

REPUBLIQUE DU CAMEROUN

Paix – Travail – Patrie

UNIVERSITE DE YAOUNDE I
FACULTE DES SCIENCES
DEPARTEMENT DE INFORMATIQUE



REPUBLIC OF CAMEROUN

Peace – Work – Fatherland

UNIVERSITY OF YAOUNDE I
FACULTY OF SCIENCES
DEPARTMENT OF COMPUTER SCIENCES

ELABORATION D'UN MODELE DE REFERENCE D'UNE ENTREPRISE BASEE SUR L'INGENIERIE DES BESOINS DES PROCESSUS METIERS

Doctorat/PhD en Informatique

Par :

ATOUBA Christian Lopez,

Sous la direction
TANGHA Claude
Professeur

Année Académique
2014-2015





AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire de Yaoundé I. Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : biblio.centrale.uyi@gmail.com

WARNING

This document is the fruit of an intense hard work defended and accepted before a jury and made available to the entire University of Yaounde I community. All intellectual property rights are reserved to the author. This implies proper citation and referencing when using this document.

On the other hand, any unlawful act, plagiarism, unauthorized duplication will lead to Penal pursuits.

Contact: biblio.centrale.uyi@gmail.com

Dédicaces

A ma défunte mère MINKO Jeannette,
À ma famille qui m'a donné la force et le réconfort nécessaire pour pouvoir arriver au bout
de ce travail.

A tous ceux qui de près ou de loin m'ont motivé et/ou assisté dans ce travail.

Remerciements

Mes remerciements vont vivement à l'endroit de mon superviseur de thèse, Monsieur le Professeur TANGHA Claude, et de mon directeur de thèse, Monsieur le Dr. ATSA ETOUNDI Roger, pour avoir supervisés et dirigés mes travaux de recherches, ainsi que pour leurs conseils certainement très utiles pour mon avenir. Je leur suis également très reconnaissant pour leur soutien et leurs encouragements permanents.

*

Mes remerciements vont à l'endroit du Laboratoire d'Études des Systèmes d'Information et des Technologies Éducatives pour le cadre d'échange qui a été mis sur pied pour nous permettre de progresser dans nos recherches.

*

Mes remerciements vont à l'endroit de Monsieur FOUA NDJODO Marcel Laurent, Chef du laboratoire ci-dessus évoqué, pour les orientations et les conseils prodigués tout au long de cette activité de recherche ardue et harassante.

*

Mes remerciements vont enfin à l'endroit de Monsieur KOUAMOU Georges, pour la motivation qu'il m'a donnée dans l'exercice de recherche, malgré le fait que nous n'ayons pas pu cheminer ensemble dans le cadre de cette thèse.

*

Mes remerciements vont à l'endroit de ma famille pour le soutien moral qu'elle m'a apportée tout au long de cette activité de recherche.

*

Mes remerciements vont particulièrement à l'endroit de ma très douce et tendre épouse, Madame ATOUBA, née MEYABEME BELINGA Cécile Carine, pour son réconfort, sa patience et son assistance.

*

Mes remerciements vont à l'endroit de Monsieur OWONO ZE Roger, mon oncle, pour le réconfort et le soutien moral et psychologique.

*

Mes remerciements particulièrement vont à l'endroit de Messieurs ZOBO Eric Patrick, PRISO NSAWÉ NDEDI, AMOUGOU NGOUMOU, ABESOLO ALO'O Ghislain, NJUH Walter, Madame NGAH virginie et Monsieur AZEGUE, tous membres du laboratoire d'Études des

Systèmes d'Informations et des Technologies Educatives (LESITE), pour leur soutien tout au long de cette activité de recherche.

*

Mes remerciements vont à l'endroit du personnel de l'ex-Division Informatique du Ministère de la Fonction Publique et de la Réforme Administrative, pour le support du surcroit de travail, pendant mes périodes pour lesquelles j'étais pris par mes travaux de thèse.

*

Mes remerciements vont à l'endroit de Messieurs AHANDA Pascal Pierre et NGOUCHEME MONGBET Ibrahim pour les échanges fructueux que nous avons eus tout au long de ce travail.

*

Mes remerciements vont à l'endroit de Mesdames et Messieurs TAMBE Jacob AYUKTAKEM, NDAM Meredith, KUMBE Joyce, Andrew EPULE, NDUMBE TETE, pour la relecture de la version anglaise de mes différentes publications.

Sommaire

CHAPITRE I: INTRODUCTION GENERALE	1
I. CONTEXTE	2
II. PROBLEMATIQUE	5
III. OBJECTIF	7
IV. CONTRIBUTION	7
V. PLAN DU MEMOIRE DE THESE.....	10
CHAPITRE II: MODELISATION DES ORGANISATIONS.....	15
I. LA MODÉLISATION DES ORGANISATIONS.....	16
I.1 Introduction	17
I. 2 LES PROCESSUS D’AFFAIRE	20
I.2.1 Modélisation de processus d’affaire	22
I.2.1.1 Concepts de base.....	22
I.2.1.2 Technologies de spécification des processus d’affaire	23
I.2.2. TECHNOlogie de modélisation des processus d’affaire	26
I.2.2.1 Techniques Ad hoc.....	26
I.2.2.2 Workflows	27
I.2.2.3 Services Web	27
I.2.2.4 Electronic Business XML (ebXML) et Ontology Web Language for Services (OWL-S).....	28
I.2.2.5 METHODES IDEF - ICAM DEFINITION Language	29
I.2.2.5.1 IDEF0	29
I.2.2.5.2 IDEF1.....	30
I.2.2.5.3 IDEF2.....	30
I.2.2.5.4 IDEF3.....	30
I.2.2.6 La méthodologie GRAI.....	31
I.2.3 LES LIMITATIONS DES SOLUTIONS ACTUELLES	32
I.2.3.1 Le Workflow	32
I.2.3.2 L'EAI - Enterprise Application Intégration.....	33
I.2.3.3 Le B2Bi - Business to Business integration	33
I.2.3.4 Les progiciels intégrés	34
I.2.3.5 Les moteurs de règles métiers	35
I.3. Conclusion	36
II. INGENIERIE DES BESOINS POUR LES PROCESSUS D’AFFAIRES	37
II.1 Introduction.....	37
II.2 – Concepts de Base.....	38
II.3 Approches d’ingénierie des besoins	40
II.3.1 - Ingénierie des besoins orientée par les buts	40
II.4 les Modèles de besoins orientés buts.....	43
II.4.1 – Le modèle i*i	45
II.4.2 - Le modèle KAOS.....	48
II.4.3 - Le modèle BESOINS NON FONCTIONNELS.....	51
II.4.4 - Le modèle MAP.....	52
II.4.5 le formalisme de Farida et Joël brunet.....	53

III. QUALITE DE SERVICE AU SEIN DES ORGANISATIONS	59
III.1 Introduction.....	59
III.2 La Qualité.....	59
III.3. LA QUALITE PRODUIT.....	61
III.3.1 La norme Iso 9126, Technologies de l'Information : "Qualité des produits logiciels"	61
III.3.2 Iso 9001v2000 "Systèmes de management de la qualité - Exigences"	62
III.3.2.1 Présentation ISO 9001v2000	62
III.3.2.2 PDCA: Plan, Do, Check, Act, "la roue de Deming"	62
III.4 Conclusion.....	63
IV. CONCLUSION DU CHAPITRE	63

CHAPITRE III : MODELE DE PROCESSUS D'INGENIERIE DES BESOINS POUR LES PROCESSUS D'AFFAIRE DES ORGANISATIONS..... 65

I. Introduction	68
II. RAPPELS DE QUELQUES CONCEPTS DE BASE	70
II.1 RAPPEL DES CONCEPTS DE BASE INHERENTS A LA REPRESENTATION DES BESOINS.....	70
II.1.1 BESOIN D'UN UTILISATEUR	71
II.1.2 DOCUMENT DE RECENSEMENT DES BESOINS	71
II.1.3 DOMAINE	72
II.1.4 CONTRAINTES.....	72
II.1.5 BUT	72
II.1.6 REGLE DE GESTION	74
II.1.7 IMPORTANCE DU BESOIN EXPRIME	75
II.1.8 FRAGMENT DE CONNAISSANCE OU BESOINS EXPRIMES.....	75
II.2 RAPPEL DES CONCEPTS DE BASE INHERENTS A LA COMPREHENSION ET A LA modelisation de la satisfaction des beneficiaires de service	76
II.2.1 Point de vue	78
II.2.2 Facteur de Qualité.....	78
II.2.3 Connaissance de la Satisfaction [14]	80
II.2.4 Observateur de Performance	80
III. FORMALISATION DE QUELQUES CONCEPTS.....	84
III.1 Introduction.....	84
III. 2.- CANEVAS POUR LA FORMALISATION DES Regles metiers	84
III.2.1 Définition: Règle Métier	84
III.2.2 Structure des Règles métiers	85
III.3.- PROPOSITION D'UNE SPECIFICATION DE REGLES METIERS.....	88
III.3.1 Types Prédéfinis	88
III.3.1.1 Types Simples	88
III.3.1.2 Types Composés	89
III.3.2 Syntaxe de Déclaration	90
III.3.3 Déclaration des Règles métiers	91
III.3.3.1 Définition du Contexte	91
III.3.3.2 Définition de la Description.....	94
III.3.3.2.1 Garde	94
III.3.3.2.2 Séquence.....	97
III.3.3.2.3 résultats	101

III.4. RELATIONS ENTRE REGLES METIERS	103
III.4.1 Axiomes	103
IV. DEMARCHE	105
IV.1 Recensement des besoins	105
IV.1.1 Recensement des besoins des cadres métiers	105
IV.1.1.1 RECENSEMENT DES BESOINS DES UTILISATEURS.....	105
IV.1.1.2 REGROUPE DES BESOINS PAR BUTS.....	106
IV.1.2 Recensement des attentes de qualités	106
IV.1.2.1 Recensement des points de vus	106
IV.1.2.2 Règles de construction de l'ensemble des Facteurs de Qualités	107
IV.2 Sélection.....	108
IV.3 CONSTRUCTION DU MODELE	110
IV.3.1 Transformation des besoins	110
IV.3.1.1 TRANSFORMATION DES BESOINS EN FRAGMENT DE CONNAISSANCE	110
IV.3.1.2 ÉLABORATION DU MODELE metier	111
IV.3.1.2.1 « PROBLEM FRAMES »	111
IV.3.1.2.2 AXIOMES DE BASE POUR L'ELABORATION DU MODELE DE REFERENCE DE L'ORGANISATION	112
IV.3.2 Construction des indicateurs de performances.....	114
IV.3.2.1 REGLE DE CONSTRUCTION DES TÂCHES	114
IV.3.2.2 Démarche.....	115
IV.3.2.2.1 Construction des Observateurs	116
IV.3.2.2.2 Construction des facteurs de performances.....	117
IV.3.2.2.3 Jonction des préoccupations	117
IV.4 VERIFICATION DE LA COHERENCE DANS LE modèle de reference	118
IV.4.1 Cohérence entre les composantes du fragment de connaissance et les attentes des bénéficiaires des services.....	118
IV.4.2. Cohérence entre les concepts DE REGLE METIER ET les composantes du fragment de connaissance.	120
IV. 5. CYCLE DE VIE DU MODÈLE	125
IV. 6. CONCLUSION.....	126

CHAPITRE IV : DYNAMIQUE DU MODELE DE REFERENCE DANS UNE ORGANISATION127

I. INTRODUCTION.....	128
II. COMPOSANTS METIERS ORIENTES ASPECTS (CMOA).....	129
III. DYNAMIQUE DE LA QUALITE DE SERVICE.....	131
IV. DYNAMIQUE D'UN FRAGMENT DE CONNAISSANCE.....	133
IV.1 Dynamique	133
IV.2 Illustration à l'aide du concept de CMOA	136
V. DYNAMIQUE DU MODELE de reference	137
V.1 Dynamique.....	137
V.2 Illustration à l'aide de CMOA.....	140
VI. CONCLUSION	141

CHAPITRE V : CAS D'UTILISATION

I. Introduction	145
-----------------------	-----

- II. Recensement des besoins146
 - II.1. Recensement des besoins des utilisateurs.....146
- III Sélection des besoins149
- IV. Modélisation des processus d'affaires.....150
 - IV.1 Transformation des Besoins150
 - IV.1.1 Transformation des besoins en fragments de connaissance.....150
 - IV.1.2 Elaboration du Modèle de REFERENCE.....174
 - IV.2 Constructions des indicateurs de performances.175
 - IV.2.1 Transformation du Fragment de connaissances en une tâche.....176
 - IV.2.2 Démarche176
 - IV.2.2.1 Construction des observateurs176
 - IV.2.2.2 Construction des facteurs de performances.....180
 - IV.3 Jonction de préoccupations.181
- V. Validation du Modèle.....181
- VI. Conclusion181

CHAPITRE VI: CONCLUSION GENERALE ET PERSPECTIVES182

RÉFÉRENCES187

LISTE DES PUBLICATIONS ISSUES DE LA THESE197

INDEX198

Résumé

Ce travail s'inscrit dans le cadre du « Human-centric Business Process Modeling ». Il s'attaque à la problématique de la conciliation formelle de l'expression des besoins des organisations à la production des modèles de références des organisations inhérentes à ces besoins. Certains travaux ont déjà été faits dans ce sens. Cependant, ils s'attaquent de manière parcellaire à cette problématique. Ce qui a induit la formation de plusieurs courants de pensées s'appuyant sur les concepts de domaine ou sur l'ingénierie de domaine pour ressortir des concepts permettant la modélisation des organisations. Parmi ces travaux, nous nous sommes intéressés aux travaux de Atsa Etoundi Roger, de Farida Semmak et de Joël Brunet. Chacun de ces travaux contribue partiellement à l'atteinte de cette solution. En effet, les travaux d'Atsa Etoundi Roger considèrent une organisation comme étant un ensemble de processus d'affaires et de ressources nécessaires à leurs exécutions. Ainsi, la modélisation d'une organisation se restreint à la modélisation de ses processus d'affaires et des ressources sous-jacentes à leurs exécutions. Ils s'appuient sur l'ingénierie de domaine et les Workflows et proposent un formalisme de représentation des processus d'affaires tout en prenant en compte leur environnement d'exécution. Les travaux de Farida Semmak et Joël Brunet, quant à eux s'appuient sur les concepts de domaine pour dégager un formalisme de représentation des besoins d'un domaine. Cependant, ces travaux ne sont pas trop appesantis sur la construction d'un modèle, à partir des besoins réels d'une organisation et des attentes de ses bénéficiaires de services. Dans ce contexte, ce travail se propose d'enrichir le formalisme de Farida Semmak et Joël Brunet d'un processus d'ingénierie des besoins qui permet de partir des besoins des organisations pour construire un modèle de référence d'une part, et de composer (fédérer) les deux formalismes pour avoir un modèle beaucoup plus riche pour la modélisation des organisations, d'autre part.

Dans cette thèse, nous présentons un processus d'ingénierie des besoins permettant de partir des besoins de l'organisation et des bénéficiaires de service, pour élaborer le modèle de référence d'une organisation, obtenu par composition du modèle de Atsa Etoundi Roger et celui issu des travaux de Farida Semmak et Joël Brunet. Ce processus d'ingénierie couvre les activités de l'ingénierie des besoins que sont : l'explicitation, la sélection et la validation des besoins étant entendu que la spécification desdits besoins a été définie dans la littérature. Ce processus s'appuie sur quatre (04) étapes fondamentales à savoir : le

recensement, la sélection, l'élaboration du modèle et la validation. Pendant que l'étape de recensement collecte les besoins et les attentes des bénéficiaires de services de l'organisation, l'étape de sélection revient au choix des besoins consistants. L'étape d'élaboration du modèle quant à elle va permettre de construire le modèle de référence, à partir de l'ensemble des besoins sélectionnés et des attentes des bénéficiaires de service. Enfin, la dernière étape va permettre de se rassurer de l'inexistence des incohérences entre les concepts de règles métiers et celui de buts dans le modèle de référence. Cette vérification se fait à partir d'un ensemble d'axiomes et constitue l'objet de la quatrième étape. Cette démarche a permis d'élucider les concepts tels que les concepts de point de vue, facteur de qualité, Observateur, Observation, Indicateur de performance, Observateur de performance, inhérents au suivi d'un processus d'affaire dans une organisation. A côté de ces quatre activités, Il a été abordé le problème d'alignement dans le temps des besoins réels de l'organisation avec le modèle de référence de cette organisation. Pour ce faire, une étude de la dynamique de ce modèle a été menée. Le but visé par cette étude a été la maîtrise de l'évolution du modèle lorsque les besoins de l'organisation et les attentes de ses bénéficiaires de services évoluent. Il a été de ce fait supposé que c'est en réponse aux attentes des bénéficiaires de service que le modèle de référence est amené à subir des modifications. Ainsi, trois aspects de cette dynamique ont été identifiés : (i) le premier porte sur l'étude de la dynamique de la qualité de service externe ; (ii) le second porte sur l'étude de la dynamique du fragment de connaissance. Il a été considéré qu'un fragment de connaissance b' est une évolution du fragment de connaissance b si la connaissance b' , appliquée à une tâche t en lieu et place de b , améliore le rendement de t . Ce rendement a été considéré comme rendement de la connaissance b' ; (iii) le troisième, sur l'étude de la dynamique du modèle, il a été considéré qu'un modèle de référence $s M'$ d'une organisation est une évolution du modèle de référence M de cette même organisation, si la qualité de service externe du modèle M' est supérieure à celle du modèle M . A l'issue de cette étude, il a été identifié trois points fixes. Le premier a trait à la qualité de services perçus par les bénéficiaires de service ; le second se rapporte à la connaissance dont l'usage lors de l'exécution d'une tâche permet à celle-ci de fournir un rendement maximal ; le troisième se rapporte au modèle pour lequel la plupart des clients et le staff dirigeant de l'organisation sont satisfaits. Ces points de stabilité matérialisent la limite de la dynamique du modèle de référence. Ils permettent d'établir que le modèle de référence n'a pas une évolution

infinie. Indiquant par là même que lorsqu'une organisation a axé son processus d'affaire sur la satisfaction des ses clients(ou la gestion axée sur les résultats), l'évolution de son processus d'affaires admet une limites au delà de laquelle aucun enrichissement ne peut plus améliorer de manière substantielle la qualité de service. Ces travaux ont été appliqués au système de liquidation des droits des personnels de l'état du Cameroun. Les résultats obtenus ont été utilisés dans le cadre de la refonte des systèmes ANTILOPE (Application Nationale pour le Traitement Informatique et la Logistique des Personnels de l'Etat) et SIGIPES (Système Informatique de Gestion Intégrée des Personnels de l'Etat et de la Solde) gérant respectivement la solde et la carrière des personnels de l'Etat.

La démarche ainsi proposée permettra à terme aux entreprises de rester compétitives dans la nouvelle économie mondiale.

Mots clefs : *Ingénierie des besoins, Modélisation Orientée buts, modélisation des organisations, Processus d'affaire, modélisation de la satisfaction.*

Summary

This work is part of the "Human-centric Business Process Modeling". It tackles the problem of reconciling formal expression of the requirements of organizations in the production of reference models inherent in these organizations needs. Some work has already been made to this respect. However, they tackle this problem partially. Thus, induces several schools of thought based on the concepts of domain or domain engineering to bring out concepts for modeling organisations. Amongst these works, we were interested in the work of Roger Atsa Etoundi, Farida Semmak and Joel Brunet. Each part of this work contributes in achieving this solution. Indeed, the work of Roger Atsa Etoundi considers an organization as a set of business processes and resources necessary for their execution. Thus, modeling an organization is restricted to modeling the business processes and resources that underlie their execution. They based themselves on domain engineering and workflows and propose a formalism for representation of business processes taking into account their execution environment. The works of Farida Semmak and Joël Brunet, in turn are based on the concepts of domain to obtain a representation formalism of the requirements of a domain. However, these studies do not focus much on the construction of a model, based on the real needs of an organization and the expectations of its service beneficiaries. In this context, this work proposes to enrich the formalism put forward by Farida Semmak and Joel Brunet of an engineering process which enables the construction of a reference model from the requirements of an organization on the one hand, and on the other, to compose (to federate) the two formalisms to obtain a much richer model for modeling the organization.

In this thesis, we present a requirements engineering process starting with the requirements of the organization and the service beneficiaries, to develop the reference model of an organization, obtained through the composition of the Atsa Etoundi Roger's model and the one resulting from the work of Farida Semmak and Joël Brunet. This engineering process covers the activities of requirements engineering which are: the elicitation, selection and validation of requirements provided that the specification of such requirements had been defined in the literature. This process is based on four (04) basic steps which are: the elicitation, selection, model development and validation. While the elicitation collects the needs and expectations of service beneficiaries of the organization, the selection involves the choice of consistent requirements. The model development

phase involves the construction of the goal oriented requirements model, from the set of selected requirements and expectations of service beneficiaries. The final step will help to reassure the non-existence of inconsistencies between the concepts of business rules and the goals in the goal oriented requirements model. This check is done through a set of axioms and constitutes the subject of the fourth phase. This approach helps to clarify concepts such as point of view, quality factor, Observer, Observation, Performance Indicator, Performance Observer, inherent in the monitoring of a business processes within an organization. Besides these four activities, the problem of aligning the requirements in time to the actual requirements of the organization with the purpose of this model of organization was addressed. Thus a study of the dynamics of this model was conducted. The purpose of this study was to control the evolution of the model when the organization's requirements and expectations of service beneficiaries evolve. It was therefore assumed that, it was in response to the expectations of service beneficiaries that the model needs is subject to change. Thus, three aspects of this process were identified: (i) The first concerns the study of the dynamics of external quality of service, (ii) the second is the study of the dynamics of the knowledge bit. It was considered that a knowledge bit b' is an evolution of the knowledge bit b if the knowledge b' , applied to a task t in place of b improves the output of t . This output was considered that the output knowledge b' , (iii) thirdly, the study of the dynamics of the model, it was considered that a requirements model M' of an organization is an evolution of this requirements model M of the same organization, if the external quality of service of the model M' is greater than that of the model M . From this study, three stability points were identified. The first relates to the quality of services received by beneficiaries of the service; the second relates to knowledge used during execution of a task to provide maximum output and the third refers to the model for which most customers and executive staff of the organization are satisfied. These stability point materialize the limit of the dynamics of the requirements model. It enabled to show that the reference model does not evolve infinitely. Hence indicating that when business processes of an organization are oriented towards customer satisfaction (or results based management), the evolution in its business process has a limit beyond which any further enhancement does not improve substantially the quality of service. This work was applied on the payroll system of the state personnel in Cameroon. The results obtained were used in the redesign of the systems : ANTILOPE (Application Nationale pour le Traitement Informatique et la Logistique des Personnels de

l'Etat) and SIGIPES (Système Informatique de Gestion Intégrée des Personnels de l'Etat et de la Solde) respectively manages the payroll and career of State personnel.

The proposed approach thus shall eventually enable companies to remain competitive in the new global economy.

Keywords: requirements engineering, goal-oriented modeling, organization modeling, business process, satisfaction modeling.

Liste des abréviations

ANTILOPE : Application Nationale pour le Traitement Informatique et la Logistique des Personnels de l'Etat

SIGIPES : Système Informatique de Gestion Intégrée des Personnels de l'Etat et de la solde

MDA : Model Driven Architecture

MDE : Model Driven Engineering

IEEE: Institute of Electrical and Electronics Engineers

ACM: Association for Computing Machinery

GORE : Goal-Oriented Requirements Engineering

AMT : Approche de Modélisation Triangulaire

AME : Approche de Modélisation en Etoile

BPM : Business Process Management

BPR : Business Process Re-engineering

MME: Méthodes de Modélisation d'Entreprise

ICAM: Integrated Computer Aided Manufacturing

IDEF : Icam DEFinition

EPC: Event Process Chains

RAD: Role Activity Diagrams

REA: Ressource-Event-Agent

BPML: Business Process Modeling Language

BPMN: Business Process Modeling Notation

WPDL: Workflow Process Description Language

PIF: Process Interchange Format

PSL: Process Specification Language

B2Bi: Business to Business integration

BPEL4WS: Business Process Execution Language for Web Services

UML: Unified Modeling Language

URI: Uniform Resource Identifier

XML: Extensible Markup Language

SOAP: Simple Object Access Protocol

UDDI: Universal Description Discovery and Integration

ebXML: Electronic Business XML

SOA: Service Oriented Architecture

WSDL: Web Service Description Language

SMTP: Simple Mail Transfer Protocol

HTTP: HyperText Transfer Protocol

FTP: File Transfer Protocol

CPAs: Collaboration Protocol Agreements

OWL-S: Ontology web Language for Services

DAML-S: DARPA agent markup language for services

SADT: Structure Analysis and Design Technique

IDEFX: Icam DEFinition version X

GRAI: Graphes à Résultats et Activités Interreliées

EAI: Intégration des Applications d'Entreprise

SI : Système d'Information

EDI : Electronic Data Interchange

SAM : Security Account Manager

TIC: Technologie de l'information et de la communication

RE : Requirement Engineering

FR : Functional Requirement

NFR: Non Fonctional requirement

BMM : Business Motivation Model

OMG : Object Management Group

GOMS : Goals, Operators, Methods, and Selection rules

ISAC : Information Sharing and Analysis Center

EKD : Enterprise Knowledge Development

GSN : General Software Note

GQM: Goal, Question, Metric

CMM: Change Management Method

KAOS: Knowledge Acquisition in automated Specification

GBRAM: Goal-Based Requirements Analysis Method

SD: Strategic dependencies

SR: strategic relation

ISO: International Standards Organization

SMQ: Système de management de la Qualité

PDCA: Plan, Do, Check, Act

CMMI: Capability Maturity Model Integration

ITIL: Information Technology Infrastructure Library

CMOA : Composant Métier Orienté Aspects

ACA : Agence de Consultants Associés

MINFOPRA : Ministère de la Fonction Publique et de la Réforme Administrative

ENS : Ecole Normale Supérieure

ENSET : Ecole Normale Supérieure de l'Enseignement Technique

CUSS : Centre Université des Sciences de la Santé

QoS: Quality of Service

Liste des Tableaux

Tableau 2.1 : Le rôle de l'analyse des buts dans les activités RE [69]	44
Tableau 3.1 : liste des prédicats de comparaison.....	95
Tableau 4.1 : sémantiques des annexes	146
Tableau 4.2 : Synthèse de l'étude diagnostique réalisée par le cabinet ACA en 2008	147
Tableau 4.3 : Synthèse des échanges réalisés en 2008 entre le cabinet ACA et les bénéficiaires des actes de liquidations.	148
Tableau 4.4 : Ensemble des besoins retenus.....	149

Liste des figures

Figure 2 : Modèle d'ingénierie des Processus d'Affaire	9
Figure 3 – Un méta-modèle de processus [16]	18
Figure 4: Extrait du méta-modèle d'i* implanté dans Tropos	45
Figure 5: Modèle SD appliqué à un système de gestion de planning de réunions [57]46	
Figure 6: Modèle SR appliqué a un système de gestion de planning de réunions [57]47	
Figure 7: Extrait du méta-modèle de KAOS [KAOS Objectiver]	48
Figure 8: Méta-modèle du goal-model de KAOS	49
Figure 9: Fragment du modèle KAOS pour le London Ambulance Service system [73]50	
Figure 10: Exemple NFR Sécurité des comptes (ICSQ94)	51
Figure 11: Extrait du méta-modèle MAP d'après [44].....	52
Figure 12: Exemple de carte MAP [58].....	53
Figure 13: Le modèle général d'un fragment de connaissance [3]	55
Figure 14: Un exemple de fragment de connaissance [3].....	55
Figure 15 - La relation entre le coût et la qualité	60
Figure 16 - PDCA : Plan, Do, Check, Act, « la roue de Deming »[7].	63
Figure 17: Illustration du processus d'ingénierie permettant la construction du modèle de référence	124
Figure 18 : Cycle de vie du modèle de référence	125
Figure 19 : Dynamique de la Qualité de Service	132
Figure 20 : Dynamique d'un besoin	136
Figure 21: Dynamique du Modèle de référence	139
Figure 22 : Illustration à base de buts du modèle de référence obtenus pour un dossier d'attribution d'un capital-décès et d'une pension de réversion	175
Figure 23 : Illustration à base de fragments de connaissances du modèle de référence obtenus pour un dossier d'attribution d'un capital-décès et d'une pension de réversion	175
Figure 24 : Illustration à base de tâches du modèle métier obtenus.	176
Figure 25 : Illustration de l'identification des tâches terminales.	177
Figure 26 : Illustration à base de tâches du modèle de métier obtenu enrichis d'observateurs.	178
Figure 27 : Illustration des facteurs de performance associés aux tâches terminales construites.....	180

Lexique

But : Un but fonctionnel définit un besoin/attente potentiel(le) que le système peut satisfaire, il exprime ce que l'utilisateur du système souhaiterait faire.

Composants métiers orientés aspects (CMOA): un composant métier orienté aspect est une implémentation abstraite d'un modèle de référence.

Connaissance de la satisfaction : la connaissance de la satisfaction des clients représente l'ensemble des informations du processus d'affaire permettant de calculer les facteurs de qualité.

Facteur de qualité : Un facteur de Qualité est une observation à partir de laquelle un client s'appuie pour donner son jugement par rapport au service qui lui a été rendu. Les facteurs de qualité sont intimement liés au service dont a bénéficié le client.

Fragment de connaissance : Le fragment de connaissance constitue une matérialisation d'un besoin exprimé.

Indicateur : Un indicateur est une information contribuant à l'appréciation d'une situation [14]. Elle représente le résultat de l'évaluation d'un observateur associé à une activité du processus d'affaire.

MultiSet : est un ensemble dans lequel un élément peut avoir plusieurs occurrences

Observateur : Un observateur est une variable attachée, à une règle métier ou à un objet métier d'un processus d'affaire, dont le contenu est mis à jour pendant l'exécution d'une activité de ce processus d'affaire faisant usage soit de la règle métier, soit de l'objet métier.

Observation : Une observation est une variable booléenne.

Point de vue : le point de vue d'un client par rapport au service qui lui est rendu dans une organisation, représente le sentiment que celui-ci éprouve par rapport au service qui lui a été rendu dans cette organisation.

Problem frames : un « problem frame » [94] (ou schéma de problème) est un schéma qui définit de manière intuitive une classe de problèmes identifiée en termes de son contexte ; des caractéristiques de ses domaines ; et de ses interfaces et des

besoins.

Qualité : la « qualité » peut se définir comme la capacité à atteindre les objectifs opérationnels visés.

Qualité externe, correspondant à la satisfaction des clients. Il s'agit de fournir un produit ou des services conformes aux attentes des clients afin de les fidéliser et ainsi améliorer la part de marché.

Qualité interne, correspondant à l'amélioration du fonctionnement interne de l'entreprise. L'objet de la qualité interne est de mettre en œuvre des moyens permettant de décrire au mieux l'organisation, de repérer et de limiter les dysfonctionnements.

Recensement des besoins des utilisateurs : le recensement des besoins des utilisateurs est l'activité qui vise, la collecte des besoins des utilisateurs, ainsi que la validation de ces derniers

Règle de gestion : Une règle de gestion définit une loi du domaine à laquelle le but doit se conformer. Elle aide à organiser un processus de gestion pour atteindre le but.

Règle métier : une « règle métier », d'après nous, sera une directive du domaine qui régit le comportement d'un métier dudit domaine.

**CHAPITRE I:
INTRODUCTION
GENERALE**

I. CONTEXTE

La systématisation de l'usage des systèmes informatiques dans les différents secteurs métiers a augmenté la complexité des problèmes à résoudre ; et mis en exergue les contraintes temporelles, humaines, matérielles, environnementales et financières dans la démarche de production des systèmes informatiques. Pour faire face à ces challenges, les recherches en ingénierie logicielle se sont concentrées sur la production de logiciels par réutilisation. Dans cette perspective, plusieurs approches ont été suggérées (Architectures orientées services, approches par composants, Model Driven Architecture (MDA), Model Driven Engineering (MDE)). Bien qu'ayant obtenu des résultats probants, les approches actuellement utilisées n'accordent pas une attention particulière au cadrage, à tout instant, du modèle qu'elles proposent avec les besoins réels des organisations. Ce problème est dû au fait que les besoins des organisations évoluent au fil du temps. Ce constat a entraîné une prise de conscience du poids des représentations « métier » dans les nouvelles approches de modélisations de systèmes informatiques. Ceci s'est consolidé par le souci de flexibilité pour faire de l'alignement stratégique de manière récurrente. Ainsi la modélisation des systèmes d'informations dans les organisations reste un problème ouvert. Dans cette perspective, ATSA Etoundi Roger dans [1] s'est appuyé sur l'ingénierie de domaine et les Workflows pour définir un formalisme de modélisation des processus d'affaires avec la prise en compte de leurs environnements d'exécution. Ce formalisme a été considéré comme une contribution dans la modélisation des organisations. Dans ce formalisme, une organisation est perçue comme ensemble de processus d'affaires couplés à leurs environnements d'exécution. Un processus d'affaires est pris pour un ensemble de tâches qui doivent être réalisées, dans un contexte précis, pour l'atteinte des buts spécifiques ; Une tâche est vue comme la donnée d'un contexte de réalisation, d'une fonction de transition entre les états et d'un but à atteindre. Dans cette perspective, l'atteinte d'un but est réalisée par l'observation des valeurs associées à chaque observateur lié à l'exécution d'une ou d'un ensemble de tâches. Un état est l'ensemble des objets de l'environnement auquel sont associées des valeurs issues des exécutions des tâches dans ledit environnement. Il est cependant relevé dans [2] la nécessité de disposer

d'une démarche permettant de construire une représentation standard d'une organisation à partir des besoins réels du domaine. Cette préoccupation a une documentation assez dense dans la littérature. Ce qui semble traduire son intérêt sur le plan scientifique. Deux travaux ont retenu notre attention : les travaux de Atsa Etoundi Roger sus évoqués et ceux de Farida Semmak et Joël Brunet sur l'ingénierie des besoins. Le formalisme proposé par Farida Semmak et Joël Brunet se situe dans la phase d'analyse d'un système informatique et se présente comme résultat de cette phase. Ce formalisme permet de représenter les besoins et utilise comme concept de base le fragment de connaissance. Le fragment de connaissance constitue une matérialisation d'un besoin exprimé. Bien que ces travaux sur l'ingénierie des besoins soient extensibles aux organisations, il est cependant nécessaire de relever que certains concepts indissociables à la modélisation d'une organisation et incidemment à la modélisation de ses processus ne semblent pas revêtir un intérêt particulier pour ces auteurs. Compte tenu du fait que ces aspects ont été clairement détaillés dans les travaux de Atsa Etoundi Roger, nous avons trouvé important de coupler les travaux de Atsa Etoundi Roger avec ceux de Farida Semmak et Joël Brunet, à l'effet de définir un modèle de référence d'une organisation. Ce travail constitue une alternative majeure à la modélisation des organisations et par ricochet celle des processus d'affaire.

Bien que, la modélisation des processus d'affaires en elle-même suscite plusieurs interrogations, nous focaliserons notre attention sur celles qui tournent autour de l'axe « Human-centric Business Process Modeling » notamment sur : - « User-oriented aspects of business process modeling », - « User-centric aspects of Business Process Modeling », ou « Business Process Modeling as a Mechanism for Communicating Business Requirements » à cause des similitudes avec les travaux précédemment cités de Atsa Etoundi Roger. Ce thème prône la modélisation des processus d'affaires à partir des préoccupations orientées utilisateurs. Au stade actuel, les recherches autour de ce thème sont théoriques et très abondantes, en ce qui concerne particulièrement le sous thème « Goal-Oriented Requirements Engineering (GORE) ». A cet égard, toutes les grandes conférences aussi bien IEEE que ACM s'y intéressent. Dans cette optique, plusieurs auteurs à l'instar de Farida Semmak et Joël Brunet ont défini, dans [3], un méta-modèle orienté buts pour

modéliser les besoins d'un domaine. Ken Decreus and Geert Poels dans [4] et Valérie Emin-Martinez dans [5] ont abordé le problème sous des angles variés.

Le problème générique qui ressort de ces recherches est la nécessité de concilier les attentes des organisations avec la représentation de leurs processus d'affaires et workflows. Cette préoccupation n'est pas une panacée de la modélisation des processus d'affaires, mais plutôt un sous thème transversal à plusieurs autres champs tels que l'ingénierie dirigée par les modèles, le « Model-Driven Product Line Engineering ». A travers « Automated and Interactive Product Derivation », le « Requirement Engineering » aborde la même problématique. Dans le contexte du « Requirement Engineering » par exemple, une approche récente semble s'imposer : l'approche KAOS. Cette approche présente en perspective, d'après [6], la nécessité de faire la conciliation des modèles de besoins obtenus à l'expression du modèle mathématique du système informatique sous-jacent. Or, Atsa Etoundi Roger a mis en œuvre un formalisme de modélisation de processus d'affaires avec prise en compte de l'environnement d'exécution de ce dernier, Farida Semmak et Joël Brunet dans [3], ont proposé un méta-modèle orienté buts pour la représentation des besoins. Notre travail consistera donc à coupler le formalisme de ATSA Etoundi Roger et celui de Farida Semmak et Joël Brunet d'une part, d'enrichir le modèle résultant d'un processus d'ingénierie de besoins permettant de le construire à partir des besoins réels d'une organisation et des attentes des bénéficiaires de services, d'autre part. Basé sur l'ingénierie des besoins, le modèle résultant permettra à coup sûr de gérer de « manière douce » et d'anticiper sur l'évolution des besoins au sein de l'organisation. Cette préoccupation constitue l'objectif poursuivi par la présente thèse. Ces recherches permettront à terme de rendre utilisable, par un profane du domaine, les travaux relatés dans [1]. Ce travail constitue une approche de modélisation des organisations avec la prise en compte de leurs environnements de travail. Il se distingue des autres approches en ceci qu'elle capte les besoins réels des organisations et déduit de ces derniers le modèle de référence de l'organisation. Il permet d'aligner, à tout instant, le modèle de référence de l'organisation aux besoins de cette organisation.

Cependant, la modélisation d'une entreprise renvoie à celle de ses processus

d'affaires. Nous nous intéressons à la modélisation des processus d'affaire qui participent à la réalisation du produit ou du service fourni aux clients de cette entreprise. Ces processus sont appelés processus noyaux ou de production [7]. Dans la suite, la terminologie processus d'affaire renverra au processus d'affaires de production.

II. PROBLEMATIQUE

Concilier l'expression des besoins des organisations, les attentes des bénéficiaires de service et la production du modèle de référence d'une organisation est l'objectif global de nos travaux. Cette préoccupation demande que les procédés de développement intègrent : (a) des mécanismes permettant l'auto-évaluation des performances internes (Qualité de services internes) de l'organisation; (b) des mécanismes permettant la prise en compte de l'évolution des exigences de qualité de la clientèle d'une organisation. Pour ce faire, il est indispensable de disposer de solutions suffisamment adaptables à souhait et d'un mécanisme permettant de passer formellement des besoins d'une organisation à la production du modèle de référence, solution de ces attentes vues comme des contraintes à satisfaire. Pour tenter d'y apporter une solution, une étude a été menée dans ce sens par Farida SEMMAK et Joël BRUNET dans [3] et a conduit à la mise en œuvre d'un méta-modèle orienté buts des besoins d'un domaine. Ce méta-modèle fait usage d'un certain nombre de concepts que sont les fragments de connaissances. Le fragment de connaissance étant représenté par un triplet <but, règle, fragment conceptuel> où le but représente l'attente d'un utilisateur, la règle est une loi du domaine sous laquelle l'attente de l'utilisateur est satisfaite, le fragment conceptuel quant à lui, est une matérialisation de la mise en œuvre de la solution inhérente à cette attente. Le fragment de connaissance constitue donc une matérialisation d'un besoin exprimé. Le méta-modèle est un ensemble hiérarchisé des fragments de connaissances. Des relations peuvent exister entre les besoins d'un même niveau. Farida SEMMAK et Joël BRUNET ont pensé qu'une fois les besoins d'un domaine, clairement représentés, qu'il était possible de déduire une représentation du système informatique sous-jacent. Le méta-modèle proposé par Farida SEMMAK et Joël BRUNET devrait être en

aval d'un processus d'ingénierie de besoins. Cette partie a été négligée jusqu'à nos jours et reste une question ouverte. De plus, la réutilisation des connaissances est un domaine qui connaît un essor accru ces dernières décennies et cela va avec la contextualisation de la connaissance. Nous pensons donc qu'il serait aussi judicieux de coupler les travaux de [3] avec ceux de [1] et d'enrichir le modèle résultant par un processus d'ingénierie capable, à partir des besoins réels d'une organisation, de produire un modèle de référence d'une organisation.

Nous pensons qu'il est important de partir des besoins réels d'une organisation dans la mesure où, suite à l'expérience de la période allant des années 70 à 80 (période pendant laquelle le développement des systèmes informatiques se faisait selon la vision des concepteurs et des ingénieurs, sans l'implication effective des utilisateurs et autres parties prenantes [8], et qui a abouti à l'abandon de nombreux systèmes qui étaient très bien construits d'un point de vue technologique, mais qui ont été considérés comme des échecs du fait qu'ils ne correspondaient pas aux besoins des utilisateurs), il n'est plus possible d'envisager le développement des systèmes informatiques sans prendre en compte les besoins réels des utilisateurs d'une organisation. De plus, par essence, le développement d'applications informatiques se veut une démarche par laquelle les ingénieurs logiciels élaborent des solutions informatiques aux problèmes posés par leurs clients. Il nécessite donc la maîtrise des attentes des utilisateurs, appelées aussi besoins, à partir desquels, le modèle du système informatique sera déduit; et la mise œuvre d'un ensemble d'autres préoccupations non métiers (politique de sécurité, persistance des données, etc.) nécessaires au produit informatique solution inhérente à ces attentes des utilisateurs. De plus, le contexte actuel de mondialisation oblige les organisations à tenir compte de la concurrence. Ces mécanismes ne constituent pas une préoccupation majeure dans les travaux [3] et [1]. Compte tenu des avancées du modèle de [1], et des travaux de Farida SEMMAK et Joël BRUNET relatifs à la définition du méta-modèle permettant la représentation des besoins d'un domaine.

Notre travail consistera à coupler les travaux de [1] et ceux de [3], puis à enrichir le modèle résultant d'un processus d'ingénierie visant, à partir des besoins réels d'une entreprise, d'élaborer un modèle de référence. Bien évidemment, Cela

constituera un enrichissement substantiel des travaux de Atsa Etoundi Roger [1, 2] et Farida Semmak et Joël Brunet [3].

L'objectif poursuivi par ce travail, est l'établissement d'un pont entre l'expression des besoins (attentes des utilisateurs, la satisfaction des bénéficiaires des services, de la qualité de service interne et externe à l'organisation [7]) et la production du modèle de référence de l'organisation. C'est-à-dire la conciliation des besoins exprimés et la modélisation du système d'informations sous-jacent auxdits besoins.

III. OBJECTIF

L'objectif poursuivi par ce travail, est l'établissement d'un pont semi-formel entre l'expression des besoins (attentes des utilisateurs, la satisfaction des bénéficiaires des services (qualité de service externe [7]), de la qualité de service interne à l'organisation) et la production du modèle de référence du système informatique solution inhérente à ces besoins. C'est-à-dire la conciliation des besoins exprimés et la modélisation du système d'informations sous-jacent auxdits besoins. Pour atteindre cet objectif la démarche préconisée devra être suffisamment proche du langage naturel afin que des profanes du génie logiciel ne ménagent aucun effort pour comprendre et valider les différents artefacts de ce processus.

