto [Kim et al., 2005]. Ils considèrent que le poids lui-même reflète la préférence de l'utilisateur. Si l'utilisateur estime que toutes les propriétés sont importantes, alors les pondérations seront réparties d'une manière égale. Ou bien, si l'utilisateur considère que certaines propriétés non-fonctionnelles sont plus importantes que d'autres alors, les poids forts sont attribués à ces propriétés et les autres auront le même poids. Leur approche est simple et facile à utiliser. Toutefois, elle se limite à la sélection unitaire de services (un service Web pour chaque tâche) et n'aborde pas la sélection de composition de services. Cette dernière est essentielle particulièrement quand il n'existe aucun service disponible qui répond aux besoins de l'activité métier. De plus, leur approche utilise le registre de services UDDI, or ce dernier, comme nous l'avons vu précédemment, est très limité pour la gestion des propriétés non-fonctionnelles qui nécessitent d'être gérées durant tout le cycle de vie des services pour une meilleure découverte.

Ran, dans [Ran, 2003], présente également un modèle de découverte de services Web dans lequel les propriétés fonctionnelles sont traitées. L'auteur propose une extension du modèle de base (fournisseur, registre et consommateur) en rajoutant une autre entité appelée *QoS Certifier*. Cette entité vérifie les propriétés de qualité de service fournies par le fournisseur du service avant que le service ne soit publié dans le registre UDDI. Il propose aussi une extension de la structure de données de ce registre afin qu'il prenne en considération les propriétés non-fonctionnelles telles que la disponibilité, la fiabilité, la performance. Une approche similaire est présentée dans [Deora et al., 2003] et se base sur les attentes des utilisateurs. Elle recueille les attentes et les évaluations des utilisateurs d'un service. Par la suite, la qualité de service est calculée uniquement au moment où une demande de service est faite et seulement en utilisant les évaluations qui ont des attentes similaires.

II.6.2. Synthèse sur la réconciliation non-fonctionnelle

De nos jours, les organisations mettent de plus en plus en place des plateformes d'automatisation des processus impliquant un traitement collaboratif des services pour répondre à un besoin commun global. Toutefois, avec l'augmentation du nombre de services Web offrant les mêmes fonctionnalités, le problème de la découverte et de la sélection des services est devenu crucial. Les propriétés non-fonctionnelles (telles que le temps de réponse, la disponibilité, la sécurité, etc.) jouent un rôle essentiel pour différencier les services, ayant les mêmes propriétés fonctionnelles, entre eux.

Nous venons d'étudier quelques approches permettant de filtrer et de classer les services en se basant sur les propriétés non-fonctionnelles. Ces approches sont diverses et même si elles permettent la sélection de services, elles se basent sur le registre de découverte de base UDDI. Cependant, comme nous l'avons vu précédemment (cf. section II.2.2), ce registre est limité et n'assure pas une gestion globale des services et particulièrement la gestion de leurs propriétés non-fonctionnelles. Nous présentons dans le Tableau 2 une synthèse des approches étudiées. La dernière colonne expose notre proposition pour la réconciliation non-fonctionnelle.

Tableau 2 : Synthèse sur les approches de réconciliation non-fonctionnelle.

	[Guangjun et al., 2011]	[Badr et al., 2008]	[Ran, 2003]	Proposition
Propriétés non-fonctionnelles des services	QoS top Ontology	NFP Categorization Ontology	QoS Category Model	standard WSQF
Sélection de services	X	X	X	X
Gestions des poids	X	X		X
Extension d' un outil de gouvernance	UDDI	UDDI	UDDI	EasierGov-NFR
Sélection de compositions de services				X
Liaison avec outil de modélisation de processus pour la réconciliation métier / technique				X

II.7. Conclusion

Dans ce chapitre, nous avons introduit les architectures SOA et leur rôle pour améliorer le système d'information de l'entreprise en offrant un couplage lâche et une meilleure agilité. Nous avons aussi présenté un état de l'art sur la modélisation des propriétés non-fonctionnelles des services.

Par la suite, nous avons présenté la gestion des processus métier (BPM) collaboratifs en général avant de détailler leur symbiose avec la SOA. Par ailleurs, nous avons montré les différentes approches de la composition de services (à savoir l'orchestration et la chorégraphie) et comment les propriétés non-fonctionnelles jouent un rôle important dans ces processus (puisque'elles constituent un élément clé pour la popularité du service). Nous nous sommes intéressés aussi à étudier les différentes approches pour modéliser graphiquement les processus métier lors de la composition et pour les annoter avec des exigences non-fonctionnelles. Dans ce contexte, nous avons, d'une part, montré aussi l'importance de la gouvernance SOA qui représente un registre de services avancé pour une meilleure découverte de services et d'autre part, étudié quelques approches pour la réconciliation et sélection de services selon les contraintes non-fonctionnelles.

Dans les chapitres suivants, nous présentons nos contributions pour l'annotation graphique des activités du processus BPM dans un contexte SOA à l'aide d'exigences non-fonctionnelles et nous détaillons notre approche d'un cadre de gouvernance SOA qui permette la gestion des services et de leurs propriétés non-fonctionnelles pour pouvoir enfin réaliser la réconciliation non-fonctionnelle de services pour les activités annotées de notre processus.

Chapitre III

Propriétés non-fonctionnelles : annotation des processus et gouvernance des services

« Chaque individu apporte au monde sa contribution unique. »

Jack Kornfield

III.1. INTRODUCTION	58
III.2. PETALS BPM-NFR : PLATEFORME DE MODELISATION ET D'ANNOTATION NON-FONCTIONNELLE DES PROCESSUS.....	60
III.2.1. Architecture générale de Petals BPM-NFR	60
III.2.2. Modélisation des processus métier	61
III.2.3. Annotation non-fonctionnelle des processus BPMN 2.0	62
III.3. EASIERGOV-NFR : GOUVERNANCE SOA POUR LA GESTION DES SERVICES ET DES PROPRIETES NON-FONCTIONNELLES	65
III.3.1. Découverte SOA.....	66
III.3.2. Gestion des services	67
III.3.3. Gestion des propriétés non-fonctionnelles	68
III.3.3.1. Annotation non-fonctionnelle des services	68
III.3.3.2. Modèle de contrat de services Web (<i>WS-Agreement template</i>).....	70
III.3.3.3. Classification des propriétés non-fonctionnelles	71
III.3.3.4. Gestion des propriétés non-fonctionnelles dans EasierGov-NFR	74
III.4. CONCLUSION	76

III.1. Introduction

Dans la littérature, bien que le terme « propriétés non-fonctionnelles » soit utilisé depuis plus de 25 ans et que la majorité ait souligné l'importance de l'utilisation de ces propriétés, il n'existe toujours pas un consensus pour les définir, les obtenir et les valider [Glinz, 2007]. Ce manque critique pour la réussite des processus et applications de l'entreprise est peut être dû à plusieurs facteurs ([Rosa et al., 2002] et [Khaled et al., 2005]) :

- les propriétés non-fonctionnelles sont généralement abstraites et la plupart du temps elles sont décrites d'une façon informelle ;
- souvent, il n'existe pas une séparation claire entre les aspects fonctionnels et non-fonctionnels d'un service ;
- les propriétés non-fonctionnelles sont souvent en conflit entre elles et en concurrence les unes avec les autres (par exemple la disponibilité et la performance) ;
- la modélisation des propriétés non-fonctionnelles est souvent complexe.

Afin de combler ce manque et d'atteindre les besoins exprimés dans le chapitre I, nous avons présenté dans le chapitre II des solutions existantes pour la description, la modélisation et la gestion des propriétés non-fonctionnelles (pour les processus métier et pour les services). Nous nous sommes également intéressés à étudier les travaux existants autour : (i) de la gouvernance SOA pour une meilleure gestion et rationalisation des services, (ii) de la réconciliation et la sélection de services selon les propriétés non-fonctionnelles. Cependant, ces travaux présentaient certaines limites (vues dans le chapitre II) qui nous sont essentielles. Ces limites concernent les trois thèmes qui constituent notre étude :

- modélisation des propriétés non-fonctionnelles et annotation des processus métier : il est nécessaire, lors de la phase de modélisation de processus, de permettre aux utilisateurs d'annoter facilement les activités par un ensemble de propriétés non-fonctionnelles. Cet ensemble de propriétés doit être assez large pour qu'ils puissent exprimer leurs besoins métier ;
- rationalisation de la gestion des propriétés non-fonctionnelles par la gouvernance SOA : il est indispensable de gérer l'ensemble des propriétés non-fonctionnelles depuis la phase de la publication des descriptions de services. La gouvernance SOA adresse ce besoin en proposant des concepts visant la modélisation des descriptions de services et la prise en considération de leurs propriétés non-fonctionnelles ;

- réconciliation entre les besoins métier et les services techniques : les approches de réconciliation existantes ne sont pas adaptées dans le cas d'une composition de services (1-N : à une activité métier, nous associons une composition de n services).

La Figure 22 permet de résumer tous ces éléments en détaillant nos besoins, en présentant l'existant et en analysant les limites et manques de ce dernier.

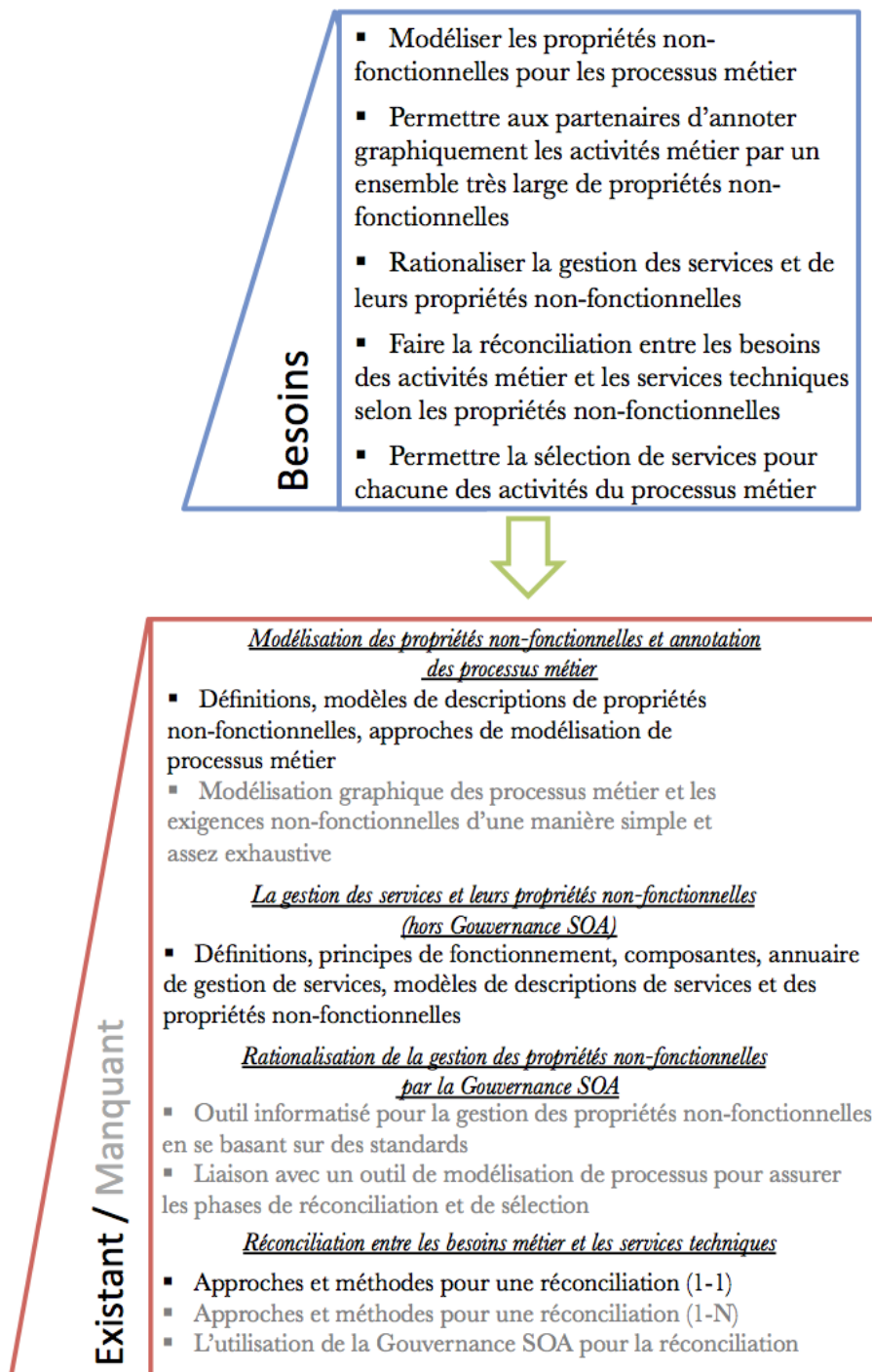


Figure 22 : Analyse croisée : besoins / existant et manquant.

Dans le présent chapitre, comme l'illustre la Figure 23, nous nous intéressons aux deux premiers thèmes de notre problématique de recherche. Nous proposons, dans la Section III.2, un enrichissement de la modélisation graphique des processus métier collaboratifs BPMN 2.0 par l'annotation non-fonctionnelles des activités. Dans la section III.3, nous présentons un cadre permettant de gérer les services et les propriétés non-fonctionnelles au sein du registre de gouvernance SOA. Ce système se base sur des standards afin de garantir principalement l'interopérabilité (aspect essentiel dans un cadre collaboratif).

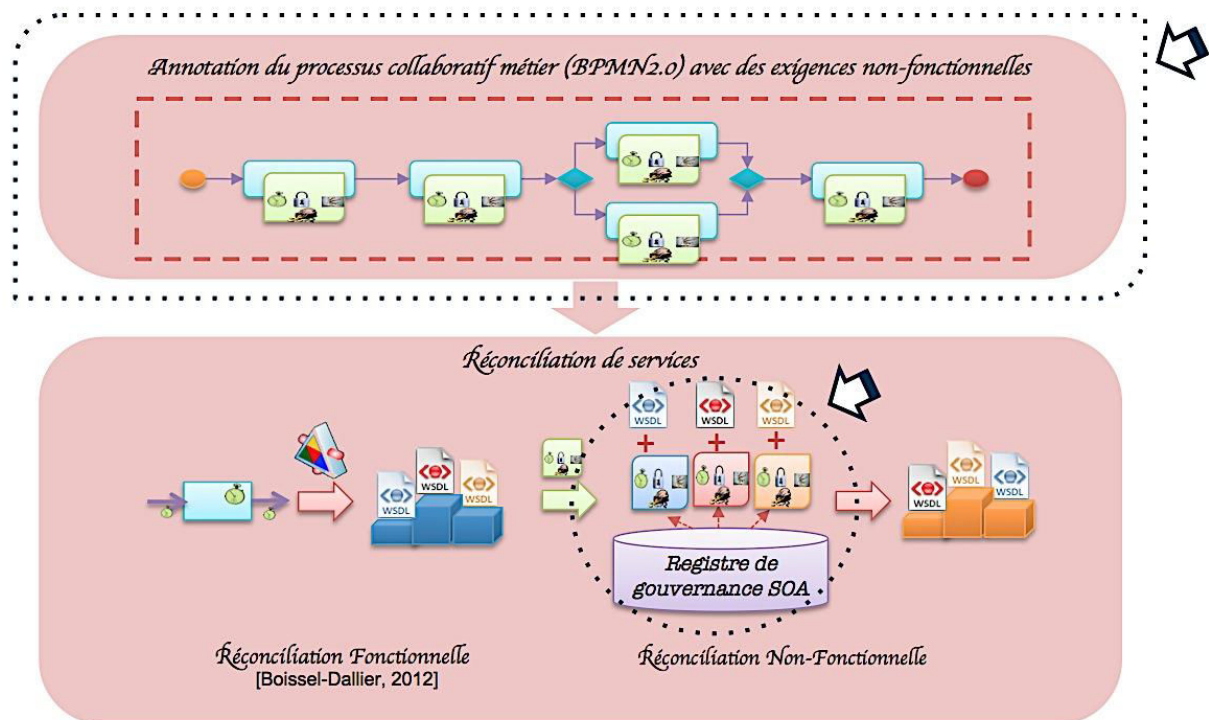


Figure 23 : Modélisation, annotation non-fonctionnelle et gouvernance SOA.

III.2. Petals BPM-NFR : plateforme de modélisation et d'annotation non-fonctionnelle des processus

Dans cette section, nous présentons notre prototype de modélisation et d'annotation graphique des activités métier nommé *Petals BPM-NFR*.

III.2.1. Architecture générale de Petals BPM-NFR

Petals BPM-NFR (Petals Business Process Management - Non-Functional Requirements) est une plateforme qui permet de modéliser des processus métier suivant le standard BPMN 2.0. Cette

plateforme enrichit la modélisation graphique habituelle des processus (établie par Petals BPM) par la possibilité d'annoter les activités métier à l'aide d'exigences non-fonctionnelles. Dans la Figure 24, nous présentons l'architecture générale de la plateforme Petals BPM-NFR.

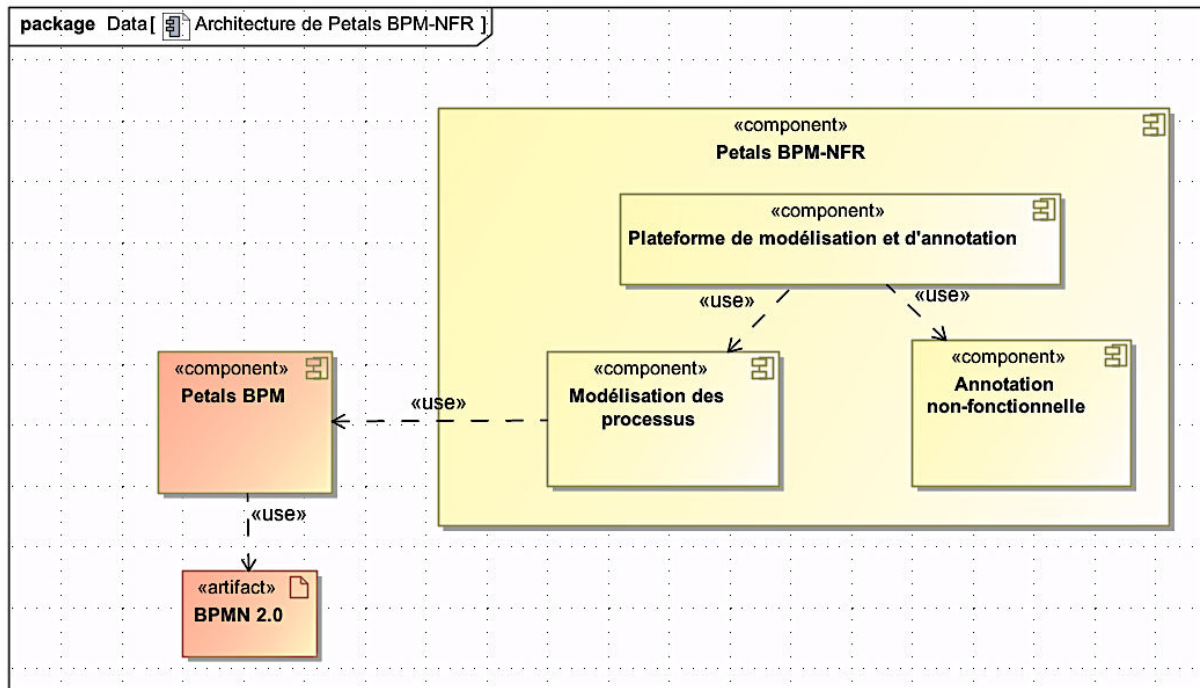


Figure 24 : Architecture générale de Petals BPM-NFR.

Deux composants principaux constituent cette plateforme : (i) «Modélisation des processus » qui permet de modéliser les processus métier collaboratifs et se base sur l'outil de modélisation graphique existant Petals BPM et (ii) « Annotation non-fonctionnelle » qui assure l'annotation non-fonctionnelle des activités métier par des exigences non-fonctionnelles. Dans les sous-sections qui suivent nous détaillons ces deux composants.

III.2.2. Modélisation des processus métier

Nous souhaitons réaliser un éditeur de modélisation de processus métier collaboratifs basé sur Petals BPM et supportant la démarche d'annotation non-fonctionnelle des activités métier.

Petals BPM est un modeleur graphique, open-source, en ligne de processus BPMN 2.0⁴. Dans la Figure 25 nous représentons une capture d'écran de son interface graphique. Il est composé de trois parties : (i) la palette BPMN 2.0 (à gauche) contenant l'ensemble des éléments permettant aux utilisateurs de créer des processus BPMN 2.0, (ii) la zone de modélisation (au centre) qui permet aux utilisateurs de dessiner leurs processus grâce à la palette d'outils, et enfin (iii) la définition des

⁴ Petals BPM : <http://bpmneditor.petalslink.com/>

propriétés (en bas) permettant de définir pour chacun des éléments du processus le nom et la description fonctionnelle.

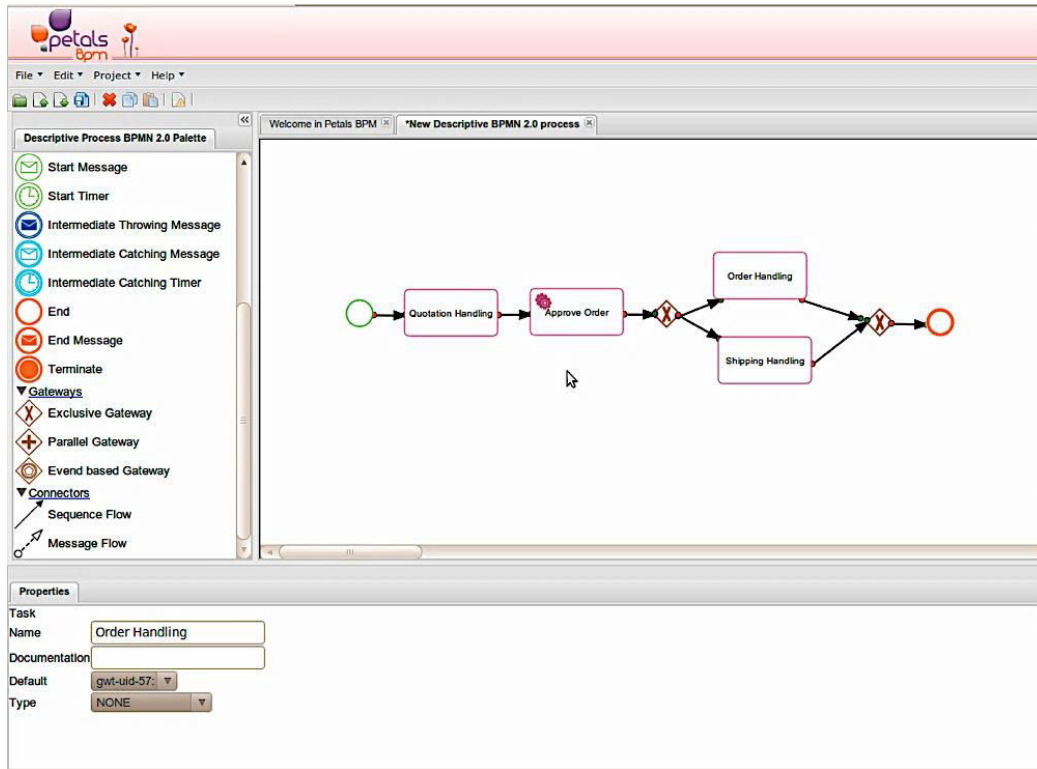


Figure 25 : Interface graphique de Petals BPM.

Comme nous venons de le voir, Petals BPM permet uniquement de décrire fonctionnellement les processus métier collaboratifs. Petals BPM – NFR enrichira la description interne de la palette graphique pour lier automatiquement la modélisation de l'activité BPMN 2.0 au composant « annotation non-fonctionnelle » que nous détaillons ci-après.

III.2.3. Annotation non-fonctionnelle des processus BPMN 2.0

Nous avons souligné dans les chapitres précédents l'importance de la prise en compte des exigences non-fonctionnelles lors de la modélisation des processus métier. Nous avons également mis en évidence l'importance de l'utilisation des standards non seulement pour assurer l'interopérabilité dans ce contexte collaboratif, mais aussi pour l'étendue et l'intégration de nos travaux open-source. Ces deux besoins ont constitué notre point de départ pour la modélisation et l'annotation non-fonctionnelle des processus BPMN 2.0. Dans le chapitre *état de l'art*, nous avons étudié quelques modèles et approches existants (cf. sous-section II.5.4 et II.5.5) et nous avons déduit qu'aucun d'eux ne couvrait à la fois nos deux besoins.

Pour répondre à cette problématique de recherche, nous avons réalisé un travail d'analyse de la littérature sur les exigences non-fonctionnelles qui peuvent potentiellement être définies au niveau métier. Nous effectuons une première analyse de la littérature sur ce sujet en nous basant sur les travaux présentant dans [Santos et al., 2010], [Cysneiros et al., 2004], [Mylopoulos et al., 1992], [Chung et al., 2000], [Rodriguez et al., 2007], [Chung et al., 2009], et [Heinrich et al., 2011]. Chacun de ces travaux fournit un ensemble d'exigences non-fonctionnelles proposé, en extrait, dans le Tableau 3 ci-après (cf. Annexe 2 pour le tableau complet). La dimension horizontale représente la liste des exigences non-fonctionnelles définies dans les travaux de recherche élaborés par les auteurs. Même si les auteurs se différencient sur quelques exigences, nous remarquons que les exigences suivantes : « disponibilité », « prix », « temps de réponse », « sécurité », « exactitude (*Accuracy*) » et « intégrité » sont majoritairement présentes dans tous les travaux. À l'exception de l'exactitude, nous constatons que ces exigences sont également présentes dans le standard WSQF. Nous effectuons une deuxième analyse représentée dans le Tableau 3 (cf. Annexe 2 pour le tableau complet) par la dimension verticale.

Tableau 3 : Extrait du tableau « analyse croisée exigences non-fonctionnelles métier / WSQF ».

		Exigences non-fonctionnelles Métier		WSQF
[Santos et al., 2010]	Performance	Time Performance	Response Time	Response Time
			Throughput	Maximum throughput
	Reliability	Space Performance		
		Availability		Availability
		Fault Tolerance		
	Security	Accuracy		
		Access Control		Authorization + Authentication
		Confidentiality	Internal Confidentiality	Confidentiality
			External Confidentiality	
	Availability		Availability	
...			...	

Légende



Exigence non-fonctionnelle couverte par WSQF



Exigence non-fonctionnelle non couverte par WSQF

Cette analyse ne remet pas en cause notre constatation. Même si la spécification de WSQF est initialement dédiée à la description non-fonctionnelle des services (niveau technique), elle couvre la majorité des exigences non-fonctionnelles proposée dans la littérature pour les processus métier. De plus, comme nous avons employé cette spécification pour l'annotation non-fonctionnelle des services, ceci facilitera la réconciliation le métier et le technique. Ces raisons nous conduisent à la proposition d'utiliser les propriétés du standard WSQF (cf. Annexe 1) pour la modélisation et l'annotation des exigences non-fonctionnelles des activités du processus métier collaboratif BPMN 2.0 [Zribi et al., 2013].

Par ailleurs, comme notre objectif principal est d'associer à chacune des activités un service parmi ceux disponibles dans notre registre EasierGov, il est également nécessaire de savoir la pertinence de chacune des exigences pour le client. Pour cela, *Petals BPM-NFR* fournit aux utilisateurs la possibilité d'exprimer leurs préférences par rapport aux exigences non-fonctionnelles choisies. Cette préférence prend la forme de poids que l'utilisateur attribue à chaque exigence.

Nous schématisons dans la Figure 26, (une reprise de la Figure 24), l'architecture générale de *Petals BPM* à laquelle nous avons rajouté le standard WSQF comme modèle d'annotation non-fonctionnelle et la gestion des préférences des utilisateurs.

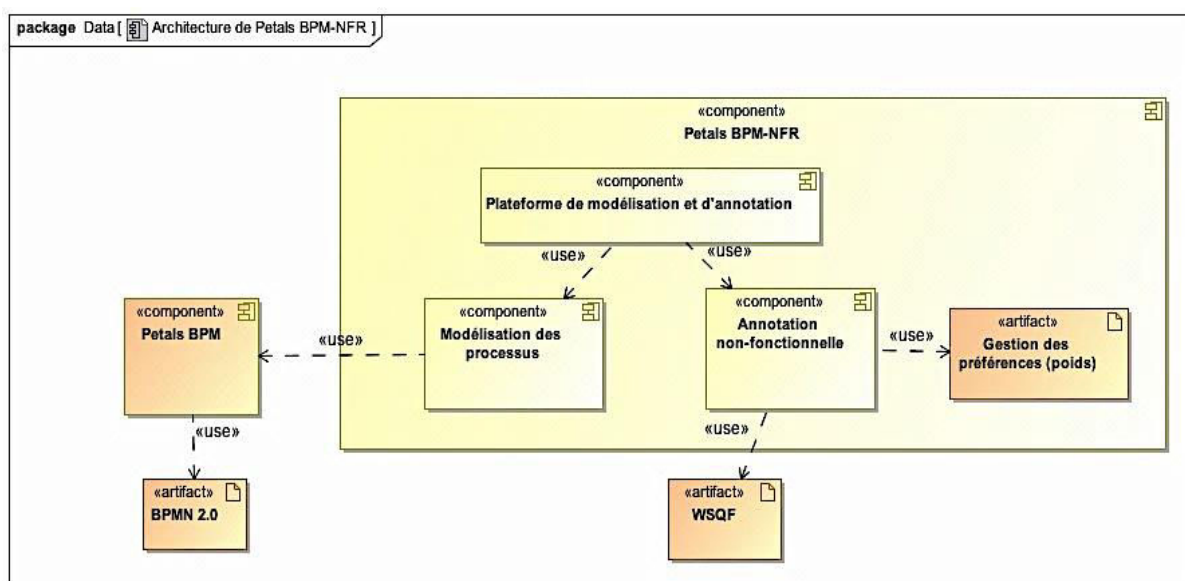


Figure 26 : Architecture de *Petals BPM – NFR* : annotation non-fonctionnelle détaillée.

III.3. EasierGov-NFR : gouvernance SOA pour la gestion des services et des propriétés non-fonctionnelles

La prolifération des services au sein du système d'information rend nécessaire la rationalisation de leur gestion. Par ailleurs, les propriétés non-fonctionnelles sont présentes durant toutes les phases du cycle de vie d'un service. Leur gestion nécessite un registre qui va au delà de cataloguer les données du service et d'en faire l'inventaire. Elle requiert également de stocker les métadonnées des services telles que des contrats de services et les propriétés non-fonctionnelles. La gouvernance SOA permet de répondre à ces problématiques.

EasierGov [Zribi et al., 2012] implémente le paradigme de Gouvernance SOA. Comme l'illustre la Figure 27, il est à la frontière des trois domaines suivants : la Découverte SOA (*SOA Discovery*), la Gestion SOA (*SOA Management*) et les Politiques de la Gouvernance (*Governance Policies*).



Figure 27 : Présentation générale d'EasierGov [Zribi et al., 2012].

- Registre de gouvernance SOA (*SOA Governance Registry*) désigne le registre de services qui assure les fonctionnalités de la découverte du service. Il sert non seulement à faire l'inventaire, à cataloguer les données du service, mais aussi à une base de données pour stocker les métadonnées des services nécessaires à la gouvernance (nous citons à titre d'exemple : *Web Service Description Language (WSDL)*, documents, les propriétés non-fonctionnelles, les contrats de service, etc.). Ce registre permet aux fournisseurs de publier leurs services et les métadonnées respectives. Ces derniers seront mis par la suite à la disposition des utilisateurs pour qu'ils puissent les utiliser.
- Gestion SOA (*SOA Management*) couvre la prise en charge des cycles de vie des services et ce de la phase de conception jusqu'à la phase d'exécution. Ceci inclut la définition des politiques et règles adéquates pour les services à chaque étape de leurs cycles de vie. Par

ailleurs, la gestion SOA comprend aussi la gestion des propriétés non-fonctionnelles des services telles que les contrats de services.

- Politiques de gouvernance (*Governance Policies*) constituent un élément fondamental pour la Gouvernance SOA. En effet, grâce à l'adoption de règles et de standards, la gouvernance assure la compatibilité entre les services de l'organisation et ceux appartenant à d'autres entreprises ou organismes. Les politiques de gouvernance sont définies à chaque étape du cycle de vie et sont fortement liées aux domaines de découverte et de gestion SOA. Chaque acteur impliqué dans la gouvernance a un rôle spécifique et des responsabilités clairement identifiées.

Les travaux de cette thèse portent sur les deux parties : la découverte SOA et la sous-partie gestion des services et leurs propriétés non-fonctionnelles de la partie gestion SOA. Ces parties constitueront les éléments de notre prototype de gouvernance nommé EasierGov-NFR (cf. Figure 28).

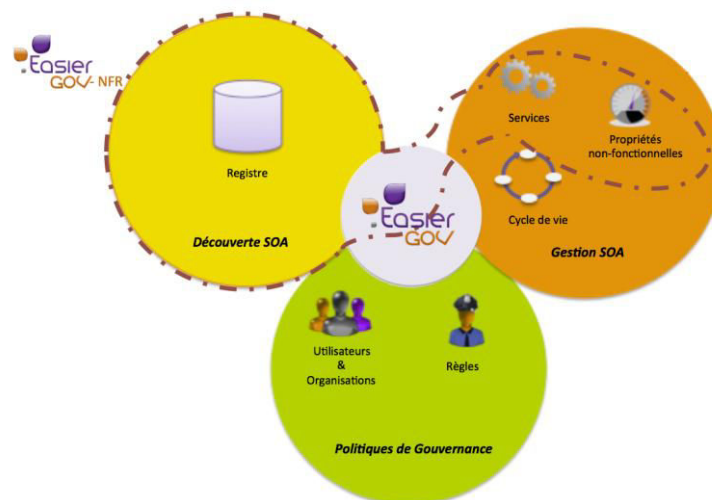


Figure 28 : Présentation générale d'EasierGov-NFR.