IV. CONTRIBUTION

Pour atteindre notre objectif, nous avons, dans le cadre de cette thèse, dans un premier temps, enrichi, dans [9,10], le concept de fragment de connaissance en y intégrant en lieu et place du fragment conceptuel, les concepts de domaine ou contexte, niveau d'importances et de contraintes dans la représentation d'un fragment de connaissance et avons obtenu la représentation suivante d'un fragment de connaissance : <contexte, but, règle de gestion, contraintes, niveau d'importance> dans laquelle la composante « contexte » ou « domaine » est définie par l'ensemble des connaissances relatives à un champ d'application ; la composante

« but » représente l'intention d'usage [3]; la composante « règle de gestion » représente une loi du domaine à laquelle doit se conformer le but [3]; la composante « contraintes », indique les conditions non fonctionnelles que le but doit satisfaire ; et en dernier lieu la composante « niveau importance » représente l'intérêt que revêt le but du point de vue de l'utilisateur. Dans un second temps, il a été défini, une démarche orientée but pour la représentation des processus d'affaire au travers des besoins de ceux-ci [9, 10]. Cette démarche s'appuie fondamentalement sur les travaux de Farida Semmak et Joël Brunet dans [3] et les enrichissements sus évoqués. Elle utilise le méta-modèle de [3] comme modèle de représentation de base des besoins des organisations. Ce processus d'ingénierie permet : (i) de capter, des utilisateurs, leurs besoins exprimés en langage naturel, (ii) de représenter lesdits besoins, (iii) d'homogénéiser ces besoins, (iv) de déduire un modèle de référence d'un processus d'affaire. Ce modèle est suffisamment souple pour être adapté aux évolutions. Ce travail a permis de minimiser le gap entre l'expression des besoins à la représentation de ceux-ci. Les concepts de but ou intention d'usage, de niveau d'importance ont été formalisés et un pont a été établi entre les travaux de [3] et ceux de [1]. C'est grâce à ce pont que nous avons pu démontrer que notre formalisme permettait de décrire de manière exhaustive le métier d'une organisation. Compte tenu, du caractère capital que revêt une règle métier et son contexte d'usage dans notre modèle de représentation formelle d'une attente d'un utilisateur, il était indispensable de procéder à la formalisation des règles métiers, composante du modèle formel de représentation de besoins exprimés. Dans un troisième temps, il a été défini dans [11, 12] une approche semi-formelle proche du langage naturel pour la spécification des règles métiers. Elle s'appuie sur les travaux de Mouhamed Diouf [13] sur la « spécification et mise en œuvre d'un formalisme de spécification de règles métiers ». Ce travail a permis la formalisation des composantes d'un fragment de connaissance que sont les règles métiers, les contraintes et le contexte ou domaine. Il a aussi contribué dans la même foulée, l'introduction des concepts de sous règles, de composition de règles, de raffinement, d'extension, de fusion et partitionnement des règles métiers. Ce travail a permis de constater qu'on pouvait aussi réécrire un processus d'affaire via ses règles métiers. Dans un quatrième temps, il a été défini dans [14], une démarche

permettant l'introduction des indicateurs de performances dans la modélisation des systèmes d'informations de gestion, permettant ainsi à modéliser la satisfaction des bénéficiaires des services offerts par ce système d'informations. Dans cette perspective, les critères de satisfaction des bénéficiaires des services du processus d'affaire guident le processus d'identification des indicateurs de performance nécessaires. Ce travail a conduit à la mise en place de l'Approche de Modélisation Triangulaire (AMT) ou Approche de Modélisation en Etoile (AME). Ce nouveau concept présente la modélisation d'un processus d'affaire avec une prise en compte de son environnement d'exécution comme étant un processus d'ingénierie tridimensionnel dont le premier axe est constitué des préoccupations sur les tâches ou environnement d'exécution ; le deuxième axe, est constitué des préoccupations relatives aux attentes réelles des cadres métiers ; le troisième axe, quant à lui est constitué des préoccupations relatives à la satisfaction des clients. Chaque axe préoccupation constitue un sous processus d'ingénierie spécialisé pour répondre aux préoccupations de l'axe pour lequel il est rattaché. La figure suivante a été déduite

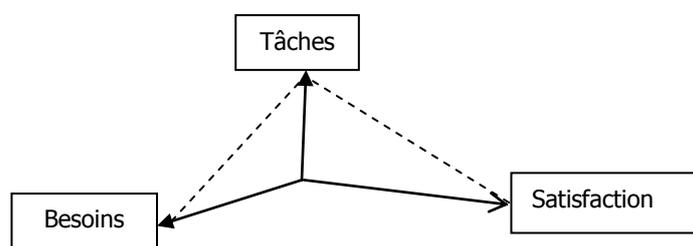


Figure 2 : Modèle d'ingénierie des Processus d'Affaire

Toutefois, notre travail a essentiellement porté sur l'axe des préoccupations relatives aux besoins des différentes parties prenantes du processus d'affaires.

Dans un cinquième temps, il a été défini dans [15], le concept de composant métier orienté aspect et avons par la même occasion, défini les règles de passage du modèle de référence au modèle de composant métier orienté aspect. Ce modèle s'appuie sur les concepts des approches aspect afin d'augmenter les capacités de réutilisations et l'adaptabilité desdits composants, ainsi que les opérations auxquelles sont sujettes lesdits composants. L'ensemble de ces phases contribuent à définir une démarche orientée satisfaction des clients d'une organisation pour la modélisation

des processus d'affaire. Ce travail, s'inscrit donc, dans le cadre de la modélisation des processus d'affaire avec prise en compte de l'environnement d'exécution de ceux-ci, et vise la mise sur pied d'une plateforme de modélisation des processus d'affaires des organisations.

L'avantage principal de cette démarche, est de viser la production automatique des systèmes informatiques, à partir des seuls besoins recueillis des utilisateurs, les attentes des bénéficiaires des services et la nécessité de pouvoir évaluer le rendement des cadres métiers de cette organisation. Compte tenu de ce qui précède, notamment la proximité avec le langage naturel, ce modèle permet de réduire le mal entendu entre utilisateurs et analystes des systèmes informatiques. Ce qui se justifie par le fait qu'il établit un pont formel entre l'expression des besoins et leurs représentations formelles. De plus, en éclatant le processus d'ingénierie des processus d'affaire, en trois sous processus d'ingénierie répondant chacun, à une préoccupation précise ; le processus proposé est adapté même aux approches de développement par aspects ; ouvrant ainsi la porte à la modélisation des composants métiers orientés aspects. Ils sont suffisamment souples dans la maintenance corrective et évolutive de ces composants. Cela provient du fait, qu'il y a dès la phase d'analyse, une séparation des différentes préoccupations. Cependant, il serait intéressant de définir une plateforme supportant cette démarche de modélisation. Cet objectif est en perspective dans cette thèse, ainsi que la finalisation des composants métiers orientés aspect.

V. PLAN DU MEMOIRE DE THESE

Compte tenu de la problématique sus évoquée et de l'objectif poursuivi, le reste du travail est articulé autour de la mosaïque composée de : un état de l'art sur les approches de modélisations des organisations et plus spécifiquement des processus d'affaire, et l'ingénierie des besoins ; à la définition d'un processus d'ingénierie des processus d'affaire d'une organisation ; à une étude de cas permettant la validation de notre modèle ; et enfin les perspectives.

PREMIER CHAPITRE: MODELISATION DES ORGANISATIONS

La section précédente nous a permis de présenter d'une manière synoptique la quintessence de cette thèse. Dans ce chapitre, il sera question de présenter de manière succincte les concepts régissant la modélisation des organisations et de nous situer par rapport à ces dernières. Ainsi, trois(03) concepts globaux ont retenu notre attention : La modélisation des processus d'affaires, la modélisation des besoins d'un processus d'affaires, à travers les processus d'ingénierie des besoins orientés buts et la qualité de service. Pour ce qui est de la modélisation des processus d'affaire, après avoir fait un bref aperçu des concepts de base, nous allons présenter les approches et les langages de modélisation des processus d'affaire. En ce qui concerne l'ingénierie de besoins, nous allons focaliser notre attention sur les approches d'ingénierie de besoins orientées buts car elles sont adaptées à la modélisation des processus d'affaire. Pour ce qui est de la qualité de service, nous avons fait un bref aperçu de cette dernière, notre objectif était d'implanter les concepts de qualité de services internes et externes et d'évoquer les normes iso qui régissent ces différentes qualités de service

DEUXIEME CHAPITRE: MODELE DE PROCESSUS D'INGENIERIE DES PROCESSUS D'AFFAIRE DES ORGANISATIONS

Définir un modèle de processus d'ingénierie pour représenter les besoins d'un domaine est l'objectif poursuivi par cette thèse. Dans les chapitres I et II, nous avons présenté les concepts généraux relatifs à notre domaine d'étude. Dans ce chapitre, nous présentons un processus d'ingénierie des besoins permettant de partir des besoins du monde réel (attentes des cadres métiers de l'organisation et exigences de qualité des bénéficiaires des services offerts par cette organisation) pour élaborer le modèle de référence conformément au formalisme de [3]. Cette approche est basée sur quatre (04) étapes fondamentales : le recensement, la sélection, l'élaboration du modèle, et la validation ; L'étape de recensement permet de collecter les besoins des cadres métiers de l'organisation et les exigences de qualités des clients de cette organisation. L'étape de sélection permet de retenir ceux des besoins de

l'organisation qui sont consistants ; L'étape d'élaboration du modèle permet dans un premier temps, de transformer les besoins de l'organisation en fragment de connaissance et les exigences de qualité de la clientèle de cette organisation en connaissance de la satisfaction de ces derniers. Dans un second temps, de construire le modèle sus évoqué, à travers un ensemble d'axiomes définis à cet effet. L'étape de validation quant à elle permet la validation du modèle construit. Dans cette étape, il sera question de se rassurer de la cohérence des différents concepts. Pour cela, il a été défini un éventail d'axiomes. Ce travail couvre trois activités de l'ingénierie des besoins que sont : l'élicitation, la sélection et la validation des besoins étant entendu que la spécification desdits besoins a été définie dans [3]. Naturellement, pour arriver au résultat escompté, des enrichissements ont été apportés au formalisme, objet des travaux dans [3]. Ils concernent le fragment de connaissance. En effet, nous avons substitué le fragment conceptuel par trois composantes : « le contexte », le « niveau d'importance », les « contraintes ». le « contexte », première composante du fragment de connaissance enrichi, décrit l'environnement d'utilisation du fragment de connaissance ; le « niveau d'importance », dernière composante du fragment de connaissance enrichi, traduit l'importance que revêt cette connaissance dans la chaîne de production du processus d'affaire ; et les « contraintes », quatrième composante du fragment de connaissance enrichi, représentent les besoins non fonctionnels associés à l'objectif visé ou but.

TROISIEME CHAPITRE: DYNAMIQUE DU MODELE DE REFERENCE

Un besoin est par essence une information dynamique. C'est-à-dire une information qui évolue au fil du temps. Ce qui induit, sur la base de cette dynamique, des réajustements/remplacements permanents des systèmes informatiques soutenant l'activité des organisations. L'incidence directe de cela, dans la modélisation des processus d'affaire, est l'obsolescence rapide et récurrente des spécifications des processus d'affaire dans les organisations. C'est-à-dire une information qui évolue au fil du temps. Il a été présenté dans le chapitre précédent une démarche permettant de capter et de construire le modèle de référence d'une organisation. Ce modèle

intègre en son sein la satisfaction des clients de cette organisation. Il est construit à partir de fragments de connaissances. Les fragments de connaissances sont issus des besoins exprimés par ladite organisation. De ce fait, le concept de besoins est intimement lié à celui de fragment de connaissance. Notre travail dans ce chapitre consiste donc à présenter la dynamique du besoin et indécemment de celle du fragment de connaissance qui lui est associé dans notre modèle de référence ci-dessus mentionné. Cette évolution vise l'amélioration de la qualité de service interne de l'organisation. Pour ce faire il a été d'abord étudié la dynamique de cette qualité de service externe au sein de l'organisation. En second lieu, il a été étudié la dynamique du besoin/fragment de connaissance par rapport à une qualité de service désirée. A la fin, la dynamique du modèle de référence résultant des besoins des cadres métiers et des attentes des bénéficiaires de service de cette organisation a été déduite. A l'issue de cette étude, il a été identifié trois points fixes. Le premier a trait à la qualité de service perçue par les bénéficiaires de service ; le second se pour lequel la plupart des clients et le staff dirigeant de l'organisation sont satisfaits. Ces points fixes matérialisent la limite de la dynamique du modèle de référence. Ils permettent d'établir que le modèle de référence n'a pas une évolution infinie. Impliquant par là même que lorsqu'une organisation a axé son processus d'affaire sur la satisfaction des ses clients(ou la gestion axée sur les résultats), l'évolution de son processus d'affaires admet une limite au delà de laquelle tout enrichissement n'améliore plus la qualité de service.

QUATRIEME CHAPITRE: CAS D'UTILISATION

Ce chapitre est consacré à la présentation d'un cas d'utilisation de notre processus d'ingénierie. Cette illustration s'appuie sur les processus de liquidation des droits dans la fonction publique camerounaise. La liquidation des droits est un ensemble de processus permettant à un agent public de bénéficier de ses droits à pension, une fois ce dernier sorti des effectifs de la fonction publique d'une part, de reverser les droits de ce dernier à ses ayants droit d'autre part. Ce chapitre présente un cas d'utilisation de la démarche présentée dans le chapitre précédent, et s'appuie sur des besoins réels collectés auprès des cadres métiers en charge de ces processus.

Compte tenu des contraintes administratives, les fiches de recensements de besoins ont été paraphées uniquement par les responsables nommés au sein de la sous direction de la liquidation des droits. Nous nous sommes beaucoup intéressés au volet qui posait le plus de problèmes à l'administration.

CONCLUSION ET PERSPECTIVES

Cette partie présentera la conclusion générale et les perspectives de ces travaux.

**CHAPITRE II:
MODELISATION DES
ORGANISATIONS**

Résumé

Le chapitre précédent nous a permis de présenter d'une manière synoptique la quintessence de cette thèse. Dans ce chapitre, il sera question de présenter de manière succincte les concepts régissant la modélisation des organisations et de nous situer par rapport à ces dernières. Ainsi, trois(03) concepts globaux ont retenu notre attention : La modélisation des processus d'affaires, la modélisation des besoins d'un processus d'affaires, à travers les processus d'ingénierie des besoins orientés buts et la qualité de service. Pour ce qui est de la modélisation des processus d'affaire, après avoir fait un bref aperçu des concepts de base, nous allons présenter les approches et les langages de modélisation des processus d'affaire. En ce qui concerne l'ingénierie des besoins, nous allons focaliser notre attention sur les approches d'ingénierie de besoins orientées buts car elles sont adaptées à la modélisation des processus d'affaire. Pour ce qui est de la qualité de service, nous avons fait un bref aperçu de cette dernière, notre objectif était d'implanter les concepts de qualité de services internes et externes et d'évoquer les normes iso qui régissent ces différentes qualités de services.

I. LA MODÉLISATION DES ORGANISATIONS

La gestion de processus d'affaire est basée sur l'observation que chaque produit fourni par une société est le fruit d'un nombre d'activités établies. Les processus d'affaire constituent l'instrument clé pour organiser ces activités et prouver la compréhension de leurs relations. Ce chapitre est consacré à la présentation d'un aperçu des principales méthodes de modélisation d'entreprise. Il est déduit en grande partie de la thèse d'Annis Ferchichi et des travaux de Farida Semmak et Joel Brunet. Nous commençons par présenter la notion de processus métier. Ensuite, nous proposons une classification des principaux langages de modélisation.

I.1 INTRODUCTION

La modélisation des organisations est un terme générique qui renvoie à la spécification des procédures d'entreprise (processus métier, processus de pilotage). Elle désigne la plupart du temps la création d'un modèle mathématique d'une procédure d'entreprise. Dans une organisation, les processus d'affaire sont les processus représentatifs des activités de cette organisation, indépendamment des moyens humains et techniques. Ces processus interfèrent de manière transversale dans le système d'information et peuvent même traverser les frontières organisationnelles internes de l'organisation. Les processus d'affaire traversent même les limites de l'entreprise pour collaborer avec les partenaires [16]. Chaque processus d'affaire est identifié par un nom, un objectif et son degré de succès qualitativement ou quantitativement mesurable. Les activités d'un processus d'affaire sont exécutées par des acteurs jouant des rôles particuliers, consommant quelques ressources et produisant d'autres. Les activités peuvent être déclenchées par des événements et peuvent à leur tour produire des événements. Les activités d'un processus d'affaire peuvent être liées par des dépendances de ressource (dépendances de producteur-consommateur) ou des dépendances de commande (une activité déclenchant une autre). Les acteurs opèrent à l'intérieur des frontières des organisations qui exécutent des fonctions métiers spécifiques. Les rôles peuvent soutenir des fonctions. La figure 3 montre un méta-modèle de processus d'affaire [16].

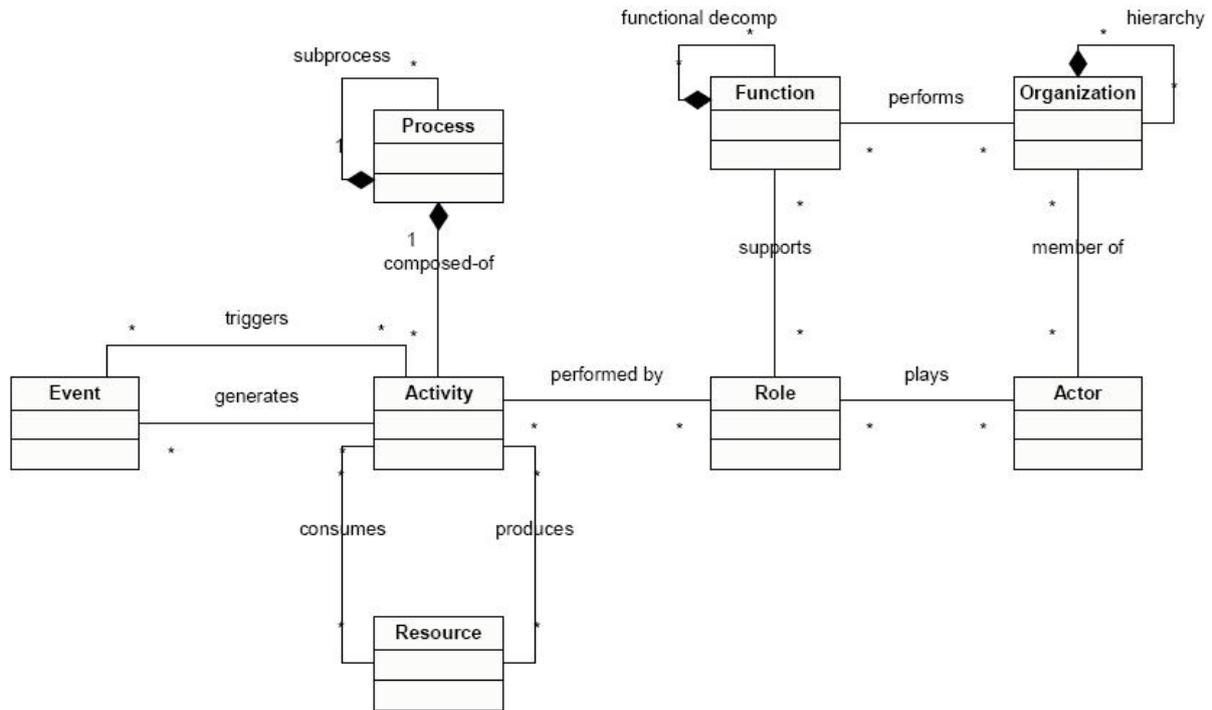


Figure 3 – Un méta-modèle de processus [16]

Habituellement, un métier ou affaire est considéré comme une organisation hiérarchique qui reflète la décomposition fonctionnelle de l’entreprise et sa chaîne de production. Les différents départements sont spécialisés dans des fonctions spécifiques (par exemple, vente, production ou comptabilité), et dans chaque département, des sous-départements, équipes ou individus sont spécialisés dans des sous-fonctions.

Le terme Business Process Management (BPM) est apparu pour la première fois au milieu des années 1990 dans le contexte de la gestion des stratégies d’entreprise en se basant sur les processus métier. Depuis, plusieurs définitions ont été énoncées. Nous avons retenu celle qui semble être la plus complète et la plus pertinente. En effet, Le BPM peut être défini comme un support des processus métier en utilisant des méthodes, des techniques et des outils logiciels afin de concevoir, modéliser, commander et analyser les processus opérationnels impliquant des humains, des organisations, des applications et tout autres sources d’information » [17]. Cette définition, dans sa deuxième partie, montre la nécessité de mettre en place un support pour les processus opérationnels. Ainsi, le terme Gestion des Processus métier (Business Process Management) inclut l’analyse, la

définition et la redéfinition des processus de l'organisation, l'allocation des ressources, la planification, la gestion des processus, la mesure de la qualité et de l'efficacité à l'aide d'indicateurs et l'optimisation des processus. En plus, l'optimisation des processus nécessite la collecte et l'analyse de mesures en temps réels (monitoring) et de mesures stratégiques (gestion des performances) et leur corrélation, en tant que base pour l'amélioration et l'innovation [18].

L'approche idéale de BPM n'est pas de forcer une organisation à se comporter d'une certaine manière formelle, mais plutôt de comprendre son comportement par des concepts et des principes qu'offre le BPM. C'est un processus de découverte de la connaissance (Knowledge Discovery Process) qui nécessite un effort considérable.

Les premières théories de gestion d'entreprise se sont concentrées sur le fonctionnement de l'entreprise et le contrôle de ses départements mais en se focalisant sur chaque département pris séparément. Avec l'apparition du BPM (Business Process Management) et BPR (Business Process Re-engineering), au lieu de se concentrer sur chaque fonction de l'entreprise séparément et de ce fait, ne pas remettre en cause la structure globale des processus ; les processus métier sont vus dans leurs globalités, espérant ainsi l'optimiser dans son entièreté. Pratiquement, cette tendance a été à l'origine de l'apparition de nouvelles structures organisationnelles telles que la gestion orientée projet où des équipes de projet sont constituées à partir de différentes unités pour gérer tous les aspects d'un projet ou encore l'organisation matricielle où les équipes métier (les colonnes de la matrice) offrent des services aux clients externes en managant les équipes réalisant les services (les lignes de la matrice) [19]. Cependant, toute modélisation d'un processus d'affaire d'une organisation par les experts du domaine fait suite à un besoin exprimé et vise l'amélioration de la production de l'organisation et la satisfaction des clients de cette organisation. Une bonne modélisation d'une organisation nécessite donc la prise en compte de ces différents éléments.

Dans la suite, notre travail sera organisé autour de trois grandes parties. La première partie portera sur la modélisation des processus d'affaire, la deuxième sur l'ingénierie des besoins des processus d'affaire et la troisième, portera sur un bref aperçu de la qualité de service. Nous terminerons ce chapitre par une conclusion

partielle.

I. 2 LES PROCESSUS D'AFFAIRE

La technologie de l'information en général et les systèmes d'information en particulier jouent un rôle important dans la gestion des processus d'affaires. En effet, plusieurs activités qu'une compagnie exécute sont supportées par des systèmes d'information. Les activités des processus métier peuvent être exécutées par les employés d'une compagnie manuellement ou bien à l'aide de systèmes d'information sans aucune intervention humaine. Une entreprise peut atteindre ses objectifs métier d'une manière efficace et effective seulement si les humains et d'autres ressources d'entreprise, tels que les systèmes d'information, travaillent ensemble. Les processus d'affaire forment un concept important pour faciliter cette collaboration effective. Dans plusieurs compagnies, il y a une séparation entre les aspects métier organisationnels et la technologie de gestion de l'information mise en place. Rétrécir cette séparation entre organisation et technologie est important, car dans les marchés actuels, les entreprises sont constamment forcées à fournir des produits meilleurs et plus spécifiques pour leurs clients. Au niveau organisationnel, les processus métier sont essentiels pour comprendre comment les compagnies opèrent. De plus, ils jouent un rôle important dans la conception et la réalisation de systèmes d'information flexibles [20]. Ces systèmes d'information fournissent la base technique pour la création rapide de nouvelles fonctionnalités afin de réaliser de nouveaux produits en vue de la satisfaction de nouvelles exigences du marché. La gestion des processus d'affaire est influencée par les concepts et les technologies de différents domaines d'administration de l'organisation et de l'informatique. Basée sur des travaux récents dans l'organisation et la gestion, la gestion des processus d'affaire trouve ses racines dans la tendance de l'orientation processus des années 90, où un nouveau chemin d'organisation des compagnies se basant sur les processus d'affaire a été proposé.

Un processus d'affaire a été défini, par Michael Hammer et James Champy dans [21], comme une collection d'activités qui prennent un ou plusieurs types d'entrée et créent une sortie qui vaut une valeur pour le client. Cette définition met l'accent sur le comportement de l'entrée/sortie d'un processus d'affaire en mentionnant ses pré-

conditions (entrées) et ses post-conditions (sorties). Le processus lui-même est décrit de façon abstraite par une collection d'activités. Notons que le terme « collection » n'implique ni un classement des activités ni toutes autres contraintes d'exécution, la définition par Hammer et Champy est tout à fait libérale quant à l'aspect du processus d'affaire. Les contraintes d'exécution entre activités sont identifiées dans [22], qui définissent un processus métier comme « un ensemble de tâches, logiquement reliées, exécutées afin d'accomplir un objectif métier pour un client particulier ou un marché ». En se basant sur cette caractérisation des processus d'affaire, [20] propose l'adoption de la définition suivante :

Définition 2.1 : Un processus d'affaire est un ensemble d'activités qui sont exécutées en coordination dans un environnement organisationnel et technique. Ces activités réalisent en commun un objectif d'entreprise (commercial). Chaque processus métier est exécuté par une seule organisation, mais il peut interagir avec d'autres processus métier réalisés par d'autres organisations.

Après une première considération des processus d'affaire, leurs composants, et leurs interactions, la vue est élargie. La gestion de processus d'affaire ne couvre pas seulement les processus d'affaire mais aussi d'autres activités.

La base de la gestion de processus d'affaire est la représentation explicite des processus avec leurs activités et les contraintes de l'exécution entre elles. Une fois que les processus d'affaire sont définis, ils peuvent faire l'objet d'analyse, d'amélioration et d'exécution. Les organisations ont un grand nombre de processus concourants. Par exemple, le conseil d'administration a ses propres processus prédéfinis pour la prise de décision, pour nommer le personnel et les membres du conseil. Le Directeur aura ses propres processus de décision et de distribution des responsabilités. Chaque département fonctionnel ou division (comptabilité, vente, production, service à la clientèle,...) aura ses propres processus. Chaque niveau de l'organisation aura ses propres objectifs, ses propres processus, ses propres mesures de performances. Ainsi pour parler de « modélisation de processus d'affaire », nous devons identifier quels sont les processus qui nous intéressent, le niveau de détail et les éventuels rapports entre ces processus.

Implémenter un processus d'affaire, c'est utiliser les ressources logicielles et matérielles adéquates pour exécuter ses activités. Les activités sont implémentées comme des processus matériels et/ou électroniques. Les processus matériels incluent la production, la transformation, la mesure et l'assemblage d'objets physiques. Un exemple de processus matériel est l'assemblage de voitures. Les processus électroniques, d'un autre côté, impliquent des systèmes logiciels pour créer, gérer et fournir de l'information. Un exemple de processus électronique est l'accès à des catalogues de produits d'une vente en ligne.

Ould distingue trois types de processus d'affaire : les processus noyau, les processus de management et les processus support [23]. En effet, les processus noyau sont liés à la fourniture du produit ou du service aux clients et représentent le cœur du métier. Par exemple, pour une société de service, ces processus sont appelés les processus de réalisation. Les processus de management sont les processus de direction qui permettent de piloter l'entité (société ou région) et d'en améliorer le fonctionnement. Les processus support contribuent au bon fonctionnement de tous les processus, ça concerne par exemple la gestion des collaborateurs. En pratique, nous analysons :

- les processus métier (processus noyau) pour améliorer la satisfaction du client ;
- les processus support pour améliorer l'efficacité de l'entreprise ;
- les processus de gestion pour améliorer la structure d'entreprise.

Comme susdit, nous focaliserons notre attention sur les processus noyaux car, ils sont gage de la satisfaction des clients de l'organisation [23]. La section suivante sera consacrée à la présentation du concept de modélisation des processus d'affaire.

I.2.1 MODELISATION DE PROCESSUS D'AFFAIRE

I.2.1.1 CONCEPTS DE BASE

Un modèle de processus métier consiste en un ensemble de modèles d'activités et de contraintes d'exécution entre eux. Un exemple de processus métier représente un cas concret dans le processus opérationnel d'une compagnie, consistant en des instances d'activités. Chaque modèle de processus métier opère comme un plan pour un ensemble d'instances dudit processus métier, et chaque modèle d'activité opère

comme un plan pour un ensemble d'instances de ladite activité. Les modèles du processus métier sont les objets principaux pour mettre en œuvre ces processus métiers. Cette mise en œuvre peut être faite par des règles d'organisation et des politiques, elle peut aussi être faite par un système logiciel en utilisant un système de gestion de processus métier. Dans ce cas, ce système logiciel est conduit par des représentations explicites du processus. Puisque par définition les processus métier sont exécutés dans une seule organisation, le cheminement d'activités peut être contrôlé par un système de gestion de processus métier comme un composant logiciel centralisé exécuté par le revendeur de la compagnie. Ce contrôle centralisé est très semblable à un conducteur qui contrôle centralement les musiciens dans un orchestre ; par conséquent, les processus métier sont aussi appelés des orchestrations de processus.

I.2.1.2 TECHNOLOGIES DE SPECIFICATION DES PROCESSUS D'AFFAIRE

Les Méthodes de Modélisation d'Entreprise (MME) ont fait leur apparition sous différentes appellations vers le début des années 80. On peut citer parmi les premiers travaux, le programme ICAM (Integrated Computer Aided Manufacturing) de l'U.S. Air Force qui a produit à cette époque la méthode IDEF (ICAM DEFINITION). La définition d'une MME diffère d'un ouvrage à un autre mais celle qui peut résumer les différentes approches est la suivante : Une méthode de modélisation d'entreprise consiste à "décrire l'organisation des processus d'un système soit dans le but de les simuler pour comparer divers scénarios, soit dans le but de les analyser et de les restructurer pour améliorer la performance dudit système" [24]. Une MME comporte les éléments suivants :

- un modèle conceptuel de référence, définissant les objectifs de l'entreprise, ses fonctionnalités, sa structure, son environnement et son comportement dynamique;
- des formalismes de modélisation, permettant de représenter les différents concepts du modèle conceptuel de référence, de préférence de façon graphique, pour des raisons de clarté;
- une démarche, qui structure les étapes d'élaboration des différents modèles

(actions à mener, validations à effectuer) [24].

Les langages de modélisation d'entreprise proviennent de différentes communautés scientifiques et permettent d'atteindre plusieurs objectifs et de représenter différentes vues de l'entreprise. Cependant représenter un processus métier est utile pour trois raisons :

- 1) décrire un processus : nous modélisons un processus pour pouvoir le décrire et le documenter. Nous pouvons avoir plusieurs cibles pour cette description :
 - a. des humains : dans ce cas la compréhension est importante;
 - b. des machines : dans ce cas le formalisme de modélisation est plus important.
- 2) analyser un processus : l'analyse d'un processus consiste en l'évaluation des propriétés d'un processus. L'amélioration des processus se base sur l'analyse des processus existants pour identifier les étapes redondantes ou non optimales. Si le processus est décrit en utilisant une méthode formelle, nous pouvons vérifier les propriétés structurelles des processus telles le couplage ou la cohésion ou encore les propriétés dynamiques telles l'absence d'inter-blocage (deadlock) ou la viabilité du processus;
- 3) exécuter des processus : à l'aide d'un outil capable aussi bien d'offrir une interface homme/machine pour l'exécution des tâches par des personnes que de s'intégrer et de faire appel aux applications du système d'information de l'entreprise, et ceci malgré leur hétérogénéité, leur disparité géographique et leur appartenance à des entités différentes.

Puisque les processus métier sont généralement complexes, les concepteurs des langages de modélisation fournissent différentes vues pour la modélisation, chacune se focalise sur un aspect du processus. Curtis identifie quatre vues [25] :

- 1) une vue fonctionnelle : cette vue représente les dépendances fonctionnelles entre les composants d'un processus (activités, sous-processus...). Ces dépendances résultent du fait que certains composants du processus consomment (ou nécessitent) des données (ou des ressources) produites par

d'autres composants (ou processus). Les notations typiques utilisées dans la vue fonctionnelle incluent des diagrammes d'échange de données;

- 2) une vue dynamique (comportementale) : la vue dynamique fournit l'ordonnancement et les paramètres d'exécution d'un processus. Cette vue permet de répondre à deux questions : quand certaines activités sont-elles exécutées (synchronisation, conditions préalables) et comment sont-elles exécutées (par exemple, en décrivant la logique d'exécution);
- 3) une vue informationnelle : cette vue inclut la description des éléments qui sont produits, consommés ou manipulés par le processus. Ces éléments incluent des données, des artefacts, des produits.
- 4) une vue organisationnelle : cette vue décrit qui exécute chaque tâche ou fonction, et où ça se passe dans l'organisation (fonctionnellement et physiquement).

Dans la suite, nous étudierons un certain nombre de langages de modélisation qui ont été développés pour décrire les processus d'affaire avec des objectifs différents. Ces langages représentent différentes vues du processus d'affaire (dynamique, fonctionnelle, informationnelle, organisationnelle) d'une façon plus ou moins formelle. Ces langages peuvent être classés comme suit :

- 1) les langages traditionnels de modélisation : Ces langages proviennent la plupart du temps des travaux de la mise en place des systèmes d'information et l'ingénierie des processus métier. Ces langages ne sont typiquement pas formels, mais peuvent se prêter à diverses analyses heuristiques ou informelles. Parmi ces langages, on trouve : IDEF, les réseaux de Petri, Event Process Chains (EPC), Role Activity Diagrams (RAD), Resource-Event-Agent (REA) et les tout récents Business Process Modeling Language (BPML) [26] et Business Process Modeling Notation (BPMN) [27] [28];
- 2) les langages de modélisation des Workflow : Un Workflow peut être défini comme l' "automatisation de processus d'affaires par échange de documents, informations et tâches entre acteurs pour action". Le workflow a pour objectif la

coordination automatisée de tâches réalisées par des intervenants humains. Le moteur de workflow transfère des documents entre les participants au processus en leur assignant des tâches (valider le document, effectuer une modification...). En général, un langage de modélisation des Workflow permet de décrire les opérations de façon à être supportées par un système de gestion de Workflow. Ces langages sont, pour la plupart, formels et exécutables. Nous pouvons citer le Workflow Process Description Language (WPDL) et les formats d'échange tels que Process Interchange Format (PIF) et Process Specification Language (PSL);

- 3) les langages d'intégration de processus : L'apparition des outils de Business to Business integration (B2Bi) a stimulé l'intérêt pour la modélisation des processus d'affaire dans les buts d'intégrer les processus de deux partenaires ou plus. De tels langages se concentrent sur les mécanismes d'intégration en termes d'abstraction, des interfaces de programmation et des formats d'échange de données. Cette intégration doit être réalisée indépendamment de la technologie. Dans cette catégorie de langages, on peut citer ebXML et Business Process Execution Language for Web Services (BPEL4WS);
- 4) les langages orientés objets : En dépit de ce que ces langages soient plus utilisés pour la modélisation des solutions logicielles, ils peuvent aussi être utilisés pour la modélisation d'entreprise grâce à un certain nombre de mécanismes. Le standard le plus utilisé est Unified Modeling Language (UML) qui permet de modéliser des processus d'affaire grâce à un mécanisme d'extensibilité permettant de spécialiser les diagrammes généraux.

Dans la suite, nous allons présenter en détail certains de ces langages et/ou technologies qui implémentent et gèrent les processus d'affaire.

I.2.2. TECHNOLOGIE DE MODELISATION DES PROCESSUS D'AFFAIRE

I.2.2.1 TECHNIQUES AD HOC

Les processus d'affaire peuvent être implémentés en utilisant les langages de programmation tels que C et Java. Dans cette approche, le processus métier est modélisé (par exemple comme un document Word). Ensuite, les activités principales

sont implémentées par des programmes. L'interaction entre ces programmes est supportée par des interconnexions entre des protocoles de communication appropriés (par exemple, les sockets TCP/IP). Le problème de cette solution est que tous les protocoles de communication utilisés doivent être compris. En même temps, toute modification dans le protocole de communication ou la logique métier requiert la recompilation des programmes.

I.2.2.2 WORKFLOWS

Un workflow [29] est l'automatisation d'un processus métier, en tout ou en partie, durant lequel des documents, informations, ou tâches sont passés d'un participant à un autre pour action, d'après un ensemble de règles procédurales [30]. Ayant la définition de workflow, il y a un grand nombre d'activités métier dans une organisation qui se classifient dans la catégorie de workflow. Elles incluent un ordre d'achat, une demande de crédit, une location de voitures, etc.

I.2.2.3 SERVICES WEB

Le concept de service Web est devenu récemment très populaire. Un service Web (ou simplement service) est une application autonome ou un composant, c'est-à-dire, une fonctionnalité sémantiquement bien définie, uniquement identifiée par un identificateur uniforme de ressource (ou Uniform Resource Identifier URI). Comme exemples typiques de services Web, nous citons les réservations de billets en lignes, la gestion des relations entre clients, la facturation, la comptabilité et la chaîne de la provision. Une définition plus précise des services Web est fournie par le dictionnaire Webopedia [31]. Il définit un service Web comme "une manière standardisée d'intégration des applications basées sur le Web en utilisant les standards ouverts XML, SOAP, WSDL, UDDI et les protocoles de transport de l'Internet. XML est utilisé pour représenter les données, SOAP pour transporter les données, WSDL pour décrire les services disponibles, et UDDI pour lister les fournisseurs de services et les services disponibles". Les services Web présentent l'application la plus concrète de SOA (Service Oriented Architecture). D'après cette définition, nous retenons qu'un service Web est décrit dans un document WSDL (Web Service Description Language), précisant les méthodes (fonctionnalités) invoquées, leur signature et leurs points d'accès du service (URL, port, etc.). Ces méthodes peuvent être

accessibles à travers des requêtes via une couche dédiée au transport de messages : SOAP (Simple Object Access Protocol). La requête et la réponse forment des messages XML transportés par HTTP (Hyper Text Transfer Protocol). Un service Web est référencé dans une sorte d'annuaire UDDI (Universal Description, Discovery and Integration) qui facilite aux utilisateurs l'accès à ce service. Un service Web peut être utilisé pour exporter des fonctionnalités à partir d'une application d'entreprise et pour les rendre accessibles via des protocoles standards. Le service Web forme donc l'interface d'accès et de dialogue avec l'application.

I.2.2.4 ELECTRONIC BUSINESS XML (EBXML) ET ONTOLOGY WEB LANGUAGE FOR SERVICES (OWL-S)

Une technologie de processus métier inclut Electronic Business XML et Web Ontology Language for Services. Elles sont brièvement décrites comme suit :

- Electronic Business XML (ebXML) : ebXML définit un ensemble de spécifications pour permettre des interactions B2B entre des organisations. La partie basique de l'infrastructure ebXML est le repository. Il sauvegarde des informations d'entreprise importantes avec les produits et services qu'elles offrent. A la couche communication, l'échange de messages se fait à travers le service de messagerie. Le dernier permet l'utilisation de protocoles communs tels que SMTP, HTTP et FTP. A la couche de processus métier, ebXML définit un schéma de la spécification du processus métier disponible en UML ou XML. Elle fournit un ensemble de spécifications de processus métier qui sont partagées entre plusieurs industries en vue de construire des processus métier personnalisés. Les interactions entre les processus métier sont représentées à travers des chorégraphies. Une chorégraphie spécifie l'ordre et les transitions entre les transactions métier. Pour modéliser la collaboration, ebXML définit les « Collaborations Protocol Agreements (CPAs) ». Un CPA est un accord entre deux partenaires qui spécifie d'avance les conditions sous lesquelles ils vont collaborer (par exemple termes de paiement).
- Ontology Web Language for Services (OWL-S) (connu anciennement par DAML-S) vise à réaliser une vision de Web sémantique pour rendre les ressources Web disponibles aussi bien par contenu que par mots clés. OWL-S organise les

services Web en ontologies. Une ontologie est définie par une spécification formelle et explicite d'une conception partagée [32]. OWL-S divise les descriptions de service en profil de service, modèle et fondement. Le profil de service fournit une description de haut niveau d'un service Web. Il exprime l'entrée requise du service et les sorties que le service va fournir au demandeur. Le modèle de service définit les opérations et leurs flots d'exécution dans le service Web. Le fondement de service fournit un mapping entre OWL-S et le standard WSDL et décrit comment le service est invoqué.

I.2.2.5 METHODES IDEF - ICAM DEFINITION LANGUAGE

IDEF est une famille de méthodes pour modéliser et analyser l'entreprise. Elle a été sponsorisée par US Air Force dans le cadre de son programme de fabrication assistée par ordinateur (Integrated Computer-Aided Manufacturing - ICAM).

Le programme, lancé dans les années 70, a visé l'augmentation de la productivité des organisations par l'application systématique de l'informatique. Le programme a identifié l'analyse de processus comme un outil important et le besoin de mise en place de meilleures techniques pour la description des processus.

I.2.2.5.1 IDEFO

IDEFO est un standard constitué d'une carte de haut niveau des processus métier majeurs utilisés par une organisation. La méthode SADT (Structured Analysis and Design Technique) a été la première à être rendue publique sous le vocable IDEFO [29]. Selon IDEFO, une activité peut être vue comme une fonction qui transforme des objets d'entrée en objets de sortie (réalisation d'une tâche). Elle adopte une approche systémique en ce sens qu'elle considère que tout système complexe est une structure composée de systèmes plus simples en interaction. Ce modèle se caractérise par son excellence en termes d'appropriation par les utilisateurs, puisque sa notation graphique et sa syntaxe sont simples et naturelles. Une activité (boîte noire) consomme des entrées pour produire des sorties, à partir de directives de contrôle (informations), en s'appuyant sur les potentialités des mécanismes (ressources).

Un diagramme IDEFO est donc un ensemble de fonctions connectées par des flux.

Il existe plusieurs niveaux de détails associés à chaque activité. Il est effectivement possible de décomposer une activité en sous-activités. Ce principe de décomposition fonctionnelle permet de jouer facilement sur le niveau d'abstraction tout en garantissant une cohérence entre les différents degrés de finesse. Cette modélisation est descendante, modulaire, structurée et hiérarchisée. Enfin, le principe de la méthode limite son usage à un point de vue fonctionnel [29] [20].

I.2.2.5.2 IDEF1

Les concepts de base de ces modèles peuvent être explicités au travers de la sémantique employée. "Une entité est un ensemble d'objets réels ou abstraits, ayant chacune une existence propre, partageant un ensemble de propriétés communes et présentant un intérêt pour l'organisation" [33]. Une relation est une association entre deux entités. Ces principaux éléments (attributs, clé, cardinalité...) permettent de définir graphiquement les besoins d'une part et de spécifier la conception de modèles de données d'autre part. Il s'agit donc d'une méthode particulièrement axée sur la vue informationnelle dans son acception conceptuelle et structurelle (même si le niveau « Physique » est également intégré à la méthode) [34].

I.2.2.5.3 IDEF2

IDEF2 est un langage de modélisation du comportement d'un système basé sur le principe des files d'attente. C'est une méthode complémentaire à IDEF0/SADT qui vise à répondre aux lacunes du point de vue de l'analyse des aspects dynamiques d'un système. Cette méthode est basée sur quatre modèles : système physique, flux des entités, contrôle du système et gestion des ressources. IDEF2 aborde donc de façon privilégiée les vues informationnelles et ressources sur des niveaux d'abstraction proches de l'exécution [31].

I.2.2.5.4 IDEF3

La méthode IDEF3 a été proposée en 1992 pour compenser les limites d'IDEF0 en matière de modélisation du comportement de l'organisation. C'est une méthode qui se limite à la saisie et à la description des processus en utilisant une notation graphique. IDEF3 modélise un processus sous forme d'étapes, appelées unités de comportement, connectées par des boîtes de jonction et des liens. Une telle représentation forme un diagramme appelé description des flux de contrôle du

processus. En d'autres termes, on retrouve une analyse macroscopique (niveau d'abstraction conceptuel) des vues fonctionnelle et organisationnelle. IDEF3 s'appuie pour ce faire sur la décomposition de l'IDEF0 en termes de typologie de système et de niveau d'avancement du projet. IDEF3 s'avère être une méthode intéressante pour la description de flux de processus. Néanmoins, ses composants n'offrent pas de description simple et formelle des conditions sur l'exécution d'un processus, toutes les informations additionnelles sont traitées sous forme de commentaires. IDEF3 ne permet pas, par exemple, de gérer les ressources et les flux de matières [31] [20].

De nouvelles méthodes ont rejoint la famille IDEF, les plus connues sont IDEF4 qui est une méthodologie orientée objet et IDEF5 qui est une méthodologie pour le développement des ontologies. La liste des méthodes IDEF s'étend d'IDEF0 jusqu'à IDEF14 regroupant ainsi quinze méthodes différentes.

I.2.2.6 LA METHODOLOGIE GRAI

La méthodologie GRAI a été développée à l'origine par le groupe de recherche GRAI du Laboratoire d'Automatique et de Productique de l'Université Bordeaux 1 depuis le début des années 80, à partir :

- d'un ensemble de concepts de base rassemblés dans le modèle conceptuel de référence qui décrit la structure d'un système de production (ou d'une entreprise) et celle de son pilotage, de façon générique ;
- de formalismes graphiques de représentation qui décrivent les modèles ;
- d'une démarche générique qui permet de mettre en œuvre les différentes méthodes.

La méthodologie GRAI permet de :

- rechercher les améliorations de performance (détection des points à améliorer et des points forts du système étudié, restructuration, re-engineering ...)
- établir les spécifications et le cahier des charges pour développer des applications informatiques, choisir des progiciels du marché ou bien servir de base à des développements logiciels ;

- concevoir et implanter un système d'indicateurs de performance ;
- réaliser des études de benchmarking de processus ou de benchmarking d'entreprise ;
- simuler le comportement des processus ;
- gérer les connaissances (acquisition, mémorisation, réutilisation) ;
- contribuer à une démarche qualité

GRAI s'attache tout particulièrement à mettre en évidence le système décisionnel de l'entreprise. Elle définit un centre de décision comme étant le croisement d'une fonction et d'un horizon / période de décision. Les centres de décision sont positionnés au sein d'une grille GRAI. Chaque cadre de décision précise les performances attendues de cette décision (les objectifs), les éléments sur lesquels on peut jouer (variables de décision), les limites de ces variables (contraintes) et une aide au choix parmi les actions possibles (critères) [37].

I.2.3 LES LIMITATIONS DES SOLUTIONS ACTUELLES

La section suivante présente les principales solutions actuelles pour la gestion des processus d'entreprise.