III.3.1. Découverte SOA

Au cœur de la gouvernance SOA émerge le concept de registre de services : cet annuaire robuste favorisant la publication, la découverte et la réutilisation des services. Les fonctionnalités de gouvernance, telles que la publication des services, la recherche, la gestion de leurs cycles de vie ou de leurs propriétés non-fonctionnelles sont fondamentales. EasierGov-NFR répond à ces besoins en offrant un registre innovant dédié à la fois aux services et à leurs propriétés non-fonctionnelles. Cette fonctionnalité est essentielle pour faire face à un nombre important de services hétérogènes qui répondent aux mêmes besoins fonctionnels. Par ailleurs, grâce à sa liaison à Petals BPM-NFR, EasierGov-NFR permet d'effectuer la réconciliation non-fonctionnelle entre les activités métier du

processus collaboratif (modélisé et annoté non-fonctionnellement) et les services disponibles. La Figure 29 présente l'architecture générale d'EasierGov-NFR.

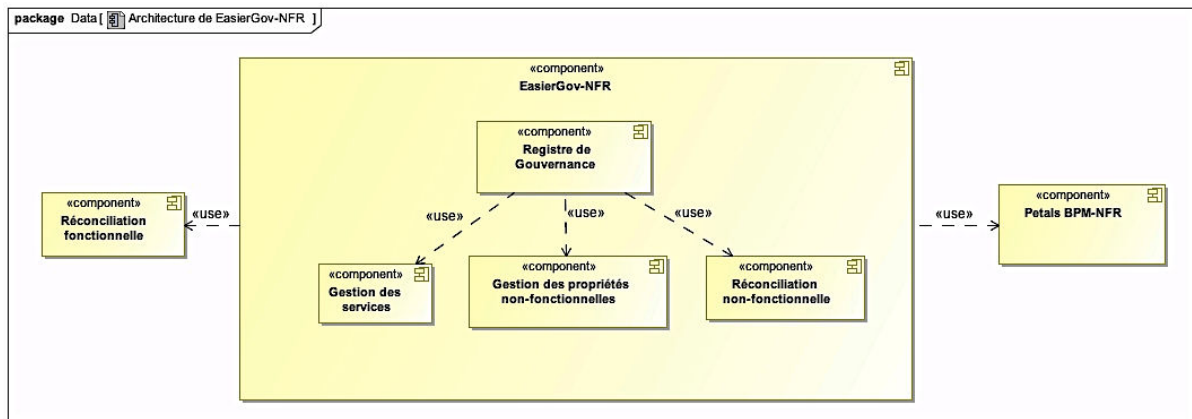


Figure 29 : Présentation de l'architecture d'EasierGov-NFR.

EasierGov-NFR est constitué de trois composants : gestion des services, gestion des propriétés non-fonctionnelles et réconciliation non-fonctionnelle. Les deux premiers sont détaillés dans les sections qui suivent et le troisième dans le chapitre suivant (cf. Chapitre IV). Comme nous l'avons détaillé depuis le début de nos travaux, la réconciliation non-fonctionnelle tente de sélectionner les services (parmi ceux fournis par la réconciliation fonctionnelle) pour les activités métier du processus modélisé dans Petals BPM-NFR. Ceci explique l'utilisation de ces deux composants (réconciliation fonctionnelle et Petals BPM-NFR) par EasierGov-NFR.

III.3.2. Gestion des services

Pour assurer le bon fonctionnement d'un système d'information et une meilleure rationalisation de la gestion des services, il est nécessaire d'appliquer des politiques adéquates. Ces politiques sont définies à chaque étape du cycle de vie par le responsable de gouvernance d'EasierGov. La gestion des services dans EasierGov-NFR commence à partir de la phase de publication. Afin d'aborder la problématique d'hétérogénéité de services métier, une des politiques qu'il est important de définir est la définition d'un langage commun pour la gestion des descriptions des services dans le registre de gouvernance. À cette fin, nous adoptons le langage standard et communément utilisé WSDL (cf. Chapitre II, sous-section II.2.2).

Dans EasierGov-NFR et pour la gestion des services, nous identifions deux rôles : celui de fournisseur de services et celui de consommateur. Ces deux rôles sont menés à suivre les politiques définies dans le registre de gouvernance. Les diagrammes de cas d'utilisation ci-après (cf. Figures 30 et 31) décrivent les fonctionnalités de chacun d'eux.

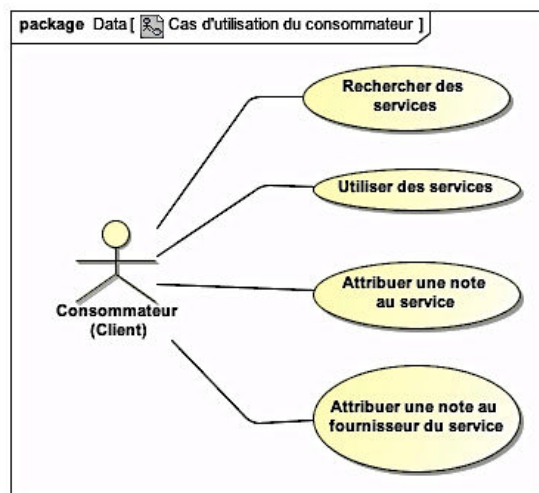
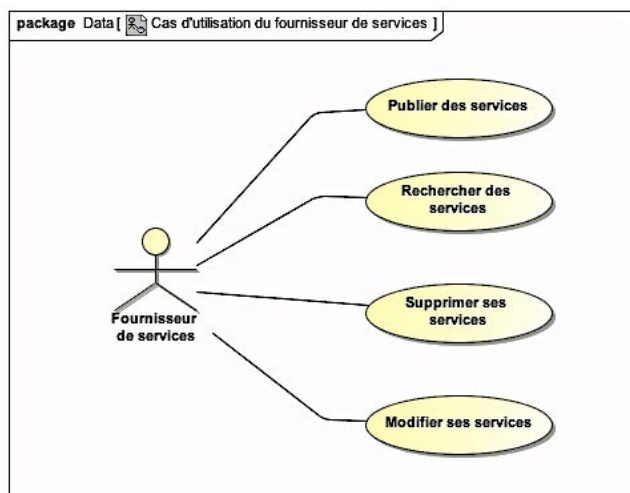


Figure 30 : Cas d'utilisation du fournisseur de service. **Figure 31 :** Cas d'utilisation du consommateur.

Le *fournisseur de services* après avoir conçu ses services, implémenté leurs fonctionnalités en suivant le standard WSDL et effectué des tests, peut les publier dans le registre de gouvernance EasierGov-NFR. Il peut également rechercher des services. En outre, le registre lui permet aussi de gérer ses services après leur publication en lui attribuant le droit de modification et de suppression.

Le *consommateur* (client) peut rechercher des services dans le registre de gouvernance et notamment les utiliser. Par la suite, le consommateur a la possibilité d'exprimer sa satisfaction par rapport à l'utilisation du service et peut le noter et/ou noter son fournisseur.

III.3.3. Gestion des propriétés non-fonctionnelles

Les services Web peuvent fonctionner en se basant uniquement sur leurs besoins fonctionnels. Cependant, pour plus d'efficacité, un service doit également fournir une certaine garantie quant à sa disponibilité, sa fiabilité, son efficacité et donc quant à ses propriétés non-fonctionnelles. Ces propriétés sont très importantes pour garantir et maintenir la confiance auprès des clients. Dans cette sous-section, nous présentons la manière avec laquelle nous pouvons gérer les propriétés non-fonctionnelles à partir du registre de gouvernance SOA : EasierGov-NFR.

III.3.3.1. Annotation non-fonctionnelle des services

La spécification de WSQF permet d'enrichir la description de services par des propriétés non-fonctionnelles. Toutefois pour pouvoir les gérer, il est bien nécessaire de savoir où nous pouvons les trouver.

Le contrat de service (appelé aussi l'accord de niveau de service, *Service Level Agreement – SLA*) représente un accord bilatéral entre le fournisseur de service et le client. Cet accord spécifie et

identifie explicitement (i) les attentes, (ii) les engagements des différentes parties autour de l'utilisation du service en terme de qualité de service et (iii) les pénalités en cas de manquement ou les récompenses en cas d'une meilleure prestation. En effet, sans contrat défini explicitement ou implicitement, il est difficile de parler de la qualité de service et donc des propriétés non-fonctionnelles [Mammeri, 2005].

Un contrat SLA est composé généralement de trois sections [Keller et al., 2003] : (i) la première section, intitulée « *Parties* », spécifie les parties impliquées dans le contrat, à savoir : les signataires (le fournisseur de service et le client) et les parties tierces qui participent au contrôle du contrat, (ii) la deuxième section est *Service « Description »* qui définit la description de service. Cette dernière contient les opérations du service, leurs messages, l'encodage du transport de ces messages, les propriétés non-fonctionnelles, la durée et la fréquence des mesures des qualités de service. Enfin, (iii) la troisième section « *Obligation* » comprend les obligations du contrat, à savoir : la période de validité, les contraintes à respecter, les pénalités en cas de sa violation et les récompenses.

Tout comme les services, les contrats SLA ont un cycle de vie. Celui appliqué à cette architecture est composé de six phases [Wustenhoff et al., 2002] et [Sun et al., 2005] :

1. *Établissement d'un modèle SLA* (appelé aussi un gabarit ou un *template*) : cette première phase consiste en la spécification, la modélisation et la publication d'un modèle de contrat SLA ;
2. *Négociation du contrat* : une fois que le service est choisi, il est important de spécifier le niveau de qualité requis ou de négocier les clauses existantes dans le modèle fourni par le fournisseur ;
3. *Etablissement du contrat* : dans cette phase, le contrat SLA est créé et doit maintenant être signé par le fournisseur du service et le client ;
4. *Exécution et surveillance des performances engagées* : cette phase consiste à surveiller l'exécution du service afin de s'assurer que toutes les clauses engagées ont été bien respectées par les deux parties ;
5. *Terminaison* : cette phase correspond à la fin de l'exécution du service ;
6. *Application des pénalités ou récompenses* : en cas de non respect et de violation des clauses du contrat, il est important d'appliquer les sanctions prédéfinies ou dans le cas contraire les récompenses engagées.

Il est évident que l'ensemble du cycle de vie d'un contrat de service est basé sur un modèle utilisé pour représenter les différentes parties de ses clauses. Ce modèle contient la liste des services offerts et intègre les capacités du fournisseur en terme de qualité de service, et donc en terme des propriétés non-fonctionnelles, auxquelles son service prétend d'assurer [Sun et al., 2005].

Il existe plusieurs spécifications pour la représentation de ce modèle SLA. Parmi lesquelles, nous citons WSOL (*Web Service Offerings Language*) [Tosic et al., 2002], WSLA (*Web Service Level Agreement*) [Keller et al., 2003] et WS-Agreement (*Web Service-Agreement*) [Andrieux et al., 2007]. Elles s'intéressent toutes à la définition du contrat SLA. Toutefois, le standard WS-Agreement se focalise plus sur la création des modèles de contrat. Il se base sur le langage XML et comporte une spécification claire, flexible et assez simple. Par ailleurs, ce standard est extensible à tout modèle écrit en XML. De plus, l'équipe R&D de Linagora Labs a réalisé un outil de surveillance (*monitoring*) appelé « EasierBSM » permettant le contrôle de l'exécution des contrats de type WS-Agreement pour vérifier leur bon déroulement et leur non-violation. Ceci nous a conduits à l'adopter et à faire l'hypothèse que les fournisseurs créent leurs SLA templates dans le respect de la spécification WS-Agreement.

III.3.3.2. Modèle de contrat de service Web (*WS-Agreement template*)

Un template SLA, selon la spécification de WS-Agreement, est effectué par le fournisseur de services et est composé de trois grandes parties. Nous résumons sa structure dans la Figure 32 ci-après.



Figure 32 : Structure d'un template SLA selon WS-Agreement [Andrieux et al., 2007].

- *Name et Context* : la première section est optionnelle et définit le nom du template SLA. Le contexte contient l'ensemble des métadonnées du contrat telles que : le fournisseur, la date d'expiration du template, etc. ;
- *Termes* : cette section contient les termes qui décrivent le contrat. Ils englobent deux sous-parties : *Service Description Terms* (SDT) et *Guarantee Terms*. Les modalités de description de service (SDT) spécifient les descriptions fonctionnelles du service à fournir. Les modalités de garantie (*Guarantee Terms*) incluent les capacités en termes de qualité de

services (propriétés non-fonctionnelles) auxquelles le service peut répondre, les pénalités ou / et les récompenses ;

- *Agreement Creation Constraints* : les contraintes de création de l'accord définissent les règles supplémentaires qui doivent être suivies lors de la proposition d'un contrat par le client.

Les propriétés non-fonctionnelles sont donc définies dans les modalités de garantie. Ces dernières sont composées de quatre éléments : *Service Scope*, *Qualifying Condition*, *Service Level Objectives* et *Business Value List*.

La portée du service (*Service Scope*) spécifie le service qui est couvert par les modalités de garantie. La condition de qualification (*Qualifying Condition*) précise les conditions préalables qui doivent être remplies pour l'application d'une garantie. Les objectifs de niveau de services (*Service Level Objectives - SLO*) définissent les propriétés non-fonctionnelles qui sont assurées par le service et qui doivent être monitorées afin de déterminer la réalisation des modalités de garanties. Enfin, la liste des valeurs métier (*Business Value List*) comprend les pénalités, les récompenses et les propriétés non-fonctionnelles qui ne nécessitent pas d'être surveillées (e.g. le prix).

Comme nous pouvons le remarquer, ce sont les sous-parties *SLO* et *Business Value List* de la partie *Guarantee Terms* qui permettent au fournisseur du service de définir les propriétés non-fonctionnelles auxquelles son service prétend de répondre. Afin de déterminer quelles propriétés peuvent être définies dans chacune de sous-parties, nous proposons dans la sous-section qui suit une classification des critères non-fonctionnels de WSQF.

III.3.3.3. Classification des propriétés non-fonctionnelles

Après l'étude et l'analyse de l'ensemble des propriétés non-fonctionnelles définies dans la spécification de WSQF, nous distinguons trois catégories distinctes : des *variables Fournisseur*, des *variables Client* et des *variables analysées*.

- Les *variables Fournisseur* sont définies dans le template SLA par le fournisseur du service. Nous les avons appelées ainsi étant donné que seul le fournisseur a le droit de gérer (ajouter, supprimer et modifier) les valeurs de ces propriétés. Nous décomposons ces variables en deux sous-catégories : des *variables monitorables* et d'autres qui sont *non monitorables*. La première sous-catégorie regroupe les propriétés non-fonctionnelles qui nécessitent d'être monitorées lors de l'exécution du service telles que le temps de réponse, la disponibilité, le cryptage, les propriétés d'interopérabilité, etc. Ceci nous permettra de vérifier que les valeurs des propriétés non-fonctionnelles engagées par le fournisseur ont bien été respectées ou non. Contrairement à ces variables, les valeurs des *variables non*

monitorables ne nécessitent pas d'être vérifiées à l'exécution telles que le prix, les pénalités, les récompenses, etc.

- Les valeurs des *variables Client* sont renseignées directement par le client du service. Par exemple, à l'issue de l'utilisation d'un service, un client peut attribuer une note à lui ou à son fournisseur. Le degré de satisfaction contribue à constituer ce qui est communément appelé, la réputation d'un service ou d'un fournisseur ;
- Les valeurs des *variables analysées* ne sont introduites ni par le fournisseur du service ni par le client. Elles sont renseignées par le registre de gouvernance. Nous retrouvons dans cette catégorie la propriété popularité du service (*Service Recognition*). Nous considérons que la valeur de cette propriété est calculée selon le nombre de clients qui ont utilisé le service.

Nous avons classé, dans le Tableau 4 ci-après, l'ensemble des propriétés non-fonctionnelles de WSQF (la dimension horizontale) dans chacune de ces catégories/sous-catégories (la dimension verticale).

Tableau 4 : Classification générale des propriétés non-fonctionnelles de WSQF.

		Variables Fournisseur		Variables Client	Variables analysées	
		Variables monitorables	Variables non monitorables			
Propriétés non-fonctionnelles de WSQF	Business	Price		X		
		Penalty and Incentive		X		
	Value	Business Performance		X		
		Service Reputation			X	
	Quality	Service Provider Reputation			X	
		Service Recognition				X
	Service	Response Time	X			
		Maximum Throughput	X			
	Level	Availability	X			
		Successability	X			
	Measurement	Accessibility	X			
		Standard Adoptability	X			
	Interoperability	Standard Conformability	X			
		Relative Proofness	X			
	Business	Message Reliability	X			
		Transaction Integrity	X			
	Processing	Collaborability	X			
		Informability	X			
	Manageability	Observability	X			
		Controllability	X			
	Security	Encryption	X			
		Non Repudiation	X			
		Authentication	X			
		Authorization	X			
		Availability	X			
		Audit	X			
		Integrity	X			
	Privacy	X				

Étant donné que le modèle de contrat du service est conçu par le fournisseur, seules les propriétés non-fonctionnelles de WSQF de la catégorie *variables Fournisseur* peuvent y être définies. Comme nous l'avons détaillé dans la sous-section précédente (cf. III.3.3.2), à l'inverse des assertions *business value list*, les assertions *SLO* contiennent les propriétés qui nécessitent d'être surveillées et vérifiées lors de l'exécution afin de s'assurer de leur conformité. Nous proposons dans le Tableau 5 un résumé des propriétés non-fonctionnelles du type *variables Fournisseur* qui peuvent être incluses dans chacune des deux parties (SLO ou business value list) du modèle de contrat WS-Agreement.

Tableau 5 : Extension de WS-Agreement par les propriétés non-fonctionnelles de WSQF de type variables Fournisseur.

			Modèle de contrat WS-Agreement	
			SLO	Business value list
Propriétés non-fonctionnelles de WSQF de type Variables Fournisseur	Business Value	Price		X
		Penalty and Incentive		X
		Business Performance		X
	Service Level Measurement	Response Time	X	
		Maximum Throughput	X	
		Availability	X	
		Successability	X	
		Accessibility	X	
	Interoperability	Standard Adoptability	X	
		Standard Conformability	X	
		Relative Proofness	X	
	Business Processing	Message Reliability	X	
		Transaction Integrity	X	
		Collaborability	X	
	Manageability	Informability	X	
		Observability	X	
		Controllability	X	
	Security	Encryption	X	
		Non Repudiation	X	
		Authentication	X	
		Authorization	X	
Availability		X		
Audit		X		
Integrity		X		
Privacy	X			

III.3.3.4. Gestion des propriétés non-fonctionnelles dans EasierGov-NFR

Dans le registre de gouvernance EasierGov-NFR, outre les fonctionnalités de découverte et de publication de services nous proposons des fonctionnalités dédiées à la gestion des propriétés non-fonctionnelles. Cette gestion couvre la prise en compte de l'ensemble des propriétés du standard WSQF pour chaque service publié par son fournisseur.

➤ *La gestion des variables Fournisseur*

Nous proposons dans EasierGov-NFR une librairie permettant aux fournisseurs de publier, pour un service donné, un ou plusieurs modèles de contrat de type WS-Agreement. Ces modèles, que nous avons étendus par WSQF, contiennent les propriétés non-fonctionnelles de type variables Fournisseur. Il est donc nécessaire de pouvoir récupérer leurs valeurs afin de les enregistrer dans notre registre de gouvernance SOA. Pour répondre à ceci, nous proposons les trois étapes suivantes :

1. Publication des services : EasierGov-NFR contient une librairie permettant aux fournisseurs de publier les descriptions de leurs services de type WSDL ;
2. Publication des templates SLA : à l'issue de la phase de publication du service, le registre de gouvernance permet au fournisseur de publier également un ou plusieurs modèles de contrats SLA de type WS-Agreement associés à ce service ;
3. Extraction dynamique et enregistrement des valeurs des propriétés non-fonctionnelles : pour chaque template SLA publié, EasierGov-NFR extrait dynamiquement les valeurs des propriétés non-fonctionnelles et les stocke dans le registre.

➤ *La gestion des variables Client*

Afin de gérer les *variables Client*, à savoir la réputation du service et la réputation du fournisseur de services, nous considérons le déroulement suivant (cf. Figure 33) :

Tout commence par l'étape de découverte du service. EasierGov-NFR fournit aux clients un accès vers la librairie de gestion de services (*services-api*) afin de leur permettre de rechercher un ou plusieurs services parmi ceux existants. Après avoir sélectionné le service qui est susceptible de répondre à ses attentes, deux possibilités se présentent :

- le client utilise le service et par la suite il peut lui attribuer une note ou à son fournisseur selon le degré de sa satisfaction ;
- le client note directement le service ou son fournisseur (par exemple dans le cas où il a préalablement utilisé le service, mais il a oublié de laisser une évaluation).

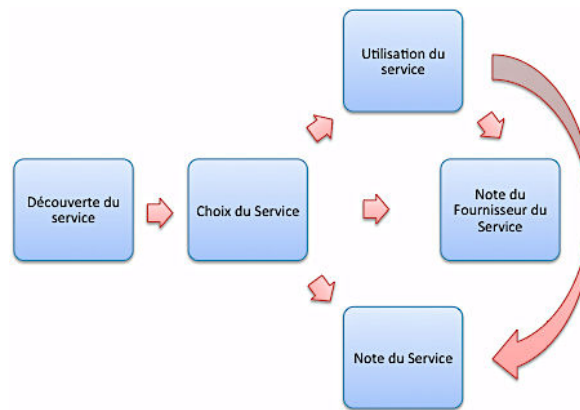


Figure 33 : *Gestion des variables non mesurables : Variables Client.*

➤ **La gestion des variables analysées**

Dans notre modèle WSQF, seule la variable *popularité du service* appartient à cette catégorie. Afin de la prendre en compte, nous considérons dans EasierGov-NFR qu'elle est calculée selon le nombre de clients qui ont utilisé le service en question.

III.4. Conclusion

Dans ce chapitre, nous avons présenté notre contribution dans le domaine de l'annotation non-fonctionnelle des processus métier BPMN 2.0 et l'apport de la gouvernance SOA dans la gestion des services et de leurs propriétés non-fonctionnelles. Nous avons mis en évidence les problématiques à résoudre et nous avons présenté :

- notre plateforme Petals BPM-NFR pour l'enrichissement de la modélisation graphique des processus métier BPMN 2.0 par l'ajout de l'aspect non-fonctionnel. Nous avons commencé par une analyse de la littérature des exigences non-fonctionnelles qui peuvent être définies au niveau métier. Par la suite, nous avons montré, la possibilité d'utiliser le standard WSQF (destiné à l'origine pour une utilisation technique) au niveau métier. Enfin, nous avons présenté l'architecture et les fonctionnalités de la plateforme Petals BPM-NFR.
- notre cadre de gouvernance SOA EasierGov-NFR qui définit un registre de services qui va au delà de cataloguer les services. Il permet également de gérer les propriétés non-fonctionnelles des services et les contrats de services. D'abord, nous avons présenté l'architecture globale du registre. Ensuite, nous avons détaillé la gestion des services dans EasierGov-NFR. Enfin, nous nous sommes intéressés à la partie relevant de la gestion des propriétés non-fonctionnelles et des contrats de services.

Dans le chapitre suivant, nous détaillons la réconciliation des activités métier du processus collaboratif BPMN 2.0 annotées à l'aide d'exigences non-fonctionnelles en assignant à chacune de ces activités un service (parmi ceux publiés dans le registre EasierGov-NFR et qui satisfont les attentes fonctionnelles) qui répond à ses exigences non-fonctionnelles.

Chapitre IV

Réconciliation non-fonctionnelle de services

« L'art de la réussite est celui des réconciliations. »

Jacques Deval

IV.1. INTRODUCTION.....	79
IV.2. RECONCILIATION DE SERVICES	79
IV.3. RECONCILIATION NON-FONCTIONNELLE DE SERVICES.....	80
IV.4. RECONCILIATION NON-FONCTIONNELLE DE SERVICES DE TYPE « 1-1 ».....	84
IV.4.1. Présentation générale de l'approche « 1-1 ».....	84
IV.4.2. Algorithme de réconciliation non-fonctionnelle de type « 1-1 »	84
IV.4.2.1. Étape 1 : Classification des valeurs des exigences non-fonctionnelles.....	86
IV.4.2.2. Étape 2 : Élimination des services candidats ayant des valeurs aberrantes	87
IV.4.2.3. Étape 3 : Normalisation et centrage des services candidats par rapport aux exigences non-fonctionnelles de l'activité métier	89
IV.4.2.4. Étape 4 : Classement et sélection des services candidats.....	95
IV.5. RECONCILIATION NON-FONCTIONNELLE DE SERVICES DE TYPE « 1-N »	98
IV.5.1. Présentation générale de l'approche « 1-N »	98
IV.5.2. Transformation d'une réconciliation non-fonctionnelle de type « 1-N » en une de type « 1-1 ».....	100
IV.5.2.1. Étape 1 : Décomposition en plusieurs sous-compositions de deux services	100
IV.5.2.2. Étape 2 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition.....	101
IV.6. CONCLUSION	106

IV.1. Introduction

Dans le chapitre précédent, nous avons présenté nos deux prototypes PetalsBPM-NFR et EasierGov-NFR. Le premier nous permet de modéliser graphiquement les processus BPMN 2.0 et d'annoter ses activités avec des exigences non-fonctionnelles. Le deuxième représente notre registre de gouvernance SOA contenant les services Web, leurs propriétés non-fonctionnelles et les modèles de contrats SLA. Notre objectif principal, est de relier le monde métier avec celui technique et donc d'assigner à chaque activité métier de notre processus collaboratif le ou les bons services Web parmi ceux existant dans notre registre de gouvernance SOA. Cette mise en correspondance est appelée « *la réconciliation de services* » que nous présentons dans la Section IV.2. La réconciliation de services se divise en deux parties : la réconciliation fonctionnelle (cf. [Boissel-Dallier, 2012]) et la réconciliation non-fonctionnelle. Dans ces travaux de thèse, nous abordons ce second type de réconciliation (Section IV.3) (cf. Figure 34).

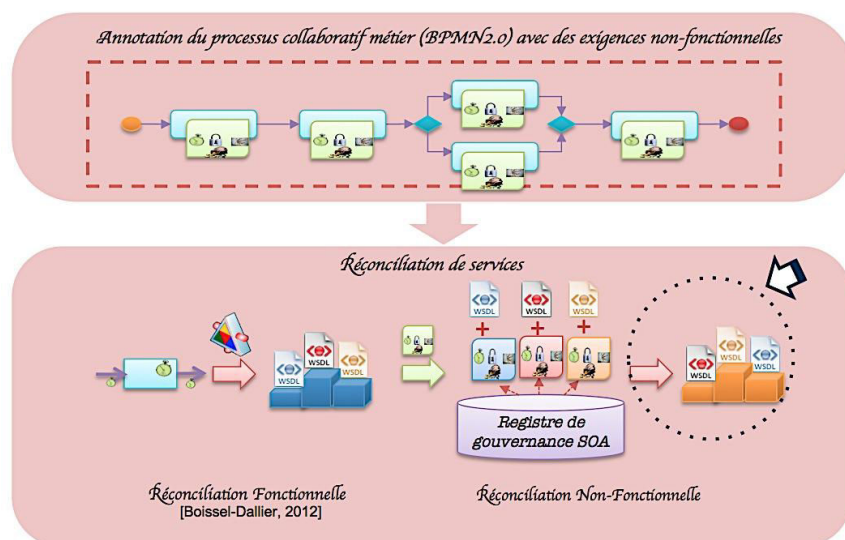


Figure 34 : Réconciliation non-fonctionnelle : sélection et classement des services.

IV.2. Réconciliation de services

L'orchestration du processus métier sous la forme de processus exécutable nécessite un mécanisme de réconciliation entre ceux-ci. Le principe général de ce mécanisme est de trouver, pour chacune des activités du processus métier collaboratif BPMN 2.0, un service ou une composition de services répondant à ses exigences fonctionnelles et non-fonctionnelles. Ces services / compositions de services doivent être classés en fonction de leur proximité avec l'activité métier cible afin que le client puisse sélectionner la solution qui lui semble la plus adéquate. Comme l'illustre la Figure 35, la

réconciliation de services se compose principalement de trois couches : (i) la spécification des exigences de l'utilisateur sur les activités du processus métier, (ii) la découverte des services qui satisfont ces exigences et leur classement selon le degré de satisfaction, et enfin (iii) la sélection de services pour chacune des activités du processus métier.

La réconciliation de services se base sur les exigences de l'utilisateur. Ces dernières peuvent être de type fonctionnel et/ou non-fonctionnel. Nous parlons de « *réconciliation fonctionnelle de services* » quand il s'agit d'exigences fonctionnelles, de « *réconciliation non-fonctionnelle de services* » dans le cas d'exigences non-fonctionnelles, et de « *réconciliation de services* » quand il s'agit des deux.

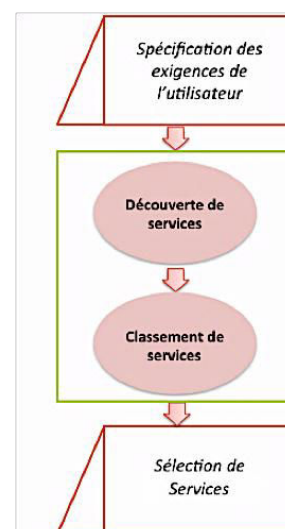


Figure 35 : Réconciliation de services.

Durant les travaux de cette thèse, nous améliorons les résultats de la réconciliation fonctionnelle établis dans la thèse de N. Boissel-Dallier [Boissel-Dallier, 2012] par l'ajout de la prise en compte des exigences non-fonctionnelles du client (cf. Figure 36). Cependant, cette amélioration prend la forme d'un deuxième filtre (qui est non-fonctionnel). Nous l'appliquons à la liste des services sélectionnés sur des critères fonctionnels dans le but de présenter au client une liste finale de services répondant à l'ensemble de ses besoins. Nous présentons dans la section qui suit le principe de la réconciliation non-fonctionnelle de services.

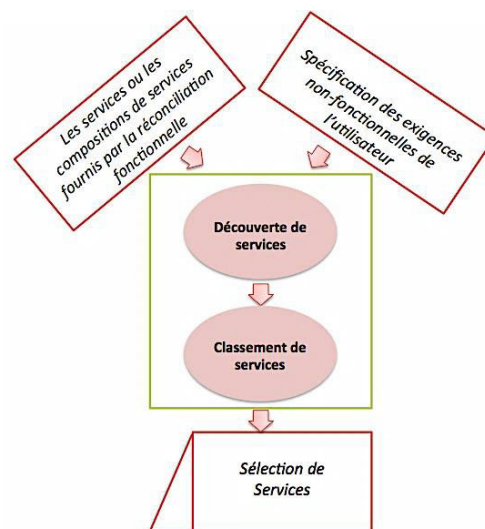


Figure 36 : Réconciliation non-fonctionnelle de services.

IV.3. Réconciliation non-fonctionnelle de services

Nous avons détaillé dans le chapitre précédent, la première couche de la réconciliation non-fonctionnelle de services, à savoir la définition des exigences non-fonctionnelles de l'utilisateur. Nous avons présenté aussi une plateforme visuelle « PetalsBPM-NFR » qui permet aux utilisateurs de modéliser et de créer leurs processus métier collaboratifs et d'annoter les activités à l'aide d'exigences non-fonctionnelles.

Nous avons aussi présenté le registre de gouvernance SOA « EasierGov-NFR » à travers lequel les fournisseurs peuvent publier leurs services enrichis par des propriétés non-fonctionnelles (via les

templates de contrats de services). Nous proposons également dans ce registre une approche de réconciliation non-fonctionnelle.

Pour la présentation de cette approche, nous considérons l'exemple d'une construction d'un pont en Lego. Il manque une pièce à ce pont (cf. Figure 37). Cette pièce est de **huit plots** (*critère fonctionnel*), de **couleur exigée grise** (*critère non-fonctionnel*), et de **nuance souhaitée Gris Fer** (*critère non-fonctionnel*). L'objectif est donc de compléter la construction du pont à partir des pièces disponibles dans la réserve de Lego.

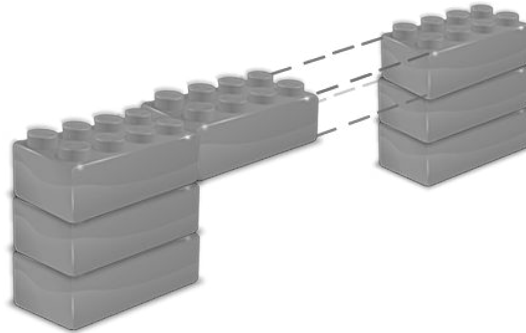


Figure 37 : Exemple d'une construction d'un pont en Lego.

Pour atteindre notre objectif, nous procédons à une recherche en deux étapes :

1) Recherche d'une pièce de huit plots

Cette recherche consiste à chercher parmi toutes les pièces à disposition, une pièce de huit plots (*critère fonctionnel*). Deux situations sont à prévoir :

- il existe une ou plusieurs pièces de Lego de huit plots. L'ensemble de ces pièces est sélectionné. Cette approche est appelée *réconciliation fonctionnelle « 1-1 »* ;
- il n'existe aucune pièce de Lego de huit plots. La difficulté est tournée par une approche de compositions de pièces (*réconciliation fonctionnelle « 1-N »*). Il s'agit d'assembler deux ou plusieurs pièces afin d'atteindre exactement, ou à défaut, de se rapprocher au plus près de la forme requise.

En effet, nous pouvons voir la réconciliation fonctionnelle « 1-N » comme une utilisation successive de la réconciliation fonctionnelle « 1-1 » sur un ensemble de combinaisons de pièces potentielles. Le temps de calcul nécessaire à l'obtention de résultats de composition est donc logiquement plus élevé que pour la réconciliation unitaire (« 1-1 »). En outre, la probabilité de proposer une composition de pièces qui réponde parfaitement au besoin risque donc d'augmenter rapidement avec l'ajout de nouvelles pièces. Par conséquent, *il est préférable de privilégier la réconciliation des plus petits ensembles possibles.*

2) Recherche d'une pièce de couleur exigée grise et de nuance souhaitée gris fer

À partir des ensembles sélectionnés à l'issue de la première recherche (filtre fonctionnel), nous effectuons une deuxième en nous basant cette fois-ci sur la couleur et la nuance requise (les deux critères non-fonctionnels).

Cette réconciliation consiste à étudier la couleur et la nuance de chacun des résultats fournis par la réconciliation fonctionnelle. Nous remarquons que dans la demande de recherche de la pièce manquante, les deux critères couleur et nuance n'ont pas le même poids : la couleur grise est *exigée* tandis que la nuance gris fer est *souhaitable*. Autrement dit, à défaut d'une pièce (ou composition de pièces) de huit plots de couleur grise et de nuance gris fer, une pièce/composition de pièces de couleur grise et de nuance qui se rapproche de celle requise et peut correspondre à l'objectif.

Nous distinguons deux cas possibles (selon le type de l'ensemble retourné par la réconciliation fonctionnelle) :

- *réconciliation non-fonctionnelle « 1-1 »* (pièces unitaires) : pour chacune des pièces fournies par la réconciliation fonctionnelle « 1-1 », nous comparons sa couleur et sa nuance avec celles requises. Dans la Figure 38, nous reprenons l'exemple présenté dans la Figure 37 afin d'illustrer la réconciliation « 1-1 ». L'ensemble sélectionné par la réconciliation fonctionnelle est composé de trois pièces de Lego de huit plots : une pièce blanche, une noire, et enfin une grise. Nous avons également à disposition les nuances de chacune : la blanche est de nuance *argile*, la noire est de nuance *noir d'ivoire*, et la grise est de nuance *gris fer*. Étant donné que la couleur grise est *exigée*, nous éliminons les pièces de couleur blanche et noire. La pièce restante est de couleur grise et de nuance *gris fer*. Ceci correspond exactement à la demande. Cette pièce est sélectionnée et le pont est construit.

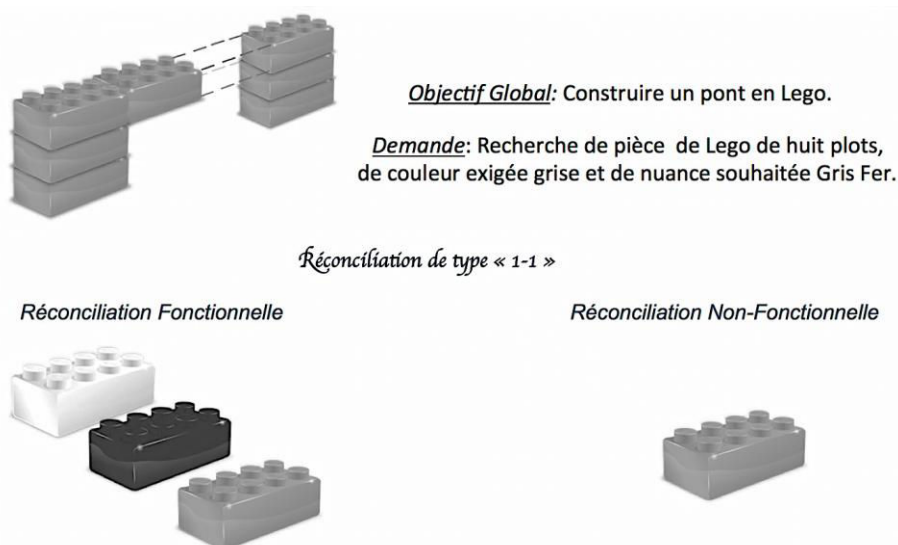


Figure 38 : Exemple d'une construction d'un pont en Lego : réconciliation « 1-1 ».

- *réconciliation non-fonctionnelle « 1-N »* (compositions de plusieurs pièces) : pour chacune des compositions fournies, nous comparons la couleur **et** la nuance de chacune de ces pièces par rapport à celles requises. Nous reprenons également l'exemple présenté dans la Figure 37 afin d'illustrer la réconciliation « 1-N » (cf. Figure 39). L'ensemble des pièces de Lego sélectionnées par la réconciliation fonctionnelle est constitué de trois compositions où chacune forme l'équivalent d'une pièce de huit plots. Nous remarquons que :
 - toutes les pièces de Lego utilisées dans chacune des compositions sont de couleur grise. Toutes les compositions répondent donc parfaitement à ce critère non-fonctionnel ;
 - l'absence d'une composition de pièces de nuance globale *gris Fer*. Les nuances de gris des pièces de chaque composition sont hétérogènes. Pour chacune des compositions, nous calculons le ratio « nuancePièce/nuanceRequise » de chacune de ces pièces. Par la suite, nous trions et nous classons les compositions selon leurs degrés de rapprochement de la nuance *gris Fer*.

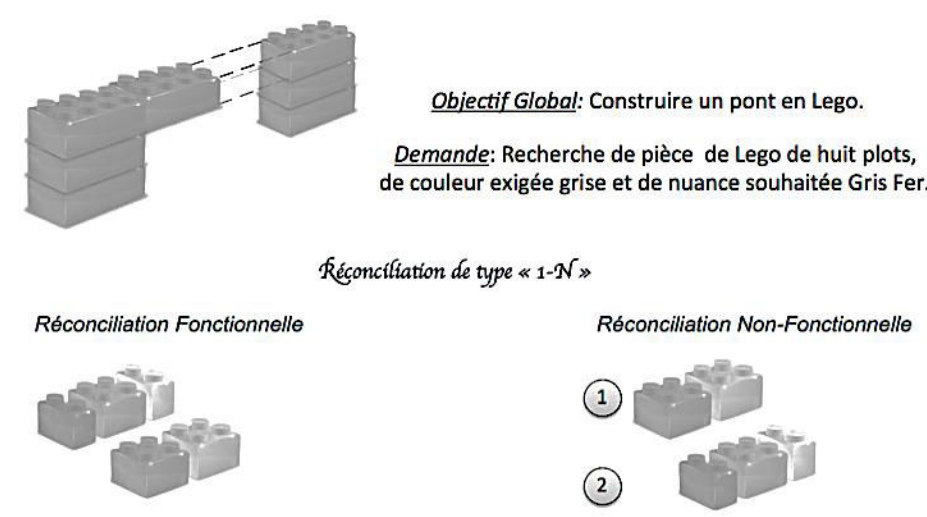


Figure 39 : Exemple d'une construction d'un pont en Lego : réconciliation « 1-N ».

Dans les sections qui suivent, nous détaillons les deux types de réconciliation non-fonctionnelle : « 1-1 » et « 1-N ».

IV.4. Réconciliation non-fonctionnelle de services de type « 1-1 »

Nous proposons dans cette section une approche pour illustrer la réconciliation non-fonctionnelle de sélection de services unitaires. Cette approche n'est nullement une nouvelle solution d'analyse multicritère. Il aurait probablement fallu en étudier plusieurs afin d'en utiliser une telle que

l'AHP (*Analytic Hierarchy Process*) [Saaty, 1988], ELECTRE (ELimination Et Choix Traduisant la REalité) [Roy, 1978], etc. Mais elles sont souvent trop figées et leur mise en œuvre est lourde pour être appliquée. Par ailleurs, nous n'avons ici pour objectif que de faire le lien entre les exigences non-fonctionnelles de l'activité métier et les propriétés non-fonctionnelles des services disponibles dans le registre de gouvernance, donc ceci n'est qu'à titre illustratif.

IV.4.1 Présentation générale de l'approche « 1-1 »

L'approche de réconciliation non-fonctionnelle de type « 1-1 » consiste à attribuer à chacune des activités du processus métier collaboratif, un service unitaire, parmi ceux qui satisfont les besoins fonctionnels, et qui répond aux exigences non-fonctionnelles (cf. Figure 40).

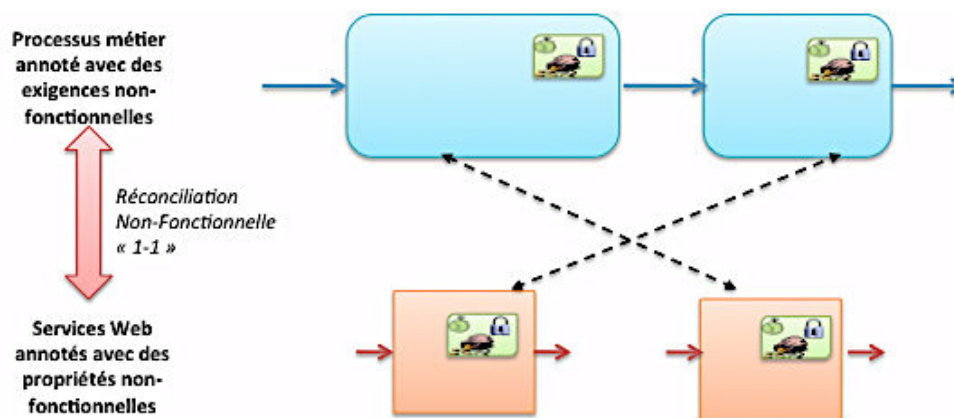


Figure 40 : Réconciliation non-fonctionnelle « 1-1 ».

IV.4.2 Algorithme de réconciliation non-fonctionnelle de type « 1-1 »

L'objectif de notre algorithme de réconciliation non-fonctionnelle est de fournir à l'utilisateur pour toute activité métier une liste triée de services Web (parmi ceux qui répondent à ses besoins fonctionnels que nous appelons services Web candidats) selon le degré de satisfaction par rapport aux exigences non-fonctionnelles.

Nous représentons les exigences non-fonctionnelles renseignées par le client pour toute activité métier, par un vecteur comme suit :

$$\text{Exigences_non-fonctionnelles} = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_n \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \end{pmatrix}$$

Où :

- P_1, P_2, \dots, P_n : l'ensemble des exigences non-fonctionnelles choisies par le client pour l'activité métier en question ;
- $\alpha_1, \alpha_2, \dots, \alpha_n$: l'ensemble des valeurs des exigences non-fonctionnelles, telles que : $\forall i \in [P_1, P_n], \alpha_i =$ la valeur de l'exigence non-fonctionnelle P_i .

Comme nous l'avons dit précédemment, nous offrons la possibilité aux clients de renseigner leurs préférences (ou pondérations) pour chacune des exigences non-fonctionnelles. Ces préférences sont également représentées par un vecteur comme suit :

$$\text{Préférences} = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_n \\ w_1 & w_2 & \dots & w_n \end{pmatrix}$$

Où :

- P_1, P_2, \dots, P_n : l'ensemble des exigences non-fonctionnelles choisies par le client pour l'activité métier en question ;
- $\omega_1, \omega_2, \dots, \omega_n$: l'ensemble des valeurs des préférences par rapport aux exigences non-fonctionnelles, telles que : $\forall i \in [P_1, P_n], \omega_i =$ la préférence de l'utilisateur par rapport à l'exigence non-fonctionnelle P_i , telle que : $\omega_i \in [0, 1]$ et $\sum_{i=1}^n (\omega_i) = 1$.

Et enfin, nous représentons l'ensemble des services Web candidats et les valeurs de leurs propriétés non-fonctionnelles choisies par l'utilisateur par une matrice comme suit :

$$\text{Services_Web_Candidats} = \begin{pmatrix} & \phi_1 & \phi_2 & \dots & \phi_n \\ S_1 & p_{11} & p_{21} & \dots & p_{n1} \\ S_2 & p_{12} & p_{22} & \dots & p_{n2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_m & p_{1m} & p_{2m} & \dots & p_{nm} \end{pmatrix}$$

Où :

- P_1, P_2, \dots, P_n : l'ensemble des exigences non-fonctionnelles choisies par le client pour l'activité métier en question ;
- S_1, S_2, \dots, S_m : l'ensemble des services Web candidats qui satisfont les exigences fonctionnelles de l'activité métier en question.

Notre algorithme de réconciliation non-fonctionnelle se déroule en quatre étapes [Zribi et al., 2013] : (i) classification des valeurs des exigences non-fonctionnelles, (ii) élimination des services Web candidats ayant des valeurs aberrantes, (iii) normalisation et centrage des services Web candidats par rapport aux exigences non-fonctionnelles de l'activité métier et enfin (iv), classement et sélection des services Web candidats. Nous détaillons toutes ces étapes dans les sections qui suivent.

IV.4.2.1. Étape 1 : Classification des valeurs des exigences non-fonctionnelles

La première étape de notre algorithme de réconciliation non-fonctionnelle est de classer les valeurs des exigences non-fonctionnelles selon leur type. Pour l'ensemble des propriétés non-fonctionnelles de WSQF, nous distinguons trois catégories de type de valeurs : le type *textuel*, le type *booléen*, et le type *numérique* qui regroupe : *pourcentage*, *temporel*, *monétaire*, *fréquence*, et *nombre*. Dans le Tableau 6, nous présentons les différents types de valeurs des propriétés non-fonctionnelles selon le standard WSQF.

Tableau 6 : Classification des valeurs des propriétés non-fonctionnelles selon leur type.

Textuel	Booléen	Numérique				
		Pourcentage	Temporel	Monétaire	Fréquence	Nombre
Message Reliability	Transaction Integrity	Availability	Business Performance	Price	Business Performance	Service Reputation
Controllability	Collaborability	Successability	Response Time	Penalty and Incentive	Maximum Throughput	Service Provider Reputation
	Informability	Accessibility				Service Recognition
	Encryption	Standard Conformability				
	Non Repudiation	Standard Adoptability				
	Authentication	Relative Proofness				
	Authorization	Observability				
	Availability					
	Audit					
	Integrity					
	Privacy					

Il est à noter que :

- nous ne considérons que les propriétés textuelles qui sont convertibles en nombre. Ces propriétés expriment un niveau d'exigence. Ce niveau peut être : très élevé, élevé, moyen, faible ou très faible. Dans ce qui suit, nous attribuons à chaque niveau une valeur. : pour le niveau très élevé, la valeur 1, 0.75 pour le niveau élevé, 0.5 pour le moyen, 0.5 pour faible et 0.05 pour le très faible ;

- les propriétés booléennes prennent deux états : soit « vrai » si valide, sinon « faux ». Dans ce qui suit, l'état vrai est représenté par « 1 » et l'état faux par « 0 » ;
- l'ensemble des propriétés de valeurs numériques peut être exprimé également par un *intervalle de valeurs numériques* (selon le choix de l'utilisateur).

Nous proposons aussi que l'ensemble des propriétés de valeurs numériques puisse être exprimé notamment par un *intervalle de valeurs numériques* (selon le choix de l'utilisateur).

Nous soulignons qu'une *propriété à maximiser de type booléen* signifie que si la valeur de la propriété est égale à « 1 » (c'est-à-dire la propriété est vraie) le maximum est atteint (par exemple le cryptage, l'authentification, etc.), contrairement à une *propriété à minimiser de type booléen* où c'est mieux quand elle est égale à « 0 ». Ce deuxième cas n'existe pas parmi les propriétés non-fonctionnelles que nous étudions, mais dans ces travaux nous souhaitons traiter l'ensemble des cas possibles.

IV.4.2.2. Étape 2 : Élimination des services candidats ayant des valeurs aberrantes

Notre objectif par cette réconciliation non-fonctionnelle est de fournir à l'utilisateur pour chacune des activités métier l'ensemble de services Web qui se rapprochent le plus de ses exigences non-fonctionnelles. Comme nous l'avons dit précédemment, nous partons d'un ensemble de services Web candidats (des services Web qui satisfont les besoins fonctionnels). Pour chacun de ces services, nous récupérons, depuis notre registre EasierGov-NFR, l'ensemble des valeurs de ses propriétés non-fonctionnelles (uniquement celles qui ont été choisies par l'utilisateur).

Cet ensemble de services Web candidats peut contenir des services Web qui satisfont suffisamment voire pleinement les exigences fonctionnelles de l'utilisateur, mais ont une ou plusieurs propriétés non-fonctionnelles qui s'éloignent totalement de ses exigences non-fonctionnelles. Les valeurs de ces propriétés non-fonctionnelles sont appelées des « valeurs aberrantes » et ces services Web ne font pas partie de ceux que nous souhaitons prendre en compte lors des calculs de réconciliation. Il convient alors de les éliminer.

Nous fournissons la possibilité aux utilisateurs d'introduire également leurs préférences par rapport à chacune des exigences non-fonctionnelles choisies. Lors de l'élimination des valeurs aberrantes, nous nous basons sur ces préférences pour déterminer le seuil de tolérance des valeurs non aberrantes (les seuils sont paramétrables dans le composant de réconciliation non-fonctionnelle) :

- $\text{Préférence} \geq 2 \times \frac{1}{\text{nombre total de propriétés choisies}}$: la préférence de l'utilisateur est d'un poids fort et le seuil de tolérance est égal à 25% ;
- $\text{Préférence} < \frac{1}{2} \times \frac{1}{\text{nombre total de propriétés choisies}}$: la préférence de l'utilisateur est d'un poids faible et le seuil de tolérance est égal à 40% ;

- Préférence non renseignée : le seuil de tolérance correspond au seuil par défaut et est égal à 30%.
- *Élimination des valeurs aberrantes dans le cas de propriété non-fonctionnelle de type numérique*

Une valeur d'une propriété non-fonctionnelle d'un service candidat est considérée comme aberrante si :

- elle est en dessous de la valeur de tolérance, dans le cas d'une propriété à maximiser ;
- elle est au dessus de la valeur de tolérance, dans le cas d'une propriété à minimiser.

Telle que : valeur de tolérance = seuil de tolérance \times valeur de la propriété dans l'activité métier

- *Élimination des valeurs aberrantes dans le cas de propriété non-fonctionnelle de type booléen ou textuel*

Une valeur d'une propriété non-fonctionnelle d'un service Web candidat est considérée comme aberrante si la préférence de l'utilisateur pour cette propriété est d'un poids fort et que la valeur de la propriété est différente de celle de l'activité métier.

- *Élimination des valeurs aberrantes dans le cas de propriété non-fonctionnelle de type intervalle de valeurs numériques*

Une valeur d'une propriété non-fonctionnelle d'un service Web candidat est considérée comme aberrante si :

- Dans le cas d'une propriété à maximiser, elle est en dessous de cette valeur :
seuil de tolérance \times valeur de la borne inférieure de la propriété dans l'activité métier
- Dans le cas d'une propriété à minimiser, elle est au dessus de cette valeur :
seuil de tolérance \times valeur de la borne supérieure de la propriété dans l'activité métier

IV.4.2.3. Étape 3 : Normalisation et centrage des services candidats par rapport aux exigences non-fonctionnelles de l'activité métier

Comme nous l'avons détaillé dans la première étape, les valeurs des propriétés non-fonctionnelles ne sont pas toutes de même type. En plus de cette différenciation de type des valeurs, ces propriétés n'ont pas la même dimension. Nous distinguons deux dimensions :

- Propriétés non-fonctionnelles « à maximiser » : plus la valeur de la propriété non-fonctionnelle d'un service Web est supérieure à la valeur de l'exigence non-fonctionnelle requise par l'utilisateur, mieux c'est. Par exemple, la disponibilité, les propriétés de la catégorie sécurité, la popularité du service Web;

- Propriétés non-fonctionnelles « à minimiser » : plus la valeur de la propriété non-fonctionnelle d'un service Web est inférieure à la valeur de l'exigence non-fonctionnelle requise par l'utilisateur, mieux c'est. À titre d'exemple, nous citons le temps de réponse, le prix.

Étant donné l'existence de toutes ces différenciations, les valeurs des propriétés non-fonctionnelles ne peuvent pas être comparées directement entre elles. Une étape de normalisation et de centrage devient nécessaire. Comme notre intérêt est d'être le plus proche possible des exigences de l'activité métier, nous proposons une normalisation et un centrage des valeurs des propriétés non-fonctionnelles des services Web candidats par rapport à l'activité métier. Le calcul des nouvelles valeurs normalisées et centrées des services Web candidats dépend du type de la propriété : *numérique ou textuel* (cf. Algorithme 1) ou *booléen* (cf. Algorithme 2) ou *intervalle de valeurs numériques* (cf. Algorithme 3).

L'algorithme 1 détaille les étapes de normalisation et de centrage pour les propriétés non-fonctionnelles de type *numérique* ou *textuel*.

Il prend en entrée l'ensemble des valeurs de l'activité métier (le vecteur *Exigences_non-fonctionnelles*) « E » et l'ensemble des services Web candidats ainsi que les valeurs de leurs propriétés non-fonctionnelles choisies lors de l'annotation (la matrice *Services_Web_Candidats*) « SC ». Pour chacune des valeurs de la matrice *Services_Web_Candidats*, si la propriété en question est une *propriété à maximiser* nous divisons par la valeur de la propriété correspondante dans le vecteur *Exigences_non-fonctionnelles*, puis nous soustrayons 1 au résultat de chacune (ligne 11 de l'algorithme 1). Sinon (c'est-à-dire si la propriété est une *propriété à minimiser*), nous divisons par la valeur de la propriété correspondante dans le vecteur *Exigences_non-fonctionnelles*. Puis, nous soustrayons le résultat de chacune de « 1 » (ligne 9 de l'algorithme 1). En fait, nous souhaitons que les valeurs normalisées des propriétés non-fonctionnelles des services normalisés et centrés soient positives si elles sont meilleures que les exigences demandées, nulles si identiques et négatives quand elles sont moins bonnes.

Toutefois, la valeur de l'exigence non-fonctionnelle de l'activité métier peut être égale à zéro. Dans ce cas particulier, on ne peut pas normaliser par rapport à la valeur de l'exigence. Comme nous souhaitons harmoniser les valeurs des services pour chacune des propriétés et les ramener entre 0 et 1, nous proposons donc de diviser par le plus grand du vecteur colonne de la matrice *Services_Web_Candidats* de la propriété en question (les lignes de 12 au 17 de l'algorithme 1).

Algorithme 1 : Normalisation et centrage des valeurs des propriétés des services candidats de type numérique

1 $E \leftarrow$ l'ensemble des exigences non-fonctionnelles de l'activité métier en question (le vecteur

2 *Exigences_non-fonctionnelles*)

3 $SC \leftarrow$ l'ensemble des services Web candidats et les valeurs de leurs propriétés non-

4 fonctionnelles choisies par l'utilisateur (la matrice *Services_Web_Candidats*)

5 **Début**

6 **Pour** i allant de 1 à n **Faire** :

7 **Pour** j allant de 1 à m **Faire** :

8 **Pour** chaque SC $[P_i][S_j]$ **Faire** :

9 **Si** $E [P_i] \neq 0$ **Alors** :

10 **Si** P_i est une propriété à maximiser **Alors** :

11 $Services_Web_Candidats_normalisés_centrés [P_i][S_j] = \frac{SC [P_i][S_j]}{E[P_i]} - 1$

12 **Sinon**

13 $Services_Web_Candidats_normalisés_centrés [P_i][S_j] = 1 - \frac{SC [P_i][S_j]}{E[P_i]}$

14 **Fin Si.**

15 **Sinon**

16 $Max_Colonne [S_j] \leftarrow$ La valeur maximale du vecteur colonne $[S_j]$ de la matrice

17 *Services_Web_Candidats* $[P_i] [S_j]$ correspondant à la propriété P_i

18 **Si** P_i est une propriété à maximiser **Alors** :

19 $Services_Web_Candidats_normalisés_centrés [P_i][S_j] = \frac{SC [P_i][S_j]}{Max_Colonne[S_j]} - 1$

20 **Sinon**

21 $Services_Web_Candidats_normalisés_centrés [P_i][S_j] = 1 - \frac{SC [P_i][S_j]}{Max_Colonne[S_j]}$

22 **Fin Si.**

23 **Fin Si.**

24 **Fin Pour.**

25 **Fin Pour.**

26 **Fin Pour.**

27 **Fin.**

L'algorithme 2 s'intéresse aux propriétés non-fonctionnelles de type *booléen*.

Il prend en entrée l'ensemble des services Web candidats ainsi que les valeurs des propriétés non-fonctionnelles choisies lors de l'annotation (la matrice *Services_Web_Candidats*) « SC ».

Pour un booléen (par définition 0 ou 1), il n'est pas nécessaire d'effectuer une normalisation, mais uniquement d'effectuer un centrage par rapport à la valeur de la propriété en question.

Pour chacune des valeurs de la matrice *Services_Web_Candidats* et afin que la valeur normalisée et centrée soit positive si elle est mieux et négative si elle est moins bonne que l'exigence métier, il est nécessaire d'effectuer l'étape suivante :

- Dans le cas d'une *propriété à maximiser* : nous soustrayons (ligne 9 de l'algorithme 2) ;
- Dans le cas d'une *propriété à minimiser* : nous soustrayons cette valeur de « 1 » (ligne 11 de l'algorithme 2).

Algorithme 2 : Centrage des valeurs des propriétés des services candidats de type booléen

```

1  $E \leftarrow$  l'ensemble des exigences non-fonctionnelles de l'activité métier en question (le vecteur
2   Exigences_non-fonctionnelles)
3  $SC \leftarrow$  l'ensemble des services Web candidats et les valeurs de leurs propriétés non-fonctionnelles
   choisies par l'utilisateur (la matrice Services_Web_Candidats)
4 Début
5   Pour  $i$  allant de 1 à  $n$  Faire :
6     Pour  $j$  allant de 1 à  $m$  Faire :
7       Pour chaque SC  $[P_i][S_j]$  Faire :
8         Si  $P_i$  est une propriété à maximiser Alors :
9            $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = SC [P_i][S_j] - 1$ 
10        Sinon
11           $Services\_Web\_Candidats\_normalisés\_centrés [P_i][S_j] = 1 - SC [P_i][S_j]$ 
12        Fin Si.
13      Fin Pour.
14    Fin Pour.
15  Fin Pour.
16 Fin.

```

L'algorithme 3 détaille les étapes de normalisation et de centrage pour les propriétés non-fonctionnelles dans le cas où l'utilisateur a choisi d'introduire un *intervalle de valeurs numériques*.

Dans ce cas, chacune des valeurs du vecteur *Exigences_non-fonctionnelles* est un intervalle de valeurs numériques et non une valeur simple comme dans les cas précédents. Le vecteur devient comme suit :

$$Exigences_non-fonctionnelles = \begin{pmatrix} \mathcal{P}_1 & \mathcal{P}_2 & \dots & \mathcal{P}_n \\ [\alpha_{1\ inf}, \alpha_{2\ sup}] & [\alpha_{2\ inf}, \alpha_{2\ sup}] & \dots & [\alpha_{n\ inf}, \alpha_{n\ sup}] \end{pmatrix}$$

Cet algorithme prend en entrée l'ensemble des intervalles de valeurs numériques de l'activité métier (le vecteur *Exigences_non-fonctionnelles*) « *E* » et l'ensemble des services Web candidats ainsi que les valeurs des propriétés non-fonctionnelles choisies lors de l'annotation (la matrice *Services_Web_Candidats*) « *SC* ».

Pour chacune des valeurs de la matrice *Services_Web_Candidats*, nous distinguons trois cas possibles :

- Elle fait partie de l'intervalle de valeurs requises par l'utilisateur : elle répond à l'exigence de l'activité métier pour la propriété en question. Toutefois, il est nécessaire de la situer par rapport aux bornes de l'intervalle demandé pour savoir de quel côté elle se rapproche. Nous distinguons deux cas possibles :
 - si la propriété est une propriété à maximiser : dans ce cas plus la valeur se rapproche de la borne supérieure, mieux c'est. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 11 ;
 - si la propriété est une propriété à minimiser : dans ce cas plus la valeur se rapproche de la borne inférieure, mieux c'est. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 13 ;
- Elle est strictement supérieure à la valeur de la borne supérieure de l'intervalle :
 - si la propriété est une propriété à maximiser : dans ce cas la valeur de la propriété du service candidat est meilleure que celle demandée par l'utilisateur. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 18 ;
 - si la propriété est une propriété à minimiser : dans ce cas la valeur de la propriété du service candidat est moins bonne que celle demandée par l'utilisateur. Pour la

normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 20 ;

- Elle est strictement inférieure à la valeur de la borne inférieure de l'intervalle :
 - si la propriété est une propriété à maximiser : dans ce cas la valeur de la propriété du service candidat est moins bonne que celle demandée par l'utilisateur. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 24 ;
 - si la propriété est une propriété à minimiser : dans ce cas la valeur de la propriété du service candidat est meilleure que celle demandée par l'utilisateur. Pour la normalisation et le centrage de cette valeur par rapport à l'intervalle de l'activité métier, nous appliquons la formule présentée dans l'algorithme 3 à la ligne 26.

Algorithme 3 : Normalisation et centrage des valeurs des propriétés des services candidats dans le cas d'un intervalle de valeurs numériques

```

1 E ← l'ensemble des exigences non-fonctionnelles de l'activité métier en question (le vecteur Exigences_non-
2   fonctionnelles)
3 SC ← l'ensemble des services Web candidats et les valeurs de leurs propriétés non-fonctionnelles choisies par
4   l'utilisateur (la matrice Services_Web_Candidats)
5 Début
6 Pour i allant de 1 à n Faire :
7   Pour j allant de 1 à m Faire :
8     Pour chaque S [Pi][Sj] Faire :
9       Si SC [Pi][Sj] ∈ E[Piinf; Pisup] Alors :
10        Si Pi est une propriété à maximiser Alors
11          Services_Web_Candidats_normalisés_centrés [Pi][Sj] =  $\frac{SC [P_i][S_{ij}] - E[P_i \text{ inf}]}{E [P_i \text{ sup}] - E[P_i \text{ inf}]}$ 
12        Sinon
13          Services_Web_Candidats_normalisés_centrés [Pi][Sj] =  $\frac{E[P_i \text{ inf}] - SC [P_i][S_{ij}]}{E [P_i \text{ sup}] - E[P_i \text{ inf}]}$ 
14        Fin Si.
15      Sinon
16        Si SC [Pi][Sj] < Eiinf
17          Si Pi est une propriété à maximiser Alors :
18            Services_Web_Candidats_normalisés_centrés [Pi][Sj] =  $\frac{SC [P_i][S_{ij}]}{E [P_i \text{ inf}]} - 1$ 
19          Sinon
20            Services_Web_Candidats_normalisés_centrés [Pi][Sj] =  $1 - \frac{SC [P_i][S_{ij}]}{E [P_i \text{ inf}]}$ 
21          Fin Si.
22        Sinon
23          Si Pi est une propriété à maximiser Alors :
24            Services_Web_Candidats_normalisés_centrés [Pi][Sj] =  $-\frac{SC [P_i][S_{ij}]}{E [P_i \text{ sup}]} - 1$ 
25          Sinon
26            Services_Web_Candidats_normalisés_centrés [Pi][Sj] =  $1 - \left(-\frac{SC [P_i][S_{ij}]}{E [P_i \text{ sup}]}\right)$ 
27          Fin Si.
28        Fin Si.
29      Fin Si.
30    Fin Pour.
31  Fin Pour.
32 Fin Pour.
33 Fin.

```

IV.4.2.4. Étape 4 : Classement et sélection des services candidats

À l'issue de l'étape 3 de l'algorithme de réconciliation non-fonctionnelle, nous avons normalisé et centré toutes les valeurs des services Web candidats (hormis ceux qui contiennent au moins une valeur aberrante) autour des exigences de l'activité métier. En effet, chaque valeur de la matrice *Services_Web_Candidats* à l'issue de l'étape de normalisation et de centrage peut être soit :

- nulle : ceci signifie que la valeur de la propriété du service Web candidat correspond exactement à la valeur de la propriété de l'activité métier ;
- strictement positive : ceci signifie que la valeur de la propriété du service Web candidat est meilleure que la valeur de la propriété de l'activité métier ;
- strictement négative : ceci signifie que la valeur de la propriété du service Web candidat est moins bonne que la valeur de la propriété de l'activité métier.

Dans l'étape 4, nous proposons de diviser les services candidats en deux zones (cf. Figure 41) :

- Zone A : cette zone contient les services candidats qui ont des valeurs positives ou nulles pour toutes leurs propriétés non-fonctionnelles. Autrement dit, les services qui sont meilleurs que l'activité métier sur toutes les exigences non-fonctionnelles.
- Zone B : cette zone contient les services candidats qui ont au moins une propriété non-fonctionnelle de valeur négative pour toutes leurs propriétés non-fonctionnelles.