I.2.3.1 LE WORKFLOW

Le Workflow [38] [39] [21] peut être défini comme « l'automatisation de processus métiers par échange de documents, informations et tâches entre acteurs pour action ». Le workflow a pour objectif la coordination automatisée de tâches réalisées par des intervenants humains. Le moteur de workflow transfère des documents entre les participants à un processus en leur assignant des tâches (valider le document, effectuer une modification ...). Cette approche pragmatique a l'avantage de l'efficacité : les concepts sont clairs, les outils relativement aisés à mettre en place. Par contre :

- Les participants au processus sont par définition des utilisateurs humains [7], on ne tient pas compte des applications du système d'information. L'intégration du workflow aux systèmes informatiques est une tâche difficile qui nécessite beaucoup de code propriétaire;
- Les documents et les tâches ne sont pas suffisants pour l'automatisation des

processus métiers. IL est nécessaire d'avoir un niveau d'abstraction supplémentaire, où l'on parle plus généralement de services et d'informations.

I.2.3.2 L'EAI - ENTERPRISE APPLICATION INTEGRATION

EAI [16] signifie "intégration des applications d'entreprise". A l'origine, les progiciels EAI avaient pour vocation la collaboration des applications d'une entreprise pour accomplir des objectifs métiers, mais dans les faits leur utilisation est beaucoup plus technique. Les fonctions de l'EAI sont :

- Connectivité : fournir les interfaces d'accès aux applications, généralement par l'utilisation de connecteurs propriétaires difficilement maintenables;
- Transformation : fournir les services de transformation de données permettant de créer un niveau d'abstraction au dessus des applications du SI - un format pivot pour représenter les données du SI (factures, bons de commande ...), et des transformations pour les mapper vers les formats propriétaires attendus par les applications;
- Routage : fournir les services permettant de localiser dynamiquement le destinataire d'un message en fonction de son contexte.
- L'EAI a eu l'avantage d'apporter une réponse au souci de réutilisabilité des entreprises, en leur permettant de capitaliser sur les applications existantes et de les faire communiquer. C'est une réponse au souci technique d'intégration mais cela ne suffit pas:
- Les fonctionnalités de gestion des processus métiers des outils d'EAI sont généralement complexes à utiliser et dissociées de l'offre technique d'intégration;
- Les processus sont définis à un niveau technique, et les décideurs n'ont aucune visibilité sur leur SI;
- Il est impossible de réconcilier cette vision avec la vision Workflow : comment faire intervenir les utilisateurs dans les processus?

I.2.3.3 LE B2BI - BUSINESS TO BUSINESS INTEGRATION

Les outils de B2Bi [16] visent à définir les processus de collaboration entre entreprises partenaires, partant du principe que les processus intra-entreprise, donc d'EAI, n'ont pas les mêmes contraintes que les processus externalisés. Le résultat est que ces outils fournissent surtout un moyen technique de surveiller une collaboration avec un partenaire : gestion de la sécurité des échanges, fiabilité du transport (le bon de commande a bien été envoyé, et reçu par le partenaire), support des protocoles EDI. Cela pose un certain nombre de problèmes:

- Ces outils doivent pouvoir collaborer avec les outils d'EAI - les processus B2B se déclinant en un processus interne par participant de la collaboration;
- Pourquoi avoir deux outils différents pour les processus internes et externalisés? Ces deux types de processus participent en réalité au même processus métier!

I.2.3.4 LES PROGICIELS INTEGRES

Les progiciels intégrés [41] sont une solution « clé en main » : des progiciels tels que ceux de SAP fournissent une solution complète de comptabilité, de facturation. Le problème qui se pose ici est celui de l'adaptabilité

- la mise en place d'un tel progiciel demande un long projet de paramétrage et d'adaptation. L'entreprise s'adapte aux processus définis par le progiciel et non l'inverse;
- pour modifier un processus, il est nécessaire d'adapter la plateforme. Cela peut s'avérer coûteux et problématique. De manière plus générale, il n'est pas aisé d'adapter un processus pour répondre à un besoin métier urgent et critique.
- les progiciels changent régulièrement de versions, en stoppant le support des versions précédentes, ce qui force la mise à jour. Le principal problème est que les outils de migration ne prennent pas en compte les adaptations du progiciel, qui sont inévitables. Il est donc nécessaire de refaire les adaptations sur la nouvelle version après migration.

I.2.3.5 LES MOTEURS DE REGLES METIERS

Les moteurs de règles métiers (Business Rules Engine) [42] permettent de modéliser les règles métiers de l'entreprise, par exemple « un bon de commande reçu de la part de tel client doit être traité par telle division du service commercial ».

Ces outils sont adéquats pour automatiser les processus métiers de l'entreprise. Ils permettent en particulier de formaliser et automatiser les prises de décisions sur la branche du processus à choisir, en fonction du contexte du processus. Par contre, ces solutions doivent obligatoirement être couplées à d'autres outils, permettant de gérer les processus.

Par ailleurs, bien que ne disposant pas de processus d'ingénierie, un formalisme de description formelle d'un processus d'affaire avec prise en compte de son environnement d'exécution a été proposé par Roger Atsa et Marcel Fouda en 2004. Ce formalisme permet de décrire toutes les composantes d'un processus d'affaire. Il s'appuie sur les types abstraits de données qui eux-mêmes découlent des types algébriques. Dans son contexte, la modélisation des processus d'affaire, pour qu'elle soit conforme à la réalité d'une entreprise nécessite de placer chaque processus dans son environnement d'exécution. Ainsi, Marcel Fouda Ndjodo, Priso Essawe Ndedi et Roger Atsa Etoundi ont proposé la représentation suivante d'un processus d'affaire :

$BP = \langle \textit{Observ}, \textit{Tasks}, \textit{follow}, \textit{goal} \rangle$ où :

- *Observ* est l'ensemble d'observateurs. Un observateur est une variable booléenne qui décrit une réalité de l'environnement réel du processus d'affaire.
- *Tasks* est l'ensemble de tâches dudit processus d'affaire
- $\textit{follow} : \textit{Tasks} \rightarrow 2^{\textit{Tasks}}$ est une fonction qui donne pour chaque tâche l'ensemble des tâches qui lui succèdent si les conditions sont remplies
- *goal* est l'observation dont on recherche la satisfaction au terme du processus.
- Une observation est, ce que l'utilisateur perçoit à la fin de l'exécution d'une tâche.

Tandis qu'une tâche elle-même sera représentée de manière formelle par :

<Name_of_task, precondition, postcondition>

- *Name_of_task* est le nom de la tâche
- *precondition* est une observation qui doit être préalablement satisfaite
- *postcondition* est une observation qui est garantie après l'exécution de la tâche

Plusieurs concepts ont été définis pour modéliser toutes les situations plausibles pendant l'exécution d'une tâche. Faute de standards, ce travail constitue donc un socle pour la modélisation des processus d'affaire adaptables, dans la limite du possible, au changement dans une organisation.

I.3. CONCLUSION

Cette présentation des principales solutions actuelles pour la gestion des processus d'entreprise montre, d'un côté, l'inexistence d'une offre globale d'outils adaptés aux entreprises. On remarque également le manque de standards, ou de normes, régissant la définition de ces outils pour la gestion des processus d'entreprise. Toutes ces méthodes sont semi-formelles dans la mesure où l'usage des graphiques est au centre de ces différentes approches. D'un autre côté, l'offre actuelle est guidée par les fournisseurs et non par le besoin des entreprises. La modélisation des organisations, à partir de leurs besoins est au centre d'une autre préoccupation du génie logiciel : l'ingénierie des besoins. En effet, ce pan scientifique est en amont de l'activité ci-dessus décrite. Elle permet la représentation des organisations à partir de leurs attentes. De cette manière, il devient plus aisé de gérer le changement dans ces organisations. La partie suivante lui sera consacrée.

II. INGENIERIE DES BESOINS POUR LES PROCESSUS D'AFFAIRES

Cette partie est consacrée à un tour d'horizon des techniques qui ont cours au niveau de l'ingénierie des Besoins. Après avoir fait un bref aperçu de ces méthodes, nous nous sommes attardés sur celles-ci qui peuvent être utilisées dans l'ingénierie des besoins d'un processus d'affaire.

II.1 INTRODUCTION

Un grand nombre d'études Lubars, M., Potts, C., Richer, C en 1993, Karen Mc Graw, Karan Harbison en 1997, The Standish Group en 1994, a montré que les échecs dans la mise en œuvre et l'utilisation des systèmes informatiques sont dus à une mauvaise compréhension des besoins auxquels ces systèmes tentent de répondre. Un exemple édifiant est celui de la refonte du système d'information du SAM londonien (gestion des ambulances d'urgence) qui, à cause d'une mauvaise compréhension des besoins, a abouti à plusieurs décès. Les efforts requis pour corriger les erreurs découlant de cette mauvaise compréhension des besoins sont très importants. Afin de corriger cette situation, il est nécessaire de définir des méthodes, des techniques et des outils pour élucider, valider et représenter de manière adéquate et structurée les besoins relatifs aux systèmes à développer. Ce travail devrait permettre, dans le futur, de développer des systèmes plus adaptés. C'est l'objectif que s'est fixé l'ingénierie des besoins.

Selon [43], «l'ingénierie des besoins est un domaine large, interdisciplinaire et ouvert, il concerne la traduction d'observations informelles du monde réel vers des langages mathématiques de spécifications ». Le développement d'un système d'information débute généralement par la construction d'une représentation abstraite des données et des traitements en suivant une approche de modélisation conceptuelle ou sémantique [44, 45, 46]. La modélisation conceptuelle permet de spécifier ce que le système devrait faire, ses fonctionnalités [46]. Elle rend compte de la sémantique de l'information à produire en élaborant une représentation

abstraite du système à construire [44].

II.2 – CONCEPTS DE BASE

De nombreuses études [47, 48, 49, 50] ont montré que les échecs dans la mise en œuvre et l'utilisation des systèmes d'informations sont principalement dûs à une mauvaise prise en compte des utilisateurs et de leurs besoins lors de la modélisation de ces systèmes. Ainsi, par exemple, l'étude réalisée par Standish Group en 1994 a montré que sur 8000 projets menés par 350 compagnies, seulement 16% des projets avaient été considérés comme «réussis» par leurs clients [48]. Des résultats similaires ont émergé d'un groupe de prospective animé par Jean Caelen de 1996 à 1998, pour étudier les émergences des nouvelles technologies de l'information et de la communication (TIC) et la genèse des produits/services dans ce secteur afin d'en tirer des enseignements [51]. Dans les projets industriels et publics étudiés, [51] fait de nombreux constats, notamment que « l'acteur instrumenté est rarement vu comme un sujet agissant, mais comme un simple utilisateur : on ne se pose pas bien souvent la question Qui, pour qui? Mais plutôt Quoi, comment ? Par exemple, on reste guidé par la tâche elle-même, plus que par la relation sujet/tâche/contexte (action située). On ne fait pas bien souvent la différence entre tâche et activité, et les systèmes sont trop souvent centrés "tâche" ils s'adressent donc plutôt à des experts ». Afin de remédier à cette situation, l'ingénierie des besoins (Requirement Engineering en anglais) propose des méthodes, des techniques et des outils pour expliciter, valider et représenter de manière adéquate et structurée les besoins relatifs aux systèmes à développer. Selon [43], le moyen le plus répandu de combler le fossé existant entre les besoins et les spécifications fonctionnelles d'un système informatique est l'affinement des besoins (« refinement of requirements» en anglais). L'affinement des besoins concerne l'identification des aspects d'un besoin qui ne peuvent être garantis ou effectués par un ordinateur seul. Ceux-ci sont alors affinés ou remplacés jusqu'à ce qu'ils soient pleinement implémentables. La connaissance du domaine d'application (« domain knowledge» en anglais) est selon ces auteurs le moyen de réduire ce fossé et de convertir les besoins en spécifications.

Dans un article de synthèse, [52] présente 25 ans de recherche sur l'ingénierie

des besoins. «Requirements engineering (RE) is concerned with producing a set of specifications for software systems that satisfy their stakeholders and can be implemented, deployed and maintained». Il explique ainsi que l'objectif premier de cette «discipline» est de produire des spécifications pour les systèmes logiciels qui satisfassent leurs utilisateurs et puissent être implémentés, déployés et maintenus. Une première tentative de classification des besoins a distingué les besoins «vitaux» (vital en anglais) des besoins «souhaitables» (désirable en anglais). Les besoins «vitaux» doivent être totalement satisfaits par le système alors que les besoins «souhaitables» peuvent le cas échéant être sacrifiés pour des raisons de délais, de coûts, etc. Afin de mieux représenter la réalité du système à construire et son intégration dans l'organisation et l'environnement, deux types de besoins sont également distingués : les besoins fonctionnels et les besoins non fonctionnels [53, 54].

Le besoin fonctionnel (FR pour functional requirement) décrit ce que le système doit faire. Il capture le plus grand nombre de scénarios d'usage possibles pour répondre aux besoins exprimés des utilisateurs. Des modèles opérationnels comme les cas d'utilisation (en anglais Use case), les diagrammes d'état sont utilisés pour les représenter. Cette analyse des besoins fonctionnels débouche sur la définition des spécifications fonctionnelles.

Le besoin non-fonctionnel (NFR pour not functional requirement) ou qualitatif décrit sous quelle contrainte le système doit le satisfaire. Parmi les besoins fonctionnels identifiés, les besoins non-fonctionnels permettent de comparer des solutions alternatives ou de définir des contraintes liées à la qualité, la sécurité, la performance, le coût. Le standard IEEE 830 de 1998 « IEEE recommended practice for software requirements specifications » définit les bonnes pratiques en termes de définition des spécifications logicielles. Il distingue outre les besoins fonctionnels, les besoins liés à la performance, les contraintes de conception (design constraints), les contraintes externes (external interface) liées aux logiciels, matériels, standards ou personnes et enfin les attributs de qualité permettant de définir la qualité à atteindre.

Aujourd'hui les organisations doivent s'adapter rapidement pour faire face à la

concurrence et à la pression économique, leur système d'information doit pouvoir également s'adapter. Les besoins identifiés ne sont plus des éléments stables, ils subissent des changements au cours du développement du système puis lors de son utilisation lorsque des changements organisationnels surviennent. [46] proposent que les modèles de besoins se réfèrent aux besoins organisationnels plutôt qu'aux fonctionnalités du système afin que les systèmes informatiques soient capables de s'adapter aux changements organisationnels. Ceci correspond à une approche orientée par les buts que nous allons maintenant présenter.

II.3 APPROCHES D'INGÉNIERIE DES BESOINS

II.3.1 - INGENIERIE DES BESOINS ORIENTEE PAR LES BUTS

Afin de résoudre le problème de l'inadéquation des systèmes vis-à-vis des besoins des utilisateurs mal définis ou évolutifs, un courant de recherche [55, 56, 57, 58] dépasse la séparation entre les besoins fonctionnels et non-fonctionnels, et introduit les modèles de buts dans la modélisation des besoins : l'ingénierie des besoins orientée par les buts ou (GORE pour Goal-Oriented Requirements Engineering en anglais). Les modèles de buts prennent leur source dans les méthodes de résolution de problèmes de l'intelligence artificielle, en particulier celles qui sont basées sur les buts comme «Means-Ends Analysis» de Newell et Simon qui définit un état courant et un état but («goal state» en anglais) et des actions qui permettent de réduire la distance entre l'état courant et l'état but. La résolution de problèmes basée sur les buts définit la solution à un problème comme la recherche d'une séquence d'actions qui mène au but désiré. D'autres méthodes de résolution de problèmes utilisent des structures de graphes ET/OU et permettent d'opérationnaliser les modèles de buts [51] à travers notamment des systèmes multi-agents [51]. Reprenant cette vision à base d'agents, [59] définit les buts comme des déclarations prescriptives d'intention, dont la satisfaction exige la coopération d'agents (ou de composants actifs) dans le système et son environnement : «Goals are prescriptive statements of intent whose satisfaction requires the cooperation of agents (or active components) in the software and its environment ». Selon [60], un but est une déclaration «optative» qui exprime ce que l'on veut, un état ou un résultat que l'on cherche à atteindre.

Enfin, selon les auteurs du formalisme MAP [61], un but peut être vu comme l'expression d'une intention, d'un souhait de réaliser un processus organisationnel. Farida Semmak et Joël Brunet ont défini un but comme étant « Un but (fonctionnel) définit un besoin/attente potentiel(le) que le système peut satisfaire, il exprime ce que l'utilisateur du système souhaiterait faire » [3].

On peut distinguer différents niveaux d'abstraction pour les buts : les buts de haut niveau (stratégiques, globaux, couvrant l'organisation) et les buts de bas niveau (opérationnels, locaux, spécifiques au design).

[46, 44] proposent de construire des systèmes «purposeful », qui «remplissent un but, un objectif dans l'organisation. La compréhension et l'explicitation des buts sont une condition nécessaire pour la conceptualisation de ces systèmes «purposeful ». L'idée est alors d'étendre la question habituelle : «que fait le système ?» à «pourquoi le système est tel qu'il est ? ». Cette question du pourquoi est abordée en termes d'objectifs organisationnels et de leur impact sur le système d'information. Le recentrage sur le pourquoi devrait permettre de produire des systèmes plus acceptables dans le futur. [62] note «qu'il est également important d'établir que le «pourquoi» du système à développer corresponde au «quoi» du système développé. L'objectif est d'assurer une meilleure adéquation (« a best fit» en anglais) entre les besoins de l'organisation (le «pourquoi») et les fonctionnalités du système (le «quoi »)».

L'ingénierie des besoins orientée par les buts s'attache donc à produire des modèles de buts répondant aux questions «Pourquoi» et «Comment» plutôt que «Quoi» doit être implémenté. La question «Pourquoi» (la justification ou «rational» en anglais) est la question centrale de cette approche [52]. L'auteur ajoute «qu'au lieu de se demander ce que le système doit faire, les méthodes orientées buts demandent » pourquoi.

Elles permettent également de définir les différentes alternatives de mise en œuvre et les critères de sélection parmi ces alternatives. En d'autres termes, l'ingénierie des besoins doit adresser les buts contextuels pour lesquels un logiciel est requis, les fonctionnalités que le logiciel doit accomplir pour atteindre ces buts et les contraintes définissant comment le logiciel qui accomplit ces fonctions doit être

conçu et implémenté. « On peut ainsi classer les différents aspects du Système d'Information en fonctionnel, non-fonctionnel et intentionnel [51].

Le concept de but est utilisé dans de nombreux domaines organisationnels tels que les sciences du management et le planning stratégique ; il semble bien adapté à l'expression des processus métiers des organisations en identifiant les buts et les objectifs des différentes parties prenantes et les activités qu'ils réalisent pour atteindre ces objectifs [44].

II.3.2 - Buts dans l'expression des processus métiers des organisations

Ainsi, selon [44], «le concept de but devient central dans la conception et la modélisation des processus métiers» (« business process modelling » en anglais) ; elle cite la définition de [63] : « a business process is a set of partially ordered activities aimed at reaching a goal », soit : «le processus métier comme un ensemble d'activités partiellement ordonnées qui visent à atteindre un but ». [64] cite plusieurs travaux [65, 66, 67] qui utilisent des relations basées sur les buts pour représenter les relations entre les systèmes et leur environnement dans le contexte de la recherche de solutions à des problèmes métiers et organisationnels. Ce choix est selon [64] «en partie motivé par le fait que les systèmes sont de plus en plus utilisés pour modifier fondamentalement les processus métiers plutôt que pour automatiser des pratiques établies de longue date ». Plus récemment, l'OMG a défini le Business Motivation Model ou BMM [51] qui fournit une structure pour développer, communiquer et gérer les plans stratégiques d'entreprise (en anglais «business plans») de manière organisée. Le BMM identifie plus particulièrement les facteurs qui motivent l'élaboration des «business plans», il identifie et définit les éléments des «business plans» et indique comment ces facteurs et éléments sont inter-reliés [51] avant de débiter la conception du système ou les développements techniques. De cette manière, la stratégie d'entreprise devient la base de l'activité et relie plus étroitement les solutions techniques aux intentions du métier. Le BMM définit le «business model» en termes (1) de fins (« ends » en anglais) - l'état futur que le métier souhaite atteindre sans inclure d'indications sur la manière de le faire-, (2) de moyens (« means » en anglais) - les méthodes et stratégies mises en œuvre pour

atteindre ces fins- et (3) d'évaluation des influences internes et externes (« assessment of influences » en anglais) qui permettra d'identifier l'impact potentiel qu'elles peuvent avoir sur le métier. Selon le BMM, le concept de fin est associé à trois concepts : une vision, des buts et des objectifs. La vision est une image globale de ce que l'organisation veut devenir, un état futur de l'organisation, elle concerne en général la perspective de l'organisation dans son entier et à long terme, par exemple : devenir le livreur de tomates favori de la ville. Pour satisfaire cette vision, il faut définir des buts plus spécifiques pour l'organisation (ou une sous partie de l'organisation), par exemple : livrer les tomates dans le temps prévu. Pour atteindre ces buts, il convient de définir des objectifs mesurables, atteignables et datés, par exemple pour le 1er Septembre 2009, livrer 95% des tomates à l'heure prévue.

Selon le BMM, le moyen représente n'importe quelle technique ou méthode pouvant être utilisée pour atteindre une fin. La définition d'un moyen ne nécessite pas de préciser les étapes nécessaires, mais seulement les possibilités offertes pour atteindre les fins. Les moyens peuvent être représentés sous forme de missions : une activité future de l'organisation permettant de rendre la vision réelle, par exemple : fournir de tomates aux clients à l'échelle de la ville, également sous forme de stratégies : ensemble d'actions pour atteindre un but par exemple : livrer les tomates aux clients à l'endroit de leur choix et enfin sous forme de tactiques : activités spécifiques permettant de réaliser une stratégie, par exemple : faire appel à des chauffeurs avec leurs propres véhicules pour livrer les tomates.

Ce méta-modèle du BMM montre l'intérêt que peuvent représenter pour l'ingénierie des systèmes d'information, les modèles de processus orientés buts qui permettent de décrire avec précision les besoins organisationnels et métier du système d'information à construire.

II.4 LES MODÈLES DE BESOINS ORIENTÉS BUTS

[68] distingue quatre types d'activités autour de l'ingénierie des besoins : l'explicitation (« elicitation » en anglais), la négociation, la spécification et la validation. [69] reprend ces quatre types d'activités d'ingénierie des besoins pour élaborer une classification des méthodes de modélisation orientée par les buts, le Tableau 2-1 ci-dessous reprend cette classification.

Tableau 2.1 : Le rôle de l'analyse des buts dans les activités RE [69]

RE Activity	Goal Analysis Contribution	Goal-Oriented Approach
- requirements elicitation	1. understanding the current organisational situation	GOMS, Goal-based Workflow, i*,EKD
	2. understanding the need for change	ISAC, F3
-Requirements negotiation	3. providing the deliberation context of the RE process	SYBIL, REMAP, The reasoning loop model
- requirements specification	4. relating business goals to functional and non-functional system components	KAOS, GBRAM, NFR Framework, Goal-scenario coupling Framework
-requirements validation	5. validating system specifications against stakeholders' goals	GSN, GQM

Dans cette classification, on observe que les méthodes et frameworks les plus nombreux se situent dans les phases d'explicitation, de négociation et de spécification. La phase d'explicitation des besoins est une phase majeure et la plupart des travaux autour du GORE la considèrent comme un point de départ à la spécification des systèmes d'information. En particulier [57] définit ces activités initiales d'ingénierie des besoins (en anglais «early-phase requirements engineering activities») comme consistant à répondre aux questions suivantes : comment le système envisagé atteindra les objectifs organisationnels ? ; Pourquoi le système est-il nécessaire ? ; Quelles alternatives peuvent exister ? ; Quelles sont les implications de ces alternatives pour les différentes parties prenantes ? ; Comment leurs intérêts et leurs préoccupations pourraient être abordés ?

Nous illustrons ci-après les deux phases majeures: la phase d'explicitation des besoins à l'aide des modèles i* et MAP et la phase de spécifications à l'aide des modèles Kaos et NFR, puis nous présenterons le formalisme relevé dans [3]

II.4.1 – LE MODELE I*I

Le modèle i*i [70, 57] est un modèle qui s'intéresse principalement à la phase initiale de découverte et d'explicitation des besoins. i*i comporte deux modèles imbriqués: celui des strategic dependencies (SD) qui décrit les relations de dépendances entre acteurs dans le contexte organisationnel, et celui des strategic rationale (SR) qui est utilisé pour décrire les intérêts des parties prenantes et leurs préoccupations, et comment ils pourraient être abordés par différentes configurations des systèmes et des environnements. Le concept central de i*i est celui d'acteur intentionnel [57]. Un acteur mène des actions pour atteindre des buts (goals), accomplir des tâches organisées (plans), produire des ressources (resources). i*i distingue deux type de buts : les goals et les soft-goals qui vont concerner les buts non fonctionnels liés à la qualité, la sécurité, etc. Les actions réalisées par les acteurs peuvent être menées en relation de dépendance avec d'autres acteurs comme le montre le méta-modèle d'i*i implanté dans la méthodologie Tropos [45] présenté dans la Figure 4.

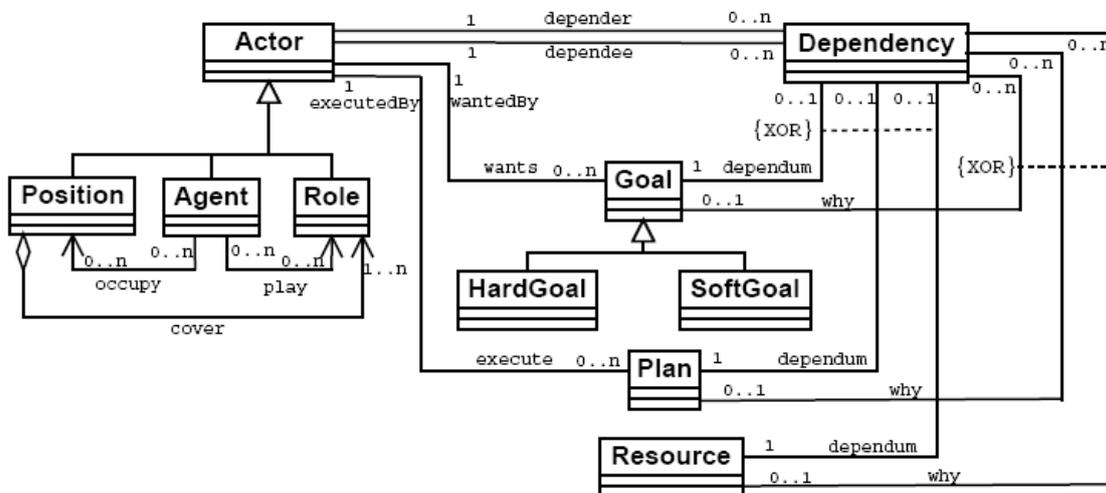


Figure 4: Extrait du méta-modèle d'i*i implanté dans Tropos

Un acteur peut être soit un agent concret, humain ou système, soit un rôle, un acteur abstrait incarnant des attentes et des responsabilités, soit encore une position, un ensemble de rôles socialement reconnus généralement joués par un agent.

Le modèle des strategic dependencies est utilisé dans la phase finale d'analyse

des besoins pour analyser les changements dans l'organisation dûs à l'introduction du système futur. La Figure 5 présente un exemple de modèle SD appliqué à un système de gestion de planning de réunion informatisé (les cercles indiquant ici les acteurs).

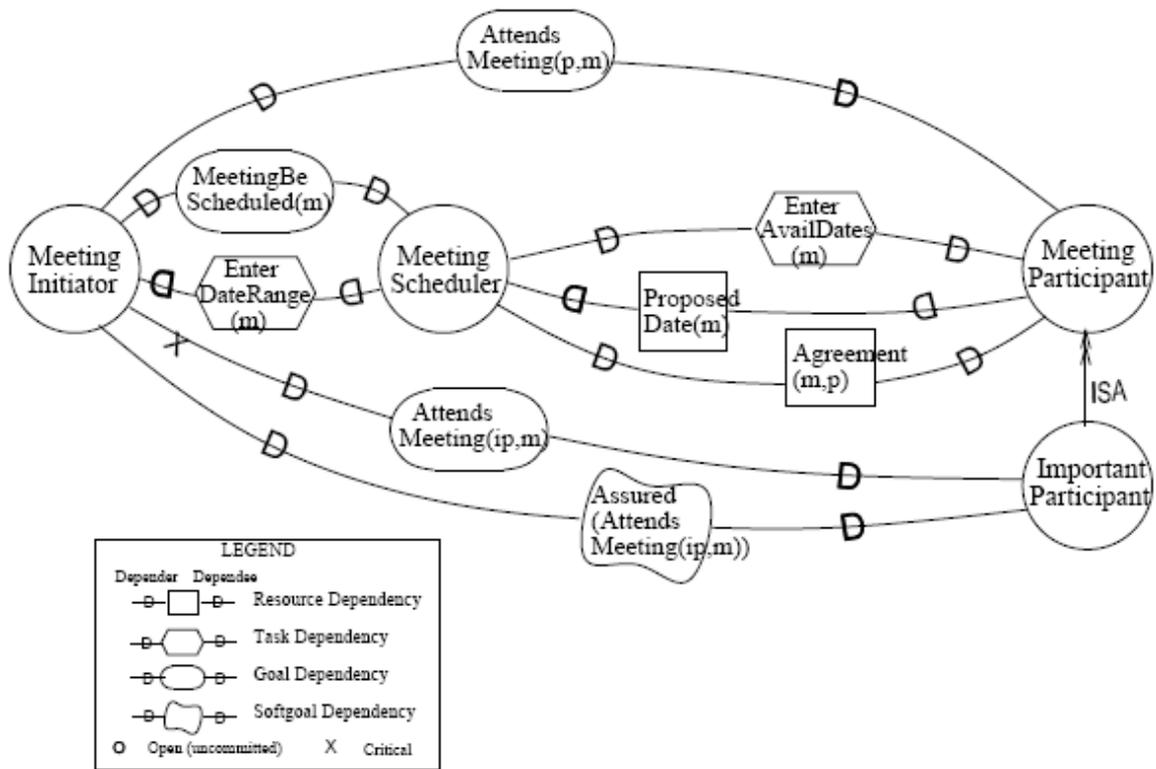


Figure 5: Modèle SD appliqué à un système de gestion de planning de réunions [57]

Le modèle des strategic rationale permet de définir de manière très précise les besoins de chaque acteur et comment les atteindre, il permet également à l'analyste d'évaluer des alternatives possibles dans la définition des processus afin de mieux prendre en compte les préoccupations des acteurs. Le modèle SR utilise deux types de liens : d'une part, les liens de décomposition qui fournissent sans une description hiérarchique des éléments intentionnels décomposés en sous-éléments et d'autre part, les liens moyens←fins (means-ends) ou intention←tâche qui permettent de savoir pourquoi un acteur s'engage dans telle tâche, poursuit tel but, a besoin de telle ressource ou souhaite tel « soft-goal ». Le modèle SR permet d'avoir une vision plus détaillée des tâches réalisées par les différents acteurs humains ou par le système. Ainsi les tâches qui incombent dans l'exemple

précédent au MeetingScheduler sont détaillées dans la Figure 3-10 présentant le modèle SR.

On trouve dans la Figure 6, une tâche principale : ScheduleMeeting qui consiste à organiser la réunion décomposée en trois tâches : ObtainAvailDates qui consiste à obtenir des dates disponibles auprès des participants, MergeAvailDates qui consiste à fusionner les différentes possibilités, ObtainAgreement qui consiste à obtenir l'accord de chaque participant sur la date proposée.

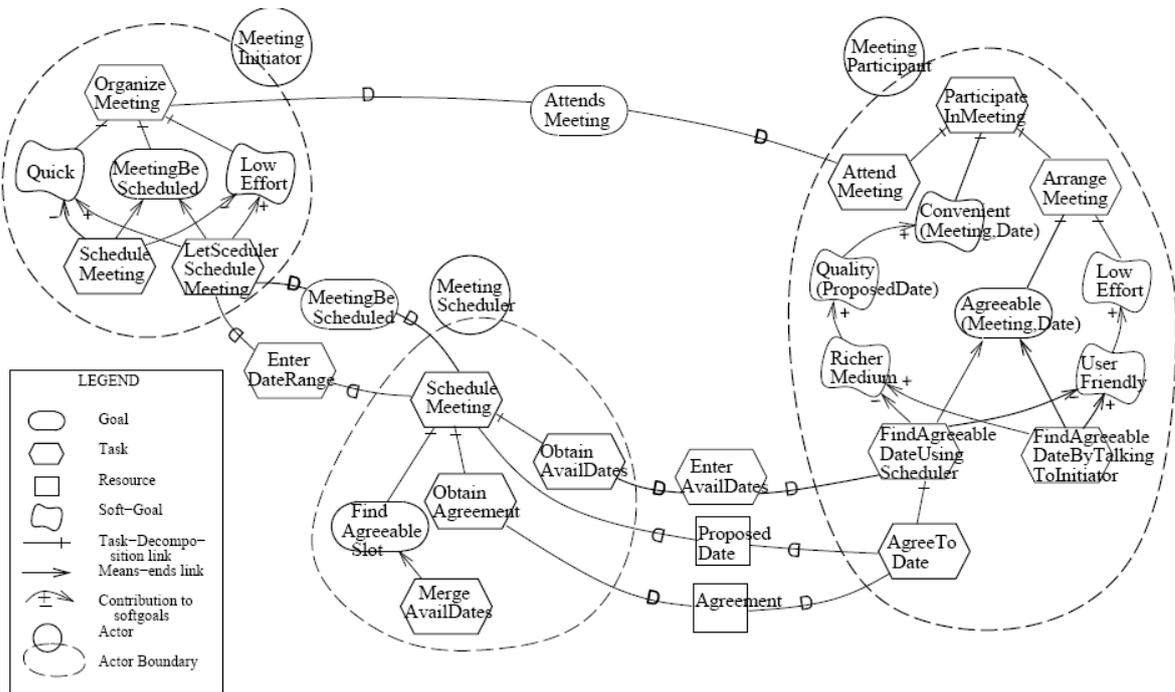


Figure 6: Modèle SR appliqué à un système de gestion de planning de réunions [57]

i*i est supporté par un environnement logiciel de modélisation que la méthodologie Tropos utilise pour représenter et raisonner sur les besoins et les choix de configuration. Tropos est associé à Formal Tropos [71], un langage formel de spécification qui offre en plus des concepts de i*i la possibilité d'ajouter des éléments temporels, des contraintes, des invariants, des pré et post-conditions. Ces modèles peuvent être validés par des techniques de validation formelle model-checking étendues à i*i et Formal Tropos par [71].

i*i combiné à Tropos et Formal Tropos, peut être utilisé pour couvrir les niveaux 1 et 3 du Tableau 3-3 de l'extraction des besoins à la validation formelle des

spécifications. L'utilisation des deux modèles (SD, SR) permet de décrire précisément d'une part les dépendances entre acteurs et d'autre part les besoins de chaque acteur et comment les atteindre.

II.4.2 - LE MODELE KAOS

Le modèle KAOS (Knowledge Acquisition in autOmedated Specification) [72, 52] consiste en un Framework formel basé sur des techniques de logique temporelle, d'intelligence artificielle et d'analyse formelle. KAOS combine différents niveaux d'expression et de raisonnement: semi-formel pour modéliser et structurer les buts, qualitatif pour sélectionner parmi les alternatives et formel pour des raisonnements plus précis. KAOS est constitué de plusieurs méta-modèles (cf. Figure 7) dont les principaux sont:

- Le goal model où les buts sont représentés et assignés à des agents;
- L'object model, modèle UML qui peut être dérivé des spécifications formelles des buts et se réfère à des objets et leurs propriétés;
- L'opération model qui définit les différents services à fournir aux agents logiciels;
- Le responsibility model qui définit, pour chaque attente (expectation) et chaque besoin (requirement), un agent responsable.

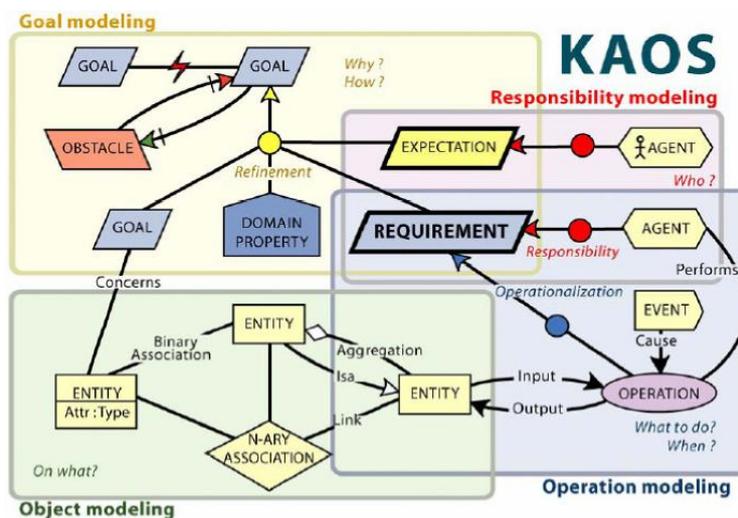


Figure 7: Extrait du méta-modèle de KAOS [KAOS Objectiver]

Dans le modèle goal model de KAOS, les buts sont représentés selon un arbre hiérarchique ET/OU [52]. Le raffinement d'un but s'achève quand chaque sous-but est réalisable par un agent qui lui a été assigné. La Figure 8 présente le méta-modèle détaillé du goal model de KAOS, on y retrouve les concepts de buts et de raffinement de buts, les états des buts (Achieve, Maintain, Avoid, SofiGoal) et les liens avec les propriétés du domaine (DomProp).

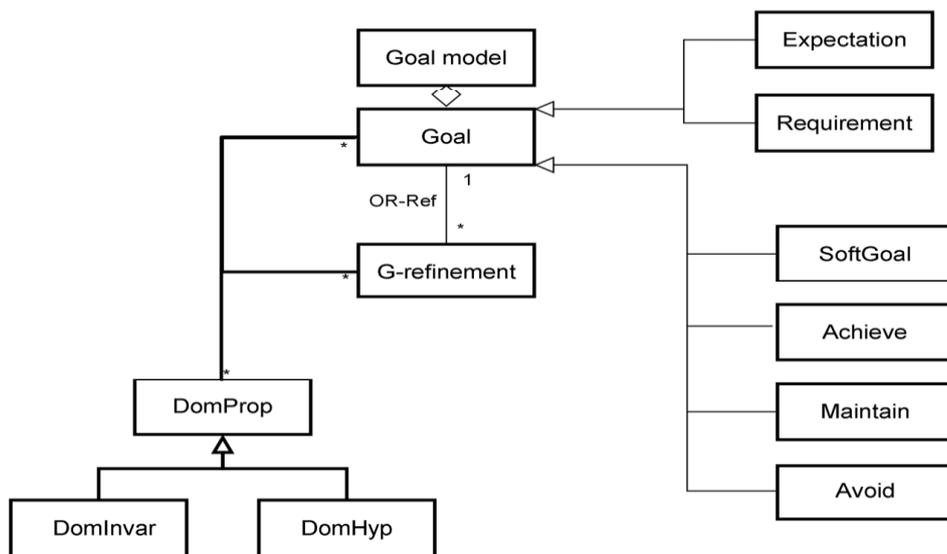


Figure 8: Méta-modèle du goal-model de KAOS

Dans l'exemple illustré dans la Figure 9, le domaine concerne le système de gestion des ambulances de Londres, afin d'allouer une ambulance sur appel téléphonique. Le goal model montre les différents buts à atteindre et les agents responsables, par exemple le but `Maintain[AmbulanceOnNormalRoute]` incombe à l'agent `AmbulanceStaff`. Sur la figure CAD représente un agent logiciel nommé `Computer Aided Despatch`, et `RA` représente l'agent `Resource Allocator` qui alloue les ressources demandées.

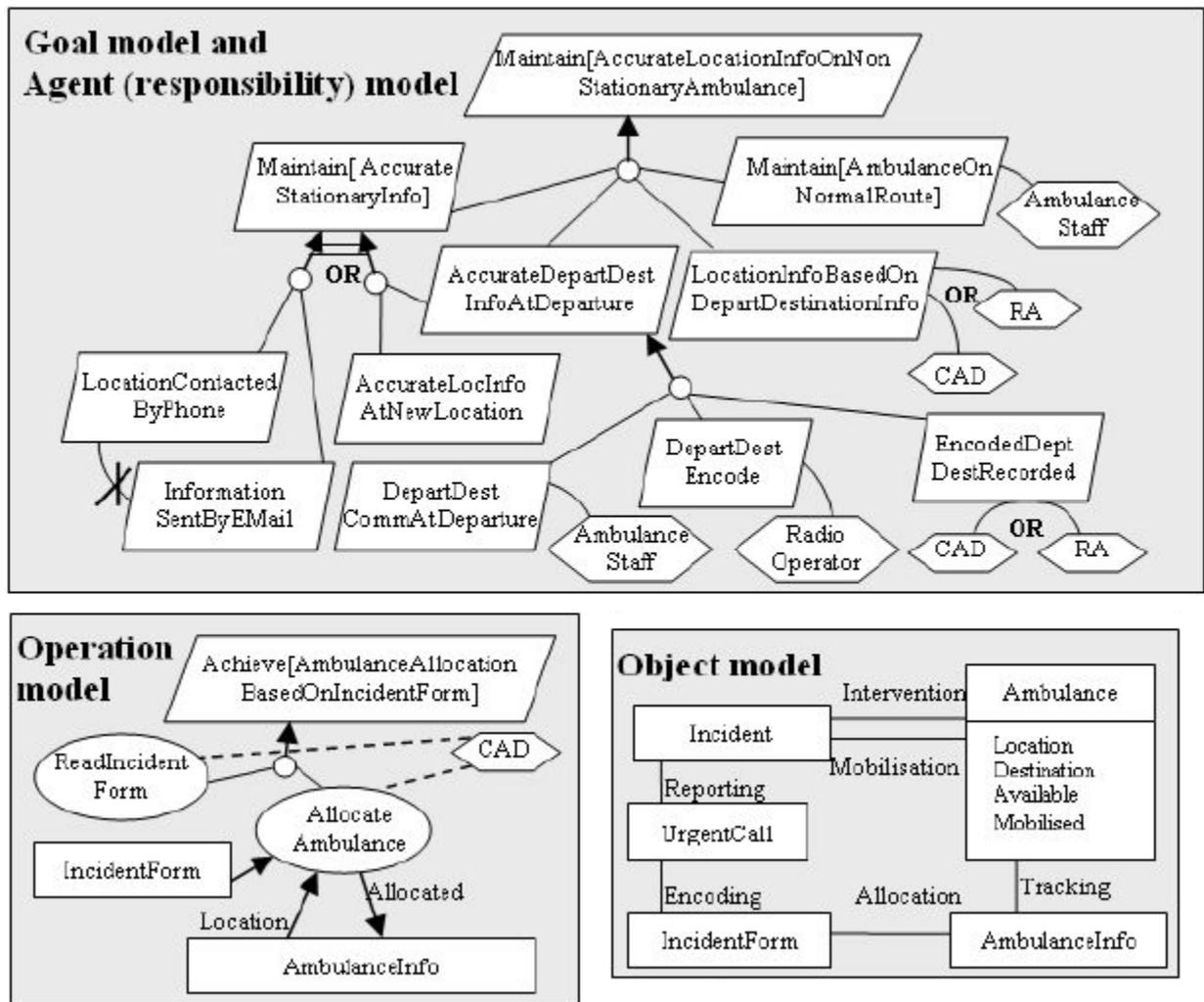


Figure 9: Fragment du modèle KAOS pour le London Ambulance Service system [73]

Un but dans KAOS est spécifié de la façon suivante [73]:

Goal Achieve [IncidentResolved]

InformalDef Every incident requiring emergency service is eventually resolved.

L'accent principal de KAOS est mis sur la preuve formelle que les exigences sont en adéquation avec les objectifs qui ont été définis pour le système envisagé [59]. L'environnement Objectiver [51] permet de concevoir et de valider les modèles KAOS. KAOS permet de formaliser de façon précise ce qu'est une intention sous la forme d'un Agent responsable d'un besoin qui est réalisé par une Opération sur une Entité.

II.4.3 - LE MODELE BESOINS NON FONCTIONNELS

Le domaine du "Non-Functional Requirements" [45] a pour objectif de modéliser et analyser les besoins non-fonctionnels, et est utilisé pour la phase de spécifications. Il repose sur la notion de soft-goals descriptibles de manière imprécise ou subjective, basé sur le concept qu'un but n'est jamais totalement atteint ou non atteint [45]. Comme exemple de soft-goals, on peut citer la rapidité, la convivialité, la sécurité, la confiance, le bien-être du patient dans le contexte d'un Système d'Informations hospitalier. Des catégories ont été définies par [74] pour évaluer de manière qualitative l'atteinte d'un but. L'idée principale est de systématiquement modéliser et affiner les besoins non-fonctionnels et d'exposer les influences négatives ou positives des différentes alternatives sur ces besoins.