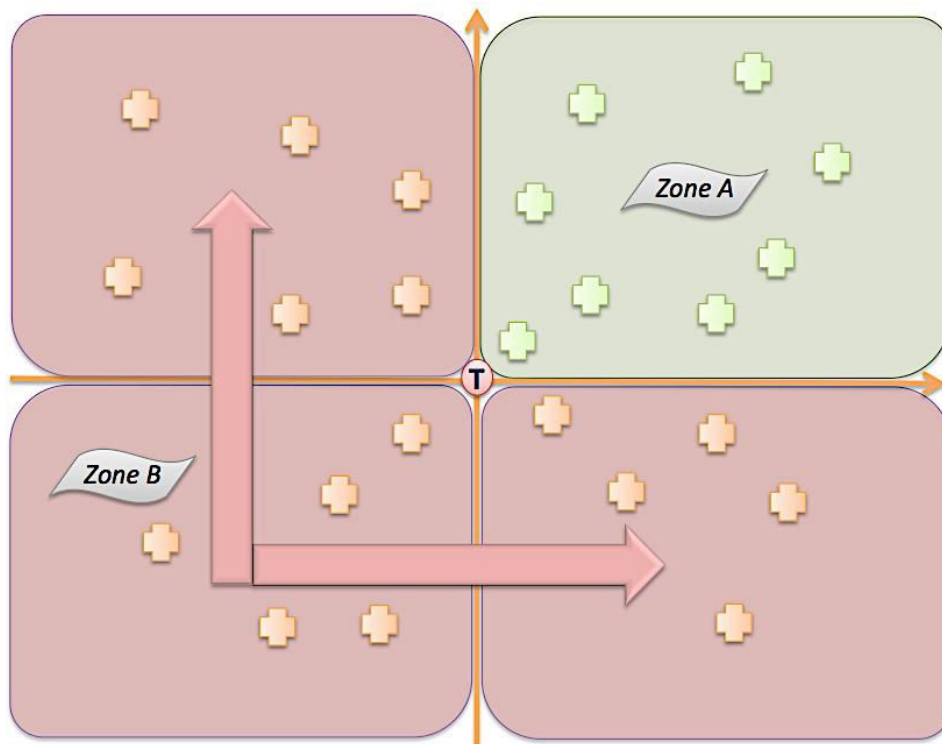


Figure 41 : Classement des services candidats dans les zones A et B.

Pour le classement et la sélection des services candidats, il est évident de commencer par classer ceux de la zone A, puis de classer à la suite les services appartenant à la zone B.

- *Dans le cas de la zone A* : classement des services Web candidats revient à calculer la distance pondérée par rapport à l'activité métier⁵. Puis les classer entre elles, le rang 1 revient à celui qui a la distance la plus grande. L'algorithme 4 détaille cette partie.
- *Dans le cas de la zone B* : cette sous-matrice comprend des services Web qui ont au moins une valeur (négative) moins bonne que celle demandée. Dans un premier temps, nous souhaitons étudier les services uniquement selon leurs valeurs défailtantes afin d'éliminer le cas où une de leurs valeurs positives récompenserait leur négativité. Pour cela, nous mettons donc à « 0 » toutes les valeurs positives des services. Tous les services vont être donc placés uniquement dans la partie négative de la zone B (cf. Figure 42).

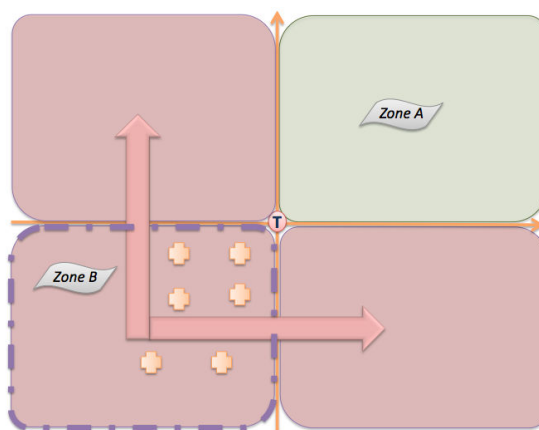


Figure 42 : Classement des services candidats de la zone B : analyse selon les valeurs négatives.

Par la suite, nous calculons la distance pondérée de chacun par rapport à l'activité métier. Le service Web candidat qui a la plus petite distance se classe en premier à la suite des services Web de la zone A.

En cas d'ex-æquo, c'est-à-dire lorsque les services Web ont exactement les mêmes moins bonnes valeurs, nous classons ces services entre eux selon les valeurs positives ou nulles initiales (les valeurs normalisées et centrées). Nous calculons la distance pondérée de ces services Web en utilisant uniquement ces valeurs. Le premier parmi eux est celui qui a la plus grande distance pondérée. L'algorithme 4 détaille cette partie.

⁵ Dans notre cas, l'activité métier se trouve à l'origine du repère dans de l'activité métier à l'issue de l'étape de normalisation et de centrage sont toutes nulles comme nous l'avons vu à l'étape précédente.

Algorithme 4 : Classement et sélection des services candidats appartenant à la zone A

1 $E_{n_et_c}$ ← l'ensemble des exigences non-fonctionnelles de l'activité métier en question après la phase de normalisation et de
2 centrage (le vecteur *Exigences_non-fonctionnelles* normalisé et centré)

3 PF ← l'ensemble des préférences (poids) attribuées par l'utilisateur à chacune des activités métier (le vecteur *Préférences*)

4 $SCA_{n_et_c}$ ← l'ensemble des services Web candidats de la zone A et les valeurs de leurs propriétés non-fonctionnelles à 56
5 l'issue de la phase de normalisation et de centrage

6 **Début**

7 **Pour** i allant de 1 à n **Faire** :

8 **Pour** j allant de 1 à m **Faire** :

9 **Pour** chaque SCA $[P_i][S_j]$ **Faire** :

10 Distance_Pondérée $[S_j] = \sqrt{\sum_{i=1}^n ((SCA_{n_et_c}[P_i][S_j] - E_{n_et_c}[P_i])^2) \times PF[P_i]}$

11 **Fin Pour.**

12 **Fin Pour.**

13 **Fin Pour.**

14 Classement des services selon la leur valeur de distance pondérée. Le premier service est celui qui a la plus grande
15 distance.

16 **Fin.**

Algorithme 5 : Classement et sélection des services candidats appartenant à la zone B

1 $E_{n_et_c}$ ← l'ensemble des exigences non-fonctionnelles de l'activité métier en question après la phase de normalisation et de
2 centrage (le vecteur *Exigences_non-fonctionnelles* normalisé et centré)

3 PF ← l'ensemble des préférences (poids) attribuées par l'utilisateur à chacune des activités métier (le vecteur *Préférences*)

4 $SCB_{n_et_c}$ ← l'ensemble des services Web candidats de la zone B et les valeurs de leurs propriétés non-fonctionnelles à
5 l'issue de la phase de normalisation et de centrage

6 **Début**

7 **Pour** i allant de 1 à n **Faire** :

8 **Pour** j allant de 1 à m **Faire** :

9 **Pour** chaque SCB $[P_i][S_j]$ **Faire** :

10 **Si** $SCB [P_i][S_j] > 0$ **Alors** :

11 $SCB [P_i][S_j] = 0$

12 **Fin Si.**

13 **Fin Pour.**

14 **Pour** chaque SCB $[P_i][S_j]$ **Faire** :

15 Distance_Pondérée $[S_j] = \sqrt{\sum_{i=1}^n ((SCA_{n_et_c}[P_i][S_j] - E_{n_et_c}[P_i])^2) \times PF[P_i]}$

16 **Fin Pour.**

17 Classement des services selon leur valeur de distance pondérée. Le premier service est celui qui a la plus petite
18 distance.

- 19 En cas d'ex-æquo de deux ou plusieurs services, nous recalculons leur distance pondérée en nous basant
20 uniquement sur leurs valeurs positives. Ils sont classés entre eux dans le sens décroissant (le premier est celui
21 qui a la plus grande distance).
- 22 **Fin Pour.**
- 23 **Fin Pour.**
- 24 **Fin.**

IV.5. Réconciliation non-fonctionnelle de services de type « 1-N »

Cette section détaille notre approche de réconciliation non-fonctionnelle de services Web de type «1-N». Nous commençons, tout d'abord, par présenter le principe général de notre approche pour détailler par la suite les algorithmes de réconciliation de compositions de services Web « 1-N ».

IV.5.1. Présentation générale de l'approche « 1-N »

La réconciliation non-fonctionnelle de type « 1-N » est appelée aussi réconciliation non-fonctionnelle de compositions de services. Elle consiste à attribuer à chacune des activités du processus métier collaboratif, une composition de services, parmi celles qui satisfont les besoins fonctionnels, qui répond à ses exigences non-fonctionnelles (cf. Figure 43).

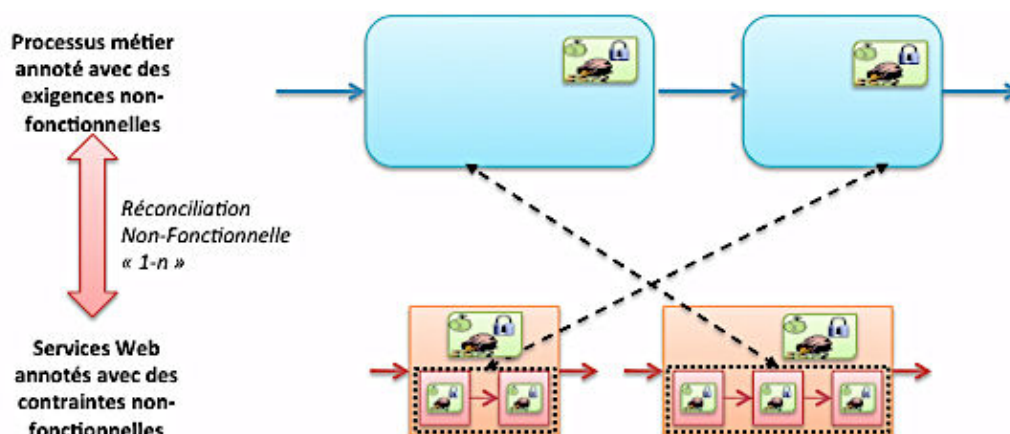


Figure 43 : Réconciliation non-fonctionnelle « 1-N ».

Nous appelons ces compositions de services « *compositions de services Web candidates* » et nous les représentons par une matrice contenant pour chacune des compositions : les valeurs des propriétés non-fonctionnelles (choisies par l'utilisateur) pour chacun de ses services Web.

$$\text{Compositions_Services_Web_Candidates} = \begin{pmatrix} & \phi_1 & \phi_2 & \dots & \phi_n \\ \text{CS}_1 & \begin{matrix} S_1 \\ S_2 \\ S_3 \end{matrix} & \begin{matrix} P_{11} \\ P_{12} \\ P_{13} \end{matrix} & \begin{matrix} P_{21} \\ P_{22} \\ P_{23} \end{matrix} & \dots & \begin{matrix} P_{n1} \\ P_{n2} \\ P_{n3} \end{matrix} \\ \text{CS}_2 & \begin{matrix} S_1 \\ S_4 \end{matrix} & \begin{matrix} P_{11} \\ P_{14} \end{matrix} & \begin{matrix} P_{21} \\ P_{24} \end{matrix} & \dots & \begin{matrix} P_{n3} \\ P_{n4} \end{matrix} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \text{CS}_m & \begin{matrix} S_x \\ S_y \end{matrix} & \begin{matrix} P_{1x} \\ P_{1y} \end{matrix} & \begin{matrix} P_{2x} \\ P_{2y} \end{matrix} & \dots & \begin{matrix} P_{nx} \\ P_{ny} \end{matrix} \end{pmatrix}$$

Comme nous le remarquons, l'approche de réconciliation non-fonctionnelle « 1-N » ressemble fortement à celle de la réconciliation non-fonctionnelle de type « 1-1 ». La seule différence entre ces deux est : dans la première, nous avons une matrice de compositions de services Web candidates et dans la deuxième, nous avons une matrice de services Web candidats. Toutefois, malgré cette différenciation, la forte ressemblance nous a menés à poser aux interrogations suivantes :

Pouvons nous transformer une problématique de réconciliation de type « 1-N » en une problématique de type « 1-1 » ? Si oui, comment faire ?

Cette transformation peut être possible si nous essayons de regrouper les services qui constituent la composition et de les considérer comme un seul grand service dont chacune de ses propriétés non-fonctionnelles est calculée en fonction de toutes les valeurs des services pour la propriété en question (cf. Figure 44)

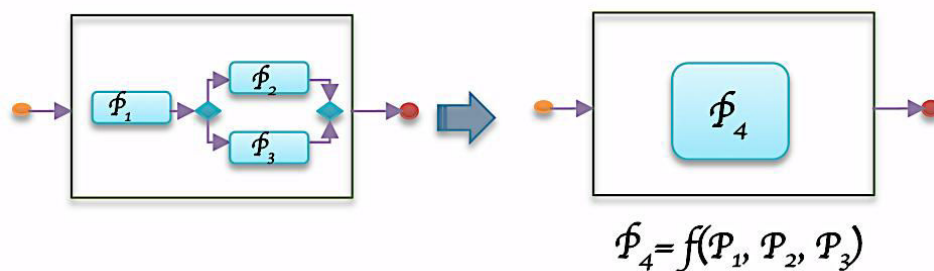


Figure 44 : Proposition de transformation d'une approche « 1-N » vers une approche « 1-1 ».

Dans ce qui suit, nous détaillons notre proposition de transformation d'une problématique de réconciliation non-fonctionnelle de compositions de services en une de sélection de services.

IV.5.2. Transformation d'une réconciliation non-fonctionnelle de type « 1-N » en une de type « 1-1 »

Le passage, d'une approche de réconciliation non-fonctionnelle de services Web en une approche de réconciliation non-fonctionnelle « 1-N », nécessite certaines transformations, comme l'illustre la Figure 45:

- \forall composition de services Web de la matrice *Compositions_Services_Web_Candidates* (CS_m) est un nouveau service Web (S_m) ;
- la valeur de chaque propriété de ce nouveau service Web (P_{nm}) doit être obtenue à partir de l'ensemble des valeurs de tous les services de la composition pour la propriété en question (elle est équivalente à la valeur de la propriété fictive de l'ensemble des services).

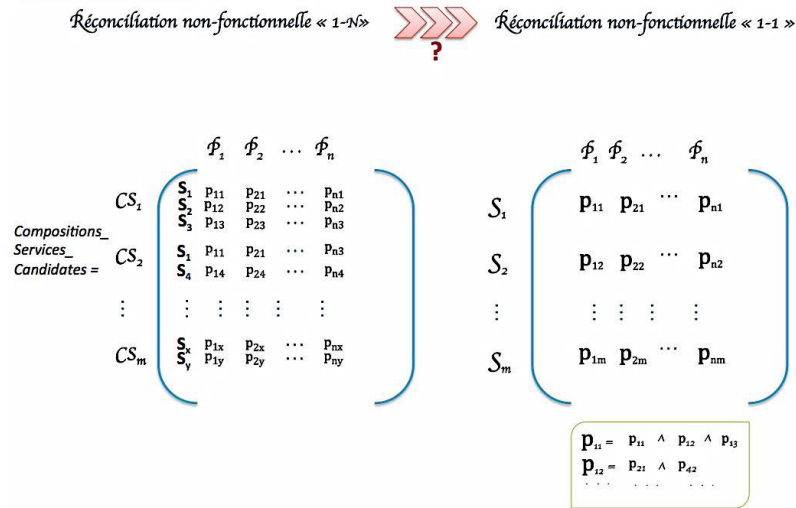


Figure 45 : Problématique de passage d'une réconciliation « 1-N » vers une de type « 1-1 ».

La résolution de cette problématique revient à déterminer les nouvelles valeurs de chaque propriété non-fonctionnelle pour chacun de ces nouveaux grands services. Pour cela, nous proposons deux étapes à suivre :

IV.5.2.1. Étape 1 : Décomposition en plusieurs sous-compositions de deux services

Les compositions de services peuvent contenir un nombre assez important de services. Afin d'analyser le déroulement de la composition en détails, nous proposons de la décomposer en une suite successives de sous-compositions qui peuvent encore se décomposer jusqu'à ce qu'obtenir dans chacune au plus deux services. Dans la Figure 46 ci-après, nous présentons un exemple de décomposition d'une composition de services.

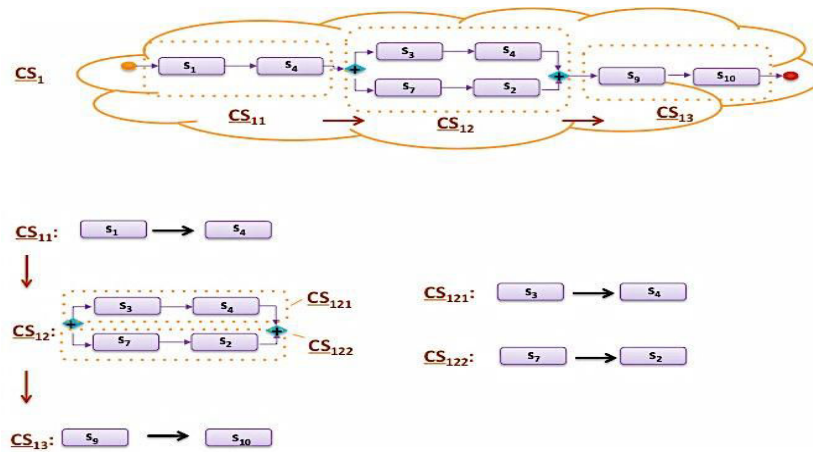


Figure 46 : Exemple de décomposition en sous-compositions de couple de services.

Nous avons commencé par décomposer la composition de services (CS_1) en trois sous-compositions (CS_{11}), (CS_{12}), et (CS_{13}). Ces trois sous-compositions de services s'exécutent séquentiellement.

- la sous-composition CS_{11} contient deux services.
- la sous-composition CS_{12} contient quatre services. Nous refaisons une autre décomposition et nous obtenons les deux sous compositions (CS_{121}) et (CS_{122}) qui s'exécutent parallèlement. Chacune des ces deux sous-compositions contient deux services. Il n'y plus donc besoin de refaire d'autres décompositions.
- la sous-composition CS_{13} contient deux services.

IV.5.2.2. Étape 2 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition

Pour calculer ces nouvelles valeurs, il faudrait prendre en considération les deux facteurs suivants : (i) le type de la propriété non-fonctionnelle (pourcentage, fréquence, temporel, monétaire ou booléen) et (ii) le type de branchement (parallèle, séquentiel, exclusif ou inclusif).

❖ Exécution des deux services

Dans le cas d'un branchement en parallèle (\diamond) (cf. Figure 47) ou d'un branchement séquentiel (\rightarrow) (cf. Figure 48), les services (S_1 et S_2) sont *exécutés tous les deux*. Par conséquent, leurs propriétés non-fonctionnelles sont *obligatoirement mises en œuvre*.

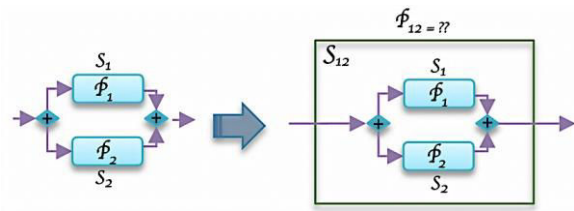


Figure 47 : Branchement en parallèle.

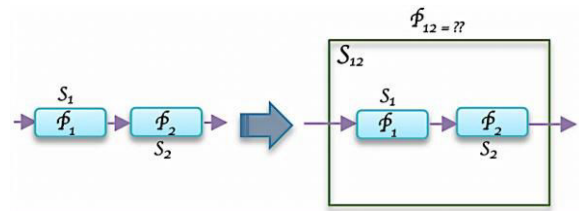




Figure 48 : Branchement séquentiel.

Pour chacune des propriétés non-fonctionnelles, la valeur globale de la propriété (P_{12}) qui correspond à l'ensemble de la sous-composition (S_{12}) est calculée comme suit :



- Temporel : dans le cas d'un branchement parallèle, étant donné que les deux services doivent s'exécuter en même temps, la valeur contraignante pour l'ensemble de la sous-composition correspond donc à la plus grande valeur des deux services. Le cas d'un branchement séquentiel où les deux services s'exécutent l'un à la suite de l'autre, il est évident que la valeur globale doit être égale à la somme des deux.
- Booléen : comme les deux services sont mis en œuvre ensemble, que ce soit pour un branchement en parallèle ou séquentiel, la validité de l'ensemble de la sous-composition ne peut être assurée que si les deux services sont valides. Dans le cas échant, l'état de la composition est donc « faux ».
- Monétaire : du fait de l'exécution simultanée des deux services, il est évident que les deux valeurs des services sont prises en compte. La valeur monétaire de l'ensemble est égale donc à la somme des valeurs des deux services.
- Pourcentage : étant donné que les deux services doivent être activés en même temps, le pourcentage de l'ensemble de la sous-composition (que ce qui pour le branchement parallèle ou séquentiel) est équivalent au produit des pourcentages des deux services.
- Fréquence : comme les services sont portés à être activés tous les deux, la fréquence de traitement de la sous-composition globale est équivalente au minimum des fréquences des deux services. Prenons l'exemple de la propriété débit maximum : nous avons deux services S_1 et S_2 fonctionnant séquentiellement (ou parallèlement). Le premier peut fabriquer 5 bouteilles par seconde et le deuxième 10 bouchons par secondes. Le nombre de bouteilles avec bouchons par seconde issu de la sous-composition de S_1 et S_2 est donc de 5.

Le tableau suivant, Tableau 7, synthétise le calcul de la valeur de la propriété non-fonctionnelle d'une sous-composition de services dans le cas d'un branchement en parallèle et d'un branchement en série (séquentiel).

Tableau 7 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas des branchements en parallèle et séquentiel.

	Branchement parallèle 	Branchement séquentiel 
Pourcentage	Produit (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})
Fréquence	Minimum (P_{S1}, P_{S2})	Minimum (P_{S1}, P_{S2})
Valeur Temporelle	Maximum (P_{S1}, P_{S2})	Somme (P_{S1}, P_{S2})
Valeur Booléenne	Produit (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})
Valeur Monétaire	Somme (P_{S1}, P_{S2})	Somme (P_{S1}, P_{S2})

❖ *Exécution d'un seul service*

Dans le cas d'un branchement exclusif ( ou ) (cf. Figure 49), le flux de séquence est passé exclusivement à un seul branchement. Autrement dit, *uniquement un service* (S_1 ou S_2) qui s'exécute obligatoirement. Ce service correspond à celui qui vérifie la condition. Ceci signifie que seules les propriétés non-fonctionnelles d'un service qui s'exécute sont mises en œuvre.

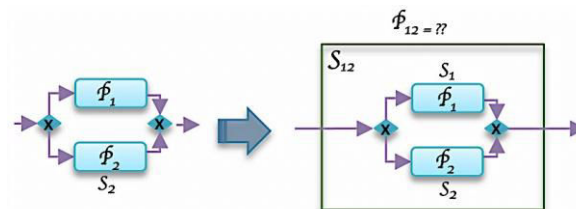




Figure 49 : Branchement exclusif.

Toutefois, nous ne pouvons connaître par avance lequel des deux services qui s'exécutera. Par conséquent, nous considérons, pour chacune des propriétés non-fonctionnelles, que la valeur de la caractéristique non-fonctionnelle globale (P_{12}) qui correspond l'ensemble de la sous-composition (S_{12}) est celle de la plus contraignante. Dans le cas où c'est une propriété à maximiser, la valeur contraignante est la plus petite des deux. Sinon (c'est-à-dire une propriété à minimiser), elle est égale à la plus grande des deux. Nous présentons dans le Tableau 8 ci-dessous la valeur de la propriété non-fonctionnelle globale de la sous-composition de services pour chacun des types des valeurs.





Tableau 8 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas d'un branchement exclusif.

	Branchement exclusif	
	 ou 	
	Propriété à maximiser	Propriété à minimiser
Pourcentage	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})
Fréquence	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})
Valeur Temporelle	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})
Valeur Booléenne	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})
Valeur Monétaire	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})

❖ *Exécution d'un seul ou des deux services*

Ce type d'exécution correspond à un branchement inclusif. Lors d'une division un (S1 ou S2) ou les deux services (S1 et S2) *sont activés*. Nous pouvons voir le branchement inclusif comme une combinaison des deux branchements parallèle et exclusif. Par ailleurs, dans ce type de branchement, on ne connaît pas par avance combien de services (et si c'est un seul lequel des deux) vont être exécutés, le calcul de la valeur de la propriété non-fonctionnelle relative à l'ensemble de la sous-composition correspond donc au pire des deux cas (entre les valeurs issues du branchement en parallèle et celles du branchement exclusif). Dans le Tableau 9 ci-dessous, nous présentons, pour chaque type de propriétés non-fonctionnelles, la valeur de la propriété globale (P12) qui correspond à l'ensemble de la sous-composition (S12).

Tableau 9 : Calcul de la valeur de la propriété non-fonctionnelle de l'ensemble de la sous-composition dans le cas d'un branchement inclusif.

	Branchement exclusif		Branchement parallèle	Branchement inclusif	
	 ou 				
	Propriété à maximiser	Propriété à minimiser		Propriété à maximiser	Propriété à minimiser
Pourcentage	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})
Fréquence	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})	Minimum (P_{S1}, P_{S2})	Minimum (P_{S1}, P_{S2})	Minimum (P_{S1}, P_{S2})
Valeur Temporelle	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})
Valeur Booléenne	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})	Produit (P_{S1}, P_{S2})
Valeur Monétaire	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})	Somme (P_{S1}, P_{S2})	Minimum (P_{S1}, P_{S2})	Maximum (P_{S1}, P_{S2})

Après avoir définis les règles de passage d'une problématique de réconciliation non-fonctionnelle de type « 1-N » en une de type « 1-1 », nous proposons ci-après une illustration de leur application. Nous reprenons l'exemple de la décomposition faite dans la Figure 46, et nous présentons (cf. Figure 50) le calcul de la propriété temps de réponse correspondante à la totalité de la composition de services (CS_1) dont la matrice est la suivante :

Réconciliation non-fonctionnelle « 1-N »

Temps de réponse (secondé)

$$CS_1 = \begin{pmatrix} s_1 & 3 \\ s_4 & 4 \\ s_3 & 2 \\ s_7 & 7 \\ s_2 & 5 \\ s_9 & 6 \\ s_{10} & 9 \end{pmatrix}$$

Nous détaillons dans la figure ci-dessous (cf. Figure 50), toutes les règles appliquées à chacune des sous-compositions pour le calcul du temps de réponse de la composition de services (CS_1).

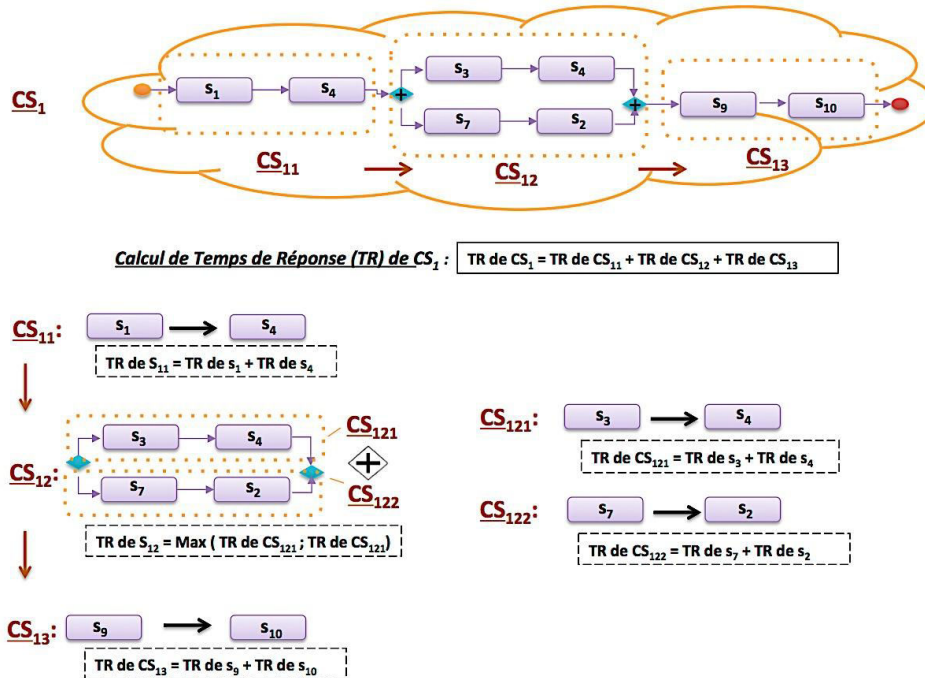


Figure 50 : Calcul du temps de réponse global d'une composition de services Web.

À partir de la matrice initiale contenant la valeur de la propriété temps de réponse pour chacun des services qui composent CS_1 et après avoir appliqué toutes les règles de calcul nous obtenons le résultat de calcul final suivant :

$$\begin{array}{c} \text{Réconciliation non-fonctionnelle « 1-1 »} \\ \text{Temps de réponse global (secondé)} \\ CS_1 = \left(\begin{array}{c} 34 \end{array} \right) \end{array}$$

Comme nous pouvons l'observer, nous obtenons une matrice de compositions de services candidates équivalente à celle de services candidats dans la réconciliation « 1-1 » (une seule valeur pour chaque candidat). Nous avons donc réussi à faire la transformation de la problématique « 1-N » en « 1-1 » et nous pouvons par la suite appliquer l'approche de réconciliation non-fonctionnelle « 1-1 » présentée dans la Section IV.4 de ce chapitre ou tout autre algorithme d'analyse multicritère de sélection de services.

IV.6. Conclusion

Dans ce chapitre, nous avons présenté en détails notre méthodologie de réconciliation non-fonctionnelle entre les activités du processus métier collaboratif et les services techniques satisfont les exigences fonctionnelles. Ce mécanisme de réconciliation non-fonctionnelle se divise en deux parties complémentaires selon le type de résultat retourné par la réconciliation fonctionnelle :

- La réconciliation non-fonctionnelle de services de type « 1-1 ». Cette approche recherche pour chaque activité du processus métier collaboratif, le service (parmi ceux qui satisfont les besoins fonctionnels) qui se rapproche le plus de ses exigences non-fonctionnelles. Pour cela, nous avons présenté une piste pour illustrer ce type de réconciliation.
- La réconciliation non-fonctionnelle de services de type « 1-N », appelée aussi réconciliation non-fonctionnelle de compositions de services. Cette approche s'intéresse à l'utilisation combinée de plusieurs services unitaires qui ensemble répondent au besoin de l'activité métier. Pour cela, nous proposons une approche détaillée qui permet de transformer cette problématique en une problématique de réconciliation « 1-1 ».

Une fois la réconciliation non-fonctionnelle de services est effectuée, les résultats obtenus permettent de trouver à chacune des activités métier, le ou les services qui satisfont ses besoins. Toutefois, le mécanisme proposé ici nécessite encore un certain nombre d'améliorations. Nous pouvons être confrontés au cas où aucun service et aucune composition de services ne couvre les

besoins de l'activité métier. Dans ce cas, il sera utile de combiner les attentes métier d'activités consécutives afin de trouver un ensemble de services répondant au besoin. Par ailleurs, notre mécanisme peut être amélioré en le basant sur une des méthodes de l'analyse multicritère.

Chapitre V

Implémentation et illustration

« Le sage n'affirme rien qu'il ne puisse prouver. »

Proverbe latin

V.1. INTRODUCTION	109
V.2. PLATEFORME A2S.....	109
V.2.1. Plateforme de modélisation : Petals BPM-NFR	110
V.2.1.1. Architecture technique	110
V.2.1.2. Implémentation avec GWT et Java	112
V.2.2. Plateforme de gouvernance SOA et de réconciliation : EasierGov-NFR	114
V.2.2.1. Rationalisation des services et de leurs propriétés non-fonctionnelles	115
V.2.2.2. Moteur de réconciliation non-fonctionnelle	117
V.2.3. Intégration dans une architecture SOA open-source	119
V.2.4. Démarche de qualité constante	120
V.3. ILLUSTRATION DE LA DEMARCHE PROPOSEE	121
V.3.1. Présentation du cas d'étude.....	121
V.3.2. Déroulement de la méthodologie	125
V.3.2.1. Modélisation et annotation non-fonctionnelle du processus	125
V.3.2.2. Réconciliation métier / technique	128
V.4. CONCLUSION.....	134

V.1. Introduction

Notre étude a permis de fournir une méthodologie permettant de prendre en considération les propriétés non-fonctionnelles depuis la modélisation des processus métier collaboratifs jusqu'à la réconciliation et sélection de services. Dans les chapitres précédents, nous avons présenté les différentes descriptions théoriques de nos cadres. Ce chapitre a pour objectifs de présenter, d'une part, l'implémentation de cette méthodologie afin qu'elle soit exploitable, et d'autre part la mise en œuvre de nos propositions sur un cas d'étude pour illustrer et valider nos travaux de recherche.

V.2. Plateforme A2S

Dans cette section, nous présentons la plateforme A2S (pour *Activities to Services*). Cette plateforme englobe les deux prototypes développés au cours de nos travaux :

- une plateforme de modélisation « *Petals BPM-NFR* » (niveau métier) qui se base principalement sur l'outil de modélisation BPMN 2.0 « *Petals BPM* » auquel nous avons implémenté une couche qui enrichit la modélisation des processus par l'annotation non-fonctionnelle ;
- une plateforme de gouvernance SOA « *EasierGov-NFR* » (niveau technique) qui s'intéresse à la rationalisation des services et de leurs propriétés non-fonctionnelles, et à la réconciliation non-fonctionnelle. Cette réconciliation hérite les résultats obtenus par la réconciliation fonctionnelle (effectuée par l'outil « *EasierSBS* »).

La Figure 51 présente une vue de haut niveau de l'architecture de la plateforme A2S. Ses composants seront détaillés dans les sections qui suivent.