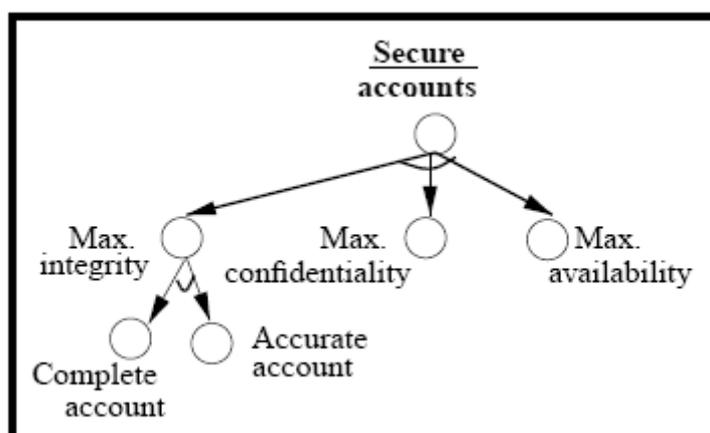


Figure 10: Exemple NFR Sécurité des comptes (ICSQ94)

L'environnement de modélisation NFR permet de raffiner les buts abstraits au moyen d'assistants qui fournissent un catalogue d'alternatives fréquemment utilisées pour aider l'analyste à préciser le but abstrait en définissant des sous-but plus spécifiques. Dans l'exemple reproduit Figure 10, pour le besoin NFR secure accounts, le système a proposé trois alternatives : en maximisant l'intégrité, la confidentialité ou la disponibilité ; pour chacune des alternatives, des buts plus précis peuvent être proposés par le système ou par l'analyste.

II.4.4 - LE MODELE MAP

MAP [61, 58] est un modèle qui peut-être utilisé comme *i** pour les phases d'explicitation. MAP propose ainsi le concept de «carte» pour décrire un modèle de processus : «une carte est un graphe orienté dont les nœuds représentent les buts et dont les arcs reliant les nœuds représentent les stratégies comme l'exprime son méta-modèle présenté Figure 11 et contrairement aux autres modèles présentés. Le modèle MAP précise explicitement et exhaustivement l'expression des intentions (ou buts) et les différentes manières de les réaliser (les stratégies).

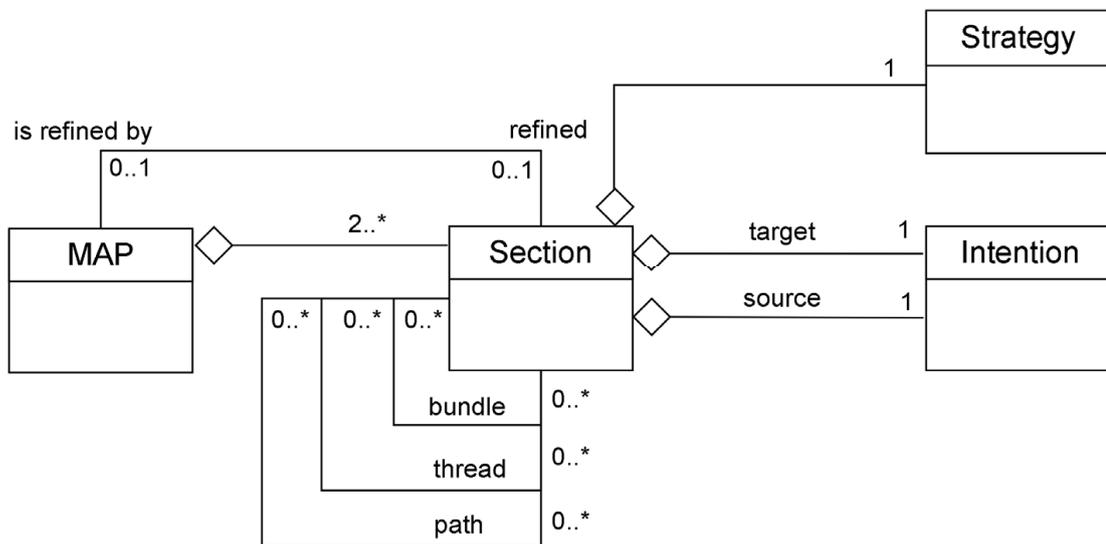


Figure 11: Extrait du méta-modèle MAP d'après [44]

On remarque sur ce méta-modèle qu'une section représente une portion de carte comprenant une intention source, une intention cible, et une stratégie permettant d'atteindre l'intention cible depuis l'intention source. Le formalisme MAP se présente sous la forme d'« un graphe dirigé et labellé dont les nœuds sont des intentions (ou buts G) à atteindre et les arcs sont les stratégies pour les atteindre. Comme il y a plusieurs stratégies pour atteindre une intention, le graphe offre plusieurs chemins du nœud de début (START) au nœud de terminaison du processus (STOP). Le formalisme IvL4P permet ainsi de représenter plusieurs processus, [...] il est donc multi-démarches» [51]. La Figure 12 représente un exemple de processus métier de livraison de produits modélisé avec le formalisme MAP.

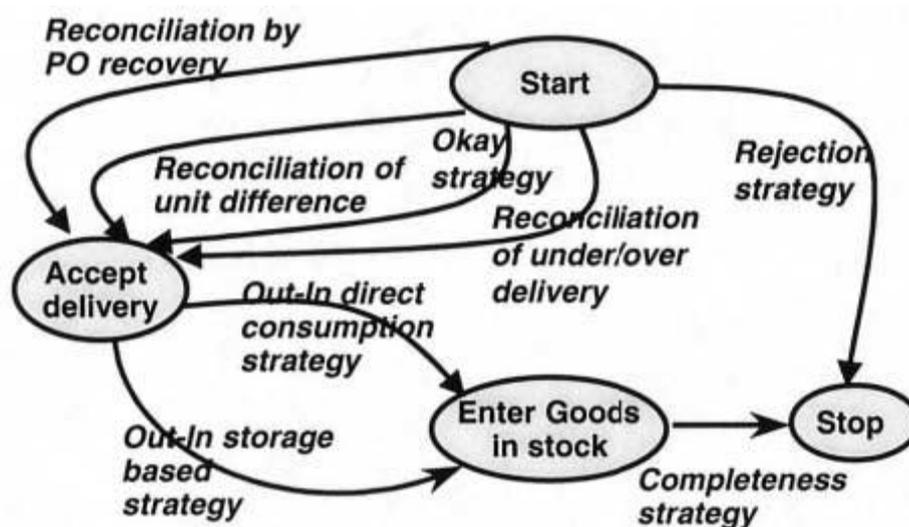


Figure 12: Exemple de carte MAP [58]

«Une carte contient une collection d'intentions reliées entre elles par des stratégies de réalisation de ces intentions lors du déroulement du processus. Une stratégie peut être une action exécutable qui aboutit à des transformations du produit ou être elle-même un processus modélisé par une sous-carte. Les cartes de processus ne contraignent pas l'utilisateur dans une démarche constituée d'étapes successives, mais permettent au contraire, un grand degré de liberté dans l'ordonnement de réalisation des intentions et dans le choix des techniques qu'il désire appliquer».

II.4.5 LE FORMALISME DE FARIDA ET JOËL BRUNET

Outre ces démarches semi-formelles, un formalisme de représentation des besoins d'un domaine se distingue : le méta-modèle orienté buts pour la représentation des besoins d'un domaine. Il est l'œuvre de Farida Semmak et Joël Brunet [3]. Il décrit un besoin comme étant un triplet <but, règle, fragment conceptuel>. Dans cette représentation, un but fonctionnel définit un besoin/attente potentiel(le) que le système peut satisfaire, il exprime ce que l'utilisateur du système souhaiterait faire ; une règle de gestion définit une loi du domaine à laquelle le but doit se conformer. Elle aide à organiser un processus de gestion pour atteindre le but. Ce travail s'inscrit dans l'activité de l'ingénierie pour la réutilisation. En effet, dans le cadre de la réutilisation, la problématique tourne autour de deux activités majeures : l'ingénierie pour la réutilisation et l'ingénierie par la réutilisation. La première activité vise à identifier, représenter et organiser la connaissance tandis que la seconde activité a

pour objectif la construction de systèmes par la réutilisation de ladite connaissance. Il s'agit donc d'une approche qui vise la réutilisation de connaissances relatives à une classe de systèmes d'un domaine spécifique.

Un domaine dans le contexte de [3] est vu comme étant l'ensemble des connaissances relatives à un champ d'application. Usuellement les connaissances à un domaine se définissent par les objets du domaine et les règles de comportement. Toutefois, il est reconnu qu'un domaine est également défini comme étant une classe de problèmes pour lesquels des solutions existent. Dans leur vision, les problèmes sont vus comme étant des besoins exprimés sous la forme d'objectifs qui doivent être satisfaits par le futur système. Ils mettent ainsi un accent sur l'objectif à atteindre et non sur les objets du système. Pour élaborer leurs modèles, Farida Semmak et Joël brunet dans [3] se sont appuyés sur deux principes permettant de déterminer les concepts du méta-modèle : le principe d'abstraction et le principe de variabilité. Le principe d'abstraction a pour objectif de mettre en évidence les propriétés partagées par une classe de systèmes et d'ignorer leurs spécificités. Ces propriétés peuvent être des objectifs communs, objets communs, et d'opérations communes à plusieurs systèmes du domaine. Le principe de variabilité a pour objectif de considérer les propriétés discriminantes des systèmes d'un domaine. Cette variabilité peut s'exprimer soit sous forme d'une hiérarchie d'objets, soit sous forme de décisions que prennent les concepteurs qui développent les systèmes.

- a) Le principe d'abstraction : l'objectif de représenter un ensemble de fragments de connaissances relatifs à une classe de systèmes d'un domaine. Pour ce faire un fragment de connaissances est représenté sous la forme d'un triplet <but, règle de gestion, fragment conceptuel> tel que défini précédemment.

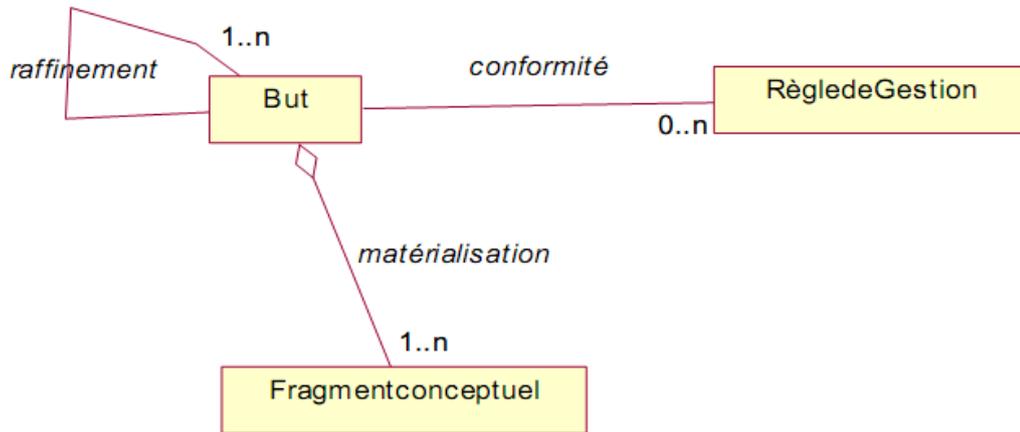


Figure 13: Le modèle général d'un fragment de connaissance [3]

Les connaissances sont décrites par des fragments de connaissances formant une arborescence dont les nœuds feuille représentent les buts auxquels sont associés des fragments conceptuels et les nœuds de plus haut niveau représentent des expressions de buts. Chaque nœud répond à un but formulé à un niveau plus ou moins d'abstrait et à chaque nœud but sont associés des règles régissant la réalisation du but. Chaque but se matérialise par un fragment conceptuel représenté par une dépendance comportementale entre les objets du domaine.

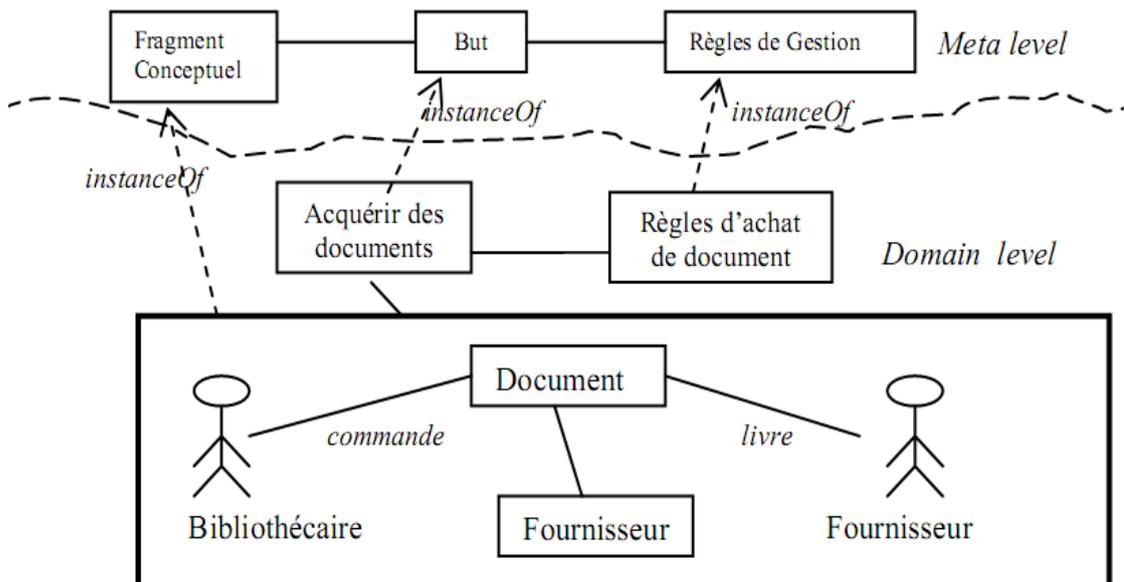


Figure 14: Un exemple de fragment de connaissance [3]

Dans le cadre de leurs travaux, les auteurs se sont exclusivement intéressés

au but fonctionnel.

- i) Un but fonctionnel définit un besoin potentiel que les systèmes doivent satisfaire, il exprime ce que l'utilisateur d'un système souhaite faire. Il a un nom et est décrit par un verbe et au moins un paramètre. Le verbe décrit l'action que l'utilisateur souhaite faire alors que les paramètres sont les entités du domaine contribuant à la réalisation du but. Le premier paramètre, obligatoire, est le type d'objet représentant une entité du domaine contribuant à la réalisation dudit but. Le second paramètre, optionnel, est de type moyen. Ce type définit la façon dont le but peut-être satisfait. Un but fonctionnel peut être soit abstrait, soit opérationnel. Un but abstrait est un but constitué de plusieurs buts qui peuvent être abstraits ou opérationnels. Un but opérationnel est un but atomique qui se matérialise dans un fragment conceptuel. Le fragment conceptuel associé à un but abstrait est obtenu par dérivation des fragments conceptuels associés aux buts opérationnels. il est donc dérivé de l'ensemble des fragments associés à ses buts opérationnels. le but racine correspond au besoin établi au plus haut niveau.
- ii) Les règles de gestion : dans les travaux relatés dans [3], une règle de gestion exprime une règle du domaine ; c'est une loi du domaine à appliquer et à laquelle la réalisation du but doit se conformer. Elle contribue à l'organisation du processus pour atteindre un but. Les règles de gestion impactent tant sur l'environnement interne qu'externe de l'organisation. Il existe trois types de règles: le type règle d'ordonnement, le type règle de déclenchement, ou le type contrainte statique. Une règle de type règle d'ordonnement exprime l'ordre dans lequel les sous-buts d'un but abstrait doivent être réalisés. Une règle de type règle d'ordonnement est associée uniquement à un but abstrait ; une règle de type règle de déclenchement est définie par un contexte, un ensemble de condition et un processus. Une règle de type contrainte statique est une règle du domaine qui doit toujours être vérifiée pour s'assurer du bon fonctionnement du système.

iii) Le fragment conceptuel : un fragment conceptuel exprime la matérialisation du but conformément aux règles de gestions. Il exprime des dépendances de comportement entre les éléments du domaine interagissant afin d'accomplir un but. Il est défini par un agrégat composé d'objets du domaine et d'acteurs, de liaisons dynamiques et d'au plus une contrainte d'ordonnancement. Il matérialise le but.

b) Le principe de variabilité : la notion de variabilité utilisée dans ce cadre est inspirée du concept « Feature » de la méthode Foda. Cette notion est appliquée sur l'ensemble des trois composantes du fragment de connaissance.

i) La variabilité au niveau du but : l'application du principe de variabilité au niveau du but permet de distinguer plusieurs systèmes par des besoins multiples et différents qu'ils doivent prendre en compte. Les auteurs relèvent que deux caractéristiques de la variabilité existent dans leur contexte : la variabilité introduite par le mot clé « Feature » et celle introduite par le mot clé « raffinementFeature ». La variabilité introduite par « Feature », lorsque la propriété « Feature » est attachée à un but, cela traduit le caractère obligatoire ou optionnel de ce but. Tandis que, la variabilité introduite par le mot clé « raffinementFeature », ne peut être définie que sur un but père pour traduire le caractère de raffinement de ses buts fils. Elle peut prendre l'une des valeurs « Or » et « X-Or ». Lorsqu'un but père a la caractéristique « Or » cela dénote que le but père est raffiné par l'ensemble de ses fils. Il peut donc être choisi une partie de ses fils. Par contre, si le but père a la caractéristique « X-Or », cela signifie que le but père est raffiné dans un seul de ses fils.

ii) Variabilité au niveau règle de gestion : L'application du principe de la variabilité aux règles de gestion autorise une représentation flexible qui permet de les combiner en fonction des besoins réels. La notion de 'Feature' est aussi utilisée dans les règles de déclenchement et les contraintes statiques. Deux caractéristiques sont dérivées de la notion 'Feature' : la propriété 'FeatureRD' et 'FeatureCS'. La propriété 'featureRD', attachée à la condition d'une règle de déclenchement, peut prendre la

valeur 'obligatoire' ou 'optionnel'. Elle permet de construire de façon flexible les conditions d'invocation du processus qui permet l'atteinte du but. La propriété 'FeatureCS', attachée à une contrainte statique, prend également la valeur 'obligatoire' ou 'optionnel'.

iii) Variabilité au niveau fragment de conceptuel : La variabilité au niveau du fragment conceptuel se situe dans les objets et les acteurs du domaine. Les techniques de paramétrisation et de généralisation sont principalement utilisées pour différencier les systèmes. La paramétrisation est l'une des plus anciennes techniques proposées dans le domaine de la programmation pour exprimer la généralité.

Le principe de variabilité permet de considérer les éléments qui discriminent les systèmes d'un domaine. Le degré de discrimination est différent selon que l'on se situe au niveau du but, des règles ou du fragment conceptuel.

Le formalisme, ci-dessus présenté est donc un outil de choix pour modéliser, aussi bien des besoins d'une ligne de produit que ceux des systèmes qui peuvent changer dynamiquement. Les organisations sont connues comme évoluant dans le temps, il est donc judicieux d'envisager une modélisation des processus d'affaires tout au moins leurs besoins via le formalisme de [3]. Cependant, le contexte de concurrence actuel amène la plupart du temps les organisations à se réajuster à part rapport aux attentes de leur clientèle. Par conséquent, introduire des mécanismes permettant d'évaluer la qualité de services offerts et le rendement de son processus d'affaire est nécessaire dans la modélisation de l'organisation. Nous avons à ce titre choisi d'utiliser le formalisme de [3] car il peut être formellement représenté. De plus, l'accent est mis sur le but/intention et non sur les agents/acteurs de l'organisation, ce qui, à notre sens, démarque cette approche des autres. Notre objectif global étant d'opérationnaliser le formalisme de [3], nous pensons que pour un début, le formalisme de Farida Semmak et Joël Brunet de [3] est adapté. Dans la partie suivante, nous allons faire un bref aperçu de la qualité de service qui est un aspect de poids dans la modélisation des organisations à caractère commercial.

III. QUALITE DE SERVICE AU SEIN DES ORGANISATIONS

III.1 INTRODUCTION

Les parties, I et II nous ont permis de montrer comment aborder les processus d'entreprise pour les comprendre et les modéliser. Cette partie nous permet de présenter les bonnes pratiques et recommandations desquelles nous allons nous inspirer pour organiser les processus. Ces bonnes pratiques sont très souvent organisées dans des normes et standards qualité. Ce chapitre présente donc la notion de qualité en entreprise et un survol des principaux standards et normes qualité.

III.2 LA QUALITÉ

La « Qualité » peut se définir comme la capacité à atteindre les objectifs opérationnels visés. La norme ISO 9001v2000 [75] la définit comme étant l'« Aptitude d'un ensemble de caractéristiques intrinsèques à satisfaire des exigences. »

Dans la pratique [40], la qualité se décline sous deux formes :

La qualité externe, elle correspondant à la satisfaction des clients. Il s'agit de fournir un produit ou des services conformes aux attentes des clients afin de les fidéliser et ainsi améliorer la part de marché. Les bénéficiaires de la qualité externe sont les clients d'une entreprise et ses partenaires extérieurs. Ce type de démarche passe ainsi par une nécessaire écoute des clients mais doit permettre également de prendre en compte des besoins implicites, non exprimés par les bénéficiaires;

La qualité interne, elle correspondant à l'amélioration du fonctionnement interne de l'entreprise. L'objet de la qualité interne est de mettre en œuvre des moyens permettant de décrire au mieux l'organisation, de repérer et de limiter les dysfonctionnements. Les bénéficiaires de la qualité interne sont la direction et les personnels de l'entreprise. La qualité interne passe généralement par une étape d'identification et de formalisation des processus internes réalisés grâce à une démarche participative.

L'objet de la qualité est donc de fournir une offre adaptée aux clients, avec des processus maîtrisés tout en s'assurant que l'amélioration ne se traduit pas par un surcoût général, auquel cas on parle de « sur qualité ». Il est possible d'améliorer un grand nombre de dysfonctionnements à moindre coût, mais, à l'inverse, plus on souhaite approcher la perfection plus les coûts grimpent.

L'opposé de la qualité, nommé « non qualité », possède également un coût. En effet, il s'avère généralement plus coûteux de corriger les défauts ou les erreurs que de « faire bien » dès le départ. D'autre part, le coût de la non-qualité est d'autant plus important qu'elle est détectée tardivement. A titre d'illustration, réaliser à nouveau un produit défectueux coûtera au final plus du double du prix de production du produit initial s'il avait été réalisé correctement. Qui plus est, la différence de prix sera moins grande si le défaut est détecté en cours de production que s'il est détecté par le client final (insatisfaction du client, traitement de l'incident, suivi du client, frais de port...).

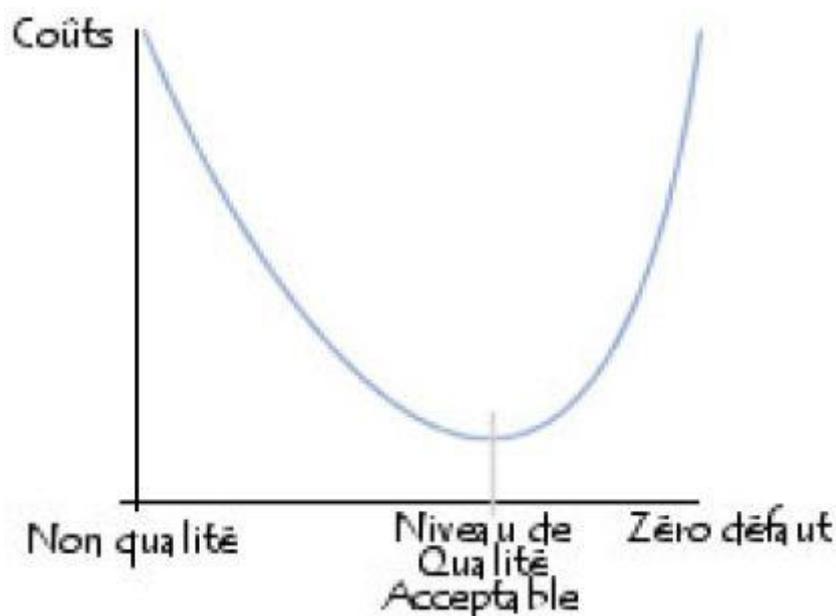


Figure 15 - La relation entre le coût et la qualité

L'amélioration de la qualité, interne et externe, permet à l'entreprise de travailler dans de meilleures conditions avec ses bénéficiaires, ce qui se traduit par une relation de confiance et des gains sur le plan financier (augmentation des bénéfices...) ou humain (clarification des rôles, des besoins et de l'offre, motivation

du personnel...).

III.3. LA QUALITE PRODUIT

Il s'agit toutefois d'un effort impliquant l'ensemble de l'entreprise et conduisant la plupart du temps à des modifications des habitudes de travail, voire des changements organisationnels. Ainsi appelle-t-on « Démarche qualité » l'approche organisationnelle permettant un progrès permanent dans la résolution des non-qualités. Il s'agit d'une démarche participative, c'est-à-dire à laquelle doit nécessairement participer l'ensemble de l'entreprise et par conséquent nécessairement portée au plus haut niveau hiérarchique.

Nous pouvons distinguer une autre typologie de la qualité:

- La qualité produit liée aux caractéristiques même du produit.
- La qualité des processus liée aux méthodes, outils et moyens utilisés pour réaliser le produit.

C'est à travers cette typologie que nous décrivons les principales normes et modèles pouvant être utilisés dans le monde de l'informatique, avec une focalisation sur les deux normes qui nous intéressent (ISO 9001v2001 et CMMI).

III.3.1 LA NORME ISO 9126, TECHNOLOGIES DE L'INFORMATION : "QUALITE DES PRODUITS LOGICIELS"

La norme ISO 9126 est un dérivé du modèle de Mc Call créé en 1977 au cours d'une étude pour l'US AIR FORCE [76]. Mc Call définit une approche de la qualité à partir de la définition de caractéristiques

- externes : facteurs de qualité;
- internes : critères de qualité;
- mesurables : métriques.

Il a donc défini 23 critères sur 11 facteurs. Chaque critère correspond à aux moins une métrique. Le modèle de Mc Call différencie utilisateur et réalisateur. La norme ISO 9126, « Technologies de l'Information: Qualité des produits logiciels », définit et décrit une série de caractéristiques qualité d'un produit logiciel (caractéristiques

internes et externes, caractéristiques à l'utilisation) qui peuvent être utilisées pour spécifier les exigences fonctionnelles et non fonctionnelles des clients et des utilisateurs. Chaque caractéristique est détaillée en sous-caractéristiques, et pour chacune d'elles, la norme propose une série de mesures à mettre en place pour évaluer la conformité du produit développé par rapport aux exigences formulées.

III.3.2 ISO 9001V2000 "SYSTEMES DE MANAGEMENT DE LA QUALITE - EXIGENCES"

III.3.2.1 PRESENTATION ISO 9001V2000

Les normes ISO 9000 ont été originellement écrites en 1987, puis elles ont été révisées en 1994 et à nouveau en 2000. La norme ISO 9001v2000 porte essentiellement sur les processus permettant de réaliser un service ou un produit alors que la norme ISO 9001 :1994 était essentiellement centrée sur le produit lui-même [75].

La présente norme internationale encourage l'adoption d'une approche processus lors du développement, de la mise en œuvre et de l'amélioration de l'efficacité d'un système de management de la qualité, afin d'accroître la satisfaction des clients par le respect de leurs exigences [77]. Lorsqu'elle est utilisée dans un Système de Management de la Qualité (SMQ), cette approche souligne l'importance de :

- 1) comprendre et de satisfaire les exigences;
- 2) considérer les processus en termes de valeur ajoutée;
- 3) mesurer la performance et l'efficacité des processus;
- 4) améliorer en permanence des processus sur la base de mesures objectives.

Le Système de Management de la Qualité (SMQ) s'appuie ainsi sur divers documents permettant de décrire, avec un niveau de précision grandissant, les activités de l'entreprise et leurs interactions

III.3.2.2 PDCA: PLAN, DO, CHECK, ACT, "LA ROUE DE DEMING"

Défini par Deming [78], PDCA est basé sur le concept de la remise en cause des méthodes de travail, des flux d'information, des fonctions, des documents de travail utilisés sur le concept de l'amélioration continue. La remise en cause comprend

l'analyse de toutes les anomalies rencontrées dans la vie d'une entreprise, vues ou pas encore vues par les clients. Ce processus constitué de 4 phases (Planifier, Faire, Vérifier, Améliorer) permet de structurer la démarche d'amélioration continue définie par ISO 9001v2000.

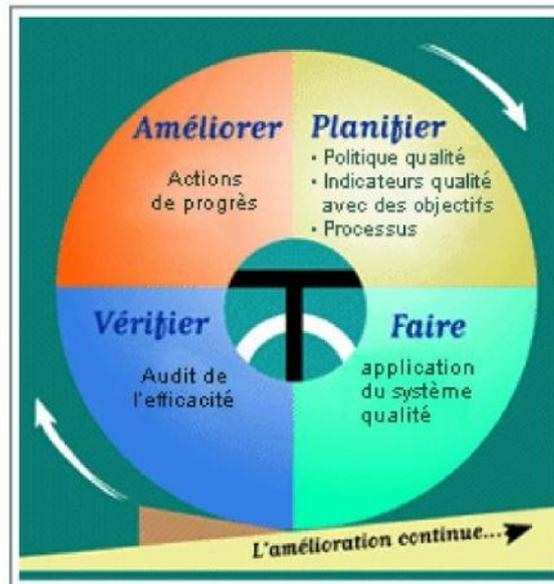


Figure 16 - PDCA : Plan, Do, Check, Act, « la roue de Deming »[7].

En effet, le cycle PDCA est une méthode qui permet d'exécuter un travail de manière efficace et rationnelle.

III.4 CONCLUSION

Tous ces modèles et normes ont leurs forces et leurs points faibles. Parmi les différents modèles, normes et corpus de connaissances présentés dans l'état de l'art, certains englobent toutes les activités d'une entreprise et adoptent une démarche descendante comme ISO 9001 d'autres comme CMMI et ITIL, sont des guides de bonnes pratiques et donc spécifiques à certaines activités adoptant une approche ascendante.

IV. CONCLUSION DU CHAPITRE

Bien qu'abordée par des axes de recherches différents, la modélisation des

processus d'affaire, l'ingénierie des besoins et la qualité de services contribuent à mettre à disposition des organisations, des systèmes informatiques fiables et capables de s'adapter à l'évolution de cette dernière. Notre travail va consister à construire un processus d'ingénierie des besoins pour les processus d'affaire qui intégrera les besoins des cadres métiers, des attentes en qualité des clients et les indicateurs de performances dont les valeurs pourront être exploitées par le staff dirigeant de ladite organisation. Le résultat de ce processus d'ingénierie est un modèle de référence d'un processus d'affaire élaboré dans le formalisme proposé par Farida Semmak et Joël Brunet [3]. Cependant, nous envisageons des ajustements afin que le modèle obtenu soit traductible, si possible, au formalisme proposé dans [1, 2] dans la mesure où nous souhaitons faire des travaux de [1, 2] une alternative aux standards de l'heure.

Le chapitre suivant sera consacré à la présentation de notre contribution à la solution de ce problème.

**CHAPITRE III : MODELE DE
PROCESSUS D'INGENIERIE
DES BESOINS POUR LES
PROCESSUS D'AFFAIRE DES
ORGANISATIONS**

Résumé :

Définir un modèle de processus d'ingénierie des besoins pour construire un modèle de référence d'une organisation est l'objectif poursuivi par cette thèse. Dans les chapitres I et II précédents, nous avons présenté les concepts généraux relatifs à notre domaine d'Etude. Dans ce chapitre, nous présentons un processus d'ingénierie des besoins permettant de partir des besoins du monde réel (attentes des cadres métiers de l'organisation et exigences de qualité des bénéficiaires des services offerts par cette organisation), pour élaborer le modèle de référence d'une organisation conformément aux travaux de Farida Semmak et Joël Brunet. Ce processus d'ingénierie des besoins comporte quatre (04) étapes fondamentales : le recensement, la sélection, l'élaboration du modèle, et la validation.

L'étape de recensement permet de collecter les besoins des cadres métiers de l'organisation et les critères de qualités des clients de cette organisation. Elle offre comme résultats l'ensemble des documents de recensement des besoins et celui des facteurs/critères de qualité.

L'étape de sélection permet de retenir ceux des besoins des cadres métiers de l'organisation qui sont consistants. Elle fourni comme résultats l'ensemble des besoins consistants de l'organisation.

L'étape d'élaboration du modèle permet de construire le modèle de référence à partir de l'ensemble des besoins consistants et des facteurs de qualité de sa clientèle. Cette élaboration en elle-même est un processus à six phases regroupées sous deux catégories : la modélisation du métier d'une organisation et la modélisation de la satisfaction des bénéficiaires des services de cette organisation. Les deux catégories constituent deux axes de préoccupations que nous avons dénommés axes de préoccupations relatives au métier pour la modélisation du métier et axe de préoccupations relatives à la satisfaction des bénéficiaires de service pour la modélisation de la satisfaction des bénéficiaires de service. La première catégorie couvre la première et la deuxième phase tandis que la deuxième catégorie, couvre la troisième, la quatrième et la cinquième phase. La première phase est réservée à la transformation des besoins retenus lors de l'étape de sélection en fragments de

connaissance. Elle vise à formaliser la règle métier d'un besoin retenu dans le modèle de description des règles métiers élaboré à cet effet. La deuxième phase est dédiée à l'élaboration du modèle métier. Ce dernier est construit à partir de ces fragments de connaissance obtenus à la phase précédente et n'intègre pas les aspects de modélisation de la satisfaction des clients de l'organisation. Le modèle métier ainsi obtenu constitue ce que nous appelons un axe de préoccupations métiers. La troisième phase est destinée à la déduction du graphe de dépendance (base de connaissance de la satisfaction) entre les tâches du processus d'affaire à partir du modèle métier produit à la phase précédente. Elle marque le début de la modélisation de la satisfaction de la clientèle. La quatrième phase vise l'adjonction des observateurs et des seuils de satisfactions aux tâches constituant le graphe de dépendance résultat de la phase précédente ou aux objets métiers d'une part, la définition des fonctions d'évaluation, par critère de qualité, associées auxdits observateurs, d'autre part. La cinquième phase vise la construction des observateurs de performances associés à chaque tâche, à partir des résultats de la phase précédente. Le résultat obtenu constitue un axe de préoccupations relatives à la modélisation de la satisfaction des bénéficiaires de service. Elle met fin à la modélisation de la satisfaction de la clientèle. La dernière phase est axée sur la jonction de ces préoccupations c'est-à-dire, l'intégration des deux axes de préoccupations. Dans cette optique, il est défini des points d'échange entre les deux axes de préoccupations. Ce n'est qu'après cette étape qu'on obtient le modèle de référence.

L'étape de validation quant à elle permet la validation du modèle ainsi obtenu. Dans cette étape, il est question de se rassurer de la cohérence des différents concepts. Pour cela, il a été défini un éventail d'axiomes à partir desquels s'appuie cette validation.

Ce travail couvre trois activités de l'ingénierie des besoins que sont : l'explicitation¹, la sélection et la validation des besoins étant entendu que la spécification desdits besoins a été définie dans la littérature. Naturellement, pour arriver au résultat escompté, des enrichissements ont été apportés à notre modèle de spécification de base proposé dans la littérature par Farida Semmak et Joël

¹ Elicitation en anglais

Brunet. Ces enrichissements concernent le fragment de connaissance. En effet, nous avons substitué le fragment conceptuel par trois composantes : «le contexte », le « niveau d'importance », les « contraintes ». Le « contexte », première composante du fragment de connaissance enrichi, décrit l'environnement d'utilisation du fragment de connaissance ; le « niveau d'importance », dernière composante du fragment de connaissance enrichi, traduit l'importance que revêt cette connaissance dans la chaîne de production du processus d'affaire ; et les « contraintes », quatrième composante du fragment de connaissance enrichi, représentent les besoins non fonctionnels associés à l'objectif visé ou but. Nous avons au niveau de la modélisation de la satisfaction des clients d'une organisation effectué un travail ontologique, les concepts de points de vue, facteurs de qualités, Observateur, Observation, Indicateurs de performances, Observateur de performance ont été définis.

I. INTRODUCTION

Dans le chapitre II, il a été présenté un ensemble de concepts généraux permettant de comprendre les concepts qui seront utilisés dans ce chapitre. Le contexte de mondialisation qui amène tous les marchés a favorisé l'augmentation de l'offre sur marché. Ainsi, pour la filière riz du marché camerounais par exemple, le riz camerounais est en compétition avec le riz thaïlandais, le riz sénégalais, le riz parfumé de France, le riz chinois. Ainsi, produire seul ne suffit plus. Pour améliorer les ventes, la nécessité de cadrer sa production aux attentes des consommateurs s'impose. Par conséquent, disposer d'un ensemble de métriques permettant de recueillir, auprès de la clientèle, le point de vue des clients par rapport à l'offre sur le marché et ajuster sa production par rapport à la catégorie de personnes qu'on aimerait atteindre devient impératif pour les organisations qui voudraient subsister. L'ingénierie des besoins est une alternative à cette préoccupation. Dans le chapitre précédent, nous avons présenté les approches d'ingénierie de besoins qui nous ont semblées pertinentes pour la modélisation des processus d'affaire. Comme la plupart d'entre elles étaient semi-formelles, nous avons jeté notre dévolu sur le formalisme suscité de F. Semmak et J. Brunet [3]. Ce formalisme a des concepts qui nous

semblent intéressants dans l'élaboration de modèles mathématiques des processus d'affaire. Cependant, ce dernier ne disposant pas de processus d'ingénierie, il est donc nécessaire de lui en adjoindre. Nous pensons qu'après la construction de ce processus, le modèle de référence, ainsi obtenu pourra être transcrit au formalisme relevé dans [1]. Définir, un tel processus pourra nécessiter de faire des aménagements dans le modèles proposé dans [3].

Nous proposons dans ce chapitre, un processus d'ingénierie permettant de construire un modèle de référence d'un processus d'affaire à partir des besoins de l'organisation et des exigences de qualités de sa clientèle. Dans la suite, nous présentons un processus d'ingénierie des besoins permettant de partir des besoins du monde réel (attentes des cadres métiers de l'organisation et exigences de qualité des bénéficiaires des services offerts par cette organisation). Pour élaborer le modèle de référence conformément au formalisme de [3]. Cette approche est basée sur quatre (04) étapes fondamentales : le recensement, la sélection, l'élaboration du modèle, et la validation ; L'étape de recensement permet de collecter les besoins des cadres métiers de l'organisation et les exigences de qualités des clients de cette organisation. L'étape de sélection permet de retenir ceux des besoins de l'organisation qui sont consistants ; L'étape d'élaboration du modèle permet dans un premier temps, de transformer les besoins de l'organisation en fragment de connaissance et les exigences de qualité de la clientèle de cette organisation en connaissance de la satisfaction de ces derniers. Dans un second temps, de construire le modèle sus évoqué, à travers un ensemble d'axiomes définis à cet effet. L'étape de validation quant à elle permet la validation du modèle construit. Dans cette étape, il sera question de se rassurer de la cohérence des différents concepts. Pour cela, il a été défini un éventail d'axiomes. Ce travail couvre trois activités de l'ingénierie des besoins que sont : l'élucidation, la sélection et la validation des besoins étant entendu que la spécification desdits besoins a été définie dans [3]. Naturellement, pour arriver au résultat escompté, des enrichissements ont été apportés au formalisme, objet des travaux dans [3]. Ils concernent le fragment de connaissance. En effet, nous avons substitué le fragment conceptuel par trois composantes : «le contexte », le « niveau d'importance », les « contraintes ». Le « contexte »,

première composante du fragment de connaissance enrichi, décrit l'environnement d'utilisation du fragment de connaissance ; le « niveau d'importance », troisième composante du fragment de connaissance enrichi, traduit l'importance que revêt cette connaissance dans la chaîne de production du processus d'affaire ; et les « contraintes », quatrième composante du fragment de connaissance enrichi, représentent les besoins non fonctionnels associés à l'objectif visé ou but.

Dans la suite, notre travail sera organisé au tour de trois points principaux. Le premier point, porte sur le rappel de quelques concepts de base nécessaires à ce processus. Le second point porte sur la présentation de la démarche. Le dernier point porte sur la conclusion.

II. RAPPELS DE QUELQUES CONCEPTS DE BASE

Notre souci, dans cette section, est de définir l'ensemble des concepts nécessaires à la compréhension de la représentation formelle d'un besoin exprimé par un utilisateur et des points de vue des bénéficiaires des services offerts par l'organisation. Pour cela, nous avons organisé cette section autour de deux points principaux. Le premier point porte sur le rappel des concepts de base inhérents à la représentation des besoins exprimés par les cadres métiers de l'organisation et à l'expression des points de vue de la clientèle par rapport au service qui leur est offert. Le second quant à lui, se rapporte à la compréhension et à la représentation des exigences de qualités issues des différents points de vue des clients de cette même organisation.

II.1 RAPPEL DES CONCEPTS DE BASE INHERENTS A LA REPRESENTATION DES BESOINS

Les concepts de base en question dans cette partie constituent des extensions de ceux proposés dans [3]. Dans la suite, l'objectif que le système doit satisfaire sera confondu au but et le but lui-même au besoin.

II.1.1 BESOIN D'UN UTILISATEUR

Nous appelons besoin d'un cadre métier d'une organisation, une activité du processus d'affaire de ladite organisation, assignée à ce cadre métier et dont il aimerait voir automatisé dans le système informatique futur [9,10]. Il traduit en quelque sorte l'objectif à satisfaire. Pour modéliser un besoin, il faut au préalable l'identifier. C'est pourquoi, nous proposons l'élaboration d'un formulaire de recensement des besoins des utilisateurs.

II.1.2 DOCUMENT DE RECENSEMENT DES BESOINS

Le document de recensement des besoins, est un formulaire à soumettre aux utilisateurs afin de recueillir de leur part, la description littérale de leurs attentes. Dans chaque formulaire, est décrite une seule attente d'un cadre métier. Le formulaire en question est structuré de la manière suivante d'après les travaux de [9,10]:

- Structure ou service : décrit la structure de l'organisation dans laquelle l'utilisateur qui exprime le besoin est affecté ;
- Cadre métier : représente le nom du cadre métier qui exprime le besoin ;
- Priorité de l'attente : la priorité d'une attente est le niveau d'importance que l'utilisateur accorde à cette attente ;
- But : l'intention d'usage de l'utilisateur ;
- Règle : représente la règle de gestion de l'intention d'usage
- Contraintes : contraintes désignent les attentes non fonctionnelles qui pourraient mettre à mal, la réalisation du but de l'utilisateur. Elles peuvent avoir trait à l'interface homme-machine, à la sécurité, etc. ;
- domaine de l'attente : le domaine de l'attente décrit le contexte dans lequel l'intention d'usage de l'utilisateur est exprimée. Ce champ est rempli par l'ingénieur logiciel ;
- Statut de l'attente : il indique l'état de l'attente. il peut prendre l'une des valeurs suivantes : proposé, rejeté, validé, prise en compte.

Le recensement des besoins des utilisateurs permet d'identifier de manière

exhaustive les attentes des utilisateurs. Les attentes recensées au niveau des utilisateurs sont par défaut à l'état proposé. Elles passent à l'état rejeté si l'attente de l'utilisateur n'a pas été approuvée par son chef hiérarchique et validé sinon. L'état pris en compte indique qu'une attente a été prise en compte dans le modèle de spécification des besoins des utilisateurs. De manière formelle, nous représenterons un besoin b d'un utilisateur par : $b:(\langle v, \chi \rangle, \langle \nu, \langle \gamma, \psi \rangle \langle \lambda, \delta \rangle \rangle, \zeta)$ où : $\langle v, \chi \rangle$ décrit la structure d'affectation désignée v et le nom χ du cadre métier ayant émis le besoin, ν le niveau d'importance de l'intention d'usage de l'utilisateur, $\langle \gamma, \psi \rangle$ décrit l'intention d'usage notée γ de ce cadre métier dans le contexte ψ , $\langle \lambda, \delta \rangle$ décrit la règle de gestion λ sous les contraintes δ , et ζ représente le statut du besoin dans le processus de pris en compte de ce dernier.

II.1.3 DOMAINE

Un domaine est le champ d'application de l'intention d'usage de l'utilisateur ou simplement le contexte dans lequel l'attente est définie ou s'applique [9].

II.1.4 CONTRAINTES

Les contraintes désignent les attentes non fonctionnelles qui pourraient mettre à mal, la réalisation du but de l'utilisateur. Elle représente l'état de l'environnement dans lequel la tâche va s'exécuter [9].

II.1.5 BUT

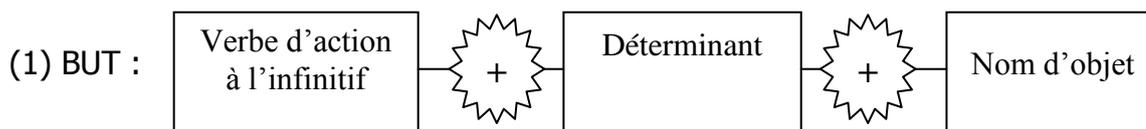
Définition 3.1 : but fonctionnel

Un but fonctionnel définit un besoin/attente potentiel(le) que le système peut satisfaire, il exprime ce que l'utilisateur du système souhaiterait faire [3].

Nous déduisons de cette définition, qu'un but fonctionnel est simplement, un texte exprimant l'intention d'usage d'un utilisateur quelconque. Ce qui implique l'usage de verbes d'action et de la voix active. Par conséquent, il y a existence, dans le texte, d'un groupe de mots traduisant l'action qui est susceptible d'être accomplie par l'utilisateur d'une part, d'autre part, l'existence d'un groupe d'objets qui subit l'action de cet utilisateur ; et si possible, le résultat final escompté. Il est dit dans [79]

qu' « un verbe d'action exprime ce que fait le sujet ou bien ce qu'il subit » et comme [80] nous renseigne que l'infinitif est « la forme la plus simple de l'expression verbale ». Nous avons opté de représenter l'action traduite par le groupe de mots évoqués précédemment, par un verbe d'action à l'infinitif. Ce qui nous donne la représentation suivante du but :

Règle (1) :



- Déterminants possibles sont : soient des articles définis (l', la, le, les), soient des articles indéfinis (d', de, du, des).
- Nom d'objet est un groupe nominal qui représente soit un ensemble d'intrants de même nature du système ; soient un ensemble d'artéfacts de même nature de ce système. Il est dérivé du nom du fragment de connaissance qui encapsule le but.