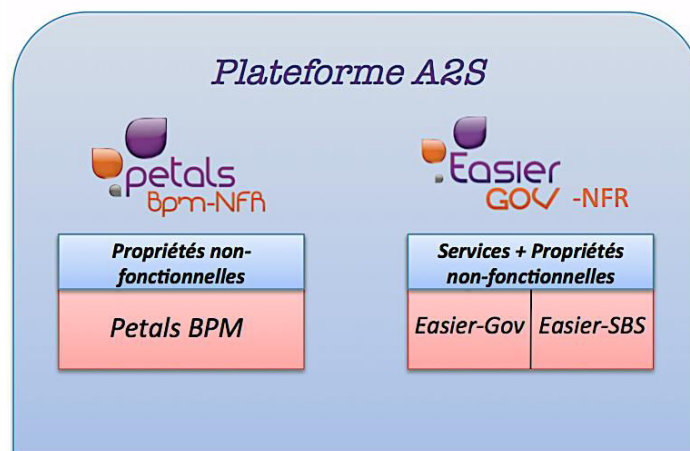


Figure 51 : Architecture de la plateforme A2S

V.2.1. Plateforme de modélisation : Petals BPM-NFR

Comme nous l'avons détaillé précédemment (cf. section III.2.2), Petals BPM-NFR représente notre plateforme graphique qui permet aux utilisateurs métier (architectes métier), d'une part, de modéliser les processus métier collaboratifs en suivant le standard BPMN 2.0, et d'autre part d'annoter les activités de ces processus par des exigences non-fonctionnelles. Cette annotation suit le standard WSQF pour la modélisation des propriétés non-fonctionnelles et permet également aux utilisateurs de définir leurs préférences (poids) par rapport à chacune des propriétés non-fonctionnelles choisies. La Figure 52 permet d'illustrer les fonctionnalités de notre plateforme Petals BPM-NFR.

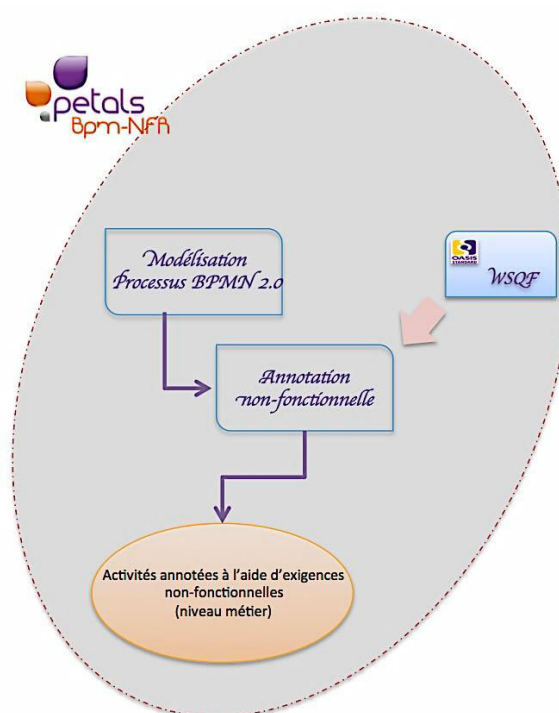


Figure 52 : Fonctionnalités de Petals BPM-NFR.

V.2.1.1. Architecture technique

Petals BPM-NFR est une application Client / Serveur et possède une architecture en couches qui se découpe en deux parties : (i) une partie client basée sur l'environnement GWT (nous présentons GWT dans la section qui suit) et (ii) une partie serveur entièrement écrite en Java. GWT propose un mécanisme d'appels de procédures distantes (*Remote Procedure Call* - RPC) qui permet la communication entre ces deux parties. Ce mécanisme de RPC utilisant la technologie de communication Ajax (pour *Asynchronous JavaScript and XML*) rend les appels de méthodes asynchrones. Cette architecture technique est schématisée par la Figure 53.

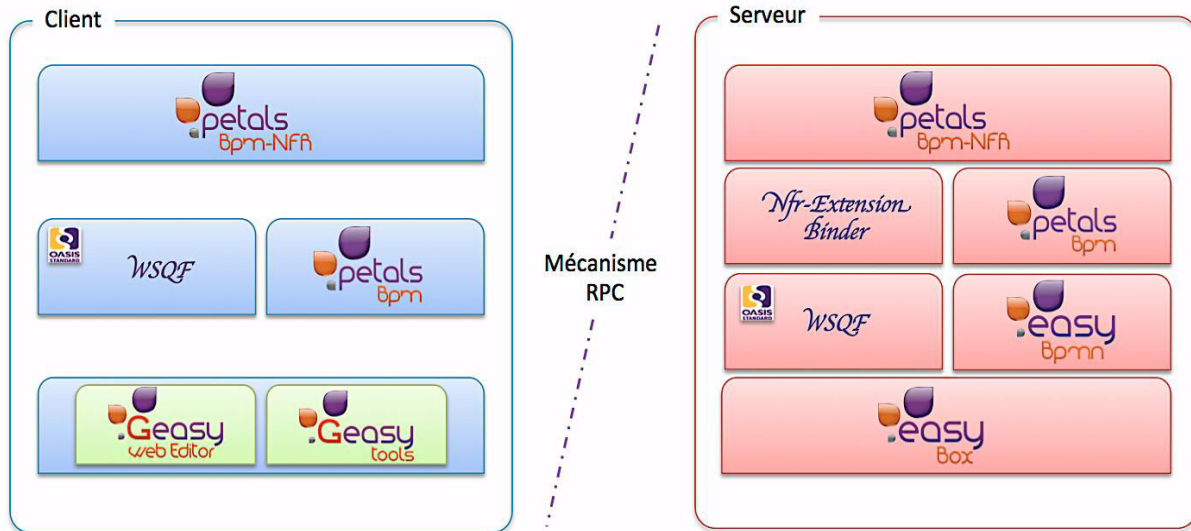


Figure 53 : Architecture technique de Petals BPM-NFR.

➤ La partie client

Petals BPM-NFR possède une interface utilisateur implémentée en GWT. Cette interface permet aux utilisateurs métier / architectes de modéliser graphiquement leurs processus, de les enrichir par des exigences non-fonctionnelles et d'exprimer leurs préférences par rapport à chacune des exigences non-fonctionnelles. La Figure 54 détaille les étapes de modélisation des processus métier collaboratifs annotés à l'aide d'exigences non-fonctionnelle en utilisant notre plateforme.

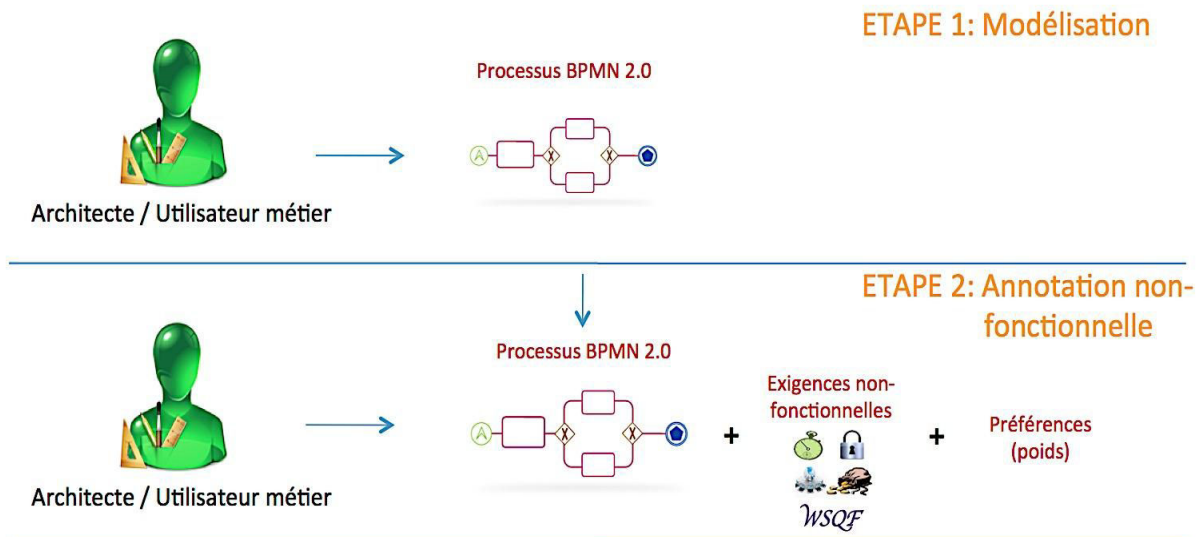


Figure 54 : Modélisation des processus métier annotés avec Petals BPM-NFR (coté client).

Petals BPM-NFR, du coté client, est basé sur le composant Petals BPM pour la modélisation graphique et sur le standard WSQF pour l'annotation non-fonctionnelle.

- *Petals BPM* : un outil Web open-source de modélisation graphique des processus métier. Il permet aux utilisateurs de créer en ligne des processus collaboratifs suivant le standard BPMN 2.0. Cette application est le résultat de l'utilisation des bibliothèques GeasyWebEditor et GeasyTools ;
- *GeasyWebEditor* : une bibliothèque basée sur le framework graphique GWT. Elle permet de faciliter la création de l'outil de modélisation graphique dans un environnement Web ;
- *GeasyTools* : un ensemble de bibliothèques graphiques GWT qui permet de créer des applications riches et interactives. Ces bibliothèques proposent un ensemble de composants graphiques de base qui seront réutilisables dans tout le projet.

➤ *La partie serveur*

Petals BPM-NFR du côté serveur possède des bibliothèques personnalisées écrites en Java et basées sur Petals BPM et NFR Extension Binder.

- *Petals BPM* : la partie serveur de Petals BPM permet de faire les modèles sous format de fichiers. Elle permet également d'exporter / importer les modèles vers / depuis le format BPMN 2.0 ;
- *EasyBPMN* : une bibliothèque qui correspond au modèle BPMN 2.0 traité par EasyBox. Elle permet également de gérer la transformation de BPMN 2.0 en un processus exécutable BPEL [Rajsiri et al., 2011] ;
- *NFR Extension Binder* : une bibliothèque qui s'intègre au serveur Petals BPM afin de permettre d'exporter / importer les extensions non-fonctionnelles au sein des fichiers BPMN 2.0 ;
- *WSQF* : une bibliothèque qui correspond au modèle Java du standard WSQF écrit par EasyBox ;
- *EasyBox* : une bibliothèque qui permet de transformer des schémas XML (eXtensible Markup Language) en modèles de classes Java équivalentes et vice versa (sérialisation et désérialisation avancées). En effet, cette bibliothèque facilite l'utilisation des fonctionnalités Jaxb (Java Architecture for XML Binding).

V.2.1.2. Implémentation avec GWT et Java

Pour implémenter les différentes fonctionnalités de notre modèleur Petals BPM-NFR, nous avons choisi l'environnement GWT (*Google Web Toolkit*) pour la modélisation graphique des processus et des exigences non-fonctionnelles. Nous avons également choisi le langage Java pour

effectuer la transformation des schémas de données en modèles de classes, ainsi que le traitement et la génération des fichiers BPMN 2.0.

La technologie GWT [Dewsbury, 2007] a été développée en 2006 par Bruce Johnson et lancée en open-source en 2007. GWT est un ensemble d'outils permettant, d'une part, de créer et d'optimiser des applications Web complexes, riches, en utilisant Java comme langage de développement et d'autre part, de compiler ce code en JavaScript et HTML. Ce code est interprétable automatiquement par les principaux navigateurs Web (*Google Chrome, Mozilla Firefox, etc.*). Par ailleurs, GWT permet aux développeurs de créer leurs propres composants graphiques, appelés *widgets* qui peuvent être facilement réutilisables et intégrés dans plusieurs projets.

Dans le cadre de la réalisation de Petals BPM-NFR, nous avons implémenté un ensemble de bibliothèques qui couvrent les fonctionnalités suivantes :

- Prise en considération des exigences non-fonctionnelles : nous avons implémenté un ensemble de bibliothèques permettant d'annoter non-fonctionnellement les processus métier collaboratifs. Ces bibliothèques concernent : (i) la transformation (EasyBox) du schéma de données du standard WSQF en des modèles de classes Java équivalents. Cette transformation permet de manipuler les propriétés non-fonctionnelles en tant qu'objets Java, (ii) l'extension du modèle BPMN 2.0 (fourni par Petals BPM) en rajoutant les objets « propriétés non-fonctionnelles » ;
- Import direct dans la plateforme graphique de diagrammes BPMN 2.0 annotés non-fonctionnellement : Petals BPM-NFR permet aux utilisateurs d'importer (*uploader*) des fichiers BPMN 2.0 annotés par des exigences non-fonctionnelles (ceci est seulement dans le cas où cette annotation respecte le modèle WSQF). Toutes les informations que ces fichiers contiennent et qui sont supportées par Petals BPM-NFR sont mises à disposition pour la modélisation en cours ;
- Export de diagrammes modélisés et annotés sous format BPMN 2.0 : Petals BPM-NFR permet aux utilisateurs d'exporter leurs processus modélisés et annotés par des exigences non-fonctionnelles sous forme de fichier BPMN 2.0. Cet export est possible si celui-ci passe une étape de validation. Deux niveaux de validation sont pris en compte : (i) une validation de conformité à la norme BPMN 2.0 et (ii) une validation de la conformité des types de valeurs des propriétés non-fonctionnelles à la norme WSQF. Le premier niveau de validation a été réalisé par Petals BPM et nous le réutilisons durant nos travaux ;
- Modélisation graphique : une interface utilisateur (*User Interface - UI*) en GWT a été développée pour Petals BPM-NFR. Elle enrichit l'interface de Petals BPM en ajoutant une partie dédiée à l'annotation graphique des activités par des exigences non-fonctionnelles (dans le respect de la spécification de WSQF). Nous avons également implémenté les

modèles internes relatifs à cette interface qui permettent d'enrichir la palette des composants BPMN 2.0 et de la lier à la zone dédiée aux propriétés dans le but de faciliter aux utilisateurs métier la modélisation. Ainsi, lors de la modélisation, à chaque fois que l'utilisateur fait un glisser – déposer (*drag and drop*) d'une activité BPMN 2.0 dans la zone de modélisation, l'interface dédiée à l'annotation non-fonctionnelle apparaît dans la zone des propriétés. La Figure 55 représente une capture d'écran de l'interface utilisateur de l'éditeur Petals BPM-NFR.

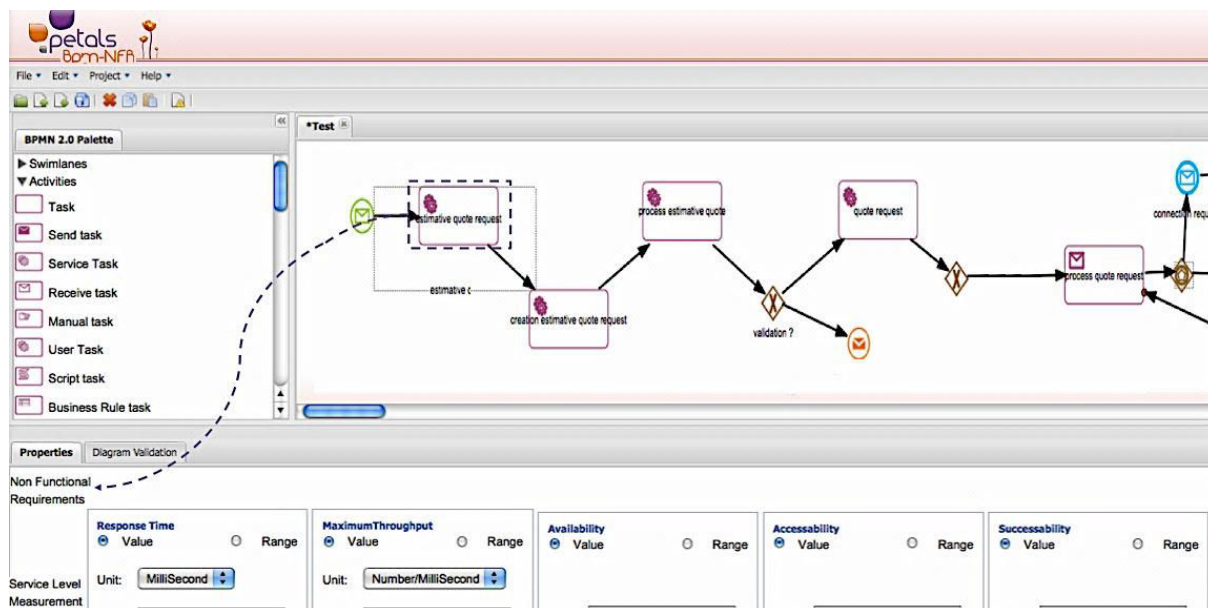


Figure 55 : Capture d'écran de la plateforme Petals BPM-NFR.

V.2.2. Plateforme de gouvernance SOA et de réconciliation : EasierGov-NFR

Comme nous avons pu le voir dans les chapitres précédents, un grand nombre de services, d'activités métier et de propriétés non-fonctionnelles est à prendre en charge. Nous souhaitons par le prototype EasierGov-NFR rationaliser la gestion des services et de leurs propriétés non-fonctionnelles et fournir le moteur de réconciliation non-fonctionnelle. Pour y parvenir, nous avons fortement basé les choix des composants d'EasierGov-NFR sur des standards (cf. Chapitre III). Nous avons également défini un ensemble de bibliothèques implémentées avec le langage Java qui permettent de gérer les fonctionnalités d'EasierGov-NFR. Chacune de ces bibliothèques expose ces fonctions dans une interface dédiée, sous la forme d'une API (*Application Programming Interface*) Java (*-api) qui sera implémentée par la suite (*-impl).

V.2.2.1. Rationalisation des services et de leurs propriétés non-fonctionnelles

EasierGov-NFR fournit deux APIs « Services API » et « SLA-API ». La première est basée sur la spécification WSDL pour la description de services, et la deuxième suit les standard WS-Agreement et WSQF.

Services API : cette bibliothèque fournit une interface que nous appelons « Services API » (cf. Figure 56).

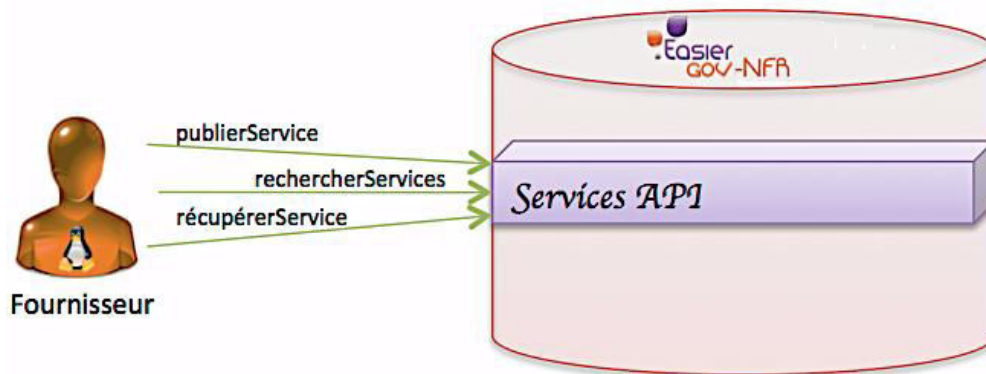


Figure 56 : Principales méthodes de la bibliothèque « services API ».

Elle contient un ensemble d'opérations permettant de gérer les services. Les principales méthodes de cette bibliothèque sont les suivantes :

- publierService (WSDL) : publier un nouveau service Web dans le registre à partir de la description de services passée en paramètre d'entrée. Cette description respecte les règles définies dans le standard de services WSDL ;
- Liste <ServiceId> rechercherServices (query) : rechercher une liste d'identifiants (id) de services à partir des paramètres d'entrées ;

- Service recupererService (idService) : obtenir la description de services (WSDL) à partir de l'identifiant demandé.

Cette librairie permet également de gérer d'autres méthodes pour les services telles que la récupération et la manipulation de l'interface du service et de son point d'accès (*endpoint*).

SLA API : Interface fournit un ensemble de méthodes permettant de créer et de manipuler les modèles de contrats de type SLA en respectant le standard WS-Agreement (cf. Figure 57) :

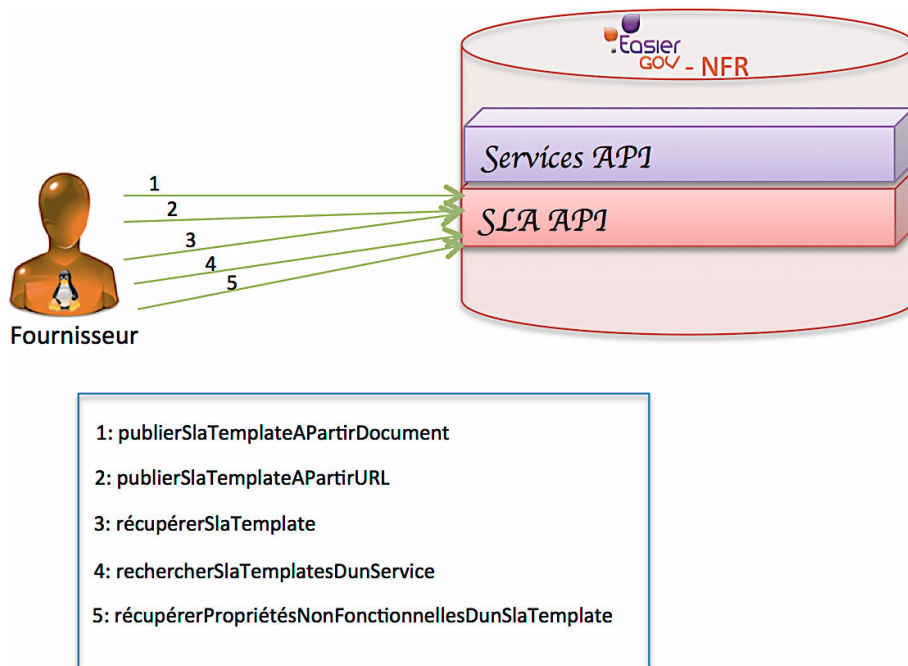


Figure 57 : Principales méthodes de la librairie « SLA API ».

- publierSlaTemplateAPartirDocument (wsagreement) : enregistrer un nouveau modèle de contrat dans le registre de gouvernance à partir du document passé en paramètre d'entrée. Ce document suit la description définie dans la norme WS-Agreement. Chaque modèle de contrat est relatif à un service défini dans le contrat par son adresse URL. Dans notre exemple (cf. Figure 58), le modèle SLA concerne le service pompier situé à l'adresse : esb://http://www.example.org/firemen//:firemen@firemenSOAP). Cette méthode permet également de parcourir le modèle de contrat, et d'enregistrer dans le registre de gouvernance, pour le service en question, les valeurs de toutes les propriétés non-fonctionnelles que le contrat contient ;


```

<wsag:Template wsag:TemplateId="A101"
  xmlns:wsag="http://schemas.ggf.org/graap/2007/03/ws-agreement"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsad="http://www.w3.org/2005/08/addressing"
  xmlns:wsqdl="http://www.nca.or.kr/2006/wsqdl"
  xsi:schemaLocation="http://schemas.ggf.org/graap/2007/03/ws-agreement wsagreement10.xsd ">
  <wsag:Name>wsagreement</wsag:Name>
  <wsag:Context>
    <wsag:AgreementInitiator xsi:type="anyType" />
    <wsag:AgreementResponder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="wsad:EndpointReferenceType">
      <wsad:Address>esb://http://www.example.org/firemen/:firemen@firemenSOAP</wsad:Address>
    </wsag:AgreementResponder>
  </wsag:Context>
</wsag:Template>

```

Figure 58 : SLA : adresse du service.

- publierSlaTemplateAPartirURL (url) : publier un nouveau modèle de contrat SLA à partir de son adresse URL (pour Uniforme Ressource Locator). Tout comme la méthode de la publication à partir de la description du modèle du contrat, cette méthode permet également de parcourir, d'analyser et d'enregistrer les valeurs des propriétés non-fonctionnelles ;
- Template récupérerSlaTemplate (idTemplate) : obtenir la description du modèle de contrat SLA depuis son identifiant ;
- Liste < idTemplate > rechercherSlaTemplatesDunService (idService) : rechercher et retourner tous les modèles de contrat relatifs au service (selon son identifiant) défini dans le paramètre d'entrée ;
- Liste < idNFR > récupérerPropriétésNonFonctionnellesDunSlaTemplate (idTemplate) : rechercher et retourner l'ensemble des propriétés non-fonctionnelles de WSQF contenu dans le template SLA.

V.2.2.2.Moteur de réconciliation non-fonctionnelle

Ce module est dédié à la réconciliation entre le niveau métier et le niveau technique en se basant sur les critères non-fonctionnels. À partir du processus métier modélisé dans Petals BPM-NFR et annoté par des exigences non-fonctionnelles, ce moteur de réconciliation permet de trouver le service ou la composition de services qui se rapproche des exigences de l'activité métier. Nous détaillons dans la Figure 59 notre démarche globale permettant d'effectuer cette réconciliation.

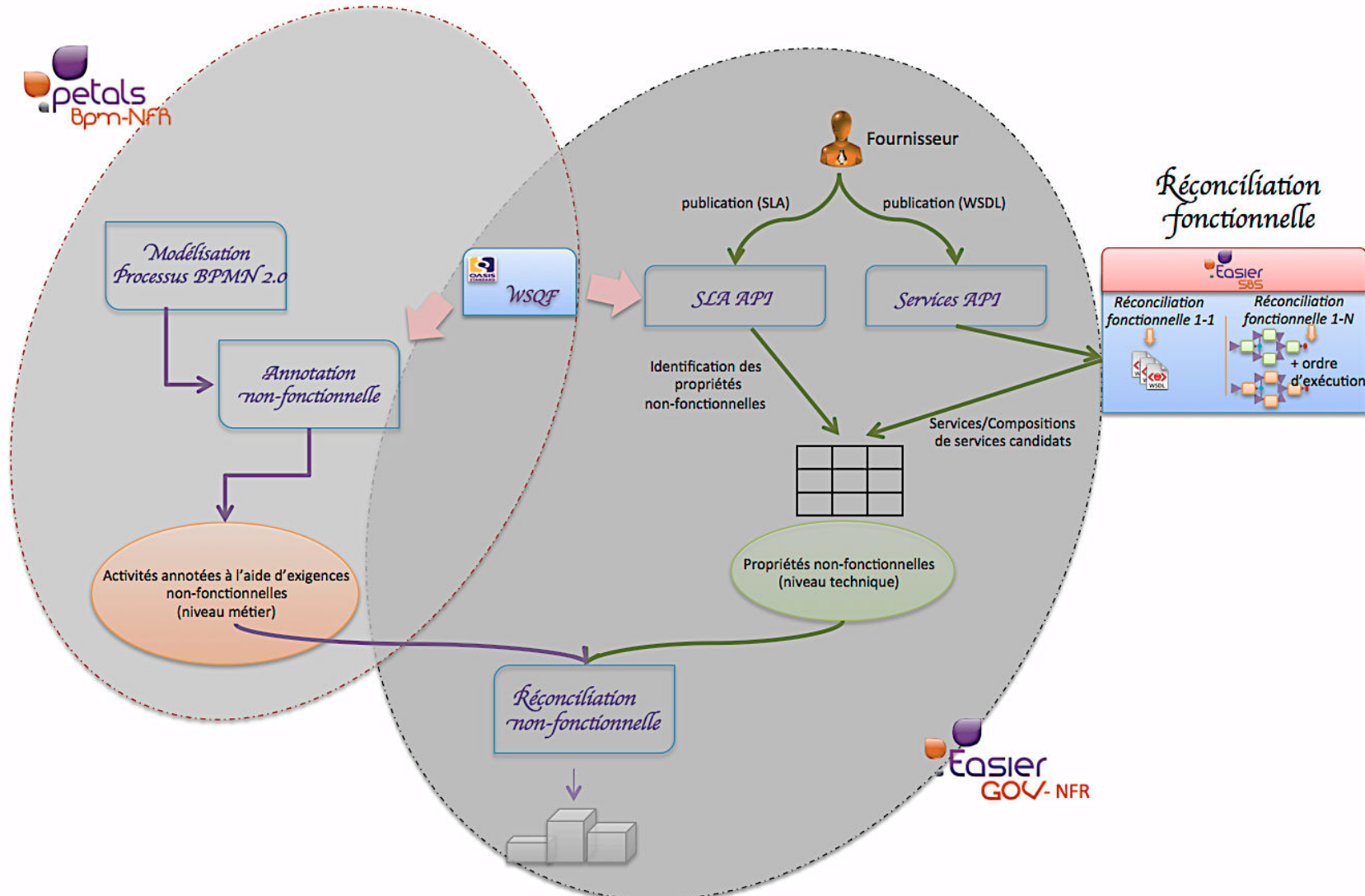


Figure 59 : Moteur de réconciliation non-fonctionnelle.

Ce moteur de réconciliation non-fonctionnelle utilise les bibliothèques de services et de SLA détaillées précédemment, afin de permettre aux fournisseurs de services de publier dans un registre unique leurs services et de leur associer (en les publiant également) un ou plusieurs modèles SLA.

À partir des services publiés dans le registre, nous appliquons le filtre fonctionnel fourni par le composant EasierSBS établi dans [Boissel-Dallier, 2012]. Ce filtre fournit pour chacune des activités métier du processus une liste de services unitaires (dans le cas d'une réconciliation de type 1-1) ou un ensemble de compositions de services (dans le cas d'une réconciliation 1-N) ;

Pour chacune des activités métier et à partir des propriétés non-fonctionnelles enregistrées dans le registre EasierGov, nous construisons une matrice contenant les services et/ou les compositions de services et leurs propriétés non-fonctionnelles, en nous limitant à celles qui ont été choisies lors de l'étape de l'annotation du niveau métier.

À partir de ces propriétés non-fonctionnelles techniques et des exigences non-fonctionnelles modélisées au niveau métier Petals BPM-NFR, nous appliquons l'algorithme de réconciliation (cf. Chapitre IV) qui correspond au type de réconciliation fonctionnelle effectuée (1-1 ou 1-N). À l'issue de cette étape, nous obtenons pour chacune des activités métier, un classement non-fonctionnel des services. Le premier est celui qui se rapproche le plus des exigences de l'utilisateur définies dans l'activité métier.

V.2.3. Intégration dans une architecture SOA open-source

Les deux plateformes Petals BPM-NFR et EasierGov-NFR s'intègrent au sein de la gamme d'applications open-sources développées par l'équipe de recherche Petals Link de Linagora afin de permettre leur implantation dans une architecture réelle.

La plateforme de modélisation Petals BPM-NFR permet, à plusieurs utilisateurs métier, de créer des processus métier collaboratifs (suivant la norme BPMN 2.0) annotés à l'aide d'exigences non-fonctionnelles suivant le standard WSQF. Petals BPM-NFR est lié à la plateforme EasierGov-NFR pour gérer la réconciliation entre les activités métier et les services techniques disponibles (réconciliation fonctionnelle par le biais d'EasierSBS). Petals BPM-NFR est également lié à l'outil Petals BPM qui comprend une bibliothèque permettant de créer, à partir des services, les processus BPEL correspondants. Ces processus sont exécutables dans l'infrastructure de services (le bus de services) Petals ESB.

La plateforme EasierGov-NFR est liée à l'outil de monitoring EasierBSM (*Business Service Monitoring*) [Lesbegueries et al., 2012] qui est également lié à Petals ESB afin de surveiller l'exécution des contrats de services et de notifier dans le cas d'une violation. La Figure 60 décrit l'intégration de Petals BPM-NFR et d'EasierGov-NFR dans l'architecture SOA existante.

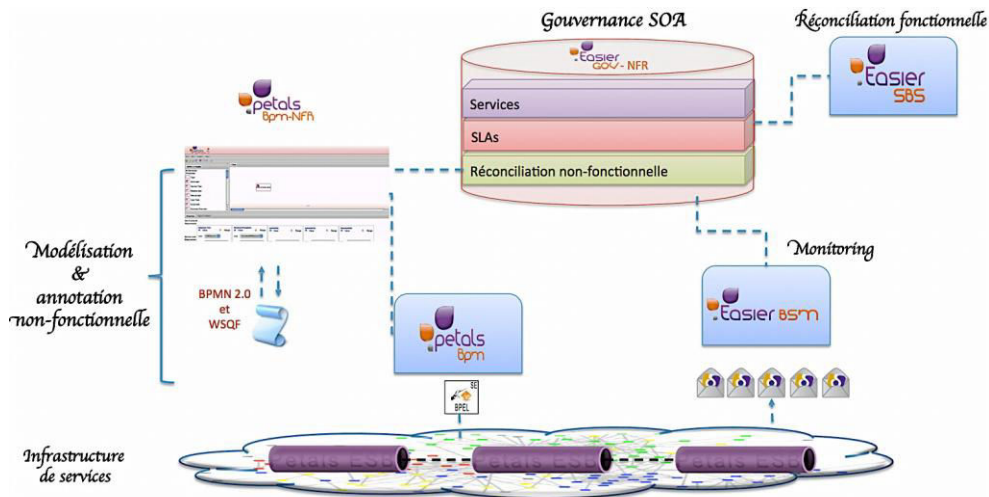


Figure 60 : Intégration de Petals BPM-NFR et EasierGov-NFR dans une architecture SOA.

V.2.4. Démarche de qualité constante

Les produits développés par l'équipe de recherche et développement Petals de Linagora respectent un ensemble de règles de développement et sont soumis à la démarche d'intégration continue. Au même titre que ces produits, les plateformes Petals BPM-NFR et EasierGov-NFR suivent le même principe d'intégration continue. Cela signifie qu'à chaque modification du code source ou d'une dépendance, les tests unitaires de chaque composant impliqué dans la plateforme en question sont relancés afin de vérifier qu'aucune régression fonctionnelle n'a eu lieu. Nous présentons dans les figures suivantes (cf. Figure 61 et Figure 62) les résultats des tests d'intégration continue respectivement pour Petals BPM-NFR et EasierGov-NFR sur la plateforme Jenkins⁶.