De manière formelle, un but b sera représenté par :

ϑ est un verbe d'action à l'infinitif
 $b: (\vartheta, \Delta_b(\tau), \tau)$, où: τ est le nom d'objet
 $\Delta_b(\tau)$ est le déterminant de τ

Définition 3.2 : But consistant

Nous dirons qu'un but est consistant lorsque son expression est conforme à la règle (1).

Définition 3.3 : but inconsistant

Un but est inconsistant lorsque son expression n'est pas conforme à la règle (1).

Lemme (1): Tout but inconsistant peut être transformé en but consistant.

Preuve : Par définition, un but est un texte traduisant l'intention d'usage d'un utilisateur. D'après [81] et [82] ce texte est compressible en un texte représentant la globalité de l'intention. Comme nous sommes en présence d'un texte traduisant une intention d'usage, alors nous pouvons identifier l'action et l'objet sur lequel cette

action s'applique. Etant donnée que l'action est connue d'après ce qui précède alors, il est évident de trouver le verbe qui traduit cette action d'une part et d'autre part de trouver le déterminant du groupe d'objets.

Le processus de transformation d'un but inconsistant en but consistant est appelé la globalisation du but. Soit b un but inconsistant, et une fonction $\theta: L \rightarrow L$, la fonction de globalisation. $\theta(b)$ représente le but consistant déduit de la globalisation de b . $\theta(b)$ est appelé but globalisé ou but global de b . En appliquant [82] et [81], $\theta(b)$ globalise l'intention d'usage de l'utilisateur. Ce qui veut dire, qu'il présente l'intention d'usage de l'utilisateur dans sa globalité. Ainsi, nous pouvons déduire que l'intention d'usage de l'utilisateur est un raffinement du but globalisé. Nous déduisons de ce fait qu'un but globalisé admet toujours au moins un sous but qui représente le besoin spécifique de l'utilisateur. Un but qui n'admet pas de sous buts est appelé : but élémentaire.

Désignons par B , un ensemble des buts consistants d'une organisation; A , l'ensemble des intentions émises par les cadres métiers de cette organisation ; et f est une fonction définie de manière suivante :

$$f: A \rightarrow B, f(b) = \begin{cases} b & \text{si } b \text{ est consistant} \\ \theta(b) & \text{sinon} \end{cases}$$

Dans cette approche le but est le concept clé. Dans la suite de cette section, nous allons simplement donner les définitions des autres composantes du fragment de connaissance. Ces autres composantes auront leur pesant d'or lorsqu'il faudra passer de notre spécification à d'autres spécifications.

Définition 3.4 : but polysémique

Un but est dit polysémique si la règle de gestion qui le régit change d'un domaine à l'autre.

II.1.6 REGLE DE GESTION

Définition 3.5 : règle de gestion

Une règle de gestion définit une loi du domaine à laquelle le but doit se

conformer. Elle aide à organiser un processus de gestion pour atteindre le but [3].

Les règles de gestion ont un impact aussi bien sur l'environnement externe (l'organisation), que sur l'environnement interne (le système) [83]. Elles sont de type règle d'ordonnancement, règle de déclenchement ou contrainte statique [3]. Une règle d'ordonnancement est une loi du domaine qui décrit l'ordre dans lequel les buts doivent être réalisés, pour une instance d'objet donné. La règle de déclenchement et les contraintes statiques gardent la même sémantique que dans [3]. Dans la suite, la notion de règle de gestion sera restreinte relative aux règles métiers.

II.1.7 IMPORTANCE DU BESOIN EXPRIME

Définition 3.6 : niveau d'importance du besoin exprimé

Le « niveau d'importance du besoin exprimé » traduisant le crédit qu'un utilisateur accorde à ce besoin exprimé [9, 10]. Il est de type très important et indispensable. Il est possible de choisir une valeur numérique pour exprimer le niveau d'importance d'une attente d'un utilisateur. Nous avons constaté par sondage effectué dans un échantillon de vingt-cinq administrations camerounaises que les cadres métiers que nous avons rencontrés avaient de la peine à attribuer des valeurs numériques pour matérialiser le crédit qu'ils accordaient à leurs attentes. Nous conseillons d'utiliser, à cet effet, une échelle d'évaluation du niveau « d'importance » d'un besoin en fonction du public cible (futurs utilisateurs du système).

II.1.8 FRAGMENT DE CONNAISSANCE OU BESOINS EXPRIMES

Un besoin exprimé ou fragment de connaissance est défini de la manière suivante d'après [9]: $\partial = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$ où ∂ est le nom du fragment de connaissance, $\langle \psi, \omega \rangle$ la définition du but ω dans le contexte ψ , $\langle \lambda, \delta \rangle$ la représentation de la règle de gestion λ sous les contraintes δ .

Nous avons remplacé dans la définition de [3], le « fragment conceptuel » par « contraintes, niveau d'importance » de l'intention d'usage de l'utilisateur, parce que dans le cadre de ce travail, nous nous intéressons exclusivement, à la formalisation de la représentation des besoins des utilisateurs. Dans le souci de prioriser les buts

des utilisateurs, nous avons introduit la composante « niveau d'importance » dans le fragment de connaissance. La composante « contrainte » a été ajoutée pour exhiber les besoins non fonctionnels qui se cachent derrière l'intention d'usage des utilisateurs. Ces besoins non fonctionnels pourront être raffinés par les ingénieurs logiciels.

Contrainte :- les verbes d'actions admissibles dans la suite sont ceux qui traduisent des actions, de préférence de gestion, susceptibles d'être automatisées par un système informatique. Par exemple, les verbes : parler, manger, sauter, courir, rire, vendre, peindre, etc., sont exclus ; - nos travaux portent exclusivement sur les buts fonctionnels ;- les buts fonctionnels sont exprimés avec des phrases à la voix active.

II.2 RAPPEL DES CONCEPTS DE BASE INHERENTS A LA COMPREHENSION ET A LA MODELISATION DE LA SATISFACTION DES BENEFICIAIRES DE SERVICE

Dans la section précédente, nous avons présenté, d'une manière générale, les différentes facettes du processus d'affaire. Il a aussi été relevé un certain nombre d'insuffisances sur chaque facette susmentionnée. Dans cette section, nous allons dans un premier temps, formaliser les différents concepts de bases indispensables à la prise en compte de la satisfaction des bénéficiaires des services. Ces concepts sont extraits de [14].

Définition (3.7) : Observateur

Un observateur est une variable attachée, à une règle métier ou à un objet métier d'un processus d'affaire, dont le contenu est mis à jour pendant l'exécution d'une activité de ce processus d'affaire faisant usage soit de la règle métier, soit de l'objet métier [14].

Définition (3.8) : Observation

Une observation est une variable booléenne [14].

Nous considérons dans la suite, *Observers* comme étant l'ensemble des

observateurs d'un processus d'affaire et $card(X)$ représentera le cardinal d'un ensemble X .

Définition (3.9) : Fonction d'Evaluation

Considérons une fonction $f_{eval}: Observers \rightarrow Indicators$, qui pour tout Obs élément de $Observers$, $f_{eval}(Obs)$ retourne une valeur numérique représentant l'évaluation du contenu de l'observateur Obs . f_{eval} est appelée fonction d'évaluation [14] et $Indicators$ représente l'ensemble des indicateurs du processus d'affaire définit ci-dessous.

Définition (3.10) : Fonction d'Observation [14]

Considérons une fonction $f_{th}: R \rightarrow \{0, 1\}$ (où : th est une valeur numérique représentant le seuil), qui pour tout réel a ,

$$f_{th}(a) = \begin{cases} 0, & \text{if } a < th \\ 1, & \text{else} \end{cases}$$

Définition (3.11): Indicateur

Un indicateur est une information contribuant à l'appréciation d'une situation [14]. Elle représente le résultat de l'évaluation d'un observateur associé à une activité du processus d'affaire. De manière formelle, un indicateur e associé à un observateur b ($b \in Observers$) sera défini ainsi qu'il suit : $e = f_{eval}(b), e \in [0, 1]$ [14].

Nous considérons dans la suite $IndiQ$ en lieu et place de $Indicators$ comme étant l'ensemble des indicateurs du processus d'affaires.

Définition (3.12) : Observation d'un Indicateur [14]

Considérons un indicateur $e \in IndiQ$ et $b \in Observers$, l'observateur associé à e (avec $e = f_{eval}(b)$), nous définissons une observation de e comme étant une fonction à seuil $g_{th}: IndiQ \rightarrow \{0, 1\}$ telle que :

$$g_{th}(e) = \begin{cases} 0, & \text{if } f_{eval}(b) < th \\ 1, & \text{else} \end{cases} \quad \text{où: } th \text{ est le seuil de satisfaction}$$

($th \in]0, 1[$),

N.B (1) : Sans nuire à la généralité, $f_{eval}(b)$ pourra de temps à autre être remplacé par une valeur réelle dans l'intervalle $[0, 1]$.

II.2.1 POINT DE VUE

Le point de vue d'un client par rapport au service qui lui est rendu dans une organisation, représente le sentiment que celui-ci éprouve par rapport au service qui lui a été rendu dans cette organisation. D'après [84], ce point de vue s'appuie sur un ensemble d'observations dont il faille connaître l'importance. La pertinence de la modélisation des points de vue des clients d'une organisation se justifie dans la mesure où c'est le client d'une organisation qui est le principal bénéficiaire du service rendu par cette organisation. Et, ne pas prendre en compte l'avis de ces derniers peut entraîner, pour cette organisation, la perte de sa clientèle au profit des organisations concurrentes. La complexité dans la prise en compte des points de vue des clients d'une organisation vient du fait que : -le point de vue est relatif et dépend de chaque client ; -le point de vue d'un client fait suite au service dont a bénéficié le client. Nous croyons, cependant, que si le processus d'affaire est orienté satisfaction des clients, il sera possible que ce dernier soit réajusté dans le temps en fonction des orientations de la clientèle. Deux concepts sont importants pour la représentation des points de vue des clients : -les facteurs de qualité ; -la connaissance de la satisfaction de la clientèle.

II.2.2 FACTEUR DE QUALITE

Un facteur de Qualité est une observation à partir de laquelle un client s'appuie pour donner son jugement par rapport au service qui lui a été rendu. Les facteurs de qualité sont intimement liés au service dont a bénéficié le client. En nous appuyant sur les travaux de [84], nous définissons un facteur de qualité *fact* de la manière suivante :

$fact = \langle value, Poids \rangle$ Où :

- *value* : est une observation ($value \in \{0, 1\}$);
- *Poids* : représente l'importance de l'observation *value* dans la satisfaction du client.

Nous considérons dans la suite *QualityFactors* comme étant l'ensemble des

facteurs de qualité sur lesquels s'appuient les jugements des bénéficiaires du service d'un processus d'affaire. nous noterons $\mathcal{P}(QualityFactors)$ est l'ensemble des parties de $QualityFactors$.

Définition (3.13) : Indicateur de Satisfaction [14]

Considérons un client cl et un service S dont il a bénéficié dans une organisation. Supposons que W est l'ensemble des facteurs de qualité sur lesquels s'appuie le jugement de cl , nous définissons une fonction $h : \mathcal{P}(QualityFactors) \rightarrow [0,1]$ telle que, W élément de $\mathcal{P}(QualityFactors)$,

$$h(W) = \frac{\sum_{fact=(val,p) \in QualityFactors} (fact.p \times fact.val)}{\sum_{fact=(val,p) \in QualityFactors} fact.p}$$

$h(W)$ est appelé indicateur de satisfaction des clients

Définition (3.14) : Fonction de Satisfaction [14]

Considérons un client cl et un service S dont il a bénéficié dans une organisation. Supposons que W est l'ensemble des facteurs de qualité sur lesquels s'appuie le jugement de cl , nous définissons une fonction à seuil $Sat_{th} : \mathcal{P}(QualityFactors) \rightarrow \{0,1\}$ telle que, W élément de $\mathcal{P}(QualityFactors)$,

$$Sat_{th} = \begin{cases} 0 & \text{if } h(w) < th \\ 1 & \text{else} \end{cases}, \quad th \in [0,1] \text{ représente le seuil de satisfaction du}$$

client cl

N.B (2): sans nuire à la généralité, nous remplacerons $h(w)$ par sa valeur numérique.

Un point de vu Pv d'un client étant le jugement que porte ce client sur le service qui lui est rendu dans une organisation, nous allons le représenter de manière formelle ainsi qu'il suit :

$$Pv_{cl} = \langle D, \langle cl, S \rangle, W \rangle \quad \text{Où :}$$

cl : représente le nom d'un utilisateur ;

S : est le service dont cl a bénéficié ;

W : est l'ensemble des observations du client cl ;

D : représente le contexte du service S

II.2.3 CONNAISSANCE DE LA SATISFACTION [14]

La connaissance de la satisfaction des clients représente l'ensemble des informations du processus d'affaire permettant de calculer les facteurs de qualité. Nous définissons de manière formelle une connaissance de satisfaction Cs de la manière suivante :

$$Cs = \langle D, S, \langle V_w, \Delta_{cl} \rangle \rangle \text{ où :}$$

D : représente le domaine de la connaissance ;

S : représente le service évalué ;

V_w : représente l'ensemble des observateurs dont l'évaluation conduit à l'ensemble des observations de W ;

Δ_{cl} : représente le seuil de satisfaction du client cl .

II.2.4 OBSERVATEUR DE PERFORMANCE

Un observateur de performance est un ensemble d'informations associées à une tâche ou à un objet métier à partir desquelles on peut mesurer le rendement d'une tâche ou l'état de progression ou de traitement d'un objet métier dans un processus d'affaire [14]. La modélisation d'un observateur de performance Op associé à une tâche t , tirée de [14], se définit de la manière suivante :

$$Op = \langle C, t, \Sigma_{obs} \rangle$$

Où : - t représente une tâche ou un objet métier ;

- Σ_{obs} : représente l'ensemble des observateurs associés à t ;

- C : représente un contexte ou domaine.

Les observateurs de performance sont mis à jour par l'exécution des activités auxquels ils sont rattachés. Dans la suite, *PerfObservers* dénotera l'ensemble des observateurs de performance et $pObs(t)$, l'ensemble des observateurs de

performance associés à t .

II.2.5 Indicateur de Performance

Considérons une activité t d'un processus d'affaire ; $Op \in PerfObservers$ l'observateur de performance associé à l'activité t ; $\mathcal{P}(PerfObservers)$ l'ensemble des parties de $PerfObservers$; nous définissons la fonction $f_{Ind} : \mathcal{P}(PerfObservers) \rightarrow [0, 1]$ qui est telle que quelque soit un observateur de performance Op ,

$$f_{Ind}(Op \cdot \Sigma_{obs}) = \sum_{obs \in Op \cdot \Sigma_{obs}} \frac{f_{eval}(obs)}{card(Op \cdot \Sigma_{obs})} \quad [14]$$

La fonction f_{Ind} est appelée indicateur de performance de l'activité t [14]. Aucune restriction n'est faite sur le nombre d'observateurs de performances associable à une activité. Il peut, par conséquent, arriver qu'une tâche ait plusieurs indicateurs de performances. Il en sera de même pour un objet métier. Nous noterons $IndiQ(t)$ l'ensemble des indicateurs associés à t où t peut être soit un objet métier, soit une tâche.

Les indicateurs de performances s'intéressent à la qualité de service interne du processus d'affaire d'une organisation. Il est relevé dans [14], l'existence de plusieurs catégories d'indicateurs :

- Indicateurs d'alerte : Ce type d'indicateur de type tout ou rien, signale un état anormal du système sous contrôle nécessitant une action, immédiate ou non. Un franchissement de seuil critique par exemple entre dans cette catégorie d'indicateur.
- Indicateurs d'équilibration : Ce type d'indicateur étroitement lié aux objectifs est un peu la boussole du décideur. Il informe sur l'état du système sous contrôle en relation avec les objectifs suivis. Seront-ils tenus ?
- Indicateurs d'anticipation : Un bon tableau de bord est aussi un instrument de prospective. Un bon tableau de bord permet de voir un peu plus loin que le bout de son écran et d'envisager avec une meilleure assise la situation actuelle. Doit-on continuer avec le plan actuel ? Le réviser ?

Il arrive très souvent que les indicateurs de performances dépendent des uns et des autres. Dans la Direction de Gestion de la Carrière du Ministère de la Fonction Publique et de la Réforme Administrative du Cameroun, les indicateurs utilisés sont des tableaux de bord. Il est constaté que le tableau de bord du Sous Directeur des personnels fonctionnaires est dépendant en grande partie de ceux des services qui lui sont rattachés ; et les tableaux de bord de ces chefs de services dépendent chacun en grande partie de ceux des bureaux qui lui sont rattachés. Tous ces tableaux de bords sont produits manuellement et ouvrant ainsi une brèche pour d'éventuelles erreurs. La modélisation de la satisfaction des clients permet de représenter ces différents types d'indicateurs afin de plus avoir besoin de les produire mais plutôt de les obtenir en guise de résultats.

Axiome 3.1 : Contraintes de Définition des Indicateurs

Les indicateurs de performances sont définis par l'approche Top-Down (du sommet de la hiérarchie des tâches vers les tâches se trouvant au dernier niveau de cette hiérarchie).

Les indicateurs de performance du sommet de la hiérarchie des tâches doivent être définis par rapport aux observations des bénéficiaires du service offert par le processus d'affaire.

Axiome 3.2 : Dépendance entre les Indicateurs de Performances

Considérons une tâche t ; $task(t)$ l'ensemble des sous tâches de la tâche t ; un indicateur Ind ; $typeof(Ind)$ (avec $Ind \in IndiQ(t)$) le type de l'indicateur Ind ; et $IndiQ_s(t)$ l'ensemble des indicateurs de type s associés aux sous tâches de t ,

$$task(t) = \phi, \forall Ind \in IndiQ(t), \exists o = (c, t, \sum_{obs}) \in pObs(t) / Ind = f_{Ind}(o. \sum_{obs}).$$

$$task(t) \neq \phi$$

$$\forall t' \in task(t), IndiQ(t') \cap IndiQ(t) = IndiQ(t') ;$$

$$\forall Ind \in IndiQ(t), \exists ! Ind' \in IndiQ(t') / typeof(Ind) = typeof(Ind') ;$$

La valeur de l'indicateur $Ind \in IndiQ(t)$ dépend de la valeur des indicateurs de performance de ses sous-taches.

Définition (3.15) : Facteurs de Performance

Un facteur de performance est un triplet $\langle Indicator, w, t \rangle$ où t est une tâche, $Indicator \in IndiQ(t)$ et w le poids de l'indicateur $Indicator$.

Axiome 3.3 : Dépendance entre les Facteurs de Performance

Deux facteurs de performances $factPer_1 = \langle Ind_1, w_1, t_1 \rangle$ et $factPer_2 = \langle Ind_2, w_2, t_2 \rangle$ sont dits dépendants si seulement si $IndiQ(t_2)$ et $IndiQ(t_1)$ sont dépendants ;

Considérons un facteur de performance $facper = \langle Ind, weight, t \rangle$, nous supposons $task(t) \neq \emptyset$. Considérons aussi que $factors(t)$ l'ensemble des facteurs de performance associés à t ; et $subfactors_s(t)$ l'ensemble des facteurs de performance associés aux sous tâches de t pour lesquels les indicateurs de performance sont de type s .

La valeur $facper.Ind$ est calculée de la manière suivante :

$$facper.Ind = \frac{\sum_{id=(i,w,s) \in subfactors(t)} id.i \times id.w}{\sum_{o=(j,v,l) \in subfactors_s(t)} o.v}$$

La valeur de $facper.w$ sera déterminée de la manière suivante :

$$facper.weight = \max \left(\bigcup_{o=(i,w,s) \in subfactors_s(t)} \{o.w\} \right)$$

Où : $s = typeof(facper.Ind)$ et max retourne le plus grand élément de la liste d'éléments reçu en paramètre.

Tout au long de ce point relatif aux rappels des concepts de base inhérents à la compréhension et à la modélisation de la satisfaction des bénéficiaire de service, il n'a pas été formalisé, dans le fragment de connaissance, les concepts de contexte, contraintes et règles métiers et ceux relatifs à la prise en compte des exigences de qualités des utilisateurs dans la modélisation des besoins d'un processus d'affaire. Les concepts d'observateurs, d'observation, but ou goal cher au formalisme relevé dans [1, 2] ont été clairement définis. La section suivante sera consacrée à la formalisation des concepts de règles métiers et incidemment ceux relatifs aux contextes, besoins non fonctionnels, ou pré et post conditions.

III. FORMALISATION DE QUELQUES CONCEPTS

III.1 INTRODUCTION

Mouhamed Diouf au cours de sa thèse [85], a défini un formalisme de représentation des règles métiers faisant usage des métadonnées. Ce formalisme intègre les concepts de niveau d'importance d'une règle métier que nous avons relevé dans [9]. De ce fait, ces travaux sont proches de ceux que nous avons menés dans [9]. C'est la raison pour laquelle, le formalisme ci-dessous présenté, s'appuie sur les travaux de [85]. Notre formalisme, enrichi les travaux de [85], d'un ensemble de concepts formels permettant de rendre manipulable à l'instar des types numériques les règles métiers.

III. 2.- CANEVAS POUR LA FORMALISATION DES REGLES METIERS

Le concept de « règle métier » a été largement développé dans le contexte des systèmes experts. Connus sous l'acronyme anglophone BRMS («Business Rules Management Systems »), les outils de gestion de règles métiers sont des dérivés des systèmes experts, systèmes qui ont connu leurs heures de gloire dans les années 1980[86]. Ils utilisent des vocables similaires et ont tous au cœur, un moteur d'inférence. Dans le cadre des approches orientées règles métiers (ARM), ce moteur d'inférence se dénomme gestionnaires des règles métiers [85]. Ils visent la modélisation du raisonnement humain, les comportements et les mécanismes cognitifs d'un expert humain dans un domaine particulier. Pour cela, ils s'appuient sur une base de faits (mémoire de travail), un moteur d'inférence (moteur de règles) et une base de règles (mémoire de production) [85]. Dans le cadre de cette thèse, nous nous sommes appuyés sur les travaux de [85] car son formalisme avait des concepts voisins de ceux développés dans [9, 10]. Les travaux ci-dessous relatés sont déduits de [11, 12].

III.2.1 DEFINITION: REGLE METIER

Dans la littérature, le concept de « règle métier » a été largement utilisé, chaque auteur y allant selon sa vision. Ainsi :

- Ronald G. Ross dans [89] et le Business Rule Community (BRG) dans [90]

soutiennent qu'une règle métier est une formulation qui définit ou contraint quelques aspects du métier. Son but est de structurer un métier (politique, savoir-faire), de contrôler ou influencer le comportement d'un métier ;

– D. Hay et K. A Healy dans [91], quant à eux, nous disent qu'une « règle métier » est une directive, qui est censée influencer ou guider le comportement d'un métier, dans le but de mettre en œuvre une politique métier qui est formulée en vue d'une réponse à une opportunité ou un risque ;

– Farida Semmak et Joël Brunet dans [3] en outre, pense qu'une règle métier définit une loi du domaine à laquelle le but doit se conformer. Elle aide à organiser un processus de gestion pour atteindre le but. Le but, quant à lui, définit une attente potentielle que le système peut satisfaire, il exprime ce que l'utilisateur du système souhaiterait faire.

Nous pensons que pour mieux cerner la sémantique d'une règle métier, qu'il est nécessaire de la situer dans un contexte ou domaine précis. Ainsi, une « règle métier », d'après nous, sera une directive du domaine qui régit le comportement d'un métier dudit domaine [9]. Son but sera de structurer un métier (politique, savoir-faire), de contrôler ou influencer le comportement d'un métier du domaine en question, dans l'optique de l'atteinte d'un résultat attendu. [85] nous renseigne que les règles métiers se subdivisent en deux grandes catégories :

– les règles structurelles (indiquant une nécessité) : ce sont des règles qui portent sur comment est-ce que le métier est organisé ou est structuré les éléments qui le composent. Les règles structurelles sont des compléments de définition ;

– les règles opératives (indiquant une obligation) : ce sont des règles qui contrôlent la manière dont les activités du métier sont conduites. Contrairement aux règles structurelles, les règles opératives sont celles qui peuvent être directement violées par les intervenants du métier.

Dans la suite, nous nous intéressons exclusivement aux règles opératives en général, aux règles de production en particulier.

III.2.2 STRUCTURE DES REGLES METIERS

Mouhamed Diouf dans [85] nous renseigne que, suivant le type de règle métier

(procédurale ou déclarative), la structure et la sémantique de cette règle métier varient. Dans cette section, nous nous focalisons sur les règles opératives en général et plus spécifiquement aux règles de production. Elles sont de la forme « if . . .then . . .[else . . .] » et sont constituées des éléments suivants :

- Déclarations des variables : permet de définir le contexte d'application de la règle métier ;
- Valeur métier (Expression) ou « niveau d'importance »: c'est une expression dont l'évaluation renvoie un entier correspondant à l'importance de la règle. Elle permet de définir un ordre dans l'exécution des règles. Cette valeur métier peut aussi être appelée priorité. Dans [9], nous avons associé le niveau d'importance au fragment de connaissance. Il serait malvenu de l'associer explicitement à la règle métier étant entendu que cette association est implicite dans [9]. Nous avons donc supprimé cette information dans la spécification que nous allons proposer dans la section suivante ;
- « If » ou condition, ou « left-hand part », ou prédicat, ou prémisse (expression) : Elle constitue la condition sous laquelle la règle peut être appliquée ; Elle est constituée d'une expression qui est évaluée à vrai ou à faux. Elle est aussi appelée « body » ;
- « Then » (Expression à effet de bord) ou conclusion ou « right-hand side » : c'est le corps de la règle ou « head ». Elle représente l'action à effectuer si la partie condition est évaluée à vrai. Dans le cadre des règles déclaratives, cette partie doit être une expression sans effet de bord.
- « Else » (optionnelle et Expression à effet de bord). Elle n'est uniquement envisageable que pour les règles procédurales si elles contiennent une partie « If ». Elle représente l'action à effectuer si la partie condition est évaluée à faux.

Les expressions sont classées en deux groupes selon qu'elles influencent ou non l'état du système. Il y a des expressions à effet de bord et des expressions sans effet de bord. Une expression est sans effet de bord si et seulement si elle ne modifie pas l'état de l'environnement. Certaines expressions, par contre, pendant l'activité d'un métier, modifient l'état de l'environnement. Elles sont dites avec effet

de bord.

Bien que dans [3, 85, 87, 88, 86, 92, 89, 90], nous ayons une vision claire de ce qu'est une règle métier, il n'en demeure pas moins que jusqu'à présent, les chercheurs semblent ne pas s'intéresser aux faits qu'on puisse faire un certain nombre d'opérations sur ces règles. Notre intérêt de définir un ensemble d'opérations sur les règles métiers se justifie par la nécessité de pouvoir :

- réutiliser les règles métiers dans le cadre de la réingénierie d'un processus d'affaires ou d'un développement par composants;
- composer les règles métiers dans le cadre de l'ingénierie dirigée par les modèles ;
- statuer sur la qualité de la formulation d'une règle après sa spécification par les experts (cadres métiers de l'organisation et experts logiciels) ;
- d'ouvrir un pant de recherche sur la description d'un processus d'affaire orientée règle métiers ;
- évaluer la qualité de la composition des modèles.

Mouhamed Diouf dans [85] résume la spécification formelle d'une règle métier de la manière suivante :

Rule = (RuleLabel, (VariableDeclaration)*, (conditions, action))

où :

RuleLabel=(comment ?, priority ?, Visible ?, refraction ?, createdTime ?,
lastModificationDateTime ?
)

Conditions = (Expression | AndExpression | OrExpression |
NegationExpression)+

Action = ((Expression)* | NegationExpression+)

Dans la suite, nous allons définir formellement la spécification d'une règle métier telle que nous la proposons dans notre approche. Nous enchaînerons par la suite par les opérations effectuables sur les règles métiers. L'expression mise en scène des objets renvoie à l'usage des propriétés de chaque objet dans une expression. Le

concept de règle métier renvoie aux règles métiers de production.

III.3.- PROPOSITION D'UNE SPECIFICATION DE REGLES METIERS

Dans cette section nous allons présenter les concepts nécessaires à la description d'une règle métier selon notre approche. Ces concepts s'apparentent à ceux qui existent dans le langage OCL [93] d'une part, et d'autre part à ceux utilisés dans les BRMS [86]. De manière générale nous garderons l'ossature d'une règle métier telle que définie dans [85]. Il est à noter dans cette section que nous ne définissons pas un langage de programmation, mais une manière de faire pour la spécification des règles métiers de telle sorte que celle-ci nous permette de pouvoir effectuer un certains nombres d'opérations sur ces critères.

III.3.1 TYPES PREDEFINIS

III.3.1.1 TYPES SIMPLES

Cinq types simples sont employés dans notre approche de spécifications :

- le type date : réservé aux données référençant les dates ; le type date permet aussi de spécifier les attributs des objets qui font référence aux dates ou aux périodes ;
- le type nombre : le type nombre concerne les données renvoyant à des valeurs numériques. Dans le cas d'espèce, les valeurs entières, décimales ou réelles sont contenues dans ce type ;
- le type règle : le type règle est utilisé pour la déclaration d'une règle métier ;
- le type « String » : renvoie à une donnée caractère ou chaînes de caractères ;
- le type « Observer » : représente un couple (*parameter, weight*) où : *parameter* représente l'observateur et *weight*, le poids de cet observateur.

Dans ce cas, il représente un facteur de performance associé à un objet métier dans lequel, il a été masqué le nom de l'objet métier et le contexte car ces informations sont contenues dans l'environnement dans lequel cet objet a été déclaré.

Une constante a de type Date, ou Nombre ou String est notée de la manière suivante : $\#a\#$ (respectivement, ' a ', " a "). Considérons a et b deux éléments de type *String*, nous définissons la fonction *SubStr* de la manière suivante :

$$String \times String \rightarrow \{Vrai, Faux\}$$

$$SubStr: \forall (a, b) \in String^2, SubStr(a, b) = \begin{cases} Vrai, si a est une partie de b \\ Faux, Sinon \end{cases}$$

Dans la suite, nous allons noter par *SimpleType* l'ensemble des types simples précédemment cités. Nous noterons aussi $\prod SimpleType$, le choix arbitraire d'un type simple dans *SimpleType*.

III.3.1.2 TYPES COMPOSES

A ces cinq types, nous avons ajouté deux suffixes : « agrégat », « views » ; et un type complexe *Something*. Il est aussi à relever que le type « Observer » peut aussi être utilisé comme type complexe. Dans ce cas, il représente un facteur de performance à associé à une règle métier dans lequel les composantes contexte et tâches ont été masqués dans la mesure où ces composantes se trouvent définies dans la règle métier. Dans les deux cas de déclaration, le type « Observer » a toujours un suffixe « views ». Un objet (respectivement attribut) de type Observer ne peut avoir ni de mention « référence », ni de mention « control » , encore moins « artifact ».

Le type *Something* fait référence à un objet physique. Exemple : un acte de carrière, une demande timbrée, une carte grise, une carte nationale d'identité, un bon de commande, une facture, etc. ; tandis que :

- le suffixe « agrégat » précédé du mot clé « **NameSpace** », fait référence à un ensemble d'objets dans la portée de la règle ;
- le suffixe « views » précédé d'un nom d'objet contenu dans le domaine circonscrit par le suffixe « agrégat », fait référence à l'ensemble des informations (propriétés ou attributs) de cet objet qui seront utilisées par la règle métier.

Les objets listés à l'aide du suffixe « agrégat » peuvent être accompagnés par les mentions ci-après :

- La mention « control » : indique que l'objet est obligatoire, mais aucune donnée de cet objet n'intervient dans la partie « description » de la règle métier. Si un tel élément vient à être absent, la donnée est déclarée invalide. Dans le dossier d'avancement d'échelon par exemple, la présence effective est nécessaire, mais elle n'a aucune incidence dans le traitement proprement dit du dossier ;
- La mention « référence » : elle renvoie un attribut, permettant d'identifier de manière unique un objet quelconque. Lors de la spécification de la règle métier, la valeur de cet attribut n'est pas connue. Un prédicat faisant usage de « référence » a la structure suivante :

... fait référence à [dans ...]

La mention « référence » peut apparaître aussi bien dans le suffixe *agrégat* que dans le suffixe *views*. Dans le suffixe *agrégat*, elle permet de faire appel à un objet déclaré hors de cette règle métier dont on voudrait éventuellement étendre le suffixe *views*. Dans le suffixe *views*, elle permet de faire appel à un objet dont on aimerait utilisé sans toutefois avoir accès à ces propriétés d'une part ; de déclarer une liste d'objets. Cette mention permet de mettre en relation les objets de l'organisation. Lorsqu'un objet X référence un attribut *w* d'un autre objet Y, l'objet X existe si et seulement si, la valeur *w* dans X existe dans les valeurs présent par *w* dans Y.

- La mention « artifact » indique que l'objet en question est produit par l'organisation.

Toutes ces mentions sont exclusives c'est-à-dire que lorsqu'elles ne sont pas combinables.

III.3.2 SYNTAXE DE DECLARATION

Contrairement aux langages de programmation (C++, Vb.Net, C#) ou les langages de spécification des règles métiers comme OCL [92], les déclarations n'ont pas une section dédiée. C'est pendant la spécification des règles que l'on spécifie les types de données à utiliser dans la règle en question.

III.3.3 DECLARATION DES REGLES METIERS

La déclaration d'une règle métier se fait en deux phases : la définition de son contexte, et la spécification de la directive du domaine qui régit le comportement du métier en question [11]. La première partie de la déclaration d'une règle métier nous permet de spécifier le contexte d'usage de chaque objet de l'organisation impliqué dans ce métier et les domaines dans lesquels cette règle métier peut encore être utilisée. Ces domaines sont listés immédiatement après le mot clé « Keywords ». La deuxième partie, quant à elle, est réservée à la description de l'action qui doit être effectuée pour réaliser le métier. Cette partie est appelée description.

III.3.3.1 DEFINITION DU CONTEXTE

La spécification de toute règle métier commence par la définition de son contexte d'usage. Le contexte d'usage d'une règle métier définit d'une part, le champ sémantique des objets utilisés dans la partie description ; et d'autre part, les domaines dans lesquels cette règle métier pourra être utilisée (champ d'exploitation de la règle). D'après [11], le contexte d'une règle métier de la manière suivante :

Contexte :

Keywords =

Domain₁, Domain₂, ..., Domain_n;

Data =

*Namespace.agrégat {object % Observer | Something [: control: artifacts]
[object % Observer | Something [: control: artifacts]] *};
object.views {field % [] SimpleType [, field % [] SimpleType] *}.*

Où:

-*field* représente soit une référence à un autre objet qui peut ne pas être dans le *Namespace*; soit un nom simple d'un attribut ou une propriété de cet objet. *field* est aussi appelé arguments du suffixe « views », ou attributs références dans le suffixe « views ».

-*[] SimpleType* représente le type de *field*

Le suffixe « agrégat » définit l'ensemble des objets accessibles lors du déroulement du métier dont la règle est en train d'être définie. Le suffixe « views » permet, pour les objets qui n'ont pas de mention « control », de définir explicitement l'ensemble des informations (propriétés ou attributs) de cet objet qui, seront obligatoirement utilisées dans la partie description de ladite règle métier. Il n'est autorisé qu'un seul agrégat par règle métier. Par ailleurs, il est important de relever que lorsqu'un objet a la mention « artifact », tous ces attributs doivent être définis avec la mention « référence » suivant la syntaxe ci-après définie. Les mentions « artifact », et « control » sont liées au contexte encours. C'est-à-dire, qu'un objet peut avoir la mention control dans un contexte X d'une règle métier X, la mention artifact dans un contexte Y d'une règle métier Y, et peut ne pas avoir une mention dans un contexte Z d'une règle métier Z. Dans la suite, nous appellerons objets du contexte, les objets n'ayant pas de mention « control » dans le suffixe « agrégat ». Dans la suite, le concept ensemble sera considéré comme un ensemble multiple (MultiSet).

Considérons un objet b de ce contexte n'ayant pas la mention « control », nous noterons : $card(b)$, le nombre de propriétés de b listées dans le suffixe « views » ; $NameSpace_C$, l'ensemble des objets référencés dans le suffixe « agrégat » du contexte C ; $card(NameSpace_C)$ représentera le nombre d'objets spécifiés dans le suffixe « agrégat » ; Sg_C , l'ensemble des signatures des objets du contexte C ; et $Keywords_C$, l'ensemble des mots listés après le mot clé « Keywords » dans le contexte C . Nous notons par $Sig_C(b)$, la signature de b définie de la manière suivante :

$$NameSpace_C \rightarrow Sg_C$$

$$Sig_C: \forall b \in NameSpace_C, \quad Sig_C(b) = \left(\prod SimpleType \left[, \prod SimpleType \right]^* \right)$$

La signature de b , $Sig_C(b)$ représente la liste des types d'attributs de b définis par le suffixe « views » .

Etant donné un autre objet a du même contexte, nous notons $Tri(a, b)$, la fonction qui retourne la signature de a ordonnée dans l'ordre défini par celle de b . Tri est définie de la manière suivante :

$$NameSpace_C^2 \rightarrow Sig_C$$

$$Tri: \forall (a, b) \in NameSpace_C^2, Tri(a, b) = \begin{cases} (Sig_C(a) \cap Sig_C(b)) \cup (Sig_C(a) - Sig_C(b)), \\ \quad card(a) > card(b) \\ (Sig_C(a) \cap Sig_C(b)), \text{ sinon} \end{cases}$$

Axiome 3.4 : Equivalence de Signature

(1) Nous dirons que a et b ont des signatures équivalentes si et seulement si :

$$Tri(a, b) = Sig_C(b).$$

(2) a et b sont dits équivalents dans leur contexte.

Axiome 3.5 : Raffinement ou Extension de Signature

(3) Nous dirons que la signature de a est un raffinement de celle de b si et seulement si :

$$Sig_C(a) \subset Sig_C(b)$$

(4) nous dirons aussi que b est une extension de a .

Axiome 3.6 : différence de signature

Nous dirons que les signatures des objets a et b sont différentes si et seulement si elles ne vérifient ni l'axiome 3.4, ni l'axiome 3.5.

Axiome 3.7: *NameSpace* et *Contextes*:

Soient deux contextes C et B ,

(5) les $NameSpace_C$ et $NameSpace_B$ seront dits équivalents si et seulement si :

1. $card(NameSpace_C) = card(NameSpace_B)$ et,
2. $\forall a \in NameSpace_B \exists b \in NameSpace_C, Sig_B(a) = Sig_C(b)$.

(6) nous dirons aussi que $NameSpace_B$ raffine $NameSpace_C$ si et seulement si :

1. $card(NameSpace_C) > card(NameSpace_B)$ et,
2. $\forall a \in NameSpace_B \exists b \in NameSpace_C, Sig_B(a) = Sig_C(b)$.

(7) C et B seront dits équivalents si et seulement si :

1. $NameSpace_C$ et $NameSpace_B$ sont équivalents ; et,
2. $Keywords_B \cap Keywords_C \neq \emptyset$.

(8) nous dirons que le contexte C raffine le contexte B si et seulement si :

1. $NameSpace_C$ raffine $NameSpace_B$; et,
2. $Keywords_B \cap Keywords_C \neq \emptyset$.

III.3.3.2 DÉFINITION DE LA DESCRIPTION

La partie « description » d'une règle métier quelconque est réservée à la spécification des directives qui doivent être déroulées pour l'atteinte des résultats attendus. De manière formelle, nous allons définir la partie description d'une règle métier par le triplet :

Description : (Garde, Séquence, Résultats)

Où :

- Garde : est la condition qui doit être satisfaite pour que les actions soient exécutées ;
- Séquence : est une suite d'actions séparées par des virgules ;
- Résultats : est une spécification du résultat attendu.

III.3.3.2.1 GARDE

La garde est une condition au sens de [85] qui doit être satisfaite pour que la partie description de la règle métier soit activée. Elle est spécifiée de manière littérale par une expression du langage humain mettant en scène les objets et attributs listés dans le contexte. De manière concrète, elle est matérialisée dans la formule « if condition then ... else ... » par « condition ». Une garde sera dite simple, si et seulement si elle est la condition de la formulation « if condition then ... ». Elle est dite complexe si et seulement si elle est la condition de la formulation « if condition then ... else ... ».

Une condition est une expression du langage naturel qui exprime une réalité qui peut être soit vraie, soit fausse. Pour pouvoir exprimer des expressions, nous avons défini un certain nombre de prédicats : les prédicats de comparaison, les prédicats de coordination, et les prédicats de la négation. Le tableau ci-après récapitule les prédicats suscités.

Type de prédicats	Prédicats	Notation
Les prédicats de comparaison	... est inférieur à...	(lt, ...,...)
	... est inférieur ou égal à...	(le, ...,...)
	... est supérieur à...	(ut, ...,...)
	... est supérieur ou égal à...	(ue, ...,...)
	... est égal à...	($\stackrel{def}{=}$, ... , ...)
existe	(\exists , ...)
	Existe plusieurs.....	($\exists p$, ...)
les prédicats de conjonction	... et ...	(\wedge , ..., ...)
	... ou ...	(\vee , ...,...)
Le prédicat de la négation	opposé de ...	(\neg ,...)

Tableau 3.1 : liste des prédicats de comparaison

Les prédicats de conjonction ne s'appliquent que sur les conditions. Il s'agit d'une conjonction de conditions à l'aide de conjonctions de coordination ('ou' ou 'et'). Pour réduire les incompréhensions entre cadre métiers et développeurs, il était important de rapprocher notre spécification du langage naturel. Il est cependant à noter que seuls les objets déclarés dans le contexte et n'ayant pas de mention « control », doivent être utilisés. Des prédicats suscités nous définissons trois types d'expression : *CompExpression*, réservée aux expressions littérales mettant en scène deux objets du contexte liés par un prédicat de comparaison; *NegExpression*, réservée aux expressions exprimant le contraire de la réalité exprimée par une condition. *NegExpression* ne s'applique qu'aux conditions ; *CrdExpression*, réservée aux expressions traduisant la composition de conditions. La représentation formelle de chacun des types d'expressions précédentes est la suivante :

$CompExpression : \langle Object_1.attr, Object_2.attr, \Delta \rangle$

$NegExpression : \langle ComExpression_1, \neg \rangle$

$CrdExpression$:

$\langle ComExpression_1, ComExpression_2 \mid CrExpression \mid NegExpression, \diamond \rangle$

Où :

- $Object_x.attr$, représente un attribut de l'objet $Object_x$, spécifié dans le suffixe « views » ;
- $ComExpression_x$ est une expression de type $CompExpression$;
- $CrExpression$ est une expression de type $CrdExpression$;
- Δ est un prédicat de comparaison ;
- \diamond est un prédicat de conjonction ;
- \neg le prédicat de la négation.

D'une manière générale, une condition sera représentée de manière formelle par :
 $Condition : \langle CompExpression \mid CrdExpression \mid NegExpression \rangle$

Nous dénommons par *Conditions*, l'ensemble des conditions et définissons la fonction $TriCond: Conditions^2 \rightarrow Conditions$ telle que pour tout couple (a, b) de $Conditions^2$, $TriCond(a, b)$ retourne une condition triée au sens et de la fonction Tri ci-dessus définie, si les termes a et b sont liés par la même conjonction de coordination. $TriCond(a, b)$ retourne a dans les autres cas. Considérons une condition β , nous noterons $Termes(\beta)$, l'ensemble des prédicats composant la condition β . Nous appellerons *squelette*, la fonction $Squelette: Expressions \rightarrow String$ qui, pour une expression quelconque a , $Squelette(a)$ retourne une chaîne obtenue en remplaçant les différents attributs des objets composant cette expression par leurs types sans toutefois agir sur les constantes.

Axiome 3.8 : Equivalence des Conditions

Considérons deux conditions a et b , nous dirons que a et b sont équivalentes si et seulement si :

(10) $squelette(a) = squelette(b)$;

(11) l'un des squelettes est dans l'ordre inverse de l'autre.

Axiome 3.9 : Raffinement ou Extension d'une Condition

(12) Considérons deux conditions a et b , nous dirons que a est un raffinement de b si et seulement si $:SubStr(squelette(a),squelette(b))$ est vrai.

(13) si (10) est satisfait, on dira que b est une extension de a .

Dans la suite, la notion de squelette sera étendue aux prédicats de base pris en compte dans les séquences.

III.3.3.2 SEQUENCE

Une séquence est une suite d'actions qui doivent être exécutées pour produire le résultat du métier. Une action est un prédicat, traduisant une opération ou traitement atomique permettant l'atteinte d'un objectif partiel ou final. Elle modifie l'état de l'environnement : c'est une expression à effet de bord. Deux types d'actions existent : les actions simples et celles complexes. Les actions simples sont celles constituées d'un seul prédicat de base. Les actions complexes sont celles qui font référence à un ensemble de prédicats de base ; à des règles métiers existantes via l'opérateur de choix déterministe par rapport à une condition ; à une combinaison de prédicats de base entre eux ; ou une combinaison de prédicats de base avec une garde. Les prédicats de base pris en compte dans notre spécification sont :

Tableau 3.2: liste des prédicats matérialisant les actions de la séquence

Liste des prédicats de bases	Description
l'ajout de[,...]* à ...	Matérialise le cumul des valeurs listées à gauche et à droite du mot clé « à ». Ce prédicat est adapté lors qu'on connaît explicitement les valeurs dont il

	faillie faire le cumul.
la somme de[,...]* dans]	Ce prédicat matérialise le cumul des valeurs des éléments d'au moins une liste.
le retrait [en (....[,...]*)] de[,...]* à	Matérialise la différence de valeurs, sous un format précis, entre les valeurs listées à gauche du mot clé « à » et celle qui se trouve à droite de ce même mot clé.
le produit de[,...]* par	Réalise le produits des valeurs située de part et d'autre du mot clé « par »
la division de[,...]* par	Matérialise la division. Les dividendes sont à gauche du mot clé « par » et le diviseur se trouve à droite de ce même mot clé.
l'édition de[,...]* sur]	Matérialise les opérations d'édition de l'opérande à gauche du mot clé « sur » celui à droite représente le support.
l'enregistrement de[,...]* dans]	Enregistrement de tous les éléments de même natures que ceux listés à gauche du mot clé « dans » se trouvant dans le contexte de la règle métier appelant de prédicat, dans l'élément se trouvant à droite du même mot clé.
la création de[,...]* dans	Matérialise l'ajout d'éléments ou l'allocation automatique d'une valeur dans la liste se trouvant à droite du mot clé « dans »
le rangement de[,...]* dans	Matérialise le stockage ou l'affectation des éléments se trouvant à gauche du mot clé « dans » dans l'élément de se

	trouvant à droite du même mot clé.
le suivant de[,...]* dans]	Matérialise le successeur des éléments listés à gauche du mot clé « dans ».
le précédent de[,...]* dans]	Matérialise le prédécesseur des éléments listés à gauche du mot clé « dans ».
la mise à jour de[,...]* par	Matérialise la modification des éléments se trouvant à gauche du mot clé « par » par celui qui se trouve à droite de ce même mot clé.
La construction de, type....[, zone (....[,...]*)] à partir de[,...]*	Construction d'un élément aux endroits indiqués dans la partie « zone », à partir d'une liste de valeur se trouvant à droite du mot clé « à partir de »
L'extraction dans (les la partie.....) de	Permet d'extraire une partie d'une valeur ou un élément d'une liste. La valeur ou l'élément extrait est stocké dans l'élément se trouvant à droite du mot clé « extraction dans ».

L'ensemble des prédicats de bases constitue le vocabulaire du domaine. Cette notion a été bien présentée dans [85]. Nous n'allons pas, par conséquent, y revenir. Toutefois, il est à noter que des extensions de ce dernier peuvent-être faites en fonction du langage naturel et le domaine, pour couvrir l'ensemble des actions atomiques de ce langage. Ces prédicats de base doivent être adaptés pour les cadrer avec l'expression du domaine sans toute fois modifier la finalité de l'action simple. Par exemple, on peut en lieu et place de « *la création de[,...] dans* », utiliser « *créer[,...] dans* » ; on peut en lieu et place de « *le retrait [en...[,...]] de[,...] à;* », utiliser « *la différence [en...[,...]] de[,...] à;* »

Dans le langage humain, plusieurs tournures du langage peuvent traduire ces prédicats. Il revient à l'ingénieur logiciel, assisté du cadre métier qui spécifie la règle

métier, d'identifier, des propos de ce cadre métier, le prédicat en question. Pour faciliter l'expression de la règle métier, nous avons, au risque de nous répéter, choisi des prédicats qui, dans une certaine mesure, peuvent traduire la même action. Ces prédicats, doivent mettre en scène certains objets déclarés dans le suffixe « agrégat ». À côté de ces prédicats de base, nous avons défini, un prédicat de choix déterministe notée $[...]\{...\}$, de la manière suivante :

$$[\%Obs_1, \%Obs_2, \dots, Obs_n]\{action_1(val_{Obs_{1,1}}, val_{Obs_{1,2}}, \dots, val_{Obs_{1,n}}), \dots, action_n(val_{Obs_{n,1}}, val_{Obs_{n,2}}, \dots, val_{Obs_{n,n}})\}$$

Où:

- $action_x$: représente une règle métier préalablement définie ;
- Obs_x : représente une propriété d'un objet déclaré par le suffixe « agrégat » ;
- $val_{Obs_{i,x}}$: représente la valeur de Obs_x pour $action_i$.

La règle métier $action_i$ sera exécutée si et seulement si les valeurs des paramètres $Obs_1, Obs_2, \dots, Obs_n$ correspondent à celles de $val_{Obs_{i,1}}, val_{Obs_{i,2}}, \dots, val_{Obs_{i,n}}$. Ce prédicat permet l'exécution, sous certaines conditions, d'une seule action parmi celles listées. Il permet aussi de représenter une action complexe à choix multiples. De manière formelle, la partie « séquence » sera représentée par :

$Sequence : (SimpleAction | ComplexAction)^+$ où :

- $SimpleAction$: est une action constituée d'un unique de base ;
- $ComplexAction$: $\langle SimpleAction | [...]\{...\} | [ComplexAction] \rangle$.

Lorsqu'on combine les prédicats de bases entre eux, le prédicat de base appelé se trouve entre parenthèses. Si par contre, on combine un prédicat de base à une garde, la garde est définie entre parenthèses et placée à la fin du prédicat.

Nous dénommons par *Sequences*, l'ensemble des séquences du processus d'affaire.

Axiome 3.10 : Equivalences des actions

Deux actions a et b sont dites équivalentes si et seulement si : $squelette(a) = squelette(b)$.

Nous définissons la fonction $processStr$ ainsi qu'il suit : $processStr : Sequences \times NameSpace_C \rightarrow SimpleType$ telle que $\forall (s, o) \in Sequences \times NameSpace_C, processStr(s, o)$ est l'ensemble des traitements que subit l'objet o dans la séquence s . $processStr(s, o)$ est aussi appelée la chaîne de modifications de l'objet o . $processStr(s, o)$ matérialise l'incidence des traitements de s sur l'objet o .

Axiome 3.11 : Equivalences entre Séquence ;

Considérons deux séquences s_1 et s_2 , nous dirons que s_1 et s_2 sont équivalentes si et seulement si : $\forall a \in NameSpace_C, processStr(s_1, a) = processStr(s_2, a)$

Axiome 3.12 : Extension, raffinement de Séquence

Considérons deux séquences s_1 et s_2 , nous dirons que s_1 est un raffinement de s_2 (ou s_2 est une extension de s_1) si et seulement si s_2 est contient le prédicat du choix déterministe et il existe une séquence s dans ce prédicat, telle que : $\forall a \in NameSpace_C, processStr(s_1, a) = processStr(s, a)$.

III.3.3.2.3 RESULTATS

Le résultat représente ce que l'on observe à la fin de l'activité du métier. Les indicateurs de résultats sont définis de la même manière que pour les attributs utilisables d'un objet qui sont utilisés dans la partie description d'une règle métier. D'une manière formelle, nous l'écrivons de la façon suivante :

Résultats : (object.views{ field %SimpleType[, field %SimpleType]})⁺ .*

Cependant, dans la spécification des résultats certains attributs des objets peuvent ne pas avoir la mention 'référence'. Lorsqu'on aimerait recueillir une liste d'objets ayant une structure particulière comme résultats d'une règle métier, la syntaxe est la suivante :

Résultats : object.views{[field%SimpleType]⁺, object fait référence à Object dans Results}

Si par contre, on aimerait retourner une liste d'objets quelconques en guise de résultat d'une règle métier, la syntaxe est la suivante :

Résultats : object.views{object fait référence à Object dans Results}

Toutefois, il est important de relever que lorsqu'on fait usage d'un élément de *Résultats* dans la partie contexte d'une règle métier, il doit obligatoirement avoir la mention « artifact » et tous ces attributs ont la « reference », d'une part ; la règle métier ayant cet objets comme résultat est implicitement appelé, d'autre part. L'extension de la clause « views » d'un tel objet n'est pas autorisée.

Nous dénommons par *ResultObjets*, l'ensemble des objets résultats de la règle métier et $card(ResultObject)$, le nombre d'objets de cet ensemble. Lorsque $card(ResultObject) = 1$, il n'est pas nécessaire de préfixer les attributs de cet objet par son nom dans la partie séquence de cette règle métier. Les résultats peuvent aussi être perçus comme étant le but à atteindre.

Axiome 3.13: Equivalence des Résultats

Considérons deux résultats *a* et *b*, nous notons *ResultObjets_a* (respectivement *ResultObjets_b*), l'ensemble des objets résultats de *a* (respectivement de *b*), nous dirons que *a* et *b* sont équivalents si et seulement si :

1. $card(ResultObjets_a) = card(ResultObjets_b)$ et,
2. $\forall o_1 \in ResultObjets_a \exists o_2 \in ResultObjets_b, Sig_a(o_1) = Sig_b(o_2)$.

Axiome 3.14 : Extension, Raffinement Résultats

Considérons deux résultats *a* et *b*, nous notons *ResultObjets_a* (respectivement *ResultObjets_b*) l'ensemble des objets résultats de *a* (respectivement de *b*), nous dirons que *a* est un raffinement de *b* si et seulement si :

1. $card(ResultObjets_a) > card(ResultObjets_b)$ et,
2. $\forall o_1 \in ResultObjets_a \exists o_2 \in ResultObjets_b, Sig_a(o_1) = Sig_b(o_2)$.

Après avoir présenté les différentes parties d'une règle métier, nous définissons de manière formelle la règle métier de la façon suivante :

$$nom_r\grave{e}gle = (Contexte_{nom_r\grave{e}gle}, description_{nom_r\grave{e}gle})$$

Où : - $Contexte_{nom_r\grave{e}gle}$: représente la partie contexte de la règle métier

- $description_{nom_r\grave{e}gle}$: représente la partie description de la règle métier

La représentation formelle des parties contexte et description d'une règle métier a été donnée dans les sections précédentes. Nous avons aussi défini un certain nombre de relations telles que l'Equivalence, le raffinement entre les concepts développés dans les sections ci-dessus. Dans la suite, nous allons utiliser ces relations pour développer les relations entre les règles métiers.

III.4. RELATIONS ENTRE REGLES METIERS

Nous avons, dans la section précédente, défini un ensemble de concepts et des relations entre ces différents concepts. Tous ces travaux avaient pour objectif de présenter de manière claire notre vision de la règle métier et préparer le terrain pour la définition des relations entre les différentes règles métiers. Les lignes qui suivent seront consacrées à ces relations. Par ailleurs, nous noterons l'équivalence entre deux concepts par : \equiv ; et le raffinement par : \cong

III.4.1 AXIOMES

Considérons deux règles métier α et β , nous désignons par $Contexte_{\alpha}$, $Contexte_{\beta}$ les contextes respectifs des règles métiers α et β et $Description_{\alpha} = (Garde_{\alpha}, Sequence_{\alpha}, Result_{\alpha})$, $Description_{\beta} = (Garde_{\beta}, Sequence_{\beta}, Result_{\beta})$ la partie description respectivement de la règle métier α et celle de β .

Axiome 3.15 : Equivalence de Règles Métiers

Nous dirons que α et β sont équivalentes si et seulement si :

$$\text{Contexte}_\alpha \equiv \text{Contexte}_\beta \text{ et } \begin{cases} \text{Sequence}_\alpha \equiv \text{Sequence}_\beta \\ \text{Garde}_\alpha \equiv \text{Garde}_\beta \\ \text{Result}_\alpha \equiv \text{Result}_\beta \end{cases}$$

Axiome 3.16: Extension, Raffinement de Règles Métiers

Nous dirons que α est un raffinement β ou que β est une extension de α si et seulement si :

$$\text{Contexte}_\alpha \cong \text{Contexte}_\beta \text{ et } \begin{cases} \text{Sequence}_\alpha \cong \text{Sequence}_\beta \\ \text{Garde}_\alpha \cong \text{Garde}_\beta \\ \text{Result}_\alpha \cong \text{Result}_\beta \end{cases}$$

Axiome 3.17 : Règle Métier Inconsistante

Nous dirons qu'une règle métier est inconsistante, si et seulement si :

1. il existe au moins un objet dans le contexte de cette règle métier qui ne soit pas utilisé dans la partie description de ladite règle métier ;
2. il existe au moins un attribut ou propriété d'un objet de son contexte qui ne soit pas utilisé dans la partie description de ladite règle métier ;

Axiome 3.18 : Règle Incomplète

Une règle métier est dite incomplète si et seulement si elle n'est pas inconsistante et il existe des propriétés ayant la mention « référence » qui ne figure pas dans les objets résultats de la même règle.

Pour rendre une règle incomplète, complète, il faut simplement supprimer les attributs ayant la mention « référence » non utilisés dans la partie séquence de la règle métier.

Axiome 3.19 : Importation Implicite de contexte,

Considérons les règles métiers α et β , telles que définies plus haut. Nous dirons que la règle métier α importe implicitement le contexte de β si et seulement si, $\text{Contexte}_\alpha \cap \text{Result}_\beta \neq \emptyset$.

Axiome 3.20: *Raffinement implicite*

Considérons les règles métiers α et β , si α importe implicitement le contexte de β alors α est un raffinement β .

IV. DEMARCHE

Les sections précédentes nous ont permis de présenter les concepts de bases inhérents à l'approche proposée ci-dessous. Ce processus d'ingénierie de besoins s'appuie sur trois étapes fondamentales : le recensement, la sélection et la construction du modèle de référence. L'étape de recensement des besoins a pour ambition de faire des interviews des parties prenantes et de recueillir leurs attentes ou leur avis sur le service qui leur est rendu. L'étape de sélection a pour vocation la transformation en fragment de connaissance, les besoins recensés d'une part, la recherche et des incongruités entre les différents fragments de connaissance, elle s'appuie sur un certain nombre d'axiomes élaborés dans travaux [9, 10, 11, 12, 13]. Ces axiomes seront présentés dans la suite. L'étape de transformation quant à elle se veut l'activité par laquelle le modèle de référence est construit. Il s'agit d'un formalisme mathématique. A cet effet, un ensemble d'axiomes permettant de guider la construction de ce modèle est défini. Une présentation exhaustive sera faite dans la suite.

Notre travail sera donc axé sur trois points principaux déduits des activités ci-dessus décrites. Le premier sera consacré au recensement des besoins ; le second sera consacré à la sélection des besoins et le dernier à la construction du modèle de référence.

IV.1 RECENSEMENT DES BESOINS

Cette étape fait usage de deux activités de base : le recensement des besoins de l'organisation et la collecte des exigences de qualité des clients de l'organisation.

IV.1.1 RECENSEMENT DES BESOINS DES CADRES METIERS

Les concepts présentés dans cette section sont issus des travaux [9,10]

IV.1.1.1 RECENSEMENT DES BESOINS DES UTILISATEURS

Le recensement des besoins des utilisateurs est l'activité qui vise, la collecte

des besoins des utilisateurs, ainsi que la validation de ces derniers. Pour ce faire, l'ingénieur logiciel doit utiliser un formulaire ayant la structuration du document de recensement des besoins tels que définis à la section 2 paragraphe 2.1. L'ingénieur logiciel devra au préalable faire valider les différents besoins les responsables compétents de l'organisation en présence des utilisateurs qui ont été consultés. Une fois validé, les ingénieurs logiciels doivent renseigner le champ domaine dans chaque fiche. Dans la suite, seule les besoins dont l'état est validé sont considérés. Nous désignerons par la suite attentes des utilisateurs ou besoins exprimés, les besoins dont l'état est validé.

IV.1.1.2 REGROUPE DES BESOINS PAR BUTS

Cette activité vise à catégoriser les attentes des utilisateurs en fonction des buts contenus dans leur formulaire de recensement des besoins. Aucune règle n'existe, l'ingénieur logiciel va se fier à sa compréhension des différents buts. Lorsque, la synonymie est décelée entre des buts, il est conseillé avant de les catégoriser de prendre l'avis de ceux qui les ont exprimés. Ce n'est que si ces derniers confirment qu'il s'agit des mêmes intentions que ce regroupement est validé. A la fin, les ingénieurs logiciels récapitulent les besoins par buts et les soumettent une fois de plus à la validation de l'organisation. Ce groupement permet de se rassurer que la sémantique des buts est la même pour tout le monde (utilisateurs et ingénieurs logiciels) et qu'il n'y a pas double emploi d'un but. Un même but est unique par domaine. C'est pourquoi le classement des besoins des utilisateurs se fait par buts. Cela nous permet de déceler des cas de doubles emplois du même but. Dans la suite, Ω représentera l'ensemble des attentes des utilisateurs de l'organisation. Soit b de Ω , $b.a$, représentera, la composante « a » du besoin b .

IV.1.2 RECENSEMENT DES ATTENTES DE QUALITES

IV.1.2.1 RECENSEMENT DES POINTS DE VUS

Cette activité a pour but de recueillir tous les points de vue des clients de l'organisation. Elle est continue et permet à une organisation, d'axer sa gestion sur la satisfaction de ses clients. Les points de vue recueillis doivent respecter la structuration précédente. Nous désignons par P_{views} , l'ensemble des points de vue

des clients de l'organisation.

IV.1.2.2 REGLES DE CONSTRUCTION DE L'ENSEMBLE DES FACTEURS DE QUALITES

Une fois collecté, les différents points de vue des clients de l'organisation, nous allons construire un MultiSet $m_Observations$ et, $Observations$ l'ensemble des observations des différents bénéficiaires du service rendu par un processus d'affaire, de la manière suivante :

$$m_Observations = \coprod_{pv=(d,cl,s,w) \in P_{views}} pv.w$$

$$Observations = \bigcup_{pv=(d,cl,s,w) \in P_{views}} pv.w$$

Où \coprod_* : représente une union d'ensemble avec répétition d'éléments. Cet opérateur permet de construire un multiset.

\bigcup_* : représente l'union ensembliste

$m_Observations$ est une collection des observations des différents bénéficiaires du service offert par le processus d'affaire, $Observations$ est un ensemble des observations des différents bénéficiaire de ce service. Nous définissons la fonction $f_{occ}:m_Observations \rightarrow N$ telle que quelque soit $obs \in m_Observations$, $f_{occ}(obs)$ retourne le nombre d'occurrences de obs dans $m_Observations$. Nous construisons l'ensemble des facteurs de Qualités ($QualityFactors$) ainsi qu'il suit :

$$QualityFactors = \bigcup_{obs \in m_Observations} (obs, f_{occ}(obs))$$

$f_{occ}(obs)$ représentera par la suite le poids de l'observation obs .

Axiome 3.19 : Points de vue divergents

Considérons deux points de vue Pv_1 et Pv_2 , $Pv_1 = (d, \langle cl, s \rangle, w) \in P_{views}$ et

$Pv_2 = (a, \langle cust, b \rangle, v) \in P_{views}$, Pv_1 et Pv_2 seront dits divergents si et seulement si $Pv_1.w \cap Pv_2.v = \emptyset$.

Axiome 3.20 : Points de vus similaires

Considérons deux points de vus Pv_1 et Pv_2 , $Pv_1 = (d, \langle cl, s \rangle, w) \in P_{views}$ et $Pv_2 = (a, \langle cust, b \rangle, v) \in P_{views}$, Pv_1 et Pv_2 seront dits similaires si et seulement si $Pv_1.w \cap Pv_2.v = Pv_1.w$ ou $Pv_1.w \cap Pv_2.v = Pv_2.v$.

Condition nécessaire 1 : Critères de Convergence

Considérons deux partitions de A_{fact} et B_{fact} de $QualityFactors$, A_{fact} est l'ensembles des facteurs de qualité pour lesquels le poids est égal à 1 ; B_{fact} est l'ensemble des facteurs de qualité pour lesquels le poids est différent de 1. Nous dirons que l'ensemble de facteurs de qualité $QualityFactors$ est convergent si et seulement si $card(B_{fact}) > card(A_{fact})$. $QualityFactors$ est dit divergent dans le cas contraire.

IV.2 SÉLECTION

L'étape de sélection ne concerne que les attentes des cadres métiers de l'organisation. Il à noter qu'aucun point de vue d'un client n'est à jeter tous les points de vue des clients sont par défaut retenus. C'est pourquoi la nécessité de les sélectionner ne s'impose pas [14].

Considérons un langage humain L (Français, Anglais, Allemand, etc.), nous définissons la fonction \approx de la manière suivante :

$$\approx: L \times L \rightarrow \{vrai, faux\}$$

$$(a, b) \in L \times L, a \approx b = \begin{cases} vrai, & \text{si } a \text{ et } b \text{ sont de sémantique voisins} \\ faux & \text{sinon} \end{cases}$$

Le symbole \neg sera utilisé pour exprimer la négation. Ainsi, $\neg b$ représentera une expression de sens opposé à celle de b.

Considérons a et b deux besoins de l'organisation, avec a et b éléments de Ω . nous avons :

$$a: (\langle v, \chi \rangle, \langle v, \langle \gamma, \psi \rangle \langle \lambda, \delta \rangle \rangle, \zeta) \text{ et } b: (\langle v', \chi' \rangle, \langle v', \langle \gamma', \psi' \rangle \langle \lambda', \delta' \rangle \rangle, \zeta')$$

- propriété 1 (besoins inconsistants) : a sera dit inconsistant si et seulement si $a.\gamma$ est inconsistant.

- propriété 2 (ambigüité des besoins) : a et b seront dits ambiguës si et seulement si $a.\gamma \approx b.\gamma', a.\lambda \neq b.\lambda'$ et $a.\psi \neq b.\psi'$.

- propriété 3 (similarité des besoins) : a et b seront dits similaires si et seulement si : $a.\gamma \approx b.\gamma'$ et $a.\lambda \neq b.\lambda'$.

- propriété 4 (contradiction entre les besoins) : a et b seront dits contradictoires si et seulement si au moins l'une des conditions suivantes est satisfaites :

$$(a) a.\gamma \approx \neg b.\gamma' \text{ et } a.\psi \approx b.\psi';$$

$$(b) a.\gamma \approx b.\gamma' \text{ et } a.\lambda \approx \neg b.\lambda'$$

- propriété 5 (identité des besoins) : a et b seront dits identiques si et seulement si

$$a.\gamma \approx b.\gamma' \text{ et } a.\psi \approx b.\psi' \text{ et } a.\lambda \approx \neg b.\lambda'.$$

- propriété 6 (Consistance des besoins) : a est dit consistant si et seulement si pour tout besoins b aucune propriété précédente n'est satisfaite.

Considérons ∇ comme ensemble des besoins retenus, et R ceux rejetés. Le processus de sélection des besoins consiste en :

Algorithme de sélection des besoins

1- $\nabla = \emptyset, R = \emptyset$

2- $\Omega \cup \nabla \cup R; \nabla = \emptyset, R = \emptyset$

3- Si a vérifie la propriété 1 : rejeter a ; $\Omega = \Omega - \{a\}$; $R = R \cup \{a\}$

- 4- S'il existe a et b vérifiant les propriétés 2 à 5 : rejeter a et b ; $\Omega = \Omega - \{a, b\}$;
 $R = R \cup \{a, b\}$
 - 5- S'il existe a et b vérifiant la propriété 6 : $\Omega = \Omega - \{a, b\}$; $\nabla = \nabla \cup \{a\}$
 - 6- Si a vérifie la propriété 6 : $\Omega = \Omega - \{a\}$; $\nabla = \nabla \cup \{a\}$
- Répéter les étapes 3 à 6 jusqu'à ce que $\Omega = \emptyset$.

Les éléments de R doivent faire l'objet de discussion avec les cadres métiers qui les ont exprimés. À la fin des discussions reprendre les étapes 2-6. Cette activité vise à déceler et à supprimer tout besoin avec un but « superflus » des utilisateurs. L'ingénieur logiciel doit se fier à sa compréhension des différentes intentions d'usage des utilisateurs. Lorsque l'une des propriétés 3-5 est vérifiée, il est conseillé de discuter avec les personnes qui ont émises ces intentions d'usage en question. Ce n'est que si ces derniers confirment qu'il s'agit des mêmes intentions que ce regroupement est validé. A la fin, les ingénieurs logiciels récapitulent les besoins par buts et les soumettent une fois de plus à la validation de l'organisation. Ce regroupement permet de se rassurer que la sémantique des buts est la même pour tout le monde (utilisateurs et ingénieurs logiciels) et qu'il n'y a pas double emploi d'un but. C'est pourquoi le classement des besoins des utilisateurs se fait par buts. Cela nous permet de déceler des cas de doubles emplois. Dans la suite, Ω représentera l'ensemble des besoins retenus.

IV.3 CONSTRUCTION DU MODELE

IV.3.1 TRANSFORMATION DES BESOINS

IV.3.1.1 TRANSFORMATION DES BESOINS EN FRAGMENT DE CONNAISSANCE

Considérons un besoin a de Ω , $a: (\langle v, \chi \rangle, \langle v, \langle \gamma, \psi \rangle \langle \lambda, \delta \rangle \rangle, \zeta)$, et C un fragment de connaissance, par définition, C peut s'écrire sous la forme $C = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$. Notre objectif est de pouvoir construire C à partir de a .

D'après la règle (1), nous avons :

$$f(a, \gamma) : (\vartheta, \Delta_b(\tau), \tau).$$

Règle 2 (translation des besoins) : $\vartheta = f(a, \gamma). \tau$; $\psi := a. \psi$; $\omega = f(a, \gamma)$;

$\lambda := a.\lambda$; $\delta := a.\delta$; et $v := a.v$.

Cette activité vise à transformer les besoins des utilisateurs en fragment de connaissance, intégrant le niveau d'importance de chaque besoin exprimé. Ce processus est repris pour chaque élément de Ω . Il s'achève lorsque tous les éléments de Ω ont été pris en compte. Les fragments de connaissance issus de la translation forment Σ l'ensemble des fragments de connaissance de l'organisation. Il est important de relever que le symbole " := " représente le processus de formalisation de l'argument se trouvant à la droite de ce symbole. C'est un processus à trois étapes : la première étape, optionnelle, est destinée à l'enrichissement et à l'adaptation des prédicats de bases avec ceux propres au domaine d'étude ; la deuxième étape, est destinée à la formalisation des règles métiers à partir des prédicats de base ; la dernière étape, est consacrée à la construction du fragment de connaissance à partir de la spécification de la règle métier et le besoin de l'utilisateur. Ainsi, si nous prenons comme exemple, λ' la règle obtenue après la formalisation de $a.\lambda$. Nous déduisons $\partial.\psi$ et $\partial.\lambda$ de la manière suivante :

$\partial.\psi = \lambda'.Contexte$ et $\partial.\lambda = \lambda'$.

IV.3.1.2 ÉLABORATION DU MODELE METIER

IV.3.1.2.1 « PROBLEM FRAMES »

Un « problem frame » [94] (ou schéma de problème) est un schéma qui définit de manière intuitive une classe de problèmes identifiée en terme de son contexte ; des caractéristiques de ses domaines ; et de ses interfaces et des besoins. Le système à développer est représenté par la "machine". Pour chaque schéma de problème, un diagramme est établi. Les rectangles simples dénotent les domaines de l'application (qui existent déjà), les rectangles avec une double barre dénotent les domaines "machine" qui est à réaliser, et les besoins sont notés par un ovale pointillé. Les lignes qui les relient représentent des interfaces, aussi appelées "phénomènes partagés". Jackson distingue les domaines "causaux" qui obéissent à certaines lois, les domaines lexicaux qui sont des représentations physiques des données, et les domaines "biddable" (qui émettent des commandes) qui sont des personnes. L'utilisation d'un schéma de problème consiste à instancier les domaines, les interfaces et les besoins.

Dans la suite, nous nous appuyerons sur les concepts de schéma de problème pour faire des démonstrations. Nous désignerons par machine, la machine sus évoquée. Cette machine transforme les fragments de connaissance (problème du monde réel) en exigences du futur système ; Σ représentera l'espace des fragments de connaissance (problèmes) de l'organisation et PF le « problem frame » de l'organisation, c'est l'ensemble des exigences du système futur de l'organisation. S^θ représentera l'ensemble de tous les objets de l'organisation du système d'information de l'organisation. Ces derniers sont mis en scène dans les règles de gestion.

En plus des axiomes des axiomes 3.19 et 3.20 sur les règles métiers, il est proposé un ensemble d'axiomes sur les buts. Les axiomes ci-dessous traitent beaucoup plus des buts.

IV.3.1.2.2 AXIOMES DE BASE POUR L'ELABORATION DU MODELE DE REFERENCE DE L'ORGANISATION

Considérons un fragment de connaissance b . b élément de Σ , $b = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$, une machine W au sens de [18] et un objet α de S^θ . Nous notons $\bar{W}(\alpha, \partial, \psi, \partial, \omega, \partial, \lambda, \partial, \delta)$, le traitement de l'objet α par la machine W , dans le contexte ∂, ψ , sous la règle ∂, λ et les contraintes ∂, δ , telles que le but ∂, ω est satisfait. Nous dirons d'une telle machine qu'elle reconnaît le besoin b . Nous construisons S^θ de la manière suivante : $S^\theta = \{a | a \in S^\theta, \bar{W}(a, \partial, \psi, \partial, \omega, \partial, \lambda, \partial, \delta)\}$. S^θ est l'ensemble des objets du système d'information de l'organisation pour lesquels l'attente ∂, ω est satisfaite sous la règle ∂, ω et la contrainte ∂, δ . Dans la suite, la notation S^θ par S^b ; le fragment de connaissance sera appelé besoin ou attente; PF_b représentera l'ensemble des exigences du système qui sont satisfaites dans le contexte ∂, ψ , sous la règle ∂, ω et la contrainte ∂, δ ; * est utilisé pour représenter une valeur indéterminée d'une composante.

IV.3.1.2.2.1 AXIOME 1 : COHERENCE DU FRAGMENT DE CONNAISSANCE

Considérons une attente b de Σ , nous dirons que $b = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$ est un besoin cohérent si et seulement si :

$$(1) -S^b \neq \emptyset;$$

(2) - $PF_b \subseteq PF$

IV.3.1.2.2.2 AXIOME 2 : CONCEPT DE SOUS-BESOINS

Soient deux besoins cohérents a et b de Σ et deux machines V et W telles que V reconnaisse a et W reconnaisse b ,

(3) nous dirons que $a = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$ est un sous-besoin de $b = (\langle \psi', \omega' \rangle, \langle \lambda', \delta' \rangle, \nu')$ ou que ω est un raffinement de ω' , si et seulement si : $S^a \subset S^b$ et $\exists e \in S^b$, tel que $\overline{W}(e, b)$ et l'exécution de $V(e, a)$ ne satisfait pas b .

(4) nous dirons que b est une généralisation de a si et seulement si b est un sous besoin de a . Nous notons ρ^a la généralisation de a . ω est appelé but spécifique de ω'

(5) nous dirons que a et b de Σ sont traçables si et seulement si, $b = \rho^a$ ou $a = \rho^b$.

(6) un besoin b est dit incomplet s'il n'a qu'un seul besoin fils. Les besoins incomplets doivent être arrangés par l'ajout des besoins fils. Si cela est impossible, alors fusionner le père et les fils comme un unique besoin.

(7) si $b = \rho^a$, alors λ est une règle d'ordonnancement des besoins fils de b .

IV.3.1.2.2.3 AXIOME 3 : BESOINS FUSIONNABLES

Soient deux besoins cohérents a et b de Σ , et trois machines M, V, W telles que M reconnait a et V reconnait b ,

(8) nous dirons que $a = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$ est fusionnables à $b = (\langle \psi', \omega' \rangle, \langle \lambda', \delta' \rangle, \nu')$ si et seulement si :

$\omega \approx \omega', \exists c = (\langle \psi'', \omega'' \rangle, \langle \lambda'', \delta'' \rangle, \nu'') \in \Sigma, \overline{W}(*, b) / s^b \cup s^a \subseteq s^c \quad e \in s^a, d \in s^b \rightarrow \overline{W}(e, c)$

et $\overline{W}(d,c)$ où : $\psi'' = \psi' \cup \psi$,

ω'' : est un but englobant ω' et ω .

λ'' : règle d'ordonnancement ;

$\delta'' = \delta' \cup \delta$,

v'' : est le plus grand niveau d'importance entre v' et v

IV.3.1.2.2.4 AXIOME 4 : IMPORTANCE DES BESOINS.

Soient deux besoins cohérents $a = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, v)$ et $b = (\langle \psi', \omega' \rangle, \langle \lambda', \delta' \rangle, v')$ de Σ avec v et v' des valeurs prises dans un ensemble ordonné. Nous dirons que a est plus important que b si et seulement : $\rho^a = \rho^b$ et $v > v'$.

IV.3.1.2.2.5 AXIOME 5 : BESOINS AMBIGÜES

deux besoins cohérents $a = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, v)$ et $b = (\langle \psi', \omega' \rangle, \langle \lambda', \delta' \rangle, v')$ de Σ , sont dits ambigües si seulement si : $\psi = \psi'$, $S^a = S^b$ et $\omega \neq \omega'$.

IV.3.1.2.2.6 AXIOME 6 : PARTITIONNEMENT DES BESOINS

Soient n besoins cohérents c_1, c_2, \dots, c_n de Σ , et un besoin cohérent $h = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, v)$ de Σ , tel que $S^h = \bigcup_{i=1}^n S^{c_i}$, $\bigcap_i S^{c_i} = \emptyset$ et $\psi^{c_i} = \psi$, alors le besoins h est partitionnable en n sous besoins h_1, h_2, \dots, h_n où $h_i = c_i$ et $\rho^{h_i} = \rho^{c_i}$.

IV.3.1.2.2.7 AXIOME 7 : BESOINS IDENTIQUES

Deux besoins cohérents $a = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, v)$ et $b = (\langle \psi', \omega' \rangle, \langle \lambda', \delta' \rangle, v')$ de Σ , seront dits identiques si et seulement si : $\psi = \psi', \omega \approx \omega'$ et $S^a = S^b$.

IV.3.2 CONSTRUCTION DES INDICATEURS DE PERFORMANCES

La construction des indicateurs de performance de performance nécessite que le modèle de référence orienté buts soit préalablement construit. Lorsque le dernier est élaboré, il faut déduire des fragments de connaissance, les tâches y afférentes.

IV.3.2.1 REGLE DE CONSTRUCTION DES TÂCHES

Les concepts de satisfaction des bénéficiaires des services, dans cette thèse ont

été élaborés par rapport aux tâches d'un processus d'affaire. De ce fait, les observateurs, concept de base de la prise en compte de la satisfaction des bénéficiaires de service, ont été associés aux tâches du processus d'affaire. A cet égard, nous avons jugé utile de proposer un pont permettant de passer de la représentation des fragments de connaissance à celle relative aux tâches sous le formalisme proposé dans [1].

Marcel Fouda et Roger Atsa Etoundi, dans [1], ont défini une tâche comme étant la donnée d'un contexte de réalisation ; d'un ensemble d'états (état initial de l'environnement², état final de l'environnement³) ; d'une fonction de transition entre les états; et d'un but à atteindre. La construction d'une tâche à partir d'un fragment de connaissance se fait ainsi qu'il suit:

- Considérer que l'environnement du processus d'affaire est l'ensemble des objets accessibles aux différentes règles métiers des fragments de connaissance pendant leur exécution ;
- La pré-condition est la garde de la règle métier du fragment de connaissance ;
- La post-condition est l'ensemble des objets résultats de la règle métier du fragment de connaissance ;
- La fonction de transition est constituée de la règle métier du fragment de connaissance ;
- Le goal est la composante « but » du fragment de connaissance.

Lorsque les tâches sont déduites des fragments de connaissance, elles forment un ensemble hiérarchique similaire du modèle métier. Pour faciliter la lisibilité des modèles, nous conseillons de préfixer les noms des tâches par "*Task_*". Cette structuration permet d'établir la dépendance entre tâche dans la réalisation du but global du processus d'affaire. La hiérarchisation des tâches permet aussi de déceler la manière dont les indicateurs de performances sont dépendants.

IV.3.2.2 DEMARCHE

Cette démarche vise à introduire les indicateurs de performances dans les processus d'affaire. Cependant, l'introduction des indicateurs est axée sur la

² Il est aussi appelé pré-condition

³ Il est aussi appelé post-condition

satisfaction du client. Notre démarche s'articule autour de quatre axes principaux : (i) construction de l'ensemble des observations des clients de l'organisation ; (ii) construction des facteurs de qualités ; (iii) la construction des ensembles d'observateurs associés à chaque observation ; (iv) construction des facteurs de performances. L'axe (i) (respectivement (ii)) a été suffisamment détaillé dans la section IV.1.2.1 (respectivement IV.1.2.2). De ce fait, ils ne seront plus abordés dans la suite.

IV.3.2.2.1 CONSTRUCTION DES OBSERVATEURS

La construction des observateurs revient à définir l'ensemble des variables à partir desquelles les indicateurs seront calculés. Considérons un facteur de qualité *fact* élément de *QualityFactor*, la construction de l'ensemble des observateurs est manuelle et à la charge de l'analyste et impose le respect des étapes ci-après :

(1) définir de bas en haut, les observateurs à partir desquels les indicateurs de performance de chaque tâche seront calculés. Toutes les tâches du processus d'affaires doivent être prises en compte. Si une tâche contribue à la satisfaction de plusieurs critères de qualité, elle aura autant d'observateurs que de critères de qualités;

(2) définir les fonctions d'évaluation f_{eval} pour chaque observateur d'une tâche élémentaire (les tâches se trouvant au dernier niveau de la hiérarchie des tâches). La fonction d'évaluation est la même pour tous les observateurs liés à un critère de qualité donné ;

(3) définir le seuil de satisfaction associé à *fact*. Le seuil de satisfaction est défini en fonction de l'observateur et de la satisfaction du critère de qualité auquel cet observateur est associé. Si une tâche contribue à la satisfaction de plusieurs critères de qualité, elle aura autant de seuil que d'observateurs ;

(4) en fonction de ce qu'on aimerait évaluer, associer aux objets métiers des observateurs permettant de renseigner sur ces informations.

Ce processus doit être repris pour l'ensemble des facteurs de qualité construit dans la section IV.1.2.2. Ce processus permet de construire la fonction $h_{map}: QualityFactors \rightarrow \mathcal{P}(Observers)$ qui pour tout facteur de qualité

$fact \in QualityFactors$, retourne l'ensemble des observateurs associés au facteur de qualité $fact$; et la fonction $h_{task}: Observators \rightarrow Tasks$ qui pour tout observateur, retourne la tâche à laquelle est associée l'observateur. Nous notons $Observer_{Qf}(t)$, l'observateur associé à la tâche t pour l'atteinte du facteur de qualité Qf .

IV.3.2.2.2 CONSTRUCTION DES FACTEURS DE PERFORMANCES

La construction des facteurs de performances doit se faire par les analystes logiciels assistés des cadres métiers de l'organisation. Cette construction s'appuie sur l'ensemble des observateurs ci-après définis auquel on ajoute le poids de chaque indicateur. Dans cette section les indicateurs étant des valeurs calculées, ils sont par conséquent déduits à partir des résultats de la section IV.3.2.1. La construction des facteurs de performance impose donc la définition des poids associés à chaque indicateur. Ainsi, étant donné un $fact = \langle value, weight \rangle \in QualityFactors$ un facteur de qualité, $elemTasks$ l'ensemble des tâches éléments du processus d'affaire et une tâche $t \in elemTasks$ (avec t une tâche terminale); la définition des poids associés aux indicateurs de performance impose le respect des étapes ci-après :

(1) si t est une tâche terminale et contribue à l'atteinte de l'observation $fact.value$ alors le facteur de performance $fp = \langle ind, w, task \rangle$ associé de t pour la satisfaction du facteur de qualité $fact$ est défini de la manière suivante :

$ind = f_{eval}(Observer(t)); w = fact.weight; ti = t;$

(2) si t ne contribue pas à l'atteinte du résultat, : $ind = f_{eval}(Observer(t)); w = 0; ti = t.$

(3) pour les tâches intermédiaires et la tâche racine de la hiérarchie des tâches du processus d'affaire, les facteurs de performance seront déterminés selon l'axiome 3.3.

(4) si un objet o possède un observateur de performance, construire l'indicateur de performance qui lui est associé suivant la règle énoncée dans IV.3.2.2.2-(1). Si l'objet n'a qu'un seul observateur, le poids n'est pas nécessaire.

IV.3.2.2.3 JONCTION DES PREOCCUPATIONS

Les fonctions d'évaluations, les indicateurs de performances, les fonctions de satisfaction ne sont pas modélisées dans le modèle de référence. On se limite à la représentation mathématique ou littérale de ces fonctions. L'ensemble des

descriptions des fonctions liées à la satisfaction des bénéficiaires des services constitue un axe de préoccupations dénommé par axe de préoccupation relatives à la qualité de service, tandis que l'ensemble des fragments de connaissances en constitue un autre dénommé par axe de préoccupations relatives aux besoins. Lier les deux axes de préoccupations consiste à définir les points d'échange de données entre les différents axes de préoccupations. Dans le cas d'espèce, les points de communication sont constitués d'objets/attributs de type « Observer ». Ces variables sont mises à jour par l'exécution de la règle métier et leur contenu est interprété par les fonctions d'évaluations. Cette jonction ne modifie pas la hiérarchie des fragments de connaissance. Elle permet d'enrichir les fragments de connaissances des observateurs de performance et les axiomes permettant de mettre à jour la valeur de l'observateur. Dans le cadre de l'élaboration du modèle des besoins d'un processus d'affaires, on se limite à la déclaration de ces observateurs. Cette déclaration se passe dans le contexte de la règle métier et plus spécifiquement dans les suffixes « agrégat » pour les observateurs de performance associés aux règles métiers ; et/ou « views », pour ce qui des observateurs attachés aux objets métiers. Toutefois, un observateur de performance associé à une règle métier peut aussi être déclaré entre les suffixes « agrégat » et « views », dans ce cas, il ne dispose pas du suffixe « views »[9].

IV.4 VERIFICATION DE LA COHERENCE DANS LE MODÈLE DE REFERENCE

Le modèle de référence est construit à partir des concepts provenant des domaines de recherche distincts. La mise en commun de ces concepts exige qu'on se rassure de la cohérence des différents concepts utilisés pour construire un modèle de référence en général, un fragment de connaissance en particulier. Le fragment de connaissance étant la base du modèle de référence, la validation dudit modèle consistera à la vérification de cohérence entre les composantes du fragment de connaissance et les attentes des bénéficiaires des services d'une part ; et les concepts de règles métiers d'autre part.

IV.4.1 COHERENCE ENTRE LES COMPOSANTES DU FRAGMENT DE

CONNAISSANCE ET LES ATTENTES DES BENEFICIAIRES DES SERVICES

L'impact de l'introduction des indicateurs de performance dans la modélisation des processus d'affaire se fait plus ressentir dans la validation des niveaux d'importances associés à chaque besoin exprimé par les cadres métiers de l'organisation. Ainsi, étant donné t une tâche, f_q un facteur de qualité, un besoin exprimé b . Nous supposons que b est associé à la tâche t . Considérons un facteur de performance f_p associé aussi à t . Nous définissons la fonction majorité $maj: Tasks \rightarrow boolean$ qui est telle que pour toute tâche t $maj(t)$ retourne 'true' si le nombre facteur de performance associé à la tâche t et ayant un poids non nul est supérieur à ceux ayant un poids nul ; et 'false' dans le cas contraire.

Axiome 3.21 : Besoin Important par rapport à un Seuil

Considérons un besoin b et un seuil del , nous dirons que le besoin b est important si son niveau d'importance est supérieur au seuil del . del est appelé seuil d'importance des besoins exprimés.

Axiome 3.22 : Cohérence entre Besoins et Indicateurs de Performance

Considérons un besoin exprimé b , une tâche t associée au besoin b et un seuil d'importance del , nous dirons que le niveau d'importance ni d'un besoin exprimé b est :

(7) cohérent aux indicateurs de performances si et seulement si $maj(t)$ a la valeur 'true' et $ni > del$.

(8) incohérent dans les cas contraires ;

(9) les facteurs de performances associés à t sont aussi appelés facteurs d'importances associés au besoin exprimé b . Il en est de même pour les autres concepts ci-dessus évoqués.

Dans un cas d'incohérence entre le niveau d'importance d'un besoin et les indicateurs de performance associés à ce besoin, l'analyste logiciel doit se référer aux cadres métier pour ajuster le niveau d'importance du besoin exprimé.

Définition (3.17) : Besoins Critiques

(1) Considérons un besoin exprimé b et une tâche t associée au besoin exprimé b , nous dirons que b est un besoin critique si et seulement si son niveau d'importance est cohérent aux indicateurs de performance associés à t .

(2) L'ensemble des besoins critiques forment le modèle de référence critique d'un processus d'affaire.

Dans les travaux de [11], il a été défini une approche de spécification des règles métier d'un processus d'affaire, compte tenu de ce qui précède, il est nécessaire que les règles métiers prennent en charge les concepts le concept d'observateurs de qualité. Pour cela, la contrainte suivante a été émise :

Contrainte : Règle Métier

(1) les Observateurs doivent être déclarés dans le contexte de chaque règle métier à travers le type *Observer*. Ils représentent un couple (*variable, weight*) ;

(2) les observateurs déclarés dans le contexte d'une règle doivent être obligatoirement mis à jour lors de l'exécution de la règle ;

(3) une règle d'évaluation des indicateurs doit être définie.

IV.4.2. COHERENCE ENTRE LES CONCEPTS DE REGLE METIER ET LES COMPOSANTES DU FRAGMENT DE CONNAISSANCE.

Dans la suite la terminologie incidence renverra à la cohérence entre les concepts.