Continuous Integration R&D

Jenkins > PetalsBPM-NFR

S	W	Name ↓	Dernier Succès	Dernier Echec	Dernière Durée
🟢	☀️	EBMWS_quality_prod_easyswsl	7 h 38 mn - #178	N/A	2 mn 31 s
🟢	☀️	EBMWS_trunk_experimental_easiergov-NFR_Linux	8 h 25 mn - #401	N/A	13 mn
🟢	☀️	EBMWS_trunk_experimental_easysbpm_Linux	1 j 9 h - #190	N/A	3 mn 58 s
🟢	☀️	EBMWS_trunk_experimental_easyytools_Linux	1 j 8 h - #43	N/A	4 mn 52 s
🟢	☀️	EBMWS_trunk_experimental_easyywebeditor_Linux	1 j 8 h - #30	N/A	5 mn 58 s
🟢	☀️	EBMWS_trunk_experimental_petalsbpm-NFR_Linux	13 mn - #1	N/A	9 mn 19 s
🟢	☀️	EBMWS_trunk_experimental_petalsbpm_Linux	8 h 11 mn - #198	N/A	34 mn
🟢	☀️	EBMWS_trunk_experimental_wsqf_Linux	7 h 37 mn - #177	N/A	1 mn 14 s

État du lanceur de constructions

#	Statut
1	Slave01
1	Slave02
1	Slave-XP

Légende: 🟢 RSS pour toutes, 🟡 RSS pour tous les échecs, 🟠 RSS juste pour les derniers échecs

Figure 61 : Résultats des tests d'intégration continue de Petals BPM-NFR.

⁶ Jenkins est un serveur open-source d'intégration continue pour les projets Java développés avec maven.

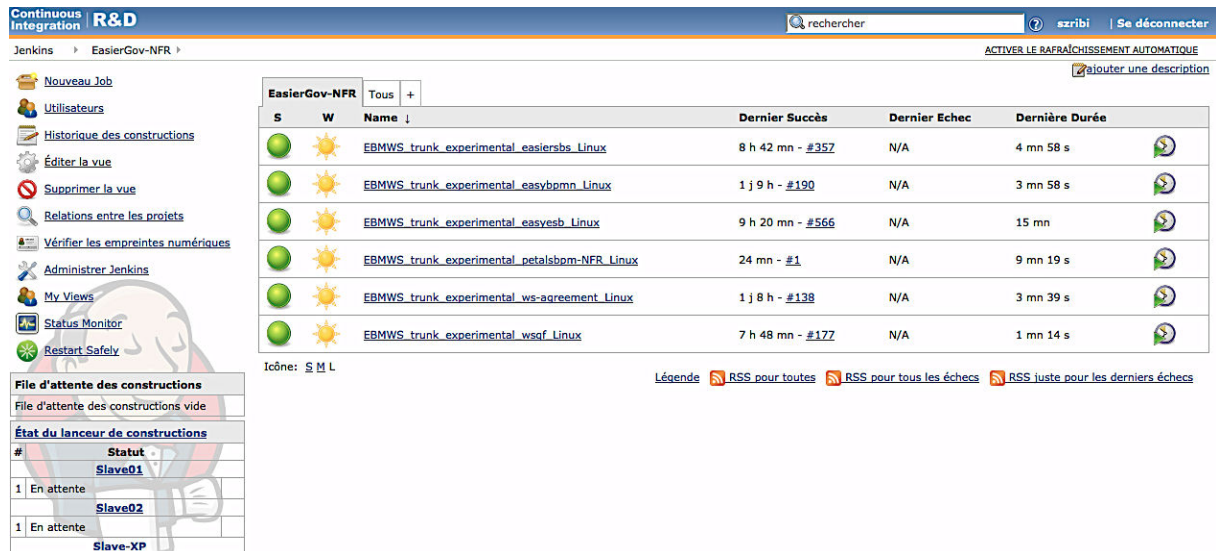


Figure 62 : Résultats des tests d'intégration continue d'EasierGov-NFR.

Comme nous pouvons l'observer, tous les voyants sont au vert, les deux plateformes Petals BPM-NFR et EasierGov-NFR assurent bien un état stable avec les autres produits Petals utilisés pour l'implémentation de chacune.

V.3. Illustration de la démarche proposée

Nous présentons dans cette section un cas d'étude fictif permettant d'illustrer l'applicabilité de notre méthodologie.

V.3.1. Présentation du cas d'étude

Sazri est une entreprise spécialisée dans la vente en ligne de chaussures. Elle souhaite trouver une entreprise de marketing prédictif avec laquelle elle envisage de collaborer afin d'améliorer ses ventes. Sazri lui fournirait la fiche du produit à promouvoir et son fichier clients. L'entreprise de marketing aurait pour rôle d'étudier les comportements clients afin d'anticiper leurs actions futures et par suite mieux cibler l'envoi de la publicité aux personnes pouvant être intéressées par le produit. Les échanges entre ces deux partenaires se dérouleraient de la manière suivante :

- 1) Sazri procède à une analyse parallèle de ses stocks et de ses tendances d'achats. Le résultat de cette analyse fournit la fiche du produit dont la vente peut être optimisée ;
- 2) L'entreprise de Marketing reçoit la fiche de ce produit et la catégorise selon ses propres critères (i.e. des bottes pour les femmes de plus de 30 ans, style classique ou non, etc.) ;
- 3) À l'issue de l'étape de la catégorisation, l'entreprise Sazri envoie son fichier clients ;

- 4) Lors de la procédure d'inscription sur le site de Sazri, cette dernière propose à ses clients la possibilité de s'inscrire soit en remplissant un formulaire, soit via leur compte réseau social Facebook ou Google +. L'entreprise de marketing profite de cet avantage afin d'élargir le fichier clients par le parcours du graphe social :
 - Si le fichier contient des inscriptions via un réseau social, l'entreprise de marketing récupère leurs contacts (dans la mesure où ceci est possible) ;
 - Sinon, le fichier clients reste inchangé.
- 5) À partir de la liste, ou de la liste enrichie, de clients, l'entreprise de marketing effectue son analyse de comportements clients et établit la liste des clients potentiellement intéressés. La liste des courriels de ces clients est par la suite envoyée à Sazri.

L'utilisation des outils classiques de modélisation graphique de processus BPMN 2.0 nous a donné le processus schématisé par la Figure 63. Dans ce processus collaboratif, nous retrouvons les différentes activités de l'entreprise Sazri (analyse des stocks, analyse des tendances d'achats) ainsi que celles demandées pour l'entreprise de marketing (catégorisation du produit, récupération du fichier clients et son élargissement par les réseaux sociaux, analyse des contacts réseaux sociaux, et envoi des courriels des clients ciblés).

C'est la tâche du composant médiateur d'invoquer ces différentes activités. Nous retrouvons également la modélisation des étapes parallélisées (analyse des stocks et analyse des tendances d'achats) et des étapes conditionnelles (analyse des inscriptions clients et celles qui ont été faites via un réseau social passeront à l'étape élargissement du fichier clients et les autres passeront directement à l'étape d'analyse du comportement).

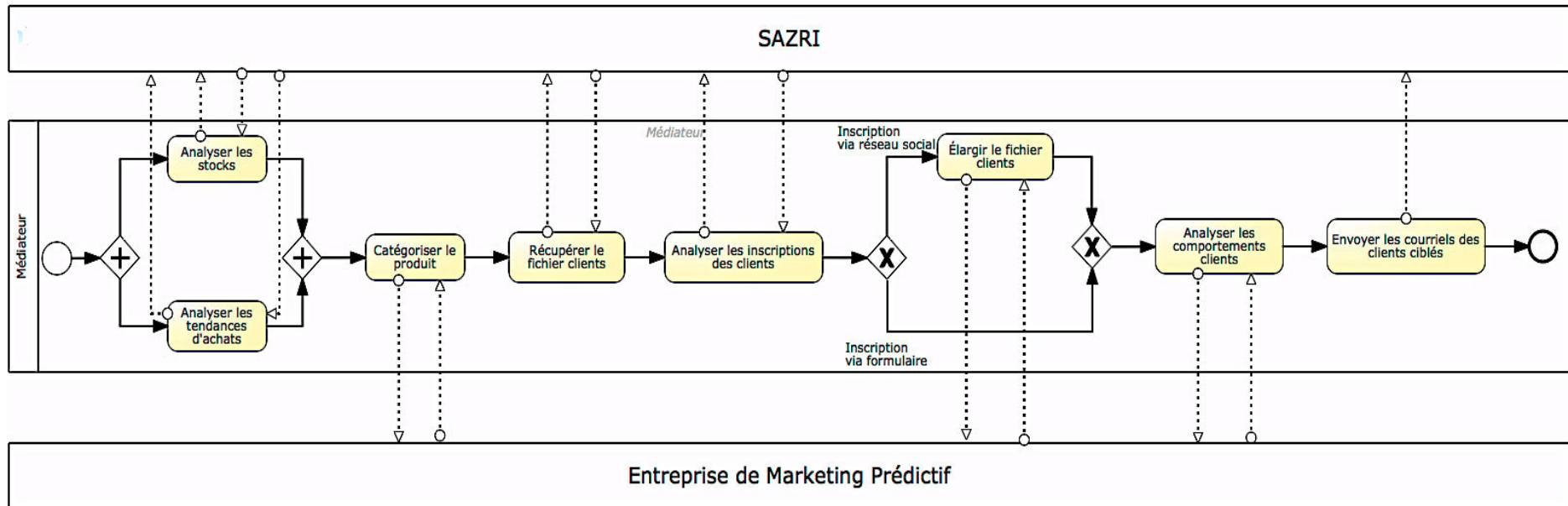


Figure 63 : Processus métier collaboratif pour l'amélioration des ventes de Sazri par le marketing prédictif.

Tableau 10 : Fonctionnalités des activités du processus métier pour l'amélioration des ventes de Sazri par le marketing prédictif.

	Activité	Fonctionnalité
Entreprise Sazri	Analyser les stocks	L'analyse des stocks consiste à étudier l'état des stocks, c'est-à-dire leur niveau, leur rotation, etc.
	Analyser les tendances d'achats	
	Récupérer le fichier clients	A l'issue de l'étape de la catégorisation du produit, le médiateur demande à l'entreprise Sazri de lui envoyer son fichier clients et le récupère.
Entreprise de marketing prédictif	Catégoriser le produit	L'entreprise de marketing devrait disposer d'un service permettant, à partir de la fiche du produit, de le catégoriser. Après traitement, ce service devrait fournir la fiche du produit catégorisé.
	Analyser les inscriptions des clients	A partir du fichier clients envoyé par l'entreprise Sazri, l'entreprise de marketing devrait disposer d'un service permettant de le parcourir et d'analyser le mode d'inscription des clients. Pour ceux qui sont inscrits via le réseau social Facebook ou Google +, le service permettant d'élargir le fichier clients devrait être invoqué. Pour ceux qui sont inscrits via le formulaire, aucun traitement à effectuer et elle passe directement à l'étape suivante, à savoir l'analyse des comportements clients.
	Élargir le fichier clients	Pour ceux qui sont inscrit via le réseau social Facebook ou Google +, l'entreprise de marketing à travers cette activité accède et analyse leur graphe social et ajoute leurs contacts et les contacts de leurs contacts et les ajoute au fichier clients initial pour obtenir le fichier clients élargi.
	Analyser les comportements clients	Cette activité prend en paramètre d'entrée le fichier clients (élargi ou initial, selon le résultat de l'activité précédente) et analyse les comportements afin d'identifier, parmi eux, ceux qui sont potentiellement intéressés par l'achat du produit sélectionné par l'entreprise Sazri. Les courriels de ces derniers seront enregistrés dans un fichier, nommé fichier clients cibles.
	Envoyer les courriels des clients ciblés	Cette activité permet d'envoyer le fichier clients cibles à l'entreprise Sazri.

Dans le Tableau 10, nous présentons les fonctionnalités de chacune des activités métier du processus collaboratif permettant l'amélioration des ventes de l'entreprise Sazri par le marketing prédictif.

Le fichier clients est ce qui est de plus précieux pour l'entreprise Sazri. Il regroupe toutes les informations concernant les clients qui lui sont essentiels et qui font vivre toutes ses activités. Il est donc indispensable, pour toute activité de l'entreprise de marketing qui utilise le fichier clients (à savoir : analyser les inscriptions des clients, élargir le fichier clients, analyser les comportements clients, et envoyer les courriels des clients ciblés), de l'accompagner d'une garantie de confidentialité et d'intégrité des données. Sazri souhaite également que ces activités proposent une procédure d'authentification.

Par ailleurs, les fêtes de Noël approchant à grands pas, l'entreprise Sazri ne souhaite pas rater cette période stratégique. Elle a donc des exigences temporelles (temps de réponse), de disponibilité, et de coût (prix). Ces exigences sont définies pour chacune des activités de l'entreprise de marketing.

Le Tableau 11 définit les exigences non-fonctionnelles de l'entreprise Sazri pour chacune des activités métier et le Tableau 12 détaille les préférences (les poids) de Sazri par rapport aux exigences non-fonctionnelles de chacune des activités métier annotées.

Tableau 11 : Exigences non-fonctionnelles de l'entreprise Sazri.

Activité	Exigence non-fonctionnelle	Temps de réponse (s)	Disponibilité (≥)	Prix (≤)	Confidentialité	Intégrité	Authentification
Catégoriser le produit		5 secondes	95%	10,00 €			
Élargir le fichier clients		40 secondes	90%	100,00 €	x	x	x
Analyser les comportements clients		1 jour	90%	100,00 €	x	x	x
Envoyer les courriels des clients ciblés		5 secondes	95%	2,00 €	x	x	x

Tableau 12 : Préférences (poids) de l'entreprise Sazri par rapport aux exigences non-fonctionnelles.

Activité	Exigence non-fonctionnelle	Temps de réponse	Disponibilité	Prix	Confidentialité	Intégrité	Authentification
Catégoriser le produit		0.3	0.5	0.2			
Élargir le fichier clients		0.05	0.05	0.05	0.4	0.4	0.05
Analyser les comportements clients			0.05	0.1	0.4	0.4	0.05
Envoyer les courriels des clients ciblés		0.05	0.1		0.4	0.4	0.05

Le processus dans la Figure 63 a permis de modéliser les propriétés fonctionnelles des activités du processus métier collaboratif. Mais les questions suivantes restent ouvertes : comment modéliser les exigences non-fonctionnelles et les préférences demandées par l'entreprise Sazri ? Comment les prendre en considération pour trouver les services qui répondent à ses attentes ?

V.3.2. Déroulement de la méthodologie

Cette section a pour objectif d'illustrer l'application de notre méthodologie au cas d'étude présenté ci-dessus afin de réconcilier les activités du processus métier collaboratifs et les services techniques disponibles. Dans cette partie, nous ne re-détaillerons pas le principe de fonctionnement de chaque étape. Nous vous invitons pour cela à vous référer aux Chapitre III et Chapitre IV.

V.3.2.1 Modélisation et annotation non-fonctionnelle du processus

La première étape de notre méthodologie consiste à modéliser le processus métier décrivant la collaboration entre Sazri et l'entreprise de marketing prédictif, puis d'annoter les activités par les exigences non-fonctionnelles.

Nous reprenons le processus présenté précédemment (cf. Figure 63) et nous le modélisons avec l'outil de modélisation graphique Petals BPM-NFR. Pour ce faire, nous créons un nouveau projet métier dédié à ce cas d'étude (Sazri / MarketingPrédictif). Par la suite, à partir de la palette, nous sélectionnons l'élément BPMN 2.0 et nous faisons un glisser / déposer dans l'espace de modélisation. Dans la Figure 64, nous montrons la modélisation du médiateur dans Petals BPM-NFR.

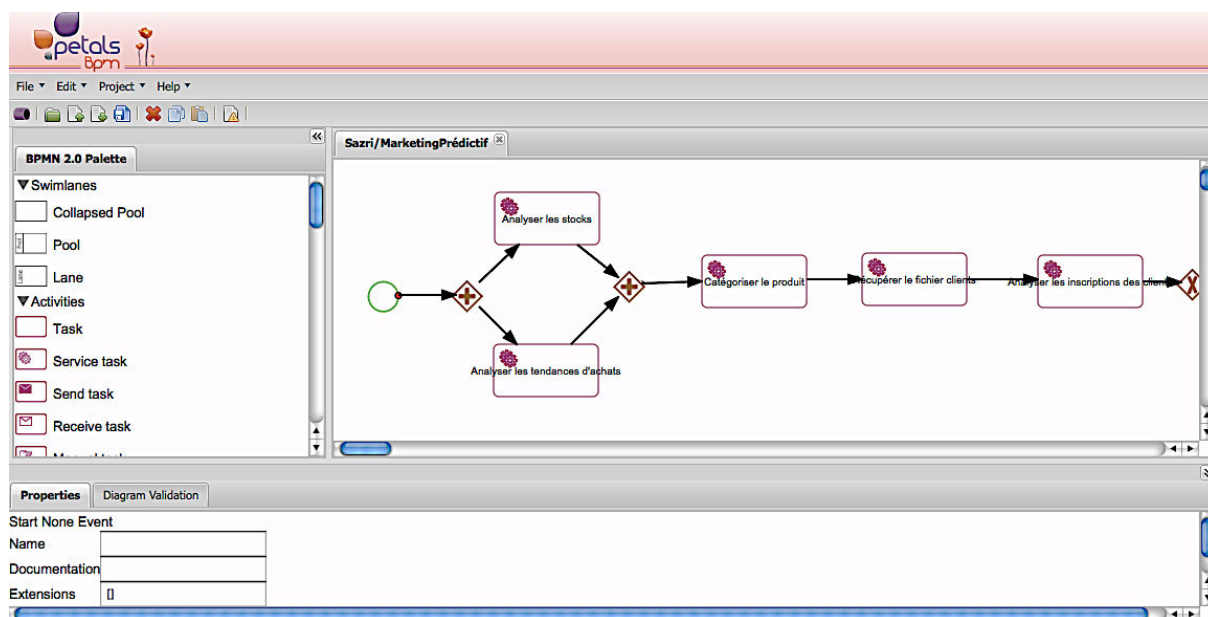


Figure 64 : Modélisation du cas d'étude Sazri/MarketingPrédicatif dans Petals BPM-NFR.

Une fois le processus métier est modélisé, il est nécessaire d'annoter les activités de l'entreprise de marketing prédictif à l'aide d'exigences non-fonctionnelles de Sazri (qui devront être respectées lors de la phase de réconciliation non-fonctionnelle). Pour cela, nous utilisons l'interface dédiée à l'annotation non-fonctionnelle dans l'outil graphique Petals BPM-NFR et pour chacune des activités de l'entreprise de marketing, nous renseignons les valeurs et les poids des exigences non-fonctionnelles demandés par l'entreprise Sazri.

L'annotation non-fonctionnelle illustrée dans la Figure 65 ne porte que sur l'activité « Élargir le fichier client ». L'ensemble des activités de l'entreprise de marketing prédictif a été également annoté.

Service Task	
Name	Élargir le fichier clients
Documentation	
Extensions	[]
Non Functional Requirements	
Service Level Measurement	Response Time <input checked="" type="radio"/> Value <input type="radio"/> Range Unit: <input type="text" value="Second"/>
	Value: <input type="text" value="40"/> Weight: <input type="text" value="0.05"/>
Security	Availability <input checked="" type="radio"/> Value <input type="radio"/> Range Value: <input type="text" value="90"/> Weight: <input type="text" value="0.05"/>
	Authentication <input checked="" type="checkbox"/> Weight: <input type="text" value="0.05"/>
Business Value Quality	Integrity <input checked="" type="checkbox"/> Weight: <input type="text" value="0.4"/>
	Privacy <input checked="" type="checkbox"/> Weight: <input type="text" value="0.4"/>
	Price <input checked="" type="radio"/> Value <input type="radio"/> Range Unit: <input type="text" value="Euro"/>
	Value: <input type="text" value="100"/> Weight: <input type="text" value="0.05"/>

Figure 65 : Annotation non-fonctionnelle de l'activité « Élargir le fichier clients ».

Il est également possible d'enregistrer le processus modélisé et annoté au format BPMN2.0. Pour cela, dans la barre de menu du Petals BPM-NFR, il suffit de sélectionner *File*, puis *Export*, et enfin dans la fenêtre qui apparaît, il faut choisir le format d'export BPMN. Nous obtenons, par la suite, le fichier BPMN 2.0 correspondant au processus que nous venons de modéliser et d'annoter. Dans la Figure 66, nous présentons un extrait de ce fichier correspondant à l'activité « Élargir le fichier clients » (cf. Annexe 3 pour le fichier au format BPMN 2.0 complet). Dans cet extrait, nous retrouvons, l'ensemble des exigences non-fonctionnelles choisies ainsi que leurs valeurs. Rappelons que les exigences non-fonctionnelles de type booléen (telles que l'*Authentication*, *Integrity* et *Privacy* choisies dans notre exemple) n'apparaissent dans le BPMN 2.0 que si elles ont été sélectionnées par l'utilisateur lors de la phase d'annotation.

```

- <bpmn20:serviceTask default="_1382291745017id52" id="_1382291720000id32" name="Élargir le fichier clients">
- <bpmn20:extensionElements>
- <wsqdl:ServiceLevelMeasurement>
- <wsqdl:MeasureFactor>
- <wsqdl:ResponseTime>
- <wsqdl:MetricValue>
  <wsqdl:Value xsi:type="xs:float">40.0</wsqdl:Value>
  <wsqdl:Type>float</wsqdl:Type>
  <wsqdl:Unit>Second</wsqdl:Unit>
</wsqdl:MetricValue>
</wsqdl:ResponseTime>
- <wsqdl:Availability>
- <wsqdl:MetricValue>
  <wsqdl:Value xsi:type="xs:float">90.0</wsqdl:Value>
  <wsqdl:Type>percent</wsqdl:Type>
</wsqdl:MetricValue>
</wsqdl:Availability>
</wsqdl:MeasureFactor>
</wsqdl:ServiceLevelMeasurement>
- <wsqdl:BusinessValueQuality>
- <wsqdl:ServiceCost>
- <wsqdl:ServicePrice>
  <wsqdl:Price unit="Euro">100</wsqdl:Price>
</wsqdl:ServicePrice>
</wsqdl:ServiceCost>
</wsqdl:BusinessValueQuality>
- <wsqdl:Security>
  <wsqdl:Property name="Authentication"> </wsqdl:Property>
</wsqdl:Security>
- <wsqdl:Security>
  <wsqdl:Property name="Integrity"> </wsqdl:Property>
</wsqdl:Security>
- <wsqdl:Security>
  <wsqdl:Property name="Privacy"> </wsqdl:Property>
</wsqdl:Security>
</bpmn20:extensionElements>
</bpmn20:serviceTask>

```

Figure 66 : Extrait du fichier BPMN 2.0 : activité « Élargir le fichier clients ».

V.3.2.2 Réconciliation métier / technique

À l'issue de l'étape de modélisation et d'annotation non-fonctionnelle du processus métier collaboratif Sazri / Marketing Prédictif, l'étape de réconciliation entre les activités métier de l'entreprise et les services techniques disponibles dans notre registre de gouvernance SOA (EasierGov-NFR) se présente.

La démarche décrite dans cette section correspond à la réconciliation de l'activité « Élargir le fichier clients ». Elle est identique pour toutes les autres activités métier du processus collaboratif. Il s'agit dans un premier temps de faire appel à l'outil EasierSBS pour réaliser la réconciliation fonctionnelle. Par la suite, pour chacun des services / compositions de services fournis par cette réconciliation, nous analysons leurs templates SLA afin de récupérer les valeurs des propriétés non-fonctionnelles. À partir de ces valeurs, nous construisons notre matrice : services candidats / propriétés

non-fonctionnelles demandées. Enfin, nous effectuons la réconciliation non-fonctionnelle pour sélectionner et classer les services qui se rapprochent des exigences de l'activité métier.

➤ *Réconciliation fonctionnelle*

À partir de notre activité métier modélisée et annotée dans Petals BPM-NFR, nous appliquons un premier filtre fonctionnel (grâce à l'outil EasierSBS) afin de trouver les services (parmi ceux qui sont publiés dans EasierGov) qui répondent fonctionnellement aux besoins de cette activité. Dans la Figure 67, les résultats de l'application du filtre pour l'activité « Élargir le fichier clients » sont présentés.

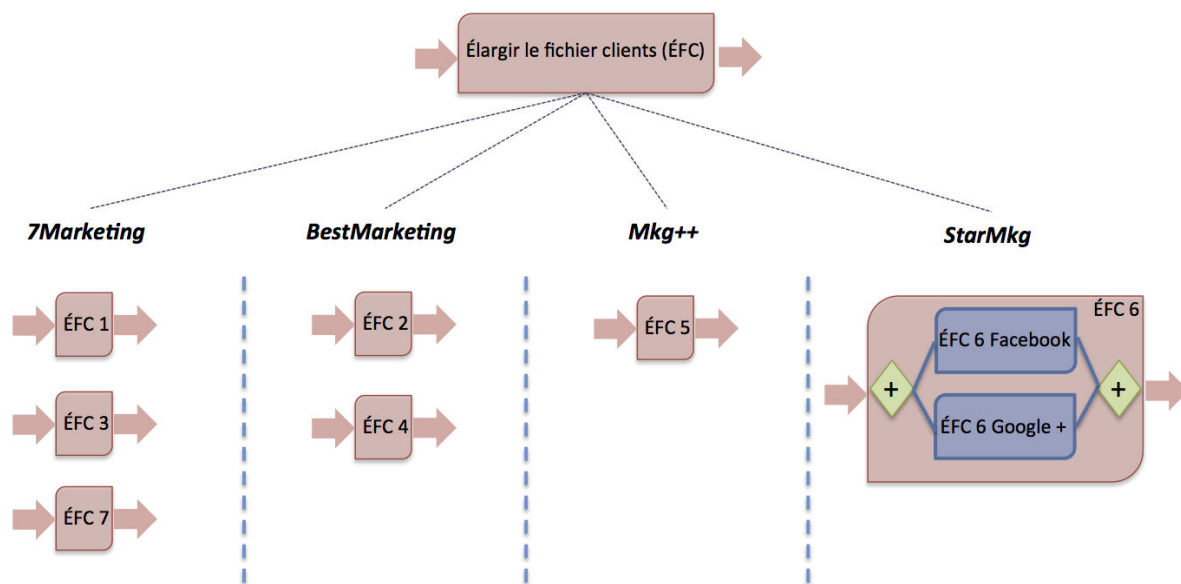


Figure 67 : Réconciliation fonctionnelle : activité « Élargir le fichier clients ».

Ainsi, comme nous le montre la Figure 67, il existe quatre entreprises candidates : 7Marketing, BestMarketing, Mkg++ et StarMkg. Les trois premières ont respectivement trois, deux et un services unitaires qui répondent à la fonctionnalité d'élargissement du fichier clients par le parcours des réseaux sociaux Facebook et Google+. Quant à la quatrième (StarMkg), elle n'en possède aucun mais de dispose de deux services distincts permettant de traiter cette fonctionnalité séparément (ÉFC 6 Facebook et ÉFC 6 Google +). Le premier offre la possibilité d'élargir le fichier clients par le parcours de Facebook et le deuxième s'occupe de Google +. La composition de ces deux services et leur exécution en parallèle aboutit au même résultat demandé.

➤ *Gestion des propriétés non-fonctionnelles*

Cette étape consiste à créer la matrice « services et compositions de services candidats ». Afin de réaliser celle-ci, pour chacun des services unitaires et des services de la composition nous analysons leurs templates SLA publiés dans EasierGov-NFR. Nous présentons dans la Figure 68 des extraits du

template SLA du service ÉFC1 contenant les propriétés non-fonctionnelles demandées au niveau métier (cf. Annexe 3 pour voir l'intégralité du template SLA).

```
<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Response Time</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Response Time"
        operator="&lt;">
        <wsqdl:MetricValue>
          <wsqdl:Value>50</wsqdl:Value>
          <wsqdl:Unit>second</wsqdl:Unit>
        </wsqdl:MetricValue>
      </wsag:CustomServiceLevel>
    </wsag:KPITarget>
  </wsag:ServiceLevelObjective>
</wsag:GuaranteeTerm>
```

```
<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Price</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Price"
        operator="&lt;">
        <wsqdl:ServiceCost>
          <wsqdl:ServicePrice>
            <wsqdl:Price>90</wsqdl:Price>
            <wsqdl:Unit>euros</wsqdl:Unit>
          </wsqdl:ServicePrice>
        </wsqdl:ServiceCost>
      </wsag:CustomServiceLevel>
    </wsag:KPITarget>
  </wsag:ServiceLevelObjective>
</wsag:GuaranteeTerm>
```

```
<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Privacy</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Privacy">
        <wsqdl:Security>
          <wsqdl:Prperty name="Privacy"></wsqdl:Prperty>
        </wsqdl:Security>
      </wsag:CustomServiceLevel>
    </wsag:KPITarget>
  </wsag:ServiceLevelObjective>
</wsag:GuaranteeTerm>
```

```
<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Integrity</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Integrity">
        <wsqdl:Security>
          <wsqdl:Prperty name="Integrity">
            </wsqdl:Prperty>
          </wsqdl:Security>
        </wsag:CustomServiceLevel>
    </wsag:KPITarget>
  </wsag:ServiceLevelObjective>
</wsag:GuaranteeTerm>
```

Figure 68 : Extraits du template SLA du service ÉFC 1 de l'entreprise 7Marketing.

Comme nous pouvons le remarquer dans la matrice ci-dessous, la propriété de l'authentification n'est pas assurée par le service ÉFC 1 car elle n'apparaît pas dans son template SLA (cf. Annexe 3).

Pour chacun des templates SLA, nous récupérons les valeurs des propriétés non-fonctionnelles (en nous limitant seulement à celles qui sont demandées par l'activité « Élargir le fichier clients », à savoir : *Response Time*, *Availability*, *Price*, *Privacy*, *Integrity*, et *Authentication*) et nous construisons notre matrice. Ainsi, nous obtenons la matrice suivante :

		<i>Response Time</i>	<i>Availability</i>	<i>Price</i>	<i>Privacy</i>	<i>Integrity</i>	<i>Authentication</i>	
Services / Compositions de services Candidats =	ÉFC 1	50 secondes	90 %	90 €	1	1	0	
	ÉFC 2	35 secondes	95 %	70 €	1	1	1	
	ÉFC 3	40 secondes	30 %	50 €	0	0	1	
	ÉFC 4	45 secondes	91 %	100 €	1	1	0	
	ÉFC 5	40 secondes	90 %	100 €	1	1	1	
	ÉFC 6	ÉFC 6 Facebook	45 secondes	100 %	45 €	1	1	1
		ÉFC 6 Google +	30 secondes	95 %	55 €	1	1	1
	ÉFC 7	36 secondes	100 %	85 €	0	0	0	

➤ Réconciliation non-fonctionnelle

À cette étape de notre démarche, nous avons d'une part les exigences non-fonctionnelles de l'activité métier, et d'autre part la matrice « candidats / propriétés non-fonctionnelles ». Nous la terminons par une phase de réconciliation non-fonctionnelle. Comme nous avons pu le voir dans la Figure 65, nous avons deux types de réconciliations : (i) une réconciliation 1-1 (pour les trois premières entreprises) et (ii) une réconciliation 1-N (pour le cas de l'entreprise StarMkg).

Nous commençons notre démarche par le traitement de la réconciliation 1-N (composition de service ÉFC 6) afin de la rendre de type 1-1. Pour y parvenir, nous commençons par faire une première étape de classification des propriétés non-fonctionnelles demandées au niveau métier.

❖ Étape 1 : Classification des valeurs des exigences non-fonctionnelles

Lors de la modélisation et l'annotation au niveau métier, six exigences non-fonctionnelles ont été choisies par l'architecte métier de l'entreprise Sazri :

- Response Time : une propriété non-fonctionnelle à minimiser de type valeur temporelle ;
- Availability : une propriété non-fonctionnelle à maximiser de type valeur numérique sous forme de pourcentage ;
- Price : une propriété non-fonctionnelle à minimiser de type monétaire ;

- Privacy : une propriété non-fonctionnelle à maximiser de type booléen ;
- Integrity : une propriété non-fonctionnelle à maximiser de type booléen ;
- Authentication : une propriété non-fonctionnelle à maximiser de type booléen.

❖ *Étape 2 : Transformation de la réconciliation 1-N en une réconciliation 1-1*

Nous appliquons les règles de transformation (cf. Chapitre 4, sous-section IV.5.2) et nous obtenons le vecteur suivant :

$$\text{ÉFC 6} = \begin{pmatrix} \text{Response Time} & \text{Availability} & \text{Price} & \text{Privacy} & \text{Integrity} & \text{Authentication} \\ 45 & 0,95 & 100 & 1 & 1 & 1 \end{pmatrix}$$

Dans la matrice *services-compositions de services candidates* obtenue, nous remplaçons les deux vecteurs lignes qui correspondaient aux deux services ÉFC Facebook et ÉFC Google + par le vecteur ÉFC 6 que nous venons d'obtenir par la transformation. Ainsi, nous obtenons la matrice des services candidats suivante :

	Response Time	Availability	Price	Privacy	Integrity	Authentication
ÉFC 1	50 secondes	90 %	90 €	1	1	0
ÉFC 2	35 secondes	95 %	70 €	1	1	1
ÉFC 3	40 secondes	30 %	50 €	0	0	1
ÉFC 4	45 secondes	91 %	100 €	1	1	0
ÉFC 5	40 secondes	90 %	100 €	1	1	1
ÉFC 6	45 secondes	95 %	100 €	1	1	0
ÉFC 7	36 secondes	100 %	85 €	0	0	0

À cette matrice nous appliquons l'algorithme de réconciliation non-fonctionnelle 1-1 en commençant par l'étape 2 (l'étape 1 de l'algorithme est la classification d'exigences non-fonctionnelles faites ci-dessus).