Définition 3.18: Règle d'ordonnance

Une règle métier α est une règle d'ordonnancement si $Sequence_{\alpha}$ contient le prédicat de choix déterministe.

Dans [9], nous avons présenté une démarche orientée buts pour la définition d'un modèle de référence d'un processus d'affaire, avec prise en compte de leur niveau d'importance et des contraintes inhérentes à ces besoins. Le niveau d'importance d'un but est le crédit que l'utilisateur accorde à ce but. Les contraintes sont des besoins non fonctionnels relatifs à ce que ce but doit satisfaire. La démarche qui a

été proposée dans [9], tourne autour de quatre activités principales : le recensement des besoins des utilisateurs, la sélection des différents buts, transformation des besoins en fragment de connaissance et en dernier ressort, l'élaboration du modèle de référence. Nous avons montré de manière formelle que cette démarche permettra de décrire de manière exhaustive un processus d'affaire. Pour ce faire, nous avons donné un formalisme permettant de modéliser le besoin d'un cadre métier et avons déduit, à partir des travaux de [3], une représentation formelle de ce que nous avons appelé fragment de connaissance ou besoin exprimé. Un besoin exprimé ou fragment de connaissance a été défini de la manière suivante : $\partial = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$ où ∂ est le nom du fragment de connaissance, $\langle \psi, \omega \rangle$ la définition du but ω dans le contexte ψ , $\langle \lambda, \delta \rangle$ la représentation de la règle de gestion λ sous les contraintes δ .

Considérons $a = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$ et $\partial = (\langle \psi', \omega' \rangle, \langle \lambda', \delta' \rangle, \nu')$ deux besoins exprimés, S^a est l'ensemble des objets du système d'information de l'organisation pour lesquels l'attente $a.\omega$ est satisfaite sous la règle $a.\lambda$ et la contrainte $a.\delta$ [9]. Il en est de même pour S^b .

Les concepts d'identité de besoins, sous-besoins, partitionnement de besoins ont été clairement définis dans [9]. Cette définition était orientée exclusivement sur l'intention d'usage. Nous n'allons pas y revenir. Nous allons utiliser les caractéristiques de ces derniers pour montrer l'incidence d'une règle métier sur le modèle de référence d'un processus d'affaire d'une organisation.

Incidence 1 : $S^a = NameSpace_{a.\lambda}$ par définition.

Incidence 2 : $a.\lambda \equiv b.\lambda'$ par conséquent a et b sont identiques

Preuve : nous supposons $a.\lambda \equiv b.\lambda'$ et montrons que $\psi = \psi', \omega \approx \omega'$ et $S^a = S^b$.

Considérons deux besoins a et b , dans les conditions du paragraphe ci-dessus, nous supposons que $a.\lambda \equiv b.\lambda'$, par définition de $a.\lambda \equiv b.\lambda'$, nous avons :

a) $Contexte_{a,\lambda} \equiv Contexte_{b,\lambda'}$ c'est-à-dire $NameSpace_{a,\lambda} \equiv NameSpace_{b,\lambda'}$ et $Keywords_{a,\lambda} \cap Keywords_{b,\lambda'} \neq \emptyset$, par suite, $S^a = S^b$;

b) $Results_{a,\lambda} \equiv Results_{b,\lambda'}$, par définition, $\omega \approx \omega'$.

c) A partir de a) et b) nous déduisons que forcément $\psi = \psi'$.

Incidence 3 : $a.\lambda \cong b.\lambda'$, a est sous besoin de b .

Preuve : supposons que $a.\lambda \cong b.\lambda'$ et montrons que $S^a \subset S^b$ et $= \rho^a$.
 Considérons deux besoins a et b , dans les conditions du paragraphe ci-dessus, nous supposons que $a.\lambda \cong b.\lambda'$; par définition de $a.\lambda \cong b.\lambda'$ nous avons :

a) $Contexte_{a,\lambda} \cong Contexte_{b,\lambda'}$ c'est-à-dire $NameSpace_{a,\lambda} \cong NameSpace_{b,\lambda'}$ et $Keywords_{a,\lambda} \cap Keywords_{b,\lambda'} \neq \emptyset$. $NameSpace_{a,\lambda} \cong NameSpace_{b,\lambda'}$ par définition nous déduisons que $S^a \subset S^b$;

b) $Séquence_{a,\lambda} \cong Séquence_{b,\lambda'}$, par définition la $a.\lambda$ est référencée dans $b.\lambda'$ donc $b = \rho^a$.

Incidence 4 (définition de besoin partitionnable): nous dirons que a est partitionnable si et seulement si $Séquence_{a,\lambda}$ contient le prédicat de choix déterministe. Le nombre de règles métiers référencées dans le prédicat de choix déterministe constitue le nombre de partitions du besoin a .

Incidence 5 (définition besoin ambiguë):

Nous dirons qu'un besoin exprimé a est ambiguë si et seulement si $a.\psi$ ne se trouve pas dans la liste des domaines listés dans le champ d'exploitation de la règle métier.

Incidence 5 (définition besoin composables):

Nous dirons que deux besoins exprimés a et b sont composables si et seulement si $a.\lambda$ est composables à $b.\lambda'$.

Ces incidences permettent de compléter la définition des concepts évoqués dans [1] qui restées superficielles.

La **figure 17** ci-dessous présente une vue synoptique du processus d'élaboration du modèle de référence.

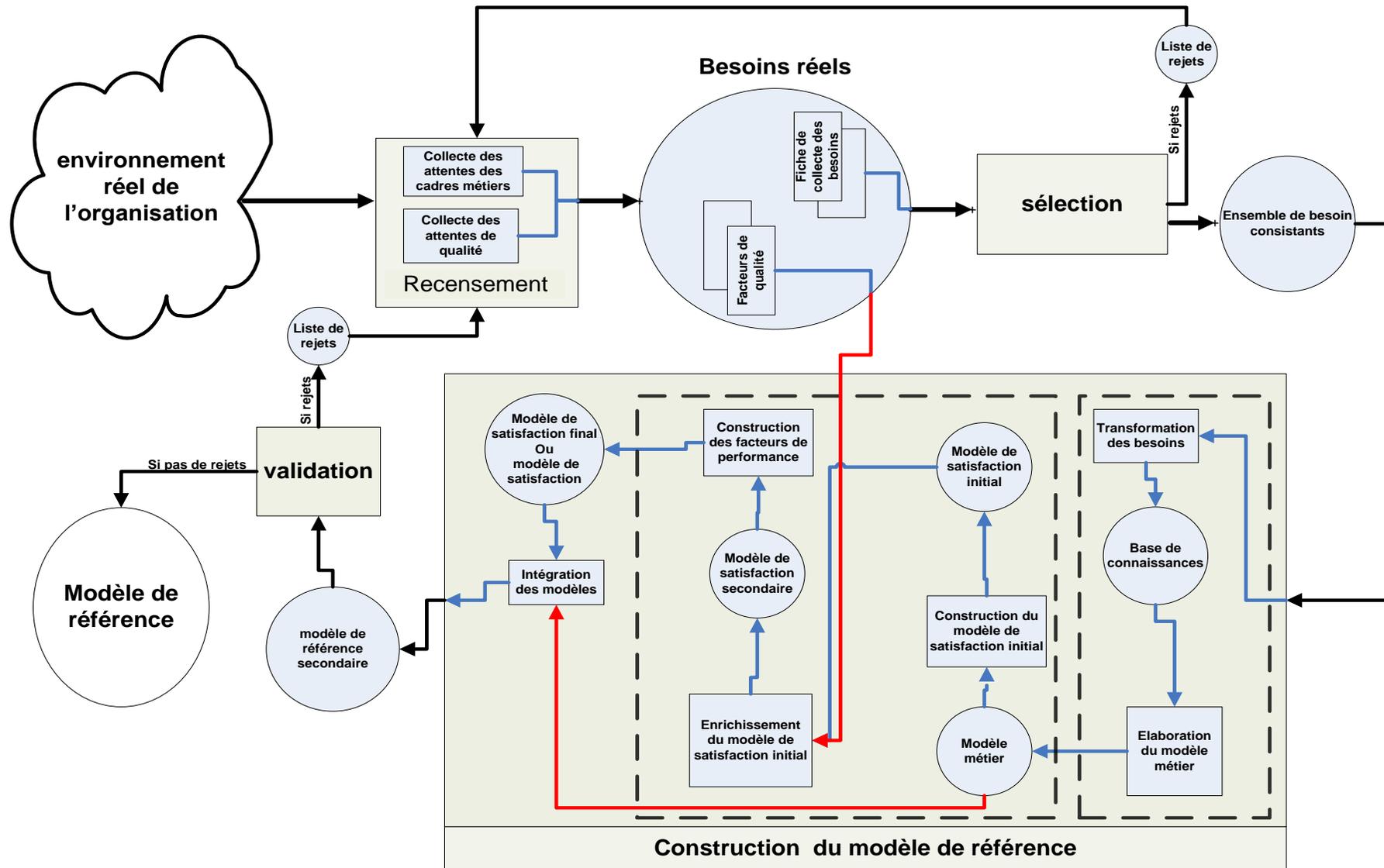


Figure 17: Illustration du processus d'ingénierie permettant la construction du modèle de référence

IV. 5. CYCLE DE VIE DU MODÈLE

Le cycle de vie du modèle de référence est une synthèse des cycles de vie du modèle métier et du modèle de satisfaction. Il concerne les étapes que doit subir ce modèle jusqu'à sa désuétude. Dans le cas de cette étude, il a été considéré cinq étapes majeures dénommées : modèle métier, évaluations, réajustements métiers, réajustements qualités et modèle de satisfaction. L'état initial du modèle de référence est constitué de deux étapes à savoir : le modèle métier et le modèle de satisfaction. Il représente la configuration du modèle de référence juste après sa construction. Les autres sont les étapes intermédiaires. L'orientation des flèches pleines indiquent le sens de successions des phases dans le cycle de vie du modèle de référence. Les flèches interrompues traduisent la dépendance entre la phase cible et la phase de base. La phase cible est celle qui se trouve du côté pointu de la flèche tandis que la phase de base est celle qui se trouve au bout opposé. Elle indique qu'en cas de modification concurrente, la phase cible doit attendre que la phase de base prenne en compte toutes les modifications avant que celle-ci ne s'active. La figure ci-dessous présente le cycle de vie du modèle de référence.

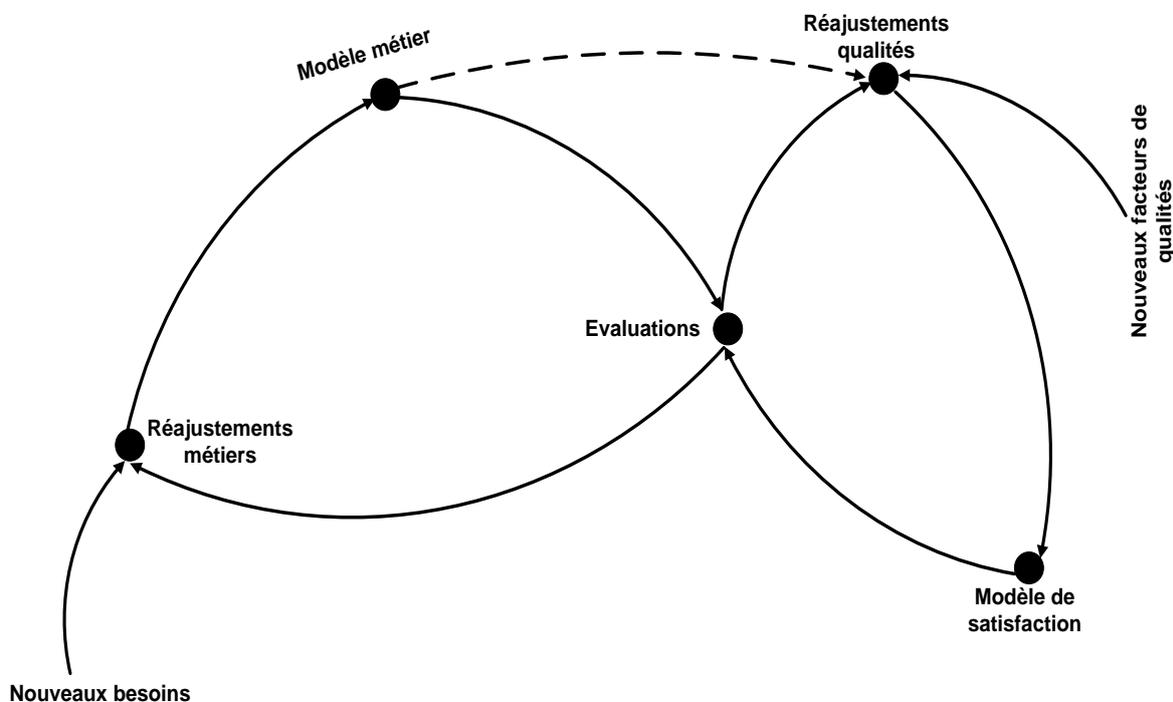


Figure 18 : Cycle de vie du modèle de référence

IV. 6. CONCLUSION

Ce chapitre nous a permis de présenter le processus d'ingénierie de besoins dédiés à la production des modèles de besoins conformes au formalisme de Farida Semmak et Joël Brunet. C'est une approche formelle, elle n'est donc pas dédiée de bout en bout à l'humain mais plutôt à des plateformes de spécifications. En réalité seule l'étape de recensement se fait en collaboration avec l'analyste du système informatique. La plateforme de spécification devrait elle-même construire le modèle de r. Ce travail n'a pas abordé le volet implémentation de ladite plateforme.

Néanmoins, nous avons observé que ce travail pouvait à plus d'un titre être utilisé dans d'autres contextes. Nous avons jugé utile de présenter ces différents bouchés. Un point a été consacré à ce sujet dans la section suivante.

CHAPITRE IV : DYNAMIQUE DU MODELE DE REFERENCE DANS UNE ORGANISATION

Résumé :

Un besoin est par essence une information dynamique. C'est-à-dire une information qui évolue au fil du temps. Ce qui induit, sur la base de cette dynamique, des réajustements/remplacements permanents des systèmes informatiques soutenant l'activité des organisations. L'incidence directe de cela, dans la modélisation des processus d'affaire, est l'obsolescence rapide et récurrente des spécifications des processus d'affaire dans les organisations. C'est-à-dire une information qui évolue au fil du temps. Il a été présenté dans le chapitre précédent

une démarche permettant de capter et de construire le modèle de référence d'une organisation. Ce modèle intègre en son sein la satisfaction des clients de cette organisation. Il est construit à partir de fragments de connaissances. Les fragments de connaissances sont issus des besoins exprimés par ladite organisation. De ce fait, le concept de besoins est intimement lié à celui de fragment de connaissance. Notre travail dans ce chapitre consiste donc à présenter la dynamique du besoin et indécemment de celle du fragment de connaissance qui lui est associé dans notre modèle de référence ci-dessus mentionné. Cette évolution vise l'amélioration de la qualité de service interne de l'organisation. Pour ce faire il a été d'abord étudié la dynamique de cette qualité de service externe au sein de l'organisation. En second lieu, il a été étudié la dynamique du besoin/fragment de connaissance par rapport à une qualité de service désirée. A la fin, la dynamique du modèle de référence résultant des besoins des cadres métiers et des attentes des bénéficiaires de service de cette organisation a été déduite. A l'issue de cette étude, il a été identifié trois points fixes. Le premier a trait à la qualité de services de perçue par les bénéficiaires de service ; le second se pour lequel la plupart des clients et le staff dirigeant de l'organisation sont satisfaits. Ces points fixes matérialisent la limite de la dynamique du modèle de référence. Ils permettent d'établir que le modèle de référence n'a pas une évolution infinie. Impliquant par là même que lorsqu'une organisation a axé son processus d'affaire sur la satisfaction des ses clients(ou la gestion axée sur les résultats), l'évolution de son processus d'affaires admet une limites au delà de laquelle aucune amélioration ne peut plus être faite.

I. INTRODUCTION

Dans le chapitre précédent, il a été présenté un modèle de référence ayant pour objectif la représentation de l'ensemble des besoins d'une organisation, captés à un moment donné. La démarche de construction de ce modèle a aussi été présentée. Cependant, aucune indication n'a été donnée quant à la gestion de l'évolution du modèle de référence au sein d'une organisation. En effet, d'après les travaux de [120], les besoins d'une organisation évoluent très rapidement. De ce fait, le modèle

de référence résultant est lui aussi en perpétuel évolution. Ainsi, on ne saurait parler de modèle de référence sans envisager son évolution. Cette préoccupation a été scindée en deux sous cas : le premier sous cas est consacré à la gestion de l'enrichissement d'une attente d'un utilisateur, et le second sous cas à la gestion de l'enrichissement du modèle de référence.

Ce chapitre s'articulera, autour de deux points principaux : la dynamique d'un fragment de connaissance et celle du modèle de référence. Une conclusion partielle terminera ce chapitre.

II. COMPOSANTS METIERS ORIENTES ASPECTS (CMOA)

Un composant métier orienté aspect est une implémentation abstraite d'un modèle de référence. C'est un ensemble d'aspects métiers. Tout comme un CMOA, un aspect métier est une implémentation abstraite du fragment de connaissance. C'est un module constitué de deux unités modulaires d'intégration [95], tandis qu'une unité modulaire d'intégration implémente une composante d'un fragment de connaissance. Etant donné qu'un fragment de connaissance est une spécification de l'attente d'un cadre métier, il est aisément admis que les constituants d'un fragment de connaissance collaborent⁴ dans la satisfaction des attentes des utilisateurs d'une organisation et incidemment, celle des clients de cette organisation. De manière formelle, une unité modulaire d'intégration, dans [95], sera défini ainsi qu'il suit :

$$Unity_name = \langle port_com, action \rangle$$

où : *action* est soit la spécification d'une règle métier, soit la spécification d'une condition ou d'une fonction ; soit la spécification d'une interface ;

port_com décrit port de communication de l'unité modulaire d'intégration. C'est par ce port que l'unité modulaire d'intégration est composée avec d'autres dans l'optique de construire un aspect plus grand appelé aspect.

La définition des interactions entre les constituants d'un fragment de connaissance

⁴ Interactions

constitue implicitement, la spécification d'une solution à l'attente d'un cadre métier. Pour matérialiser les interactions entre les différentes composantes d'un fragment de connaissance, deux opérateurs ont été proposés dans [95] : le premier est l'intégration verticale noté Δ_v et le deuxième, l'intégration horizontale noté Δ_h . Considérons deux unités modulaires d'intégrations U_1 et U_2 , $U_1\Delta_v U_2$ si et seulement si le flux de données en entrée de U_1 prend ses valeurs dans les résultats intermédiaires de U_2 ; et $U_1\Delta_h U_2$ si et seulement si U_2 utilise les résultats de U_1 pendant son exécution ou encore que l'exécution de U_2 est dépendant de celle de U_1 [95]. Pour la manipulation des construits d'unités modulaires d'intégration deux opérateurs supplémentaires ont été ajoutés : $+$ qui représente l'opérateur de prise en compte d'une unité modulaire d'intégration et l'opérateur \triangleright qui représente quant à lui l'opérateur de substitution d'une unité modulaire d'intégration dans un construit d'unités modulaires d'intégration et d'un aspect initial. Cependant, les unités modulaires d'intégration ne sont pas composées n'importe comment. Les auteurs recommandent que toute construction parte d'un construit d'une règle métier et des indicateurs de performances. Ce construit se dénomme aspect initial [95]. C'est dans cet aspect initial que sont uniquement utilisés Δ_h et Δ_v . Un aspect initial quelconque *smallAspect* sera défini ainsi qu'il suit $smallAspect = \langle Unity_1[,Unity_2]^+, Rule, \{port_com\}^+ \rangle$. Tout autre construit b d'unités modulaires d'intégration et d'un aspect initial sera représenté de la manière suivante : $b = b_0 + \sum_{i=1}^m b_i$, $m \in \mathbb{N}^*$ où : b_0 est l'aspect initial et les b_i ($i \in \mathbb{N}^*, i \leq m$) représentent les préoccupations qui ont été tour à tour prises en compte dans l'aspect en question. Il est cependant important de relever que ce travail s'appuie sur les concepts des hyperespaces, utilisés dans le cadre du développement par séparation des préoccupations (Approches par Aspect, Approches par Hyperespace, etc.). Les fonctions $dimtype:Unities \rightarrow String$ telles que pour toute unité modulaire d'intégration u , $dimtype(u)$ retourne une chaîne de caractères représentant la dimension de préoccupations d'appartenance de u [95], *Unities* est l'ensemble des unités modulaires d'intégration; et l'opérateur de choix alternatifs \square définit de la

manière suivante : $[\]: BCAspects \times Aspects \rightarrow BCAspects$ où $BCAspects$ est l'ensemble des composants métiers, $Aspects$ l'ensemble des aspects métiers. Ces concepts seront utilisés dans l'illustration de la dynamique du modèle de référence et celle du fragment de connaissance. Dans la suite, BE désignera l'univers des besoins, FC l'univers des fragments de connaissance, Mod l'univers des modèles, $P(FC) = \{x / x \subseteq FC\}$ l'ensemble des parties de FC , et ε ($\varepsilon \in [0,1]$) le pas de convergence.

III. DYNAMIQUE DE LA QUALITE DE SERVICE

Le mot «Qualité» est de plus en plus utilisé dans les entreprises, que ce soit dans le secteur alimentaire, industriel ou dans le monde informatique. Elle se décline en un ensemble de caractéristiques à satisfaire des besoins exprimés et implicites (ISO 8402-94). Dans ce contexte, la qualité de service renvoie à la qualité de service externe. Elle correspond à la satisfaction des clients [4]. Il s'agit de fournir un produit ou des services conformes aux attentes des clients afin de les fidéliser. Etudier la dynamique de la qualité de service revient à étudier la dynamique de l'ensemble des critères à partir desquels la qualité de service est appréciée. Pour ce faire, nous construisons les fonctions suivantes :

- Constructeur : $f_{be}: QoS \rightarrow BE$, qui pour tout $x \in QoS$, $f_{be}(x)$ renvoie l'ensemble des besoins de l'organisation contribuant à l'atteinte de la qualité de service x . Il faut relever que les besoins de l'organisation prennent en compte les attentes de bénéficiaires de services ;
- Fonction d'amélioration de la qualité de service : $f_{QoS}: QoS \rightarrow QoS$ telle que pour $x \in QoS$, $x \subseteq f_{QoS}(x)$ où QoS est un ensemble de parties de l'ensemble de critères de qualités. Les travaux contenus dans [4] montrent comment construire cet ensemble.

Lemme : l'ensemble des critères de qualité est un ensemble fini (c.f ISO 8402-94).

Définition 4.1: Point de stabilité pour l'évolution de la qualité de service

Soit $x \in QoS$ un ensemble de critères de qualité, nous dirons que x est un point de stabilité pour l'évolution de la qualité de service si et seulement si $f_{QoS}(x) \subseteq x$.

Dans [4], les auteurs proposent une démarche de construction des critères de qualité à partir des points de vus recueillis auprès des clients de cette organisation. Le modèle suivant est une illustration de la dynamique de la qualité de service au sein d'une organisation.

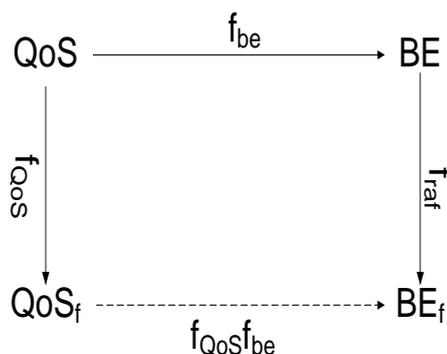


Figure 19 : Dynamique de la Qualité de Service

Dans l'illustration précédente, QoS est l'ensemble des critères de qualité ; BE l'ensemble des besoins de l'organisation dont la satisfaction conduit à l'atteinte d'une qualité de service souhaitée ; les flèches en pointillé représentent les fonctions déduites ; BE_f représente l'ensemble des besoins de l'organisation dont la satisfaction permet d'honorer aux des bénéficiaires de service ; QoS_f est le plus grand ensemble de facteurs de qualité des bénéficiaires de service. QoS_f est appelé point de stabilité de la qualité de service.

Preuve : Montrons que f_{QoS} admet un point de stabilité. Soit $QualityFactors$ l'ensemble des facteurs de qualités, par définition $QoS = P(QualityFactors)$ est l'ensemble des parties de $QualityFactors$. Les facteurs de qualités sont construits à partir d'un ensemble finis de points vus, d'après [4]. De plus, chaque point de vus est constitué d'un ensemble finis d'observations faites. Une observation est le résultat de l'évaluation de la satisfaction d'un critère de qualité. Comme l'union finie d'ensembles finis est un ensemble fini, alors f_{QoS} admet un plus grand élément ; désignons par QoS_f cet élément. De ce fait, quelque soit $x \in QoS$, $f_{QoS}(x) \subseteq QoS_f$.

Dans la suite le point de stabilité de la qualité de service sera appelé point fixe de l'évolution de la qualité de service.

IV. DYNAMIQUE D'UN FRAGMENT DE CONNAISSANCE

Un fragment de connaissance est une représentation semi-formelle d'un besoin d'une organisation. Il décrit une connaissance nécessaire à l'exécution d'une activité atomique d'un processus d'affaire d'une organisation. Il intègre dans sa représentation son environnement d'exécution, les indicateurs pour évaluer la qualité de service interne ou celle externe. Il a été pensé que les organisations affinent leurs besoins dans l'optique de faire aligner ces derniers au modèle résultant de la représentation de ces besoins. Ce qui forcément améliore la qualité de services de leur processus d'affaire. Ainsi, la dynamique d'un fragment de connaissance est étroitement liée à l'amélioration de la qualité de service dans une organisation. La section suivante sera dédiée à la présentation de la dynamique d'un fragment de connaissance.

IV.1 DYNAMIQUE

Un fragment de connaissance est un construit d'une attente d'une organisation. Pour ce faire, un fragment de connaissance ne pourra évoluer que si le besoin à partir duquel il est construit a été évolué. Parler de la dynamique d'un fragment de connaissance revient implicitement à parler de la dynamique du besoin auquel il est associé. Ces deux concepts sont indissociables dans ces travaux. Dans la suite, l'expression « évolution du besoin » pour être utilisée en lieu et place de celle d'« évolution du fragment de connaissance » ; tout comme on utilisera « connaissance » en lieu et place de « fragment de connaissance ». Comme il est dit dans [120] que les besoins des organisations sont instables du fait de leur évolution rapide et qu'une entreprise qui ne tient pas compte de la satisfaction de sa clientèle dans son processus d'affaire est amenée à disparaître [99] ; il a été supposé que c'est le souci de demeurer compétitives et en adéquation avec les attentes de sa clientèle que les organisations sont amenées à revoir leur processus d'affaire. Ainsi,

ne sera considérée comme évolution que tout enrichissement du besoin permettant d'améliorer la qualité de service externe de l'organisation. Ce processus d'enrichissement est un processus permettant à une organisation de mieux affiner ses besoins. La qualité de service interne, au niveau d'une tâche, est perçue comme étant le rendement de la connaissance utilisée par cette tâche lors de son exécution. Il va de soit que toute tâche a un rendement maximal. Compte tenu de cela, il est clairement admis que le rendement de la connaissance admet une valeur maximale. Lorsque cette dernière est atteinte, par hypothèse, tout enrichissement de cette connaissance n'améliore plus le rendement de cette connaissance. La connaissance en question est considérée comme point fixe de l'évolution du besoin. L'amélioration du rendement de la connaissance consiste à calculer la différence des deux rendements. C'est-à-dire, le rendement de la connaissance enrichie moins le rendement de cette connaissance avant son enrichissement. Ainsi, il y aura amélioration du rendement si cette valeur est supérieure à une valeur ε arbitrairement choisie. C'est sur la base de ces hypothèses que le raisonnement ci-après a été construit.

Pour étudier la dynamique d'un fragment de connaissance, il est construit les fonctions de base suivantes :

- (1). Le constructeur : $f_{FC} : BE \rightarrow FC$ qui pour tout besoin r , $f_{FC}(r)$ est le fragment de connaissance associé au besoin r ;
- (2). La fonction de satisfaction : $sat : FC \rightarrow [0,1]$ telle que pour tout fragment de connaissance fc , $sat(fc)$ retourne le rendement d'une tâche faisant usage de la connaissance fc ;
- (3). Les fonctions d'extensions :
 - La fonction d'extension du besoin : $f_{raf} : BE \rightarrow BE$, telle que pour tout besoin $\forall r \in BE, f_{raf}(r) = \begin{cases} r', & \text{si } \exists r' \in BE / r' = f_{evo}^+(r) \text{ et } sat(r') - sat(r) \geq \varepsilon, \\ r, & \text{sinon} \end{cases}$,
 f_{raf} est aussi une fonction d'évolution du besoin. nous noterons f_{raf}^+ ,
 la fonction $\underbrace{f_{raf} \circ f_{raf} \circ \dots \circ f_{raf}}_{n \text{ fois}}, (n \in \mathbb{N}^*)$;

- La fonction d'extension du fragment de connaissance : est la composition du constructeur d'un fragment de connaissance et de la fonction d'extension d'un besoin. Il est noté : $f_{FC} \circ f_{raf}$.

(4). La fonction de base : Cette fonction est utilisée par la fonction d'extension du besoin pour construire un besoin enrichi. Elle est définie ainsi qu'il suit :

- $f_{evo} : BE \rightarrow BE$ qui pour tout besoin r , $f_{evo}(r)$ représente le besoin obtenu après un enrichissement du besoin r . f_{evo} est une fonction d'enrichissement des besoins. nous noterons f_{evo}^+ , la fonction

$$\underbrace{f_{evo} \circ f_{evo} \circ \dots \circ f_{evo}}_{n \text{ fois}}, (n \in \mathbb{N}^*);$$

Considérons un pas de convergence ε dans les conditions précédentes, nous dirons :

- (1). qu'un besoin enrichi $f_{raf}(r)$ est une évolution de r si seulement si $f_{raf}(r)$ est différent de r ;
- (2). que r et $f_{evo}(r)$ sont de même catégories si et seulement si $f_{FC}(f_{evo}(r)).r\grave{e}gle \cong f_{FC}(r).r\grave{e}gle \vee f_{FC}(f_{evo}(r)).r\grave{e}gle \equiv f_{FC}(r).r\grave{e}gle$ et $0 \leq sat(f_{FC}(f_{evo}(r))) - sat(f_{FC}(r)) \leq \varepsilon$.

Définition 4.2: Point de satisfaction maximale de la connaissance.

Soit un besoin r de BE , nous dirons que r_f de BE est un point de satisfaction de la connaissance pour l'évolution du besoin r si et seulement si pour toute évolution $f_{evo}(r_f)$, $0 \leq sat(f_{FC}(f_{evo}(r_f))) - sat(f_{FC}(r_f)) \leq \varepsilon$, $f_{FC}(f_{evo}(r_f)).r\grave{e}gle \cong f_{FC}(r_f).r\grave{e}gle$ et $sat(f_{FC}(f_{evo}(r_f))) - sat(f_{FC}(r)) \geq \varepsilon$. Nous notons \bar{r} , un point de satisfaction maximale de la connaissance de l'évolution de r .

La dynamique du fragment de connaissance couplé au besoin dont il est tiré se schématise tel que présenté par la figure 1 ci-dessous.

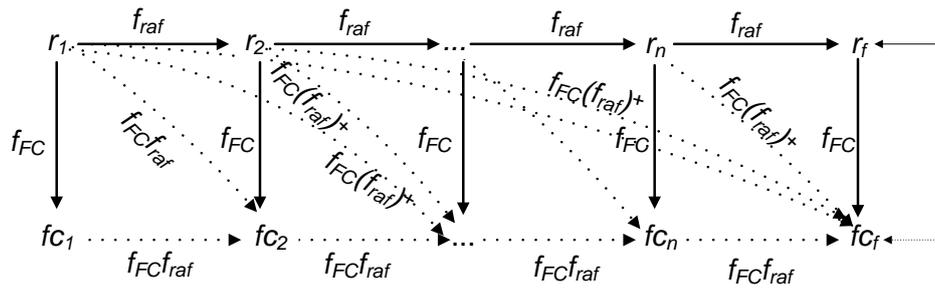


Figure 20 : Dynamique d'un besoin

$n \in \mathbb{N}^*$ représente un entier naturel non nul quelconque, $i \in \mathbb{N}^*$ représente les instants d'enrichissement de besoins, r_i les extension successives du besoin exprimé, fc_i une connaissance déduite d'un besoin r_i , les traits pleins représentent l'action des fonctions de base sur un concept donné, les flèches en pointillées présentent les fonctions déduites des fonctions de base, $f \in \mathbb{N}^*$ avec $f \geq n$ représente l'étape d'évolution à partir duquel on atteint le point de satisfaction maximale de la connaissance. Ce dernier est désigné dans ce schéma par r_f .

Dans la suite le point de satisfaction maximale de la connaissance sera appelé point fixe de l'évolution des besoins ou point fixe de l'évolution de la connaissance.

IV.2 ILLUSTRATION À L'AIDE DU CONCEPT DE CMOA

Considérons un fragment de connaissance $b = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$. Où : $b.\psi$ est le contexte de b , $b.\lambda$ est la règle de b , $b.\delta$ les contraintes de b , $b.\omega$ le but de b , et $b.\nu$ le niveau d'importance de b . Soit b' l'aspect métier associé à b . Nous supposons que b'_0 est l'aspect initial associé à l'aspect métier b' à un instant t_m (m entier naturel, $m > 0$). D'après [95], b' s'écrit : $b' = b'_0 + \sum_{i=1}^m b'_i$ (où $b'_i, 1 \leq i \leq m$ représentent les m enrichissements qui ont été apportés à b'_0 à l'instant t_m). Soit un enrichissement quelconque e à apporter à b' à l'instant t_{m+1} et e'' l'unité modulaire d'intégration associée à cet enrichissement. Deux cas sont possibles d'après [95] :

1er cas : $\exists e' \in \{b_1, b_2, \dots, b_m\}$ tel que $\dim type(e') = \dim type(e'')$,

$$b'' = b' + e''$$

$$= b'_0 + \sum_{i=1}^m b'_i + e''$$

$$b'' = b'_0 + \sum_{i=1}^{m+1} b'_i \text{ (par simple numérotation des unités modulaires$$

d'intégration)

2e cas : $\nexists e' \in \{b_1, b_2, \dots, b_m\}$ tel que $\dim type(e') = \dim type(e'')$, or d'après [95], une telle unité ne peut plus être prise en compte donc : $b' = b' + e''$

Voici brièvement présentée, la dynamique d'un besoin au sein d'une organisation. Il est cependant important de souligner que l'évolution d'un fragment de connaissance présentée dans cette section se fait de manière isolée c'est-à-dire hors du modèle de référence. Le paragraphe suivant table sur cette dynamique lorsque le fragment de connaissance se trouve intégré dans un modèle métier.

V. DYNAMIQUE DU MODELE DE REFERENCE

Le modèle de référence défini dans le cadre de la modélisation des besoins d'une organisation présente la situation à un moment donné des attentes de cette organisation et celles de ses bénéficiaires de services; Enrichi de quelques petites améliorations issus du processus de globalisation présenté dans la section II.1.7. Cependant, il ne semble pas possible de prédire des évolutions futures des besoins d'une organisation. Ce qui nous a amené à penser qu'il en est de même pour le modèle de référence. Ainsi, il n'est pas possible de savoir, à un moment donné quel sera le type d'évolution que le modèle subira à un moment donné. C'est ce qui a sans doute motivé l'idée selon laquelle il fallait étudier la dynamique du modèle dans le temps afin de savoir s'il existe des caractéristiques particulières. Cette étude est d'autant plus importante qu'il faille préserver la cohérence du modèle de référence.

V.1 DYNAMIQUE

La section précédente a été consacrée à la gestion des améliorations locales. C'est-à-dire des améliorations au niveau du fragment de connaissance. Cette section

sera consacrée aux améliorations/enrichissements majeures susceptibles d'entraîner une altération de la cohérence du modèle. La dynamique d'un modèle de référence est liée à celle de ses constituants (fragments de connaissance). Etant entendu que le fragment de connaissance provient des besoins, la dynamique du modèle de référence résultant de ces derniers est étroitement liée à celle du sous ensemble des besoins à partir duquel ses fragments de connaissance sont construits. Pour essayer d'encadrer l'évolution du modèle de référence, il a été supposé, que le modèle de référence d'une organisation évolue en réponse aux exigences de sa clientèle. Ce qui améliore la qualité de service perçue par l'utilisateur client. Cette qualité de service est aussi appelée qualité de service externe. Parce que les tâches ont des rendements maxima, le processus d'affaire a aussi une qualité de service externe qui est maximale. Cette qualité de service correspond à celle fournie par un processus dont les connaissances le constituant offrent les rendements maxima. On parlera d'extension du modèle si l'enrichissement opéré permet de modifier de manière substantielle la qualité de service perçue par l'utilisateur. L'amélioration de la qualité de service perçue par les usagers consiste à calculer la différence des deux qualités de services externes. C'est-à-dire la qualité de service externe du modèle enrichi moins la qualité de service externe de ce modèle avant son enrichissement. Ainsi, il y aura amélioration de cette qualité de service si cette valeur est supérieure à une valeur ε arbitrairement choisie. C'est sur la base de ces hypothèses que le raisonnement ci-dessous a été construit.

Pour étudier la dynamique du modèle de référence, les fonctions de base suivantes sont construites ainsi qu'il suit :

- (1). $f_p : P(BE) \rightarrow P(FC)$, telle que pour tout $b \in P(BE)$, $f_p(b) = \{f_{FC}(r) / r \in b\}$
- (2). $f_M : P(FC) \rightarrow Mod$, telle que pour tout $C \in P(FC)$, $f_M(C)$ retourne le modèle de référence construit à partir des fragments de connaissances éléments de C . $P(FC)$ désigne l'ensemble des parties de FC et Mod l'univers des modèles de besoins orientés buts déductibles à partir d'un sous ensemble de fragment de connaissances ;

- (3). $f_{Mev} : P(BE) \rightarrow P(BE)$ qui pour tout besoin $r \in P(BE)$, $f_{Mev}(r) \subseteq BE$ et $\exists y \in f_{Mev}(r) / f_{raf}(y) \in f_{Mev}(r)$ représente le besoin obtenu après un enrichissement du besoin r . f_{evo} est une fonction d'enrichissement des besoins. nous noterons f_{Mev}^+ , la fonction $\underbrace{f_{Mev} \circ f_{Mev} \circ \dots \circ f_{Mev}}_{n \text{ fois}}, (n \in \mathbb{N}^*)$;
- (4). qu'un sous ensemble de besoins enrichis $f_{Mev}(r)$ est une évolution du sous ensemble r de BE si seulement si $sat(f_p(f_{Mev}(r))) - sat(f_p(r)) \geq \varepsilon$ et $\forall a \in r, (a \in f_{Mev}(r)) \vee (\exists a' \in f_{Mev}(r) / a' = f_{raf}(a))$. ε est appelé le pas de convergence du modèle de référence. sat calcule la satisfaction des clients de l'organisation.

La dynamique du modèle de référence couplé au sous ensemble de besoins dont il est tiré se schématise tel que présenté par la figure 2 ci-dessous.

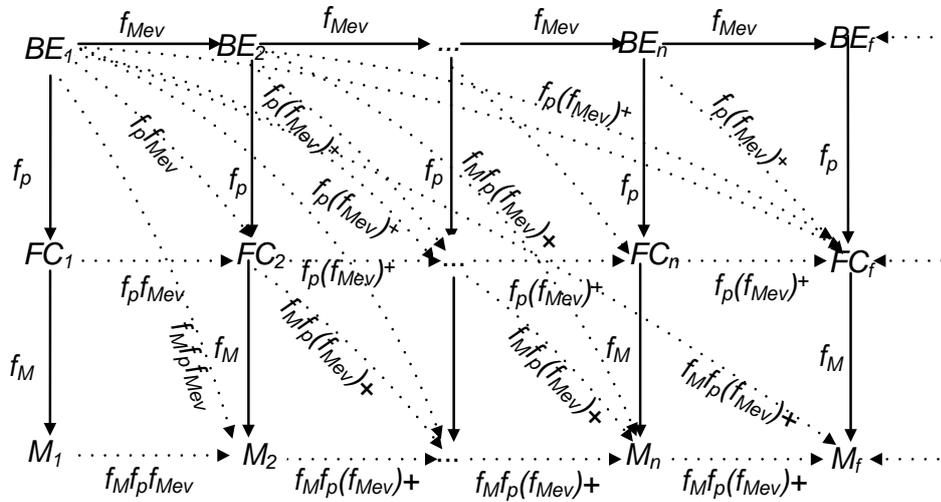


Figure 21: Dynamique du Modèle de référence

$n \in \mathbb{N}^*$ représente un entier naturel non nul quelconque, $f \in \mathbb{N}^*$ avec $f > n$ représente l'étape d'évolution à partir duquel on atteint le point fixe, les $BE_i (i \in \mathbb{N}^*)$ représente les différentes évolutions de BE par ordre chronologique, les $FC_i (i \in \mathbb{N}^*)$ sont les

sous ensembles de fragments de connaissance déduits des BE_i , les $M_i (i \in N^*)$ représentent les différents types de modèles de besoins déduits de chaque FC_i . Les flèches en pointillées indiquent les fonctions d'extension déduites des fonctions de base. M_f représente le point satisfaction maximal de l'évolution du modèle. Il est constitué de fragments de connaissance issus des points fixes des différents besoins.

Preuve : Démontrons que si la dynamique du besoin admet un point fixe alors le modèle de référence admet lui aussi un point stabilité. Considérons un modèle de référence M et BE un sous ensemble de besoins à partir duquel M est construit. Nous supposons que la dynamique du besoin admet un point fixe. Montrons par l'absurde que la dynamique du modèle de référence admet un point fixe. Nous notons \overline{BE} , un sous ensemble de besoins ayant des rendements maxima, \overline{M} le modèle de référence déduit de \overline{BE} . Nous supposons qu'il existe \overline{M}' tel que $sat(\overline{M}') \geq sat(\overline{M})$, alors nous pouvons trouver un besoin b de \overline{BE} tel que $f_{raf}(b) \in \overline{BE}$ et $sat(f_{raf}(b)) \geq sat(b)$ ce qui est absurde par définition de \overline{BE} .

V.2 ILLUSTRATION À L'AIDE DE CMOA

Nous allons, dans ce paragraphe, nous limiter à présenter les conditions à remplir pour pouvoir étendre le modèle de telle sorte que ce dernier demeure toujours cohérent. Pour cela nous allons, nous appuyer sur les concepts de composants métiers orienté aspects. Trois types d'améliorations ont retenu notre attention : le premier type concerne le raffinement d'un fragment de connaissance, le second est consacré à l'extension d'un fragment de connaissance et le dernier à la substitution du fragment de connaissance. Ces trois cas d'extensions matérialisent les différents cas de figure d'extension du modèle de référence. De manière concrète, considérons un modèle de référence quelconque M et M' le composant métier orienté aspect qui lui est associé. D'après [95], M' s'écrit : $M' = \langle a[a]^+, op \rangle$ où a est un aspect métier et op un opérateur de composition. Soit un fragment de connaissance $b = (\langle \psi, \omega \rangle, \langle \lambda, \delta \rangle, \nu)$ dans les conditions de la section précédente, l'évolution du modèle est possible dans les cas suivant:

1er cas : $\exists b' = (\psi', \omega', \lambda', \delta', \nu') \in M, b.\omega \cong b'.\omega' / \exists b' \in M / b.\lambda \cong b'.\lambda'$, nous supposons b'' est l'aspect métier associé b . $M'' = M' + b''$.

2e cas : $\exists b' \in M$ et si on peut construire un fragment de connaissance c (avec $c = (\langle \psi'', \omega'' \rangle, \langle \lambda'', \delta'' \rangle, \nu'')$) telle que $\psi'' = \psi \cup \psi'$, $\omega'' = \theta(\omega)$ et λ'' la règle d'ordonnancement, $\delta'' = \delta \cup \delta''$ et ν'' est le plus grand niveau d'importance entre ν et ν' . Nous considérons aussi b'' , b''' et c' les aspects métiers associés respectivement aux fragments de connaissance b , b' et c . L'enrichissement du modèle se fait ainsi qu'il suit :

$$M'' = M'.b''' \triangleright c'' \text{ où } : c'' = c' \sqcup (M'.b''', b'').$$

3e cas : il s'agit du remplacement d'un fragment de connaissance par un autre. Considérons un fragment de connaissance $c = (\langle \psi'', \omega'' \rangle, \langle \lambda'', \delta'' \rangle, \nu'')$, nous supposons qu'on voudrait substituer b par c dans M . Nous supposons qu'il existe un fragment de connaissance $d = (\langle \psi', \omega' \rangle, \langle \lambda', \delta' \rangle, \nu')$ tel que $b.\lambda \cong d.\lambda'$ si $c.\lambda'' \cong d.\lambda'$ et $b.\lambda \cong c.\lambda''$ ou $b.\lambda \equiv c.\lambda''$ alors la substitution de b par c dans M est possible. $M'' = M'.b'' \triangleright c'$.