❖ *Étape 2 : Élimination des services candidats ayant des valeurs aberrantes*

Dans la matrice services candidats de l'activité « Élargir le fichier clients », il existe deux services qui ont au moins une valeur aberrante : (i) le service ÉFC 3 (disponibilité 30%, or l'architecte métier demande au moins 90% ; le service n'assure ni la confidentialité ni l'intégrité, or ces deux propriétés sont de poids d'exigences forts), et le service ÉFC 7 (il n'assure pas les propriétés de

confidentialité ni d'intégrité). Nous éliminons ces deux services de notre traitement de réconciliation non-fonctionnelle (leur élimination se fait de la matrice services candidats).

❖ *Étape 3 : Normalisation et centrage des services candidats par rapports aux exigences non-fonctionnelles de l'activité métier*

En suivant les différentes étapes de normalisation et de centrage décrites dans le chapitre IV, nous obtenons la matrice de services candidats normalisés et centrés par rapport aux exigences non-fonctionnelles de l'activité « Élargir le fichier clients » suivante :

	<i>Response Time</i>	<i>Availability</i>	<i>Price</i>	<i>Privacy</i>	<i>Integrity</i>	<i>Authentication</i>
<i>ÉFC 1</i>	-0.25	0	0.1	0	0	-1
<i>ÉFC 2</i>	0.125	0.0555	0.3	0	0	0
<i>ÉFC 4</i>	-0.125	0.0111	0	0	0	-1
<i>ÉFC 5</i>	0	0	0	0	0	0
<i>ÉFC 6</i>	-0.125	0.0555	0	0	0	-1

Services Candidats normalisés et centrés =

❖ *Étape 4 : Classement et sélection des services candidats*

À l'issue de l'étape de normalisation et centrage des services candidats, nous divisons la matrice obtenue à l'étape précédente en deux zones A et B. La zone A contient tous les services qui sont meilleurs que les exigences de l'activité métier sur toutes les propriétés non-fonctionnelles. Si nous reprenons notre cas d'étude, nous retrouvons dans cette sous-matrice les services ÉFC 2 et ÉFC 5. La zone B contient les autres services, c'est-à-dire : ÉFC 1, ÉFC 4 et ÉFC 6.

Nous appliquons l'algorithme de calcul de distance pondérée et de classement dédié à chacune de ces sous-matrices, et nous obtenons les résultats donnés par la Figure 69.

```
*****Step 4: Sorting of Candidate Services*****

Zone A : ÉFC2 ÉFC5
    Weighted Distance between ÉFC2 and Business Task Requirements: 0.005435570987654323
    Weighted Distance between ÉFC5 and Business Task Requirements: 0.0

Zone B : ÉFC1 ÉFC4 ÉFC6
    Weighted Distance between ÉFC1 and Business Task Requirements: 0.05312500000000006
    Weighted Distance between ÉFC4 and Business Task Requirements: 0.05078125
    Weighted Distance between ÉFC6 and Business Task Requirements: 0.05078125
    ÉFC6 : Weighted Distance (+) 1.5432098765432115E-4
    ÉFC4 : Weighted Distance (+) 6.172839506172797E-6

Candidate Services Ranking
1_ÉFC2
2_ÉFC5
3_ÉFC6
4_ÉFC4
5_ÉFC1
```

Figure 69 : Résultats de la réconciliation non-fonctionnelle de l'activité « Élargir le fichier clients ».

Ainsi, comme nous le montre la Figure 69, pour les deux services candidats de la zone A : le service ÉFC 2 a une distance pondérée supérieure à celle d'ÉFC 5, ce qui explique son classement en premier. En effet, la distance pondérée du service candidat ÉFC 5 est nulle, cela signifie que toutes les valeurs de ses propriétés non-fonctionnelles (uniquement celles qui ont été choisies au niveau métier) sont identiques à celles des exigences non-fonctionnelles de l'activité métier. Quant au service ÉFC 2, il répond mieux que ce qui a été demandé au niveau métier. Il a un temps de réponse inférieur, plus de disponibilité, et il est moins cher.

Pour les services de la zone B : les services ÉFC 4 et ÉFC 6 ont des distances pondérées identiques. C'est pour ça que nous avons calculé leurs distances pondérées (+) c'est-à-dire, en ne prenant en considération que leurs valeurs des propriétés non-fonctionnelles ≥ 0 . Ce calcul a montré que la composition de services ÉFC 6 est plus proche des exigences non-fonctionnelles métier que le service ÉFC 4. Finalement, le service ÉFC 1 se classe en dernier car il a la plus grande distance pondérée de toute la zone B.

V.4. Conclusion

Dans ce chapitre, nous avons présenté les détails de l'implémentation réalisée. Cette implémentation porte sur (i) notre approche de modélisation et d'annotation non-fonctionnelle du processus métier collaboratif « Petals BPM-NFR » et (ii) notre démarche de gouvernance SOA

permettant la gestion des services et de leurs propriétés non-fonctionnelles, et la réconciliation non-fonctionnelle « EasierGov-NFR ».

Ce chapitre nous a également permis d'illustrer nos contributions sur un cas d'étude concernant la collaboration entre deux entreprises (où un SI de médiation gère cette collaboration). Ainsi, après avoir détaillé le cas d'étude et expliqué les diverses attentes non-fonctionnelles de l'architecte métier, nous avons montré qu'il était judicieux de disposer d'une plateforme de modélisation permettant à la fois de modéliser les activités métier graphiquement et de les annoter par des exigences non-fonctionnelles afin de répondre au mieux aux attentes métier. Ensuite, nous avons modélisé et annoté non-fonctionnellement le processus métier collaboratif en utilisant la plateforme Petals BPM-NFR et en respectant l'approche proposée dans le Chapitre III. Enfin, à partir de ce processus et en utilisant EasierGov-NFR, nous avons appliqué notre démarche (dans le respect de celle proposée dans les Chapitres III et IV) de gouvernance SOA pour effectuer la réconciliation non-fonctionnelle entre le niveau métier et le niveau technique.

Cependant, plusieurs améliorations sont possibles, notamment par rapport à l'ergonomie de nos deux prototypes. De plus, bien que nous ayons essayé d'optimiser notre démarche de gouvernance SOA, cette dernière peut être largement améliorée et étendue par l'ajout de la réconciliation N-M (plusieurs activités métier – plusieurs services).

Conclusion générale et perspectives

« *Que conclure à la fin de tous mes longs propos ? (...)* »

Voltaire

1. Rappel du contexte et des objectifs de la thèse

La collaboration inter-organisationnelle devient essentielle à la croissance, à l'innovation et à la recherche de l'excellence de toute entreprise. Le système d'information de médiation est au cœur de la réussite de cette collaboration et se doit d'être aussi bien robuste que flexible pour supporter les problématiques métier. Les exigences non-fonctionnelles ne peuvent être écartées ou même négligées lors de la réflexion sur la conception d'un tel SI de médiation. En effet, leur prise en considération est essentielle pour toute démarche de réconciliation entre le métier et le technique au sein du SI dans la mesure où elles peuvent modifier amplement le choix des services. Ceci nous a menés aux questionnements suivants : quelle solution conceptuelle puisée dans les exigences non-fonctionnelles pourrait répondre aux besoins d'annotation des processus métier collaboratifs ? Comment assurer la rationalisation de la gestion des services et de leurs propriétés non-fonctionnelles ? Comment réconcilier non-fonctionnellement les activités du processus métier et les services / compositions de services disponibles (parmi ceux qui répondent aux besoins fonctionnels) ?

2. Synthèse sur les contributions de ces travaux

Afin de répondre à ces questions et de relier cette problématique de passage du niveau métier au technique, nous avons amorcé notre réflexion par une étude de l'existant concluant sur une analyse synthétique présentant les limites et les carences des pratiques actuelles autour de la modélisation et de l'annotation non-fonctionnelle graphique des processus métier collaboratifs, de la rationalisation de la gestion des propriétés non-fonctionnelles des services et enfin, de la réconciliation entre les besoins métier et les services techniques.

Comme première contribution, nous avons présenté une nouvelle démarche de modélisation et d'annotation graphiques des activités du processus métier collaboratif BPMN 2.0 par des exigences non-fonctionnelles (cf. Chapitre III) que nous avons implémenté au sein de l'outil Petals BPM-NFR. Dans ce manuscrit, nous avons souligné le besoin de l'interopérabilité dans un contexte collaboratif,

l'utilisation des standards s'avère essentielle pour garantir ce besoin. Après avoir établi une analyse comparative des différentes exigences non-fonctionnelles existantes et par absence d'un standard permettant d'annoter non-fonctionnellement un processus BPMN 2.0, nous avons opté pour la spécification d'Oasis « WSQF » pour l'annotation. Si elle est initialement dédiée à la modélisation des propriétés non-fonctionnelles des services Web, nous avons montré comment elle peut aussi être également appliquée au niveau métier.

Le deuxième apport des travaux de cette thèse consiste à rationaliser la gestion des services et leurs propriétés non-fonctionnelles par la gouvernance SOA (cf. Chapitre III). En effet, la prolifération des services au sein du système d'information rend cette rationalisation nécessaire. Nous avons présenté une approche d'un registre évolué de gouvernance SOA basée sur la spécification USDL pour la gestion des services et sur le standard WSQF pour la gestion de leurs propriétés non-fonctionnelles. Ce registre de gouvernance, EasierGov-NFR, permet aux fournisseurs non seulement de publier et de gérer leurs services, mais également de gérer leurs métadonnées : les propriétés non-fonctionnelles et les modèles de contrats de services. Nous avons proposé que ces modèles respectent le standard WS-Agreement et que ce dernier soit étendu par la spécification de WSQF afin de permettre aux fournisseurs d'ajouter les propriétés non-fonctionnelles techniques relatives à leurs services.

En outre, nous avons montré que ce registre de gouvernance SOA, grâce à sa liaison directe à l'outil de modélisation et d'annotation non-fonctionnelle graphiques, permet le support de la réconciliation non-fonctionnelle entre les activités du processus métier collaboratif et les services / compositions de services techniques disponibles dans le registre et qui répondent aux besoins fonctionnels. En effet, nous avons défini dans ce registre une librairie de réconciliation non-fonctionnelle. À travers cette librairie, nous avons présenté, dans le chapitre IV, une idée pour illustrer l'analyse multicritère afin d'établir une réconciliation non-fonctionnelle unitaire (1-1) entre les activités métier et les services techniques. La troisième contribution de cette thèse est dédiée à la réconciliation non-fonctionnelle 1-N (cf. Chapitre IV). En effet, cette réconciliation permet de sélectionner une composition de services (qui ensemble répondent aux besoins demandés) pour une activité métier. Cependant, dans une optique de réutilisation, de minimisation des coûts et de facilitation de la mise en œuvre, nous avons proposé une approche permettant de transformer une problématique de réconciliation 1-N en une problématique de réconciliation 1-1. Pour cela, nous avons proposé un mécanisme comportant un ensemble de règles à suivre et à respecter afin d'établir ce passage.

Les différents mécanismes présentés ont été implémentés en respectant les règles industrielles de développement définies par l'équipe Petals de Linagora Labs. En outre, nous avons montré l'applicabilité de notre approche à travers un cas d'étude fictif (cf. Chapitre V) entre deux entreprises : la première, spécialisée dans la vente en ligne et la deuxième dans le marketing prédictif afin d'augmenter les ventes de la première.

3. Perspectives

« La bataille contre l'ignorance se gagne tous les jours et elle finit par ouvrir sur des perspectives insoupçonnées. » Dalai Lama

Les solutions conceptuelles présentées dans ce manuscrit de thèse autour de la (i) modélisation et d'annotation non-fonctionnelle et de la (ii) gouvernance SOA pour la rationalisation des services et de leurs propriétés non-fonctionnelles et pour la réconciliation non-fonctionnelle, ont donné lieu à des propositions et à des réflexions qui ouvrent des perspectives de travaux futurs pouvant être envisagés :

Étendre la modélisation et l'annotation non-fonctionnelle

Dans nos travaux, nous basons notre approche de modélisation et d'annotation non-fonctionnelle principalement sur les activités du processus métier collaboratif. Cependant, l'expression des exigences non-fonctionnelles est limitée. Or, les architectes métier / utilisateurs peuvent avoir besoin d'annoter une partie du processus (un ensemble d'activités), tous les éléments du processus BPMN 2.0, voire le processus en entier. Il serait souhaitable, lors de la phase de modélisation des processus collaboratifs, de prendre en considération ces types d'annotation non-fonctionnelle et de donner la possibilité aux utilisateurs d'annoter graphiquement leurs besoins non-fonctionnels sans se limiter aux activités. Une amélioration de l'ergonomie de notre plateforme Petals BPM-NFR lors de l'annotation non-fonctionnelle peut être également envisageable.

Étendre la gestion des services et leurs propriétés non-fonctionnelles

Les approches actuelles de gestion des propriétés non-fonctionnelles des services se basent sur la définition d'une manière explicite de leurs valeurs par les fournisseurs dans les modèles de contrats SLA. Des réflexions pourraient être conduites sur la conception d'approches de contrôle pour la détermination de certaines valeurs des propriétés non-fonctionnelles en se référant à l'environnement de l'exécution des services. Par exemple, la disponibilité d'un service sera déduite ou mise à jour automatiquement à l'issue de plusieurs observations lors de son exécution.

Étendre la réconciliation non-fonctionnelle métier / technique

Nous avons fait dans nos travaux l'hypothèse que les exigences non-fonctionnelles demandées au niveau métier sont toutes renseignées au niveau technique par le fournisseur du service. Cependant, ceci peut ne pas être toujours le cas. Par conséquent, il est donc nécessaire de prévoir et de gérer ces cas de figure lors de la réconciliation non-fonctionnelle. En relation avec nos réflexions présentées au

niveau de la modélisation et de l'annotation non-fonctionnelle, il faudrait envisager d'ajouter un troisième type de réconciliation : le type N-M. La réconciliation N-M devrait permettre de réconcilier « n » activités métier avec « m » services techniques.

Comme nous l'avons présenté dans ce manuscrit, nos travaux s'intègrent dans le projet pluriannuel MISE qui vise à concevoir un système d'information de médiation générique facilitant la collaboration, peu importe le domaine ou les partenaires mis en jeu. La Figure 70 représente les différents travaux réalisés, en cours et qui démarrent sur ce projet au cours des trois itérations et de douze thèses. Cette thèse s'intègre avec la seconde itération du projet qui vise à automatiser autant que possible la conception du système d'information de médiation. Elle se situe au niveau de la transformation du modèle dynamique et aide à la décision, depuis la définition de la cartographie de processus collaboratifs (processus métier) en modèles de système d'information de médiation (des processus techniques).

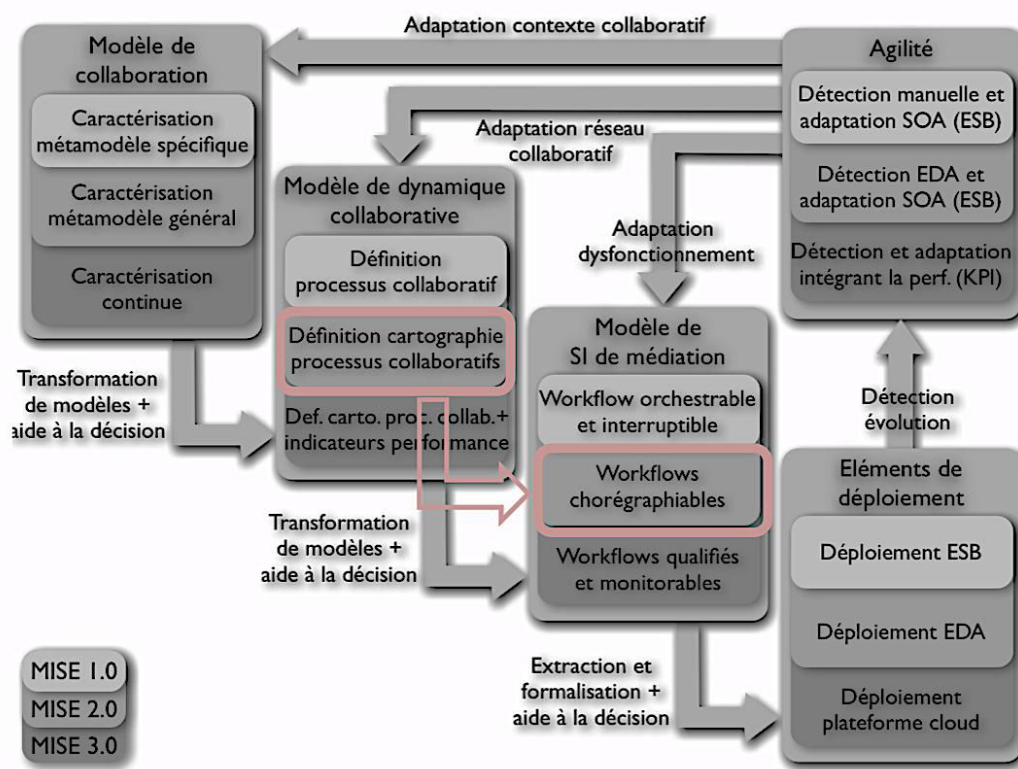


Figure 70 : Positionnement par rapport aux travaux du projet MISE.

La recherche scientifique et l'innovation industrielle ont constitué les maillons les plus importants de nos réflexions tout au long de ces travaux de thèse de doctorat. En effet, au-delà de

l'approfondissement de nos connaissances scientifiques, nous avons également essayé de chercher des nouvelles possibilités pour l'amélioration industrielle.

C'est dans cette quête pour favoriser l'originalité scientifique et valoriser l'innovation industrielle, nous avons essayé de positionner nos contributions (cf. Figure 71). L'utilisation d'un même standard (WSQF) pour l'annotation non-fonctionnelle au niveau des processus métier et au niveau technique (via les modèles de contrats de services) constitue un avantage industriel majeur. Cet avantage permet de minimiser fortement la complexité de la réconciliation et par suite une réduction des coûts importante. L'enrichissement de modèles de contrats SLA de type WS-Agreement par WSQF constitue aussi un apport scientifique du fait de l'utilisation d'un modèle normalisé contenant une liste assez exhaustive de propriétés non-fonctionnelles. Enfin, de nombreux travaux de recherche existent dans la littérature autour de la sélection non-fonctionnelle de services. Toutefois, les activités des processus métier peuvent ne pas être couvertes par un seul service. Pour cela, nous avons proposé un mécanisme de réconciliation non-fonctionnelle de compositions de services « 1-N » via l'utilisation de la gouvernance SOA.

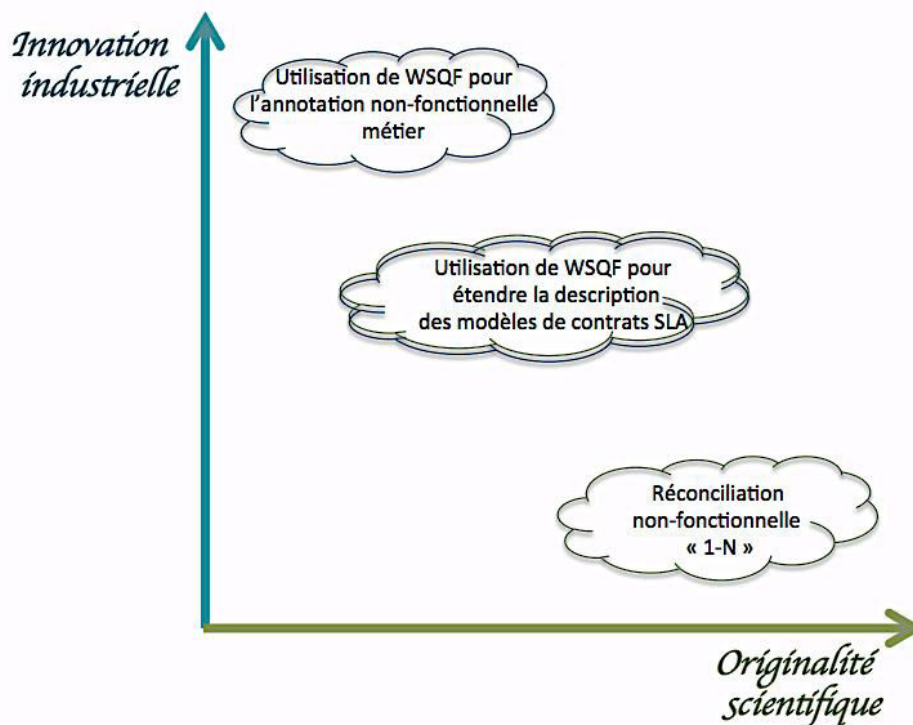


Figure 71 : *Innovation industrielle vs Originalité scientifique.*

BIBLIOGRAPHIE

- [Afshar et al., 2007] Afshar, M., Cincinatus, M., Hynes, D., Clugage, K., & Patwardhan, V. (2007). *Soa governance: Framework and best practices*. White Paper, Oracle Corporation May.
- [Alonso et al., 2004] Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). *Web services*, (pp. 123-149). Springer Berlin Heidelberg.
- [Andrieux et al., 2007] Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., & Xu, M. (2007, March). *Web services agreement specification (WS-Agreement)*. In Open Grid Forum (Vol. 128).
- [Badr et al., 2008] Badr, Y., Abraham, A., Biennier, F., & Grosan, C. (2008, October). *Enhancing web service selection by user preferences of non-functional features*. In Next Generation Web Services Practices, 2008. NWESP'08. 4th International Conference on (pp. 60-65). IEEE.
- [Baligand, 2008] Baligand, F. (2008). *Une approche déclarative pour la gestion de la qualité de service dans les compositions de service* (Doctoral dissertation, École Nationale Supérieure des Mines de Paris).
- [Bennett et al., 2011] Bennett, S., Erl, T., Gee, C., Laird, R., Manes, A. T., Schneider, R. & Venable, C. (2011). *SOA Governance: Governing Shared Services On-Premise & in the Cloud*. PrenticeHall/PearsonPTR.
- [Bénaben et al., 2007] Bénaben, F., Touzi, J., Rajsiri, V. et Pingaud, H. (2007) : *L'Interopérabilité des systèmes d'information comme moyen vers l'intégration de l'écosystème industriel*, 7e Congrès international de génie industriel, Trois-Rivières, Québec.
- [Bénaben et al., 2008] Bénaben, F., Touzi, J., Rajsiri, V., Truptil, S., Lorré, J. P., & Pingaud, H. (2008). *Mediation information system design in a collaborative SOA context through a MDD approach*. In Proceedings of the First International Workshop on Model Driven Interoperability for Sustainable Information Systems (MDISIS'08) held in conjunction with the CAiSE (Vol. 8).
- [Bénaben et al., 2010] Bénaben, F., Mu, W., Truptil, S., Lorré, J. P. et Pingaud, H. (2010): *Information Systems design for emerging ecosystems: A model-driven approach diving into abstraction layers*, IEEE-DEST'10, Springer-Verlag, Dubai, EAU.
- [Bénaben, 2012] Bénaben, F. (2012). *Conception de Systèmes d'Information de Médiation pour la prise en charge de l'Interopérabilité dans les collaborations d'organisations*. Habilitation à diriger les recherches, INPT - EMAC, Albi.

- [Bénaben et al., 2013] Bénaben, F., Boissel-Dallier, N., Pingaud, H., & Lorre, J. (2013). *Semantic issues in model-driven management of information system interoperability*. International Journal of Computer Integrated Manufacturing, 26(11), 1042-1053.
- [Bia-Figueiredo et al., 2011] Bia-Figueiredo, M., Gillette, Y., & Morley, C. (2011). *Processus métiers et SI-Gouvernance, management, modélisation-3e édition: Gouvernance, management, modélisation*. Dunod.
- [Bieberstein, 2006] Bieberstein, N. (Ed.). (2006). *Service-oriented architecture (SOA) compass: business value, planning, and enterprise roadmap*. FT Press.
- [Boissel-Dallier, 2012] Boissel-Dallier, N. (2012). *Aide à la conception d'un système d'information de médiation collaboratif : de la cartographie de processus métier au système exécutable*, Thèse de doctorat, Institut National Polytechnique, Toulouse.
- [Booch et al., 1998] Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The Unified Modeling Language (UML)*. World Wide Web: [http://www.rational.com/uml/\(UML Resource Center\)](http://www.rational.com/uml/(UML Resource Center)), 94.
- [Boutillier et al., 2002] Boutillier, M., Labye, A., & Lagoutte, C. (2002). *Financement et gouvernement des entreprises*. Revue d'économie politique, 112(4), 499-544.
- [Brandenburg et al., 2006] Brandenburg, H., & Wojtyna, J. P. (2006). *L'approche processus: mode d'emploi*. Editions Eyrolles.
- [Bray et al., 1997] Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (1997). *Extensible Markup Language (XML)*. World Wide Web Journal, 2(4), 27-66.
- [Bresciani et al., 2002] Bresciani, P. & Giorgini, P. (2002). The Tropos *analysis process as graph transformation system*. In Proceedings of the Workshop on Agent-oriented Methodologies at OOPSLA, Seattle, USA, 1-12.
- [Brown et al., 2006] Brown, W. A., Moore, G., & Tegan, W. (2006). *SOA Governance-IBM's Approach*. Somers, NY.
- [Cardoso et al., 2004] Cardoso, J., Sheth, A., Miller, J., Arnold, J., & Kochut, K. (2004). *Quality of service for workflows and web service processes*. Web Semantics: Science, Services and Agents on the World Wide Web, 1(3), 281-308.
- [Cattan, 2000] Cattan, M. (2000). *Management des processus: une approche innovante*. AFNOR.
- [CentraSite, 2010] CentraSite. (2010). *A little structure goes a long way Software AG*. Available on: <http://www.softwareag.com/corporate/products/wm/soagovernance/centrasite/partner/default.asp>

- [Chappell, 2004] Chappell, D. A. (2004). *Enterprise service bus*. O'Reilly Media, Inc.
- [CHOReOS, 2010] CHOReOS Project. (2010). *Large Scale Choreographies for the Future Internet*. Available on: www.choreos.eu/
- [Christensen et al., 2001] Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). *Web Services Description Language (WSDL)*.
- [Chung et al., 1995] Chung, L., & Nixon, B. A. (1995, April). *Dealing with non-functional requirements: three experimental studies of a process-oriented approach*. In *Software Engineering, 1995. ICSE 1995. 17th International Conference on* (pp. 25-25). IEEE.
- [Chung et al., 2000] Chung, L., Nixon, B., Yu, E., & Mylopoulos, J. (2000). *Non-functional Requirements*. *Software Engineering*.
- [Chung et al., 2009] Chung, L., & do Prado Leite, J. C. S. (2009). *On non-functional requirements in software engineering*. In *Conceptual modeling: Foundations and applications* (pp. 363-379). Springer Berlin Heidelberg.
- [Cosminexus, 2010] Cosminexus, H, (2010). Available on: <http://www.hitachi.co.jp/Prod/comp/soft1/global/prod/cosminexus/>
- [Crusson, 2003] Crusson, T. (2003). *Business Process Management: de la modélisation à l'exécution- Positionnement par rapport aux Architectures Orientées Services*. White paper. Intalio white paper
- [Cysneiros et al., 2004] Cysneiros, L.M., & Yu, E. (2004, May). *Addressing Agent Autonomy in Business Process Management – with case studies on the patient discharge process*. Proc. of the Information Resources Management Association Conference, New Orleans, USA.
- [Dan et al., 2008] Dan, A., Johnson R.D., & Carrato, T. (2008, May). *SOA service reuse by design*. In *Proceedings of the 2nd international workshop on Systems development in SOA environments* (pp.25-28). ACM, New York, USA.
- [Debauche et al., 2004] Debauche, B., & Mégard, P. (2004). *BPM: Business Process Management: pilotage métier de l'entreprise*. Hermès science publications.
- [Demirors et al., 2003] Demirors, O., Gencel, C., & Tarhan, A. (2003). *Utilizing Business Process Models for Requirements Elicitation*. In *Euromicro Conference: New Waves in Systems Architecture. Proceedings. 29th* (pp. 409-412).IEEE Computer Society.
- [Deora et al., 2003] Deora, V., Shao, J., Gray, W. A., & Fiddian, N. J. (2003). *A quality of service management framework based on user expectations*. In *Service-Oriented Computing- ICSOC 2003* (pp. 104-114). Springer Berlin Heidelberg.

- [Derler et al., 2007] Derler, P., & Weinreich, R. (2007). *Models and tools for SOA governance*. In Trends in Enterprise Application Architecture (pp. 112-126). Springer Berlin Heidelberg.
- [Desouza, 2006] Desouza, K. (Ed.). (2006). *Agile Information Systems*. Routledge.
- [Dewsbury, 2007] Dewsbury, R. (2007). *Google web toolkit applications*. Pearson Education.
- [De Backer et al., 2005] De Backer, M., & Snoeck, M. (2005). *Deterministic Petri net languages as business process specification language*. DTEW Research Report 0577, 1-21.
- [De Courcy, 1992] De Courcy, R. (1992). *Les systèmes d'information en réadaptation*. Québec, Réseau International CIDIH et facteurs environnementaux, 1(5), 7-10.
- [De Oliveira Barata, 2003] De Oliveira Barata, M.. (2003). *Étymologie du terme « gouvernance »*, http://ec.europa.eu/governance/docs/doc5_en.pdf
- [Erl, 2004] Erl, T. (2004). *Service-oriented architecture: a field guide to integrating XML and web services*. Prentice Hall PTR.
- [Galaxy, 2009] Galaxy MuleSoft, (2009). Available on: <http://www.mulesoft.org/documentation/display/GALAXY/Home>
- [GIF, 2010] *HP SOA Governance Interoperability Framework (GIF) - Hewlett-Packard Development Company, L.P.* (2010). Available on: https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-130-27^2804_4000_100__
- [Glinz, 2007] Glinz, M. (2007, October). *On non-functional requirements*. In Requirements Engineering Conference, 2007. RE'07. 15th IEEE International (pp. 21-26). IEEE.
- [Godart et al., 2009] Godart, C., & Perrin, O. (2009). *Les processus métiers: Concepts, modèles et systèmes*. (p. 363). Hermès.
- [Gorton et al., 2006] Gorton, S., & Reiff-Marganiec, S. (2006). *Towards a Task-Oriented, Policy-Driven Business Requirements Specification for Web Services*. Proc. of 4th International Conference on Business Process Management. (pp. 465-470). Vienna, Austria.
- [Greeff et al., 2004] Greeff, G., & Ghoshal, R. (2004). *Practical E-manufacturing and supply chain management*. Newnes.
- [Guo et al., 2011] Guo, G., Yu, F., Chen, Z., & Xie, D. (2011). *A method for semantic Web service selection based on QoS ontology*. Journal of Computers, 6(2), 377-386.

- [Heinrich et al., 2011] Heinrich, R., Kappe, A., & Paech, B. (2011). *Modeling Quality Information within Business Process Models*. In Proceedings of the SQMB'11 Workshop, TUM-I1104 (pp. 4-13).
- [Hepp et al., 2007] Hepp, M., & Roman, D. (2007). *An Ontology Framework for Semantic Business Process Management*. Proc. of International Conference on Wirtschaftsinformatik (Commercial Informatics), WI, Karlsruhe, Germany.
- [Jardim-Goncalves et al., 2006] Jardim-Goncalves, R., Grilo A., & Steiger-Garcao, A. (2006). *Challenging the interoperability between computers in industry with MDA and SOA*. Computers in Industry, vol 57, Issues 8-9, p. 679-689, ISSN 0166-3615.
- [Juric et al., 2006] Juric, M. B., Mathew, B., & Sarang, P. G. (2006). *Business Process Execution Language for Web Services: An Architect and Developer's Guide to Orchestrating Web Services Using BPEL4WS*. Packt Publishing Ltd.
- [Keller et al., 2003] Keller, A., & Ludwig, H. (2003). *The WSLA framework: Specifying and monitoring service level agreements for web services*. Journal of Network and Systems Management, 11(1), 57-81.
- [Kim et al., 2005] Kim, I. Y., & De Weck, O. L. (2005). *Adaptive weighted-sum method for bi-objective optimization: Pareto front generation*. Structural and multidisciplinary optimization, 29(2), 149-158.
- [Khaled et al., 2005] Khaled, C. V. E. O. H., & Khan, K. (2005, March). *Addressing non-functional properties in software architecture using adl*. In The Sixth Australian Workshop on Software and System Architecture (p. 6).
- [Koehler et al., 2007] Koehler, J., & Vanhatalo, J. (2007). *Process anti-patterns: How to avoid the common traps of business process modeling*. IBM WebSphere Developer Technical Journal, 10(2), 4.
- [Konstantas et al., 2006] Konstantas, D., Bourrières, J. P., Léonard, M., & Boudjlida, N. (2006). *Interoperability of enterprise software and applications*. Springer Science & Business Media.
- [Krafzig et al., 2005] Krafzig, D., Banke, K., & Slama, D. (2005). *Enterprise SOA: service-oriented architecture best practices*. Prentice Hall Professional.
- [Kritikos et al., 2006] Kritikos, K., & Plexousakis, D. (2006, December). *Semantic qos metric matching*. In Web Services, 2006. ECOWS'06. 4th European Conference on (pp. 265-274). IEEE.
- [Lesbegueries et al., 2012] Lesbegueries, J., Ben Hamida, A., Salatgé, N., Zribi, S., & Lorré, J. P. (2012, July). *Multilevel event-based monitoring framework for the petals enterprise service bus: industry article*. In Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems (pp. 48-57). ACM.

- [Liu et al., 2004] Liu, Y., Ngu, A. H., & Zeng, L. Z. (2004, May). *QoS computation and policing in dynamic web service selection*. In Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (pp. 66-73). ACM.
- [Lu et al., 2006] Lu, R., Sadiq, S., Padmanabhan, V., & Governatori, G. (2006): *Using a Temporal Constraint Network for Business Process Execution*. In Proceedings of the 17th Australasian Database Conference, volume 49 (pp.157–166). ACM Conference Series.
- [Luzeaux et al., 2008] Luzeaux, D., & Ruault, J. R. (2008). *Systèmes de systèmes, concepts et illustrations pratiques*, number ISBN 978-2-7462-1875-8. Hermes Science, June.
- [Mammeri, 2005] Mammeri, Z. (2005). *Qualité de Service dans les réseaux : problématique, solutions et challenges*, Ecole d'Été Temps réel ETR'2005 - Nancy, vol. ISBN: 2 905267-47-X. pp. 207-223.
- [Marks, 2008] Marks, E. A. (2008). *Service-oriented architecture (SOA) governance for the services driven enterprise*. John Wiley & Sons.
- [Master, 2009] *Petals Master SOA Governance Solution* (2009). Available on: <http://petalsmaster.ow2.org/>
- [Matsumura, 2007] Matsumura, M. (2007). *The Definitive Guide to SOA Governance and Lifecycle Management*. WebMethods Whitepaper, March.
- [Meyer, 2011] Meyer, D., M. (2011). *Build your own activiti task explorer with CDI and JSF 2*. Available on: <http://www.bpm-guide.de/2011/09/17/build-your-own-activiti-task-explorer-with-cdi-and-jsf-2/>
- [Miller et al., 2003] Miller, J., & Mukerji, J. (2003). *MDA Guide Version 1.0. 1*. Object Management Group, 234, 51.
- [Morley et al., 2005] Morley, C., Hugues, J., Leblanc, B., & Hugues, O. (2005). *Processus métiers et systèmes d'information*. Dunod, Paris.
- [Mu, 2012] Mu, W. (2012). *Caractérisation Métier et Logique d'Une Situation Collaborative*, Thèse de doctorat. INP Toulouse-Ecole des Mines d'Albi-Carmaux.
- [Murata, 1989] Murata, T. (1989). *Petri nets: Properties, analysis and applications*. Proceedings of the IEEE, 77(4), 541-580
- [Mylopoulos et al., 1992] Mylopoulos, J., Chung, L., & Nixon, B. (1992). *Representing and using nonfunctional requirements: A process-oriented approach*. Software Engineering, IEEE Transactions on, 18(6), 483-497.

- [Newcomer et al., 2004] Newcomer, E., & Lomow, G. (2004). *Understanding SOA with web services (Independent Technology Guides)*. Addison-Wesley Professional.
- [OASIS, 2000] ORG, U. (2000, September). UDDI Executive White Paper. OASIS.
- [OASIS, 2007] OASIS Standard. (2007) *WSBEPL Ver. 2.0*. Available on: <http://docs.oasis-open.org/wsbpel/2.0/os/wsbpel-v2.0-OS.html>.
- [OASIS, 2010] OASIS (2010). *Web Services Quality Factors (WSQF)*. Available on: <http://docs.oasis-open.org/wsqm/wsqf/v1.0/WS-Quality-Factors-v1.0-cd02.html>.
- [O'Brien et al., 2007] O'Brien, L., Merson, P., & Bass, L. (2007, May). *Quality attributes for service-oriented architectures*. In Proceedings of the international Workshop on Systems Development in SOA Environments (p. 3). IEEE Computer Society.
- [OMG, 2011] OMG, B. P. M. (2011). *Business Process Model and Notation (BPMN) Version 2.0 Specification*. Available on: <http://www.omg.org/spec/BPMN/2.0/>
- [Papazoglou et al., 2007] Papazoglou, M. P., Traverso, P., Dustdar, S., & Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. *Computer*, 40(11), 38-45.
- [Papazoglou et al., 2008] Papazoglou, M., Traverso, P., Dustdar, S., & Leymann, F. (2008). *Service-oriented computing: A research roadmap*. *International Journal of cooperative Information Systems*, vol. 17, p.223-255.
- [Pavlovski et al., 2008] Pavlovski, C. J., & Zou, J. (2008, January). *Non-functional requirements in business process modeling*. In Proceedings of the fifth Asia-Pacific conference on Conceptual Modelling-Volume 79 (pp. 103-112). Australian Computer Society, Inc.
- [Pingaud, 2009] Pingaud, H. (2009). *Rationalité du développement de l'interopérabilité dans les organisations*. In Management des technologies organisationnelles. Presses des MINES.
- [Peltz, 2003] Peltz, C. (2003). *Web services orchestration and choreography*. *Computer*, 36(10), 46-52.
- [Petri, 1962] Petri, C. A. (1962). *Kommunikation mit automaten*.
- [Rajsiri, 2009] Rajsiri, V. (2009). *Knowledge-based system for collaborative process specification*. Thèse en systèmes industriels. INP Toulouse-Ecole des Mines d'Albi-Carmaux..
- [Rajsiri et al., 2011] Rajsiri, V., Fleury, N., Crosmarie, G., & Lorré, J. P. (2011). *Event-based business process editor and simulator*. In Business Process Management Workshops (pp.707–

- 718). Springer Berlin Heidelberg.
- [Ran, 2003] Ran, S. (2003). *A model for web services discovery with QoS*. ACM Sigecom exchanges, 4(1), 1-10.
- [Raymond, 2007] Raymond, G. (2007). *Architecture logique, principes, structures et bonnes pratiques/* 2007]
- [Rodriguez et al., 2007] Rodriguez, A., Fernandez-Medina, E., & Piattini, M. (2007). *A BPMN extension for the modeling of security requirements in business processes*. IEICE transactions on information and systems, 90(4), 745-752.
- [Rosa et al., 2002] Rosa, N. S., Cunha, P. R., & Justo, G. R. (2002). *Process NFL: a language for describing non-functional properties*. In System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on (pp. 3676-3685). IEEE.
- [Rouse, 2007] Rouse, W. B. (2007). *Agile information systems for agile decision-making*. Agile Information Systems (pp. 16-30). Elsevier, New York.
- [Roy, 1978] Roy, B. (1978). *ELECTRE III: Un algorithme de classements fondé sur une représentation floue des préférences en présence de critères multiples*. Cahiers du CERO, 20(1), 3-24.
- [Saaty, 1988] Saaty, T. L. (1988). What is the analytic hierarchy process? (pp. 109-121). Springer Berlin Heidelberg.
- [Santos et al., 2010] Santos, E., Pimentel, J., Castro, J., Sánchez, J., & Pastor, O. (2010). *Configuring the variability of business process models using non-functionality requirements*. In Enterprise, Business-Process and Information Systems Modeling (pp. 274-286). Springer Berlin Heidelberg.
- [Schepers et al., 2008] Schepers, T. G., Iacob, M. E., & Van Eck, P. A. (2008, March). *A lifecycle approach to SOA governance*. In Proceedings of the 2008 ACM symposium on Applied computing (pp. 1055-1061). ACM.
- [Stephen, 2004] Stephen, A. (2004). *Introduction to BPMN*. White, IBM Cooperation, 2008-029.
- [Systinet, 2009] Systinet (2009). *Introduction to Systinet SOA Governance and Lifecycle Management*. Available on: <https://www.oasis-open.org/committees/download.php/17235/Systinet%20Registry%20NASA%20Presentation%203-15-06.pdf>
- [Sun et al., 2005] Sun, W., Xu, Y., & Liu, F. (2005, June). *The role of XML in service level agreements management*. In Services Systems and Services Management. Proceedings of ICSSSM'05. 2005 International Conference on (Vol. 2, pp. 1118-1120). IEEE.

- [Théroutde, 2002] Théroutde, F. (2002). *Formalisme et système pour la représentation et la mise en œuvre des Processus de pilotage des relations entre donneurs d'ordre et fournisseurs* (Doctoral dissertation, Grenoble, INPG).
- [Tidwell, 2000] Tidwell, D. (2000). *Web services-the web's next revolution*. IBM developer Works.
- [Touzi, 2007] Touzi, J. (2007). *Aide à la conception de système d'information collaboratif, support de l'interopérabilité des entreprises* (Doctoral dissertation, INPT – EMAC).
- [Tosic et al., 2002] Tosic, V., Patel, K., & Pagurek, B. (2002). *Wsol - web service offerings language*. In *Web Services, E-Business, and the Semantic Web* (pp. 57-67). Springer Berlin Heidelberg.
- [Tran et al., 2008] Tran, T., Haase, P., Motik, B., Grau, B. C., & Horrocks, I. (2008, July). *Metalevel Information in Ontology-Based Applications*. In *AAAI* (Vol. 8, pp. 1237-1242).
- [Truptil, 2011] Truptil, S. (2011). *Etude de l'approche de l'interopérabilité par médiation dans le cadre d'une dynamique de collaboration appliquée à la gestion de crise*. Thèse en systèmes industriels, INPT – EMAC.
- [Van Der Aalst et al., 2003] Van Der Aalst, W. M., Ter Hofstede, A. H., & Weske, M. (2003). *Business process management: A survey* (pp. 1-12). Springer Berlin Heidelberg.
- [WebSphere, 2010] IBM (2010). *WebSphere Service Registry and Repository*. Available on; <http://www-01.ibm.com/software/integration/wsrr/>
- [Weske, 2012] Weske, M. (2012). *Business process management: concepts, languages, and architectures*. Springer.
- [Wustenhoff et al., 2002] Wustenhoff, E., & BluePrints, S. (2002). *Service level agreement in the data center*. Sun Microsystems Professional Series.
- [Wheaton, 2007] Wheaton, M. (2007). *Decorating your soa services with governance enforcement contracts*. Agenda.
- [Wiederhold et al., 1992] Wiederhold, G. (1992). *Mediators in the architecture of future information systems*. *Computer*, 25(3), 38-49.
- [WSO2, 2010] WSO2 (2010). *Governance Registry*. Available on: <http://wso2.com/products/governance-registry>
- [W3C, 2000] W3C (2000, May), *Simple Object Access Protocol (SOAP)*.
- [W3C, 2004] W3C (2004), *Web Service-Architecture (WS-Architecture)*. Available on: <http://www.w3.org/TR/ws-arch/>

- [W3C, 2005] W3C (2005), *Web Service-Choreography Description Language (WS-CDL)*. Available on: <http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/>
- [W3C, 2006] W3C (2006), *Web Service Policy (WS-Policy)*. Available on: <http://www.w3.org/Submission/WS-Policy/>
- [Zeng et al., 2004] Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J., & Chang, H. (2004). *QoS-aware middleware for web services composition*. *Software Engineering, IEEE Transactions on*, 30(5), 311-327.
- [Zribi et al., 2012] Zribi, S., Bénaben, F., Hamida, A. B., & Lorré, J. P. (2012). *Towards a Service and Choreography Governance Framework for Future Internet*. In *Enterprise Interoperability V* (pp. 281-291). Springer London.
- [Zribi et al., 2013] Zribi, S., Bénaben, F., Lorré, J. P. & Pingaud, H. (2013). *Enhancing Services Selection by Using Non-Functional Properties within BPMN in SOA Context*. *PRO-VE 2013*: 305-313.

ANNEXE 1 : PROPRIETES NON-FONCTIONNELLES DE WSQF

Tableau 13 : Propriétés non-fonctionnelles de la catégorie « Business Quality Value ». [OASIS, 2010]

Valeur de la qualité métier (Business Quality Value)	Prix (Price)	le prix correspond au coût défini par le fournisseur du service Web et que les utilisateurs sont amenés à payer pour pouvoir utiliser ce service.
	Pénalité et récompense (Penalty and Incentive)	la pénalité définit la compensation financière suite au non-respect et à la violation de l'accord établi entre l'utilisateur et le fournisseur pour l'utilisation du service Web. Contrairement à la pénalité, la récompense peut être spécifiée dans le contrat et concerne une récompense financière suite à un déroulement mieux que prévu de l'exécution du service Web.
	Performance métier (Business Performance)	la performance métier est mesurée par le temps nécessaire pour que la réalisation d'un service métier soit complète tout en tenant compte du temps de traitement ainsi que du débit du réseau.
	Popularité du service (Service Recognition)	la popularité du service correspond à son succès en fonction du nombre d'utilisateurs.
	Réputation du service (Service Reputation)	la réputation du service représente une évaluation du service Web donnée par les utilisateurs une fois qu'ils l'ont utilisé.
	Réputation du fournisseur du service (Service Provider Reputation)	la réputation du fournisseur du service correspond à une évaluation donnée par les utilisateurs envers le fournisseur du service Web.

Tableau 14 : Propriétés non-fonctionnelles de la catégorie « Service Level Measurement Quality ». [OASIS, 2010]

Mesure du niveau de la qualité de service (Service Level Measurement Quality)	Temps de réponse (Response Time)	le temps de réponse correspond à la durée entre l'envoi d'une requête par l'utilisateur et la réception de la réponse.
	Débit Maximum (Maximum Throughput)	le débit maximum se réfère à la quantité maximale que le fournisseur de services peut traiter dans un intervalle de temps donné.
	Disponibilité (Availability)	la disponibilité représente le temps au cours duquel le service Web, en état de fonctionnement, est disponible.
	Accessibilité (Accessibility)	L'accessibilité représente le pourcentage du temps au cours duquel le service Web, en état de fonctionnement et disponible sur le réseau, est accessible
	Successibilité (Successability)	la successibilité représente le pourcentage de succès lors d'une exécution complète du service Web.

Tableau 15 : Propriétés non-fonctionnelles de la catégorie « Interoperability ». [OASIS, 2010]

Intéropérabilité (Interoperability)	Adoptabilité des normes (Standard Adoptability)	l'adoptabilité des normes représente le pourcentage d'implémentation des fonctionnalités du service Web en se basant sur des standards.
	Conformité aux standards (Standard Conformability)	la conformité aux standards représente le pourcentage de la conformité de l'implémentation des fonctionnalités en se basant sur un standard par rapport aux règles définies par ce dernier.
	Pérennité relative (Relative Proofness)	la pérennité relative indique si les services Web échangent convenablement lors de l'utilisation de deux plateformes différentes.

Tableau 16 : Propriétés non-fonctionnelles de la catégorie « Business Processing ». [OASIS, 2010]

Traitement métier (Business Processing)	Fiabilité des messages (Message Reliability)	la fiabilité des messages fait référence au niveau de fiabilité lors de la transmission des messages.
	Intégrité des transactions (Transaction Integrity)	l'intégrité des transactions consiste à savoir si un service Web peut traiter et gérer les transactions.
	Capacité de collaboration (Collaborability)	la capacité de collaboration représente la capacité d'une plateforme de services à définir, contrôler et gérer les flux de services entre les participants afin de permettre aux services Web de participer dans une collaboration (orchestration ou chorégraphie à titre d'exemples).

Tableau 17 : Propriétés non-fonctionnelles de la catégorie « Manageability ». [OASIS, 2010]

Gérabilité (Manageability)	Gestion des informations (Informability)	la gestion des informations détermine si les informations primitives fournies lors de la phase d'implémentation du service Web sont suffisantes pour pouvoir le gérer dans le cas de troubles par exemple.
	L'observabilité (Observability)	l'observabilité mesure la possibilité de fournir des informations sur l'état du service Web. Elle peut être évaluée par l'efficacité avec laquelle les mesures sont prises lors de la collecte d'informations.
	Contrôlabilité (Controllability)	la contrôlabilité mesure si suffisamment de fonctionnalités sont fournies afin de pouvoir assurer le contrôle des services Web.

Tableau 18 : Propriétés non-fonctionnelles de la catégorie « Security ». [OASIS, 2010]

Sécurité (Security)	Cryptage (Encryption)	le cryptage correspond au chiffrement des données du service Web pour empêcher l'accès aux utilisateurs non autorisés aux informations confidentielles.
	Non-répudiation (Non-Repudiation)	la non-répudiation garantit qu'aucun des acteurs ne peut remettre en cause la transaction établie (l'envoi et la réception des messages). Ceci en utilisant le principe des signatures numériques.
	Authentification (Authentication)	l'authentification consiste à savoir si le service Web nécessite une étape d'identification pour que l'utilisateur et/ou le fournisseur puissent accéder à ses fonctionnalités.
	Disponibilité (Availability)	la disponibilité permet de s'assurer que les utilisateurs autorisés peuvent accéder au service Web en cas de besoin.
	Autorisation (Authorization)	l'autorisation consiste à attribuer à un utilisateur / fournisseur les droits nécessaires pour pouvoir accéder à un service / message.
	Audit (Audit)	l'audit représente la capacité (i) de retracer et de vérifier les activités et les événements émis lors de l'utilisation des services, et (ii) de contrôler que toutes les obligations se sont bien déroulées.
	Intégrité (Integrity)	l'intégrité permet d'être alerté quand une personne non autorisée tente de créer, de modifier, ou de supprimer une donnée.
	Confidentialité (Privacy)	la confidentialité représente la garantie de l'inintelligibilité des informations personnelles et de les protéger contre toute divulgation à des tiers non autorisés.

		<i>Exigences non-fonctionnelles métier</i>		<i>WSQF</i>	
[Santos et al., 2010]	Performance	Time Performance	Response Time	Response Time	
			Throughput	Maximum throughput	
		Space Performance			
	Reliability		Availability	Availability	
			Fault Tolerance		
			Accuracy		
	Security		Access Control	Authorization + Authentication	
			Confidentiality	Internal Confidentiality	Confidentiality
				External Confidentiality	
			Availability		Availability

[Cysneiros et al., 2001]	Dynamic NFR	Performance		Response Time + Availability
		Clarity		
		Cost		Price
		Safety		
		Quality		
		Operational restrictions		
		Maintainability		Controllability
		Portability		Collaborability
		Usability		
	Integrity	Integrity		
	Static NFR	Accuracy		
		Traceability		
		Confidentiality		Confidentiality
		Quality		
		Reliability		Message Reliability
		Security		Security Category

[Mylopoulos et al., 1992] [Chung et al., 2000] [Chung et al., 2009]	*ilities	Understandability		
		Usability		
		Modifiability		
		Interoperability		Interoperability
		Reliability		Message Reliability
		Portability		Collaborability
		Availability		Availability
		Maintainability		Controllability
		Adaptability		
		Traceability		
	*ities	Security		Security Category
		Clarity		
		Integrity		Integrity
		Modularity		
	Other	Cost		Price
		Efficiency		
		Accuracy		
		Low coupling		

[Rodriguez et al., 2007]	Security Requirement	Non repudiation		Non repudiation
		Attack Harm Detection		
		Integrity		Integrity
		Privacy		Privacy
		Access Control		Authentication + Authorization
[Heinrich et al., 2011]	Activity Process	Accuracy		
		Security		Security Category
		Maturity		
		Fault Tolerance		
		Time behavior		Response Time
		Analyzability		
		Changeability		
		Stability		
		Testability		Controllability
		Compliance		Standard Adoptability
		Interoperability		Interoperability Category

Légende



Exigence non-fonctionnelle couverte par WSQF



Exigence non-fonctionnelle non couverte par WSQF

ANNEXE 3 : CAS D'ETUDE SAZRI / ENTREPRISE DE MARKETING PREDICRTIF

```

--<bpmn20:definitions exporter="PetalsBPM" expressionLanguage="http://www.w3.org/1999/XMLSchema" id="_1382291714842id1" targetNamespace="http://com.ebmwebsourcing.petalsbpm
/model" typeLanguage="http://www.w3.org/2001/XMLSchema">
-<bpmn20:collaboration id="_1382291714850id2" isClosed="false">
  <bpmn20:participant id="_1382291715832id6" processRef="_1382291715835id7"/>
</bpmn20:collaboration>
-<bpmn20:process id="_1382291715835id7" isClosed="false" isExecutable="false" processType="None">
  -<bpmn20:laneSet>
    -<bpmn20:lane id="_1382291917964id103" name="">
      <bpmn20:flowNodeRef>_1382291744919id45</bpmn20:flowNodeRef>
      <bpmn20:flowNodeRef>_1382291737835id36</bpmn20:flowNodeRef>
      <bpmn20:flowNodeRef>_1382291720000id32</bpmn20:flowNodeRef>
      <bpmn20:childLaneSet id="_1382291917964id104"/>
    </bpmn20:lane>
  </bpmn20:laneSet>
  <bpmn20:startEvent id="_1382291737835id36" name=""/>
  -<bpmn20:serviceTask default="_1382291745017id52" id="_1382291720000id32" name="Élargir le fichier clients">
    -<bpmn20:extensionElements>
      -<wsqdl:ServiceLevelMeasurement>
        -<wsqdl:MeasureFactor>
          -<wsqdl:ResponseTime>
            -<wsqdl:MetricValue>
              <wsqdl:Value xsi:type="xs:float">40.0</wsqdl:Value>
              <wsqdl:Type>float</wsqdl:Type>
              <wsqdl:Unit>Second</wsqdl:Unit>
            </wsqdl:MetricValue>
          </wsqdl:ResponseTime>
          -<wsqdl:Availability>
            -<wsqdl:MetricValue>
              <wsqdl:Value xsi:type="xs:float">90.0</wsqdl:Value>
              <wsqdl:Type>percent</wsqdl:Type>
              <wsqdl:MetricValue>
            </wsqdl:Availability>
          </wsqdl:MeasureFactor>
        </wsqdl:ServiceLevelMeasurement>
      -<wsqdl:BusinessValueQuality>
      -<wsqdl:ServiceCost>
        -<wsqdl:ServicePrice>
          <wsqdl:Price unit="Euro">100</wsqdl:Price>
        </wsqdl:ServicePrice>
      </wsqdl:ServiceCost>
      </wsqdl:BusinessValueQuality>
    </bpmn20:extensionElements>
  </bpmn20:serviceTask>
  -<bpmn20:endEvent id="_1382291744919id45" name="">
    <bpmn20:eventDefinitionRef>_1382291744920id46 </bpmn20:eventDefinitionRef>
  </bpmn20:endEvent>
  <bpmn20:sequenceFlow id="_1382291745017id52" name="" sourceRef="_1382291720000id32" targetRef="_1382291744919id45"/>
  <bpmn20:sequenceFlow id="_1382291740748id42" name="" sourceRef="_1382291737835id36" targetRef="_1382291720000id32"/>
</bpmn20:process>
<bpmn20:terminateEventDefinition id="_1382291744920id46"/>
-<bpmndi:BPMNDiagram resolution="0.0">
  -<bpmndi:BPMNPlane bpmnElement="_1382291714850id2" id="_1382291714850id2_diagram">
    -<bpmndi:BPMNShape bpmnElement="_1382291720000id32" id="_1382291720000id32_diagram" isExpanded="false" isHorizontal="false" isMarkerVisible="false"
    isMessageVisible="false">
      <dc:Bounds height="50.0" width="100.0" x="453.0" y="126.0"/>
    </bpmndi:BPMNShape>
    -<bpmndi:BPMNShape bpmnElement="_1382291737835id36" id="_1382291737835id36_diagram" isExpanded="false" isHorizontal="false" isMarkerVisible="false"
    isMessageVisible="false">
      <dc:Bounds height="30.0" width="30.0" x="351.0" y="148.0"/>
    </bpmndi:BPMNShape>
    -<bpmndi:BPMNEdge bpmnElement="_1382291740748id42" id="_1382291740748id42_diagram">
      <di:waypoint x="381.0" y="161.0"/>
      <di:waypoint x="453.0" y="154.0"/>
    </bpmndi:BPMNEdge>
    -<bpmndi:BPMNShape bpmnElement="_1382291744919id45" id="_1382291744919id45_diagram" isExpanded="false" isHorizontal="false" isMarkerVisible="false"
    isMessageVisible="false">
      <dc:Bounds height="30.0" width="30.0" x="724.0" y="173.0"/>
    </bpmndi:BPMNShape>
    -<bpmndi:BPMNEdge bpmnElement="_1382291745017id52" id="_1382291745017id52_diagram">
      <di:waypoint x="553.0" y="158.0"/>
      <di:waypoint x="724.0" y="185.0"/>
    </bpmndi:BPMNEdge>
  </bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
</bpmn20:definitions>

```

Figure 72 : Fichier BPMN 2.0 de l'activité « Élargir le fichier clients » annotée.

```

<?xml version="1.0" encoding="UTF-8"?>
<wsag:Template wsag:TemplateId="EFC1"
  xmlns:wsag="http://schemas.ggf.org/graap/2007/03/ws-agreement"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wsad="http://www.w3.org/2005/08/addressing"
  xmlns:wsqdl="http://www.nca.or.kr/2006/wsqdl"
  xsi:schemaLocation="http://schemas.ggf.org/graap/2007/03/ws-agreement wsagreement10.xsd">
  <wsag:Name>wsag:elargirLeFichierClientsAgreementTemplate</wsag:Name>
  <wsag:Context>
    <wsag:AgreementInitiator xsi:type="anyType" />
    <wsag:AgreementResponder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="wsad:EndpointReferenceType">
      <wsad:Address>gsb://http://www.example.org/elargirLeFichierClients/:elargirLeFichierClients@elargirLeFichierClientsSOAP
      </wsad:Address>
    </wsag:AgreementResponder>
    <wsag:ServiceProvider></wsag:ServiceProvider>
    <wsag:ExpirationTime>2014-07-27T12:00:00</wsag:ExpirationTime>
    <wsag:TemplateId>wsag:EFC1_2707</wsag:TemplateId>
    <wsag:TemplateName>wsag:EFC1</wsag:TemplateName>
  </wsag:Context>
  <wsag:Terms>
    <wsag:All>
      <wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
        <wsag:ServiceLevelObjective>
          <wsag:KPITarget>
            <wsag:KPIName>Response Time</wsag:KPIName>
            <wsag:CustomServiceLevel
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Response Time"
              operator="&lt;">
              <wsqdl:MetricValue>
                <wsqdl:Value>50</wsqdl:Value>
                <wsqdl:Unit>second</wsqdl:Unit>
              </wsqdl:MetricValue>
            </wsag:CustomServiceLevel>
          </wsag:KPITarget>
        </wsag:ServiceLevelObjective>
      </wsag:GuaranteeTerm>
      <wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
        <wsag:ServiceLevelObjective>
          <wsag:KPITarget>
            <wsag:KPIName>Maximum Throughput</wsag:KPIName>
            <wsag:CustomServiceLevel
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Throughput"
              operator="&gt;">
              <wsqdl:MetricValue>
                <wsqdl:Range>35</wsqdl:Range>
                <wsqdl:Unit>number/second</wsqdl:Unit>
              </wsqdl:MetricValue>
            </wsag:CustomServiceLevel>
          </wsag:KPITarget>
        </wsag:ServiceLevelObjective>
      </wsag:GuaranteeTerm>
      <wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
        <wsag:ServiceLevelObjective>
          <wsag:KPITarget>
            <wsag:KPIName>Availability</wsag:KPIName>
            <wsag:CustomServiceLevel
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Availability"
              operator="&gt;">
              <wsqdl:MetricValue>
                <wsqdl:Value>0.9</wsqdl:Value>
              </wsqdl:MetricValue>
            </wsag:CustomServiceLevel>
          </wsag:KPITarget>
        </wsag:ServiceLevelObjective>
      </wsag:GuaranteeTerm>
      <wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
        <wsag:ServiceLevelObjective>
          <wsag:KPITarget>
            <wsag:KPIName>Successability</wsag:KPIName>
            <wsag:CustomServiceLevel
              xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Successability"
              operator="&gt;">
              <wsqdl:MetricValue>
                <wsqdl:Value>0.89</wsqdl:Value>
              </wsqdl:MetricValue>
            </wsag:CustomServiceLevel>
          </wsag:KPITarget>
        </wsag:ServiceLevelObjective>
      </wsag:GuaranteeTerm>
    </wsag:All>
  </wsag:Terms>
</wsag:Template>

```

```

<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Price</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Price"
        operator="&lt;">
        <wsqdl:ServiceCost>
          <wsqdl:ServicePrice>
            <wsqdl:Price>90</wsqdl:Price>
            <wsqdl:Unit>euros</wsqdl:Unit>
          </wsqdl:ServicePrice>
        </wsqdl:ServiceCost>
      </wsag:CustomServiceLevel>
    </wsag:KPITarget>
  </wsag:ServiceLevelObjective>
</wsag:GuaranteeTerm>

<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Privacy</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Privacy">
        <wsqdl:Security>
          <wsqdl:Prperty name="Privacy"></wsqdl:Prperty>
        </wsqdl:Security>
      </wsag:CustomServiceLevel>
    </wsag:KPITarget>
  </wsag:ServiceLevelObjective>
</wsag:GuaranteeTerm>

<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Integrity</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Integrity">
        <wsqdl:Security>
          <wsqdl:Prperty name="Integrity">
            </wsqdl:Prperty>
          </wsqdl:Security>
        </wsag:CustomServiceLevel>
      </wsag:KPITarget>
    </wsag:ServiceLevelObjective>
  </wsag:GuaranteeTerm>

<wsag:GuaranteeTerm wsag:Name="ÉlargirLeFichierClientsGuarantee">
  <wsag:ServiceLevelObjective>
    <wsag:KPITarget>
      <wsag:KPIName>Non Repudiation</wsag:KPIName>
      <wsag:CustomServiceLevel
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="Non Repudiation">
        <wsqdl:Security>
          <wsqdl:Prperty name="Non Repudiation">
            </wsqdl:Prperty>
          </wsqdl:Security>
        </wsag:CustomServiceLevel>
      </wsag:KPITarget>
    </wsag:ServiceLevelObjective>
  </wsag:GuaranteeTerm>
</wsag:All>
</wsag:Terms>

<wsag:CreationConstraints>
  <wsag:Item wsag:Name="">
    <wsag:Location></wsag:Location>
    <wsag:ItemConstraint />
  </wsag:Item>
  <wsag:Constraint xsi:type="anyType" />
</wsag:CreationConstraints>
</wsag:Template>

```

Figure 73 : Template SLA du service ÉFC 1 de l'entreprise 7Marketing.

RÉSUMÉ

Depuis plusieurs années, la mondialisation du marché, l'augmentation des exigences des clients, et la recherche permanente d'une baisse des coûts induisent un accroissement fort de la complexité du monde industriel. Ainsi, la collaboration inter-organisationnelle devient essentielle pour maintenir et renforcer la compétitivité des entreprises. Dans cette optique, le système d'information peut, selon sa flexibilité et sa robustesse, freiner ou faciliter cette collaboration. Cette problématique est le point de départ du projet MISE (*Mediation Information System Engineering*) qui propose une solution pour la conception et la réalisation d'un système d'information collaboratif. Cette solution se base sur une démarche d'ingénierie dirigée par les modèles, couplée à une approche *Business Process Management* (BPM) et reposant sur les *Architectures Orientées Services* (SOA). Cette démarche se déroule en deux étapes : (i) la génération d'une cartographie de processus répondant à une situation collaborative (niveau métier) et (ii) sa transformation en un système exécutable (niveau technique). Les travaux présentés dans ce manuscrit s'inscrivent dans cette démarche. Ils s'intéressent en particulier au passage du niveau métier vers le niveau technique en se focalisant sur les aspects non-fonctionnels. Nous présentons une approche permettant (i) d'annoter les activités métier à l'aide d'exigences non-fonctionnelles lors de la modélisation des processus collaboratifs, (ii) de rationaliser la gestion des services et leurs propriétés non-fonctionnelles au moyen de la gouvernance SOA et enfin (iii) de faire une réconciliation non-fonctionnelle entre les activités et les services. Nous avons validé les concepts de notre approche au travers de deux prototypes : *Petals BPM-NFR* pour la modélisation et l'annotation non-fonctionnelle des processus métier et *EasierGov-NFR* pour la gouvernance des services et de leurs propriétés non-fonctionnelles ainsi que la réconciliation non-fonctionnelle. Un cas d'étude illustre notre approche pour prouver son applicabilité. Ces travaux s'inscrivent au sein du projet européen FP7 CHOReOS.

Mots clés : Réconciliation non-fonctionnelle, Gouvernance SOA, Interopérabilité, Système d'information, Gestion des processus métier, Services Web.

ABSTRACT

Over the last years, the globalization, the increase of customers requirements, and the constant search for lower costs, have induced a steep complexity of organization's management. Therefore, inter-organizational collaboration becomes essential to maintain and enhance the competitiveness of enterprises. In this context, the information system may, depending on its flexibility and its robustness, hamper or facilitate this collaboration. One approach to resolve this problem has been taken by the MISE (*Mediation Information System Engineering*) project. It aims to provide an effective implementation of a collaborative information system, based on Model-Driven Engineering, coupled with a Business Process Management approach (BPM) and supported by Service-Oriented Architecture (SOA). This solution consists in two steps: (i) the generation of a business processes map from the description of a collaborative situation (business level), and (ii) the transformation of these process models into an executable system (technical level). The works detailed in this PhD take roots in this solution and improve it by adding non-functional aspects management during the transition from the business level to the technical level. We present an approach that: first (i) annotates the business activities with Non-Functional Requirements (NFR) in the processes modelling stage, second (ii) rationalizes the management of services and their non-functional properties through SOA governance, and third (iii) makes a non-functional reconciliation between activities and services. We validate our approach through two prototypes: *Petals BPM-NFR* for modelling and non-functional annotation of business process, and *EasierGov-NFR* for the governance of services and their non-functional properties, and the non-functional reconciliation. A case study illustrates our approach to demonstrate its applicability. This work is supported by the FP7 European project CHOReOS.

Keywords: Non-functional reconciliation, SOA governance, Interoperability, Information system, Business process management, Web services.