VI. CONCLUSION

La gestion de l'évolution des besoins au sein d'une organisation est une préoccupation qui est assez reprise dans le monde des systèmes d'information. Cette dynamique du besoin tend à être à l'origine de l'obsolescence rapide des spécifications des processus d'affaire. L'étude présentée dans ce cadre constitue une alternative à cette préoccupation. Dans ce chapitre, il a été question de faire une étude de la dynamique de ce modèle de référence. La finalité de cette étude est de vérifier que le modèle de référence pourra évoluer en même temps que les besoins de l'organisation. Nous avons mis en exergue la nécessité, pour une organisation, d'aligner son processus d'affaire aux exigences de sa clientèle. C'est le souci de faire cet alignement qui constitue le déclencheur de l'évolution des besoins. Ce travail

constitue aussi une contribution à la réingénierie des processus d'affaire. L'étude que nous avons menée dans ce chapitre a été subdivisée en deux phases : (i) la première phase a consisté à étudier la dynamique de la qualité de service externe ; (ii) la deuxième phase a été consacré à l'étude de la dynamique du fragment de connaissance hors du modèle de référence. Pour ce faire, il a été considéré que l'évolution du besoin était due à aux changements opérés au sein de l'organisation, pour l'atteinte d'une qualité de service externe précise; (iii) la troisième phase s'est focalisée sur l'étude de la dynamique du modèle de référence dans son ensemble. De ce côté, il a été supposé que le déclencheur de l'évolution de ce modèle était la nécessité d'améliorer la qualité de service externe de l'organisation. Pendant cette étude, il a été construit trois fonctions d'extensions : f_{raf} , f_{Mev} et f_{QoS} ; et trois constructeurs : f_{FC} , f_M , et f_{be} . La fonction d'extension du besoin f_{raf} modélise l'action de raffinement du besoin jusqu'au besoin pour lequel tout enrichissement n'améliore plus la qualité de service interne. Ce besoin est appelé point fixe de l'évolution du besoin et est noté r_f ou tout simplement \bar{r} . La fonction d'extension du modèle f_{Mev} garanti l'évolution du modèle de référence jusqu'au modèle pour lequel la qualité de service rendue aux usagers clients est la meilleure possible. Le modèle offrant cette qualité de service est appelé point fixe de l'évolution du modèle et est noté M_f ou simplement \bar{M} . La fonction d'extension f_{QoS} garantit l'évolution de la qualité de service externe. Le constructeur f_{FC} permet de construire un fragment de connaissance à partir d'un besoin quelconque reçu en paramètre tandis que le constructeur f_M permet de construire un modèle de référence à partir d'un ensemble de fragments de connaissances reçu en paramètre. Le constructeur f_{be} permet de construire l'ensemble des besoins de l'organisation qui, satisfaits permettent l'atteinte d'une qualité de service précise. L'action des fonctions d'extension f_{raf} , et f_{Mev} n'a été possible que par la définition d'un seuil de satisfaction. C'est-à-dire une valeur à partir de laquelle, il est considéré qu'il y a eu amélioration de la qualité de service soit interne, soit externe, au sein d'une organisation. Dans cet article, une étude n'a pas été menée pour savoir si le choix du seuil de satisfaction influait sur la convergence des fonctions d'extensions du modèle

de référence. Dans les prochains jours nous allons nous atteler à faire cette étude. Il est cependant important de souligner que cette étude a permis d'ouvrir un nouvel axe de recherche. Notamment, la réingénierie des processus d'affaire basée sur la satisfaction des clients d'une organisation. Nous comptons aussi approfondir les recherches sur cet axe dans les jours à venir.

Le chapitre suivant sera consacré à la validation de ces travaux. Nous allons présenter un cas d'utilisation de cette méthodologie dans le système de liquidation des droits.

CHAPITRE V : CAS D'UTILISATION

Résumé :

La liquidation des droits est un ensemble de processus permettant à un agent public de bénéficier de ses droits à pension, une fois ce dernier sorti des effectifs de la fonction publique d'une part, de reverser les droits de ce dernier à ses ayants droits d'autre part. Ce chapitre présente un cas d'utilisation de la démarche présentée dans le chapitre précédent. Et s'appuie sur des besoins réels collectés auprès des cadres métiers en charge de ces processus. Compte tenu des contraintes administratives, les fiches de recensements de besoins ont été paraphées uniquement par les responsables nommés au sein de la sous direction de la liquidation des droits. Nous nous sommes intéressés beaucoup au volet qui posait le plus de problèmes à l'administration.

I. INTRODUCTION

La liquidation des droits est un ensemble de processus métiers permettant à l'administration de faire bénéficier à ses ex-employés des droits à pensions d'une part, de reverser leurs droits à leurs ayant-droits en cas de décès de l'ex-employé. Ces processus d'affaires s'appuient sur un ensemble de textes juridiques notamment :

- les lois :
 - o N° 69/LF/18 DU 10 NOVEMBRE 1969 instituant un régime d'assurance de pensions de vieillesse, d'invalidité et de décès ;
 - o N° 90/063 DU 19 DECEMBRE 1990 modifiant et complétant certaines dispositions des lois n° 69/LF/18 du 10 novembre 1969 et 84/007 du 04 juillet 1984 instituant un régime d'assurance pensions de vieillesse, d'invalidité et décès ;
 - o N° 2001/018 DU 18 DECEMBRE 2001 transférant à l'Etat la couverture et la gestion des prestations familiales de ses agents relevant du code du travail ;
- Les Décrets :
 - o N° 74/759 DU 26 AOUT 1974 portant organisation du régime des pensions civiles ;
 - o N°92/221/PM DU 08 MAI 1992 fixant les modalités d'admission à la retraite des personnels de l'Etat relevant du Code du Travail, modifié et complété par le décret n°93/334/PM du 13 avril 1993 ;
 - o N°92/220/PM DU 08 MAI 1992 fixant les modalités de transfert à l'Etat de la gestion du régime d'assurance, de pension de vieillesse, d'invalidité et décès des agents de l'Etat relevant du Code du Travail modifié et complété par le décret n°93/933/PM du 16 avril 1993 ;
 - o N° 94/036 DU 01 MARS 1994 modifiant et complétant certaines dispositions du décret n° 74/759 du 16 aout 1974 portant organisation du régime des pensions civiles ;
 - o N° 2000/684/PM DU 13 SEPTEMBRE 2000 fixant les conditions et modalités d'attribution du capital-décès ;
- La CIRCULAIRE N° 009/CAB/PM DU 29 DECEMBRE 1994 relative à la cessation de service des agents publics ;
- Les guides :
 - o GUIDE DE LIQUIDATION DES PRESTATIONS DES PENSIONS de vieillesse, d'invalidité et de décès des agents de l'Etat relevant du code du travail ;
 - o GUIDE DU DECOMPTEUR DES PRESTATIONS de vieillesse d'invalidité et de décès des personnels de l'Etat relevant du code du travail.

Il comporte onze procédures, notamment les processus d'affaires relatifs à :

- La retraite normale d'un fonctionnaire ;
- La retraite normale d'un agent de l'état relevant du code du travail ;
- Le remboursement de retenue d'un fonctionnaire ;
- La retraite par anticipation ;

- Le capital-décès ;
- Le capital décès quintuplé ;
- La pension de réversion aux ayant-droits ;
- La pension de réversion aux ayants cause ;
- La pension de survivant ;
- La pension d'invalidité ;
- Les rectificatifs d'actes.

Face aux interpellations récurrentes des usagers et des députés de la nation sur le service rendu aux usagers, le ministre de la fonction publique et de la réforme administrative a commis en 2008, une étude diagnostique du système de liquidation des droits. C'est la raison pour laquelle nous nous sommes principalement intéressés à ce sujet. Ce travail est à plus d'un titre bénéfique, dans la mesure où d'une part, l'état camerounais est entrain de migrer vers les systèmes informatiques dans lesquels est gérée la carrière de ses personnels, vers une solution intégrée renfermant toutes les fonctionnalités des systèmes SIGIPES et ANTILOPE. La mise en œuvre de cette solution requiert les règles métiers des différentes procédures ; et il est difficile de trouver un cadre métier maîtrisant les rouages des métiers de liquidation des droits. Sans nuire à la généralité nous avons considéré une procédure administrative de liquidation des droits comme étant un processus d'affaire. Et nous nous sommes limités à l'ensemble des activités qui concourent à la mise sur le circuit du projet d'acte de liquidation des droits relatifs à une procédure administrative ci-dessus évoquée.

Ce travail va permettre aux différentes administrations de disposer d'un outil permettant aux cadres métiers de connaître, hors application informatique, les métiers de la liquidation des droits. Ce qui permettra à coup sûr de faire face à la mobilité des personnels.

Dans la suite, nous allons dérouler le processus d'ingénierie des besoins présentés au chapitre précédent. Il faut noter que les annexes ont été retirées de ce document à la demande des membres du jury.

II. RECENSEMENT DES BESOINS

II.1.RECENSEMENT DES BESOINS DES UTILISATEURS

Nous avons réalisé un ensemble d'interviews pour recenser les besoins des cadres métiers en charge de la liquidation des droits au sein du Ministère de la Fonction Publique et de la Réforme Administrative. Cependant, les contraintes administratives ont voulu que seuls les cadres métiers nommés puissent parapher les documents de recensement des besoins des utilisateurs. Ces documents sont joints en annexe tel que présenté dans le tableau ci-dessus.

Tableau 4.1 : sémantiques des annexes

ANNEXES	Description
---------	-------------

ANNEXE 1	L'annexe 1, présente les besoins recensés des cadres métiers pour la mise en circulation d'un projet d'acte d'admission à la retraite normale d'un fonctionnaire ⁵ .
ANNEXE 2	L'annexe 2, présente les besoins recensés des cadres métiers pour la mise en circulation d'un projet d'acte de remboursement de retenues d'un fonctionnaire.
ANNEXE 3	L'annexe 3, présente les besoins recensés des cadres métiers pour la mise en circulation d'un projet d'acte d'admission à la retraite par anticipation d'un fonctionnaire.
ANNEXE 4	L'annexe 4, présente les besoins recensés des cadres métiers pour la mise en circulation d'un projet d'acte d'attribution d'un Capital-Décès et d'une pension de réversion aux ayants du défunt fonctionnaire, mort en activité.
ANNEXE 5	L'annexe 5, présente les besoins recensés des cadres métiers pour la mise en circulation d'un projet d'acte d'attribution d'un Capital-Décès quintuplé et d'une pension de réversion aux ayants d'un défunt fonctionnaire, mort en mission.
ANNEXE 6	L'annexe 6, présente les besoins recensés des cadres métiers pour la mise en circulation d'un projet d'acte d'attribution d'une pension d'invalidité.
ANNEXE 7	L'annexe 7, présente les besoins recensés des cadres métiers pour la mise en circulation d'un projet d'acte de capital-décès avec remboursement des retenues.

Compte tenu du fait que nous assistions les cadres métiers lors du renseignement du document de recensement des besoins, tous les besoins recensés étaient d'emblés consistants d'une part, le recensement des besoins s'est fait de manière successive, les cas de contradictions ont été résolus d'autre part.

II.2 Collecte des points de vue des usagers clients du Ministère de la Fonction Publique et de la Réforme Administrative

Pour la collecte des attentes des bénéficiaires des services, nous nous sommes fiés aux résultats de l'étude diagnostique du système de liquidation des droits qu'avait commandé le Ministre de la Fonction Publique et de la Réforme Administrative (MINFOPRA) en 2008. Plusieurs échanges entre les membres d'ACA⁶ (Agence de Consultants Associés) et les bénéficiaires des actes de liquidations des droits ont été réalisés afin de recueillir l'avis des usagers clients du MINFOPRA par rapport au service qui leur est fourni. Les tableaux 4.2 et 4.3 présentent chacun en ce qui le concerne une synthèse des résultats obtenus.

Tableau 4.2 : Synthèse de l'étude diagnostique réalisée par le cabinet ACA en 2008

⁵ Un fonctionnaire est un personnel de l'état régit par le statut général de la fonction publique camerounaise. Au Cameroun, ne peut prétendre à ce statut que celui qui a été recruté par voie de concours direct ou de bourse ouvert par le Ministre de la Fonction Publique et de la Réforme Administrative, en dehors des ENS, ENSET et du CUSS.

⁶ Cabinet ayant réalisé l'étude diagnostique du système de liquidation des droits

N°	Problèmes	Nombre de dossiers concernés
1	Dossier en instance pour défaillance du système informatique (DIDSI)	5 500
2	Dossier en instance pour manque d'un acte de carrière (DIMAC)	10 000
3	Dossier en instance pour manque d'une pièce autre qu'un acte de carrière (DIMPAAC)	500

Tableau 4.3 : Synthèse des échanges réalisés en 2008 entre le cabinet ACA et les bénéficiaires des actes de liquidations.

N°	Problèmes	Nombre de dossiers concernés
4	Le traitement d'un dossier de liquidations met trop de temps	1000

En plus, à la même date, le Premier Ministre Chef du Gouvernement a instruit, à travers la feuille de route assignée au Ministre de la Fonction Publique et de la Réforme Administrative, de ramener le délai moyen de traitement d'un dossier de liquidation des droits à un (01) mois⁷ (30 jours).

Compte tenu du fait que l'ensemble des bénéficiaires a pris cause pour les problèmes 4, relatés dans le tableau 4.3 précédent, modélisons l'avis du bénéficiaire de service de la manière suivante :

- a) Représentation de l'observation : nous représentons une observation de la manière suivante : *durée du traitement ≤ 30 jours* (avec *durée du traitement*, la durée qui s'écoule entre la date de réception d'un dossier et sa mise en circuit) ;
- b) Représentation du point de vue d'un bénéficiaire des actes de liquidation de droits. Dans le paragraphe précédent, nous avons considéré à la lumière de l'étude du cabinet ACA, un échantillon de mille (1 000) usagers. Considérons le *i*^{ème} usager de cet échantillon, son point de vue P_{v_i} sera représenté de la manière suivante :

$$P_{v_i} = (\text{Usager}_i, \text{durée du traitement} \leq 30 \text{ jours}, \text{GRH}), 1 \leq i \leq 1000$$

Où :

- Usager_i , représente le nom du *i*^{ème} usager de l'échantillon ;
- *durée du traitement ≤ 30 jours*, l'observation du Usager_i ;
- *GRH*, le domaine contenant le service évalué.

⁷ Ce délai a été consacré dans la feuille de route du Ministère de la Fonction Publique et de la Réforme Administrative de 2008

- c) Construction de P_{Views} , l'ensemble des besoins de vus des bénéficiaires des actes de liquidation de droits : la construction de cet ensemble nécessite la définition préalable des observations. P_{Views} est construit de la manière suivante :

$$P_{Views} = \{P_{v_1}, P_{v_2}, \dots, P_{v_{1000}}\}$$

Où :

- P_{v_i} , $1 \leq i \leq 1000$ représente le point de vus du $i^{ième}$ usager. La présentation de P_{v_i} a été faite au point b).

- d) Construction des ensemble $m_Observations$ et $Observations$:

$$m_Observations = \left\{ \underbrace{\text{durée du traitement} \leq 30 \text{ jours}, \dots, \text{durée du traitement} \leq 30 \text{ jours}}_{1000} \right\}$$

$$Observations = \{ \text{durée du traitement} \leq 30 \text{ jours} \}$$

- e) L'ensemble des facteurs de qualité se présente de la manière suivante :

$$QualityFactors = \{(\text{durée du traitement} \leq 30 \text{ jours}, 1000)\}$$

III SELECTION DES BESOINS

Dans ce contexte, les buts sont sélectionnés individuellement. Compte tenue des actions entreprises pendant la phase de recensement des besoins, nous avons, lorsque deux besoins avaient le même but et que leurs règles étaient différentes, raffiné l'ensemble d'objets de leurs buts pour le restreindre aux objets relatifs à la procédure pour laquelle le besoin a été exprimé. Nous avons mis en exergue (En gras, italique) le groupe d'objets. Ces informations sont contenues dans l'annexe 8. Le tableau suivant présente l'ensemble des besoins retenus après l'exécution de l'algorithme de sélection des besoins sur la collection des documents de recensement des besoins remplis par les cadres métiers de la Sous-direction de la liquidation des droits.

Tableau 4.4 : Ensemble des besoins retenus

Désignation de l'ensemble des besoins	Liste des éléments de l'ensemble des besoins
▽	$\{$ ElaLibDossRRF, ElaTextDossAdminRRF, AlignerConjoints, AlignerEnfants, AlingerHeritier, ElabEGS, AppretDossCD, AppretDossCDQ, AppretDossCDRR, AppretDossPI, AppretDossRAF, AppretDossRNF, AppretDossRRF, CalculerMontantCD, CalculerMontantCDQ, CalculerMontantCDRR, CalculerMontantPRD, CalculerPension, CalculerPensionDossRAF, CalculerRemb, ElabFichVisaDossCD, $\}$

	ElabFichVisaDossCDQ, ElabFichVisaDossCDRR, ElabFichVisaDossRAF, ElabFichVisaDossRNF, ElabFichVisaDossRRF, ElaFichTraitActePI, ElaFichTraitDossCD, ElaFichTraitDossCDQ, ElaFichTraitDossCDRR, ElaFichTraitDossRAF, ElaFichTraitDossRNF, ElaFichTraitDossRRF, ElaLibActeAdminPI, ElaLibDossCD, ElaLibDossCDQ, ElaLibDossCDRR, ElaLibDossRAF, ElaTextActeAdminPI, ElaTextDossAdminRNF, ElaTextDossCD, ElaTextDossCDQ, ElaTextDossCDRR, ElaTextDossRAF, ProdActeDossAttribPRD, ProdActeDossCD, ProdActeDossCDQ, ProdActeDossCDRR, ProdActeDossRAF, ProdActeDossRNF, ProdActeDossRRF, ProdActePI, RecepDossCD, RecepDossCDQ, RecepDossCDRR, RecepDossPI, RecepDossRAF, RecepDossRNF, RecepDossRRF, RepartCD, RepartCDQ, RepartPRD
	}

IV. MODELISATION DES PROCESSUS D'AFFAIRES

IV.1 TRANSFORMATION DES BESOINS

IV.1.1 TRANSFORMATION DES BESOINS EN FRAGMENTS DE CONNAISSANCE

Soit un besoin a quelconque de Ω , la représentation $a.r\grave{e}gle$ d note la r gle m tier du besoin a . Nous avons choisi de pr senter  tape par  tape, la transformation des besoins recens s pour le traitement d'un dossier de capital-d c s et attribution d'une pension de r version aux ayants-droits (cf. : annexe 4) en fragment de connaissance. Simplement parce que ces donn es nous permettrons de pr senter tous les cas de transformation avec notre formalisme de description des r gles m tiers. Les besoins recens s pour le traitement des autres dossiers de liquidation des droits se trouvent   l'annexe 8.

Cas 1 : Nous consid rons que $a = RecepDossCD$ (r ception du dossier de capital-d c s et attribution d'une pension de r version aux ayant-droits, cf. : Annexe 4), et $C_a = (Contexte, but, R gle, contraintes, Importance)$, un fragment de connaissance. La Construction de C_a   partir de a se fait ainsi qu'il suit :

Etape 1 : Extension du vocabulaire

Cette phase vise l'enrichissement des pr dicats de base avec ceux propres au domaine d' tude.

Type de pr�dicats	Pr�dicats	Notation
Pr�dicats de comparaisonest monogame	$(mono, \dots)$
est femme	(fam, \dots)

Etape 2 : Formalisation de la r gle m tiers "*R gle*"

Posons $C_a.r\grave{e}gle$, la règle métier du fragment de connaissance C_a , nous construisons $C_a.r\grave{e}gle$ de la manière suivante :

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits

Data =

Désignation	Éléments
<i>NameSpace.agregat</i>	{ Demande timbrée %Something, Copie de l'acte de décès %Something : control, Dossier de Capital-Décès %Something }

Désignation	Éléments
<i>Dossier de Capital – Décès.views</i>	(CopieActe%Something fait référence à une copie d'acte de naissance d'un enfant mineur, CopieMariage%Something fait référence à la copie de l'acte de mariage d'un conjoint)
<i>Demande Timbrée.views</i>	(Expéditeur %String)

Les objets ayant la mention *control*, sont ceux qui sont obligatoires et pour lesquels aucune donnée n'est nécessaire dans la partie *description* de la règle métier. Les objets ayant la mention *artifart* sont des objets qui sont soit produits par le système soit préalablement enregistrés et dont les données sont utilisées dans la partie *description* de la règle métier. Les données n'ayant pas de mentions sont des objets dont la présence n'est importante que sous certaines conditions. Par exemple, l' *Arrêté de fin de disponibilité*, n'est important que s'il existe un *Arrêté de mise en disponibilité* ; les « *Certificat de monogamie* », « *Certificat de polygamie pour chaque conjoint survivant* », « *Certificat de non remariage et non concubinage notoire pour chaque conjoint survivant* », « *Certificat de non séparation de corps et de non divorce pour chaque conjoint survivant* », « *Copie du jugement d'hérédité* » ne sont importants que si le conjoint décédé est un homme.

Phase 2 : Définition de la partie Description "*Description*"

i) La garde : Pour qu'on puisse recevoir un dossier de capital-décès, il faut obligatoirement une demande timbrée, une copie de l'acte de décès et l'existence d'au moins un conjoint ou un enfant mineur. Nous représentons cela de la manière suivante :

$(\forall, (\exists, dossier.CopieMariage), (\exists, dossier.CopieActe))$

ii) La séquence

Créer Récipissé.numéro

Mettre à jour Récépissé."nom du bénéficiaire" par

"Demande Timbrée".Expéditeur

Mettre à jour Récépissé. Libellé par "Demande de Capital – Décès"

iii) Le résultat "*Résultat*"

Désignation	Eléments
<i>Results: Récépissé.views</i>	{ Numéro%String, Nom du bénéficiaire%String, date d'enregistrement%Date, Libellé%String }

Etape 3: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Le fragment de connaissance ainsi obtenu, garde la même dénomination que le besoin.

Cas 2 : Nous considérons que $a = ElabEGS$ (élaboration de la liste des actes de carrières majeures dont a bénéficié le concerné tout au long de sa carrière, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Importance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits

Data =

Désignation	Eléments
<i>NameSpace.agregat</i>	{ CommandeEGS %Something }

Désignation	Eléments
<i>CammandeEGS.views</i>	(ActeIntégration%Something fait référence à intégration, ActesReclassement%Something fait référence à Reclassement, ActeGrade%Something fait référence à Avancement de grade, date de cessation d'activité%Date, ActeEngagement%Something fait référence à un acte d'Engagement décisionnaire, ActeContrat%Something fait référence à un Contrat, DernierActeAvancement%Something fait référence à dernier acte d'avancement)

Phase 2 : Définition de la partie Description "*Description*"

- i) La garde : la liste des actes de carrière majeur dont a bénéficié le concerné est définie en fonction de l'existence de ces actes majeurs. Nous sommes donc en présence d'une garde complexe (*if ... then ... elseif ... else.. endif*). Nous représentons les conditions de cette garde de la manière suivante :

Condition 1 : elle matérialise le cas où tous les actes majeurs suscités existent. C'est-à-dire que l'intéressé a été tour à tour décisionnaire, contractuel, puis fonctionnaire. Dans le statut de fonctionnaire l'intéressé a été reclassé et a bénéficié d'un acte d'avancement de grade.

$$\left(\wedge \left(\wedge \left(\wedge (\exists, \text{CommandeEGS. ActeIntégration}), \left(\wedge (\exists, \text{CommandeEGS. ActeGrade}), (\exists, \text{CommandeEGS. ActeReclassement}) \right) \right), (\exists, \text{CommandeEGS. ActeEngagement}) \right) \right), (\exists, \text{CommandeEGS. ActeContrat})$$

Condition 2 : elle matérialise le cas où l'intéressé est entré dans la fonction publique en qualité de Décisionnaire, puis est devenu fonctionnaire et a régulièrement bénéficié des actes majeurs suscités.

$$\left(\wedge \left(\wedge (\exists, \text{CommandeEGS. ActeIntégration}), \left(\wedge (\exists, \text{CommandeEGS. ActeGrade}), (\exists, \text{CommandeEGS. ActeReclassement}) \right) \right), (\exists, \text{CommandeEGS. ActeEngagement}) \right)$$

Condition 3 : elle matérialise le cas où l'intéressé est entré dans la fonction publique en qualité de contractuel, puis est devenu fonctionnaire et a régulièrement bénéficié des actes majeurs suscités.

$$\left(\wedge \left(\wedge (\exists, \text{CommandeEGS. ActeIntégration}), \left(\wedge (\exists, \text{CommandeEGS. ActeGrade}), (\exists, \text{CommandeEGS. ActeContrat}) \right) \right), (\exists, \text{CommandeEGS. ActeReclassement}) \right)$$

Condition 4 : elle matérialise le cas où l'intéressé est entré dans la fonction publique en qualité de fonctionnaire et a régulièrement bénéficié des actes majeurs suscités.

$$\left(\wedge \left(\wedge (\exists, \text{CommandeEGS. ActeIntégration}), (\exists, \text{CommandeEGS. ActeGrade}) \right), (\exists, \text{CommandeEGS. ActeReclassement}) \right)$$

Condition 5 : elle matérialise le cas où l'intéressé est entré dans la fonction publique en qualité de fonctionnaire et a bénéficié d'au moins un acte de reclassement.

$$(\wedge (\exists, \text{CommandeEGS. ActeIntégration}), (\exists, \text{CommandeEGS. ActeReclassement}))$$

Condition 6 : elle matérialise le cas où l'intéressé est entré dans la fonction publique en qualité de fonctionnaire et a bénéficié d'un unique acte d'avancement de grade.

$$(\wedge (\exists, \text{CommandeEGS. ActeIntégration}), (\exists, \text{CommandeEGS. ActeGrade}))$$

- ii) La séquence : le contenu de la séquence dépend de la satisfaction de l'une de ces gardes.

Cas 1 : *Condition 1* satisfaite

Enregistrer CommandeEGS.ActeEngagement dans EGS
Enregistrer CommandeEGS.ActeContrat dans EGS
Enregistrer CommandeEGS.Intégration dans EGS
Enregistrer CommandeEGS.ActesReclassement dans EGS
Enregistrer CommandeEGS.ActeGrade dans EGS
Enregistrer CommandeEGS.ActeCessation dans EGS
Enregistrer CommandeEGS.DernierActeAvancement dans EGS

Cas 2 : Condition 2 satisfaite

Enregistrer CommandeEGS.ActeEngagement dans EGS
Enregistrer CommandeEGS.Intégration dans EGS
Enregistrer CommandeEGS.ActesReclassement dans EGS
Enregistrer CommandeEGS.ActeGrade dans EGS
Enregistrer CommandeEGS.ActeCessation dans EGS
Enregistrer CommandeEGS.DernierActeAvancement dans EGS

Cas 3 : Condition 3 satisfaite

Enregistrer CommandeEGS.ActeContrat dans EGS
Enregistrer CommandeEGS.Intégration dans EGS
Enregistrer CommandeEGS.ActesReclassement dans EGS
Enregistrer CommandeEGS.ActeGrade dans EGS
Enregistrer CommandeEGS.ActeCessation dans EGS
Enregistrer CommandeEGS.DernierActeAvancement dans EGS

Cas 4 : Condition 4 satisfaite

Enregistrer CommandeEGS.Intégration dans EGS
Enregistrer CommandeEGS.ActesReclassement dans EGS
Enregistrer CommandeEGS.ActeGrade dans EGS
Enregistrer CommandeEGS.ActeCessation dans EGS
Enregistrer CommandeEGS.DernierActeAvancement dans EGS

Cas 5 : Condition 5 satisfaite

Enregistrer CommandeEGS.Intégration dans EGS
Enregistrer CommandeEGS.ActesReclassement dans EGS
Enregistrer CommandeEGS.ActeCessation dans EGS
Enregistrer CommandeEGS.DernierActeAvancement dans EGS

Cas 6 : Condition 6 satisfaite

Enregistrer CommandeEGS.Intégration dans EGS
Enregistrer CommandeEGS.ActeGrade dans EGS
Enregistrer CommandeEGS.ActeCessation dans EGS
Enregistrer CommandeEGS.DernierActeAvancement dans EGS

Cas 7 : le cas échéant

Enregistrer CommandeEGS.Intégration dans EGS
Enregistrer CommandeEGS.ActeCessation dans EGS
Enregistrer CommandeEGS.DernierActeAvancement dans EGS

iii) Le résultat "Résultat"

Désignation	Eléments
<i>Results : EGS.views</i>	{ <i>EGS%Something fait référence à EGS dans Results</i> }

Etape 2 : déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Le fragment de connaissance ainsi obtenu, garde la même dénomination que le besoin.

Cas 3 : Nous considérons que $a = AlignerConjoints$ (alignement des conjoints, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Importance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits

Data =

Désignation	Eléments
<i>Namespace.agregat</i>	{ Certificat de monogamie %Something, Certificat de polygamie pour chaque conjoint survivant %Something, Certificat de non remariage et non concubinage notoire pour chaque conjoint survivant %Something, Certificat de non séparation de corps et de non divorce pour chaque conjoint survivant %Something, Copie de l'acte de mariage d'un conjoint %Something, CopieDécès%Something fait référence à Copie de l'acte de décès

Désignation	Eléments
<i>Copie de l'acte de mariage d'un conjoint.views</i>	(Nom du conjoint de l'autre%String, Date de naissance du conjoint survivant%Date, Numéro %String, date de signature%Date, RegimeMatrimonial%String, Genre du conjoint%String
<i>CopieDécès.views</i>	(Datedécès%Date fait référence à Date du décès dans copie de l'acte de décès

Phase 2 : Définition de la partie Description "*Description*"

- i) La garde : la liste des conjoints survivants est définie en fonction du genre du conjoint survivant, du régime matrimoniale, et de l'existence des pièces justificatives. Nous sommes donc en présence d'une garde complexe (*if ...*

then ... elseif ... else.. endif). Nous représentons les conditions de cette garde de la manière suivante :

Condition 1 : elle matérialise le cas où le conjoint survivant serait un homme.

$$\left(\top, (fam, Copie de l'acte de mariage d'un conjoint. Genre du conjoint) \right)$$

Condition 2 : matérialise le cas échéant.

$$\left(\vee, \left(\wedge, \left(\begin{array}{l} (\exists, \text{ertificat de non remariage et non concubinage notoire pour chaque conjoint survivant}) \\ (\exists, \text{Certificat de non séparation de corps et de non divorce pour chaque conjoint survivant}) \\ (\exists, \text{Certificat de polygamie pour chaque conjoint survivant}) \end{array} \right), \left(\wedge, \left(\begin{array}{l} (\exists, \text{ertificat de non remariage et non concubinage notoire pour chaque conjoint survivant}) \\ (\exists, \text{Certificat de non séparation de corps et de non divorce pour chaque conjoint survivant}) \end{array} \right), (\exists, \text{Certificat de monogamie}) \right) \right) \right)$$

ii) La séquence : le contenu de la séquence dépend de la satisfaction de l'une de ces conditions.

Cas 1 : Condition 1 satisfaite

Enregistrer Copie de l'acte de mariage d'un conjoint. "Nom du conjoint",
 Copie de l'acte de mariage d'un conjoint. "Date de naissance du conjoint",
 Copie de l'acte de mariage d'un conjoint. Numéro ,
 Copie de l'acte de mariage d'un conjoint. "date de signature",
 Copie de l'acte de mariage d'un conjoint. RegimeMatrimonial,
 Copie de l'acte de mariage d'un conjoint. "Genre du conjoint",
 '0' dans *ListeConjoints*

Cas 2 : Condition 2 satisfaite

Enregistrer Copie de l'acte de mariage d'un conjoint. "Nom du conjoint",
 Copie de l'acte de mariage d'un conjoint. "Date de naissance du conjoint",
 Copie de l'acte de mariage d'un conjoint. Numéro ,
 Copie de l'acte de mariage d'un conjoint. "date de signature",
 Copie de l'acte de mariage d'un conjoint. RegimeMatrimonial,
 Copie de l'acte de mariage d'un conjoint. "Genre du conjoint"
 (différence en (Jours) de Copie de l'acte de mariage d'un conjoint. "date de signature" et CopieDécès. Datedécès) dans *ListeConjoints*

iii) Le résultat "*Résultat*"

Désignation	éléments
<i>Results : ListeConjoints.views</i>	$\left\{ \begin{array}{l} Nom\ du\ conjoint \%String, \\ Date\ de\ naissance\ du\ conjoint \%Date, \\ Numéro \%String, \ date\ de\ signature \%Date, \\ RegimeMatrimonial \%String, \\ Genre\ du\ conjoint \%String, \ Nombre\ de \\ \text{jours de mariage}, \\ ListeConjoints \%Something \ \text{fait} \\ \text{référence à ListeConjoints dans Results} \end{array} \right\}$

Etape 2: déduction du fragment de connaissance C_{α}

Le fragment de connaissance C_{α} est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.R\grave{e}gle = "R\grave{e}gle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorit\acute{e} de l'attente"$.

Cas 4 : Nous considérons que $a = AlignerEnfants$ (alignement des enfants âgés de moins 21 ans, cf. : Annexe 4), et $C_a = (Contexte, but, R\grave{e}gle, contraintes, Importance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits

Data =

Désignation	éléments
<i>Namespace.agregat</i>	{ Copie de l'acte de naissance pour enfant mineur%Something, CopieActeCessation% <i>Something fait référence à</i> Copie de l'acte de cessation d'activité, Copie de l'acte de reconnaissance%Something }

Désignation	éléments
<i>Copie de l'acte de naissance d'un enfant mineur légitime.views</i>	(<i>Nom de l'enfant%String, Date de Naissance%Date, Numéro%String, Nom de l'autre parent%String fait référence à nom conjoint dans Copie de l'acte de mariage d'un conjoint</i>)
<i>CopieActeCessation.views</i>	(<i>Date de cessation de service%Date</i>)
<i>Copie de l'acte de reconnaissance d'un enfant mineur.views</i>	(<i>Nom de l'enfant%String, Date de Naissance%Date, Nom du parent%String fait référence à Nom du concerné dans Intégration</i>)

Phase 2 : Définition de la partie Description "*Description*"

- i) La garde : la liste des enfants mineurs du concerné est définie en fonction du genre du conjoint survivant. Nous sommes donc en présence d'une garde complexe (*if ... then ... else... endif*). Nous représentons les conditions de cette garde de la manière suivante :

Condition 1 : elle matérialise le cas où le conjoint serait un homme.

$$\left(T, (fam, CopieMariage.Genre \text{ du conjoint}) \right)$$

Condition 2 : matérialise le cas échéant.

- ii) La séquence : le contenu de la séquence dépend de la satisfaction de l'une de ces conditions.

Cas 1 : Condition 1 satisfaite

Enregistrer Copie de l'acte de naissance d'un enfant mineur légitime. Numéro , Copie de l'acte de naissance d'un enfant mineur légitime. "Nom de l'enfant" , Copie de l'acte de naissance d'un enfant mineur légitime. "Date de naissance" , Copie de l'acte de naissance d'un enfant mineur légitime. Nom de l'autre parent, '0' dans ListeEnfants

Cas 2 : Condition 2 satisfaite

Enregistrer Copie de l'acte de reconnaissance. Numéro , Copie de l'acte de reconnaissance. "Nom de l'enfant" , Copie de l'acte de reconnaissance. "Date de naissance" , Copie de l'acte de reconnaissance. "Nom de l'autre parent", '0' dans ListeEnfants

Enregistrer Copie de l'acte de naissance d'un enfant mineur légitime. Numéro , Copie de l'acte de naissance d'un enfant mineur légitime. "Nom de l'enfant" , Copie de l'acte de naissance d'un enfant mineur légitime. "Date de naissance" , Copie de l'acte de naissance d'un enfant mineur légitime. Nom de l'autre parent, '0' dans ListeEnfants

- iii) Le résultat "*Résultat*"

Désignation	éléments
<i>Results:ListeEnfants.views</i>	$\left\{ \begin{array}{l} \text{Numéro}\%String, \text{ Nom de l'enfant}\%String, \text{ Nom} \\ \text{de l'autre parent}\%String, \text{ Date de} \\ \text{Naissance}\%Date, \text{ Nom du parent}\%String, \text{ Part} \\ \text{de l'enfant}\%Nombre, \text{ ListeEnfants} \\ \%Something \text{ fait référence à ListeEnfants dans} \\ \text{Results} \end{array} \right\}$

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 5 : Nous considérons que $a = CalculerMontantCD$ (calcul du montant du capital décès, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Importance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
<i>NameSpace.agregat</i>	<i>Grille Salariale%Something fait référence à barème de salaire , DossierCapitalDécès%Something fait référence à Dossier de Capital – Décès , DernierActe%Something fait référence à dernier acte d'avancement, CopieDécès%Something fait référence à Copie de l'acte de décès, Intégration%Somthing : Control, StatutGénéral%Something fait référence à Statut général de la Fonction Publique</i>

Désignation	éléments
<i>DernierActe.views</i>	(<i>Indice Solde%Nombre, Nom du concerné%String fait référence à Nom du défunt dans Copie de l'acte de décès</i>)
<i>Grille Salariale.views</i>	(<i>Indice%Nombre fait référence à Indice solde dans dernier acte d'avancement, Salaire de Base%Nombre</i>)
<i>CopieDécès.views</i>	(<i>Date de décès%Date, Age au décès%Nombre</i>)
<i>DossierCapitalDécès.views</i>	(<i>Nombre d'enfants mineurs%Nombre</i>)
<i>StatutGénéral.views</i>	(<i>Cadre%String fait référence à Cadre dans dernier acte d'avancement, Age de départ à la retraite%Nombre</i>)

Phase 2 : Définition de la partie Description "*Description*"

i) La garde :

(It, CopieDécès.Age au décès, StatutGénéral. "Age de départ à la retraite")

ii) La séquence :

Ranger (*le produit de Grille Salariale."Salaire de Base"* par '*12'*) dans *MCP*

Ranger (*le produit de '75000'* par *DossierCapitalDécès."Nombre d'enfants mineurs"*) dans *MAJ*
{{(It, CopieDécès."Date de décès, #13/09/2000 #)}}

Ranger (*le produit de '200000'* par *DossierCapitalDécès."Nombre d'enfants mineurs"*) dans *MAJ*
{{(ue, CopieDécès."Date de décès, #13/09/2000 #)}}

Ranger *MAJ* dans *MCP{{(ut, MAj , MCP)}}*

iii) Le résultat "*Résultat*"

Désignation	éléments
<i>Results:MontantCD.views</i>	{ <i>MCP%Nombre, MAJ%Nombre</i> }

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;

- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 6 : Nous considérons que $a = RepartCD$ (Répartition du capital décès, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Importance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "**Contexte**"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
<i>Namespace.agregat</i>	$\left\{ \begin{array}{l} DossierCapitalDécès\%Something \text{ fait référence à} \\ \text{de Capital - Décès} , CopieDécès\%Something \text{ fait} \\ \text{référence à Copie de l'acte de décès,} \\ ListeConj\%Something :artifact , MontCD\%Something : \\ artifact, Listeenf\%Something :artifact \end{array} \right\}$

Désignation	éléments
<i>CopieDécès.views</i>	(<i>Date de décès\%Date</i>)
<i>DossierCapitalDécès.views</i>	(<i>Date de dépôt \%Date, Nombre de conjoints\%Nombre, Nombre d'enfants mineurs\%Nombre</i>)
<i>ListeConj.views</i>	(<i>nbJour\%Nombre fait référence à Nombre de jours de mariage</i>)
<i>MontCD.views</i>	(<i>MtCD\%Nombre fait référence à MCP dans MontantCD, MtMAJ\%Nombre fait référence à MAJ dans MontantCD</i>)
<i>ListEnfant.views</i>	(<i>DateNaiss\%Date fait référence à Date de naissance dans ListeEnfants, PartEnf\%Nombre fait référence à Part de l'enfant dans ListeEnfants</i>)

Phase 2 : Définition de la partie Description "**Description**"

i) La garde :

Condition 1 :

(lt, (la différence en (années, mois, jours) de CopieDécès."Date de décès" et DossierCapitalDécès."Date de dépôt"), '4 ans 3 mois')

Condition 2 :

(ue, (la différence en (années, mois, jours) de CopieDécès."Date de décès" et DossierCapitalDécès."Date de dépôt"), '4 ans 3 mois')

ii) La séquence :

Cas 1 : Condition 1 satisfaite

Ranger (la division de (le produit de MontCD.MtCD par ListeConj.nbjour) par (le

produit de '3' par (la somme de ListeConj.nbjour dans ListeConj)) dans PCC {(ut, DossierCapitalDécès."Nombre de conjoints", '0')}

Ranger (la division de (la somme de (le produit de '2' par (la division de MontCD.MtCD par '3')), MontCD.MtMAJ) par DossierCapitalDécès."Nombre d'enfants mineurs") dans PCEL {(^, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (ut, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger (la division de (la somme de (le produit de '3' par (la division de MontCD.MtCD par '3')), MontCD.MtMAJ) par DossierCapitalDécès."Nombre d'enfants mineurs") dans PCEL { (^, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (≡, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger '0' dans PCEL {(≡, DossierCapitalDécès."Nombre d'enfants mineurs", '0')}

Ranger PCEL dans ListEnfant.PartEnf. PartEnf

Cas 2 : le cas échéant

Ranger '0' dans PCC

Ranger (la division de (la somme de (le produit de '2' par (la division de MontCD.MtCD par '3')), MontCD.MtMAJ) par DossierCapitalDécès."Nombre d'enfants mineurs") dans PCEL {(^, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (ut, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger (la division de (la somme de (le produit de '3' par (la division de MontCD.MtCD par '3')), MontCD.MtMAJ) par DossierCapitalDécès."Nombre d'enfants mineurs") dans PCEL { (^, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (≡, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger PCEL dans ListEnfant.PartEnf {(lt, (la différence en (années, mois, jours) de DossierCapitalDécès."Date de dépôt" et ListeEnf.DateNaiss), '22 ans')}

iii) Le résultat "Résultat"

Désignation	éléments
Results: MontantRepartitionCD.views	{ PCEL%Nombre, PCC%Nombre }

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 7 : Nous considérons que $a = ProdActeDossCD$ (production du projet d'acte d'attribution d'un capital décès, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Inportance)$, un fragment de connaissance

pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "**Contexte**"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
<i>Namespace.agregat</i>	<p><i>ListeConj%Something :artifact , MontCD%Something : artifact, Listeenf%Something :artifact, ActeIntégration%Something :artifact, MontRepartCD%Something :artifact</i> <i>TexteFixeCD%Something fait référence à template des parties fixes du projet d'acte de capital-décès, DossierCapitalDécès%Something fait référence à de Capital – Décès , CopieDécès%Something fait référence à Copie de l'acte de décès, DernierActe%Something fait référence à Dernier acte d'avancement, StatutGénéral%Something fait référence à Statut général de la Fonction Publique</i></p>

Désignation	éléments
<i>CopieDécès.views</i>	(<i>Date de décès%Date, LieuDécès%string fait référence à Lieu de décès dans Copie de l'acte de décès</i>)
<i>DossierCapitalDécès.views</i>	(<i>Date de dépôt %Date, Nombre de conjoints%Nombre, Nombre d'enfants mineurs%Nombre</i>)
<i>ListeConj.views</i>	(<i>Genre%String fait référence à Genre du conjoint dans ListeConjoints</i> <i>nbjour%Nombre fait référence à Nombre de jours de mariage dans ListeConjoints</i>)
<i>MontCD.views</i>	(<i>MtCD%Nombre fait référence à MCP dans MontantCD, MtMAJ%Nombre fait référence à MAJ dans MontantCD</i>)
<i>ListEnfant.views</i>	(<i>DateNaiss%Date fait référence à Date de naissance dans ListeEnfants</i>)
<i>DernierActe.views</i>	(<i>Indice Solde%Nombre, Nom du concerné%String fait référence à Nom du défunt dans Copie de l'acte de décès, SituationAdmin fait référence à Situation administrative dans Dernier acte d'avancement, Grade%String fait référence à grade dans Dernier acte d'avancement</i>)
<i>ActeIntégration.views</i>	(<i>SituationCivile%String fait référence à Situation civile dans Intégration, Nom du concerné%String fait référence à Nom du défunt dans Copie de l'acte de décès</i>)
<i>TexteFixeCD.views</i>	(<i>TexteFixeentete%String, TexteFixeVisas%string, TexteFixeArticle1%String, TexteFixeArticle2%String, TexteFixeArticle3%String, TexteFixeArticle4%String, TexteFixeArticle5%String, TexteFixeampliations%String, TexteFixelieu%String</i>)
<i>StatutGénéral.views</i>	(<i>Corps%String fait référence à Corps dans Dernier acte d'avancement, Réstatutparticulier%string</i>)

Phase 2 : Définition de la partie Description "**Description**"

i) La garde :

(v, (ut, DossierCapitalDécès."Nombre de Conjoints", '0'), (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'))

ii) La séquence :

Construire *TexteFixeCD.TexteFixeEntete*, type "Entete Capital-décès", zone (Nom du concerné, grade) à partir de *ActeIntégration."nom du concerné", DernierActe.Grade*

Construire *TexteFixeCD.TexteFixeVisas*, type "Visas", zone ("statut particulier du corps") à partir de *StatutGénéral.Réfstatutparticulier*

Construire *TexteAmpliation*, type "Ampliation Capital-Décès" à partir de *TexteFixeCD.TexteFixeAmpliation*

Construire *TexteLieu*, type "Lieu" à partir de *TexteFixeCD.TexteFixeLieu*

Construire *TexteFixeCD.TexteFixeArticle1*, type "Article premier Capital-Décès", zone ("Situation Administrative", "Situation Civile", "Date de décès") à partir de *DernierActe.SituationAdmin, ActeIntégration.SituationCivile, ActeDécès."Date de décès"*

Ranger *TexteFixeCD.TexteFixeArticle1* dans *TexteArticle1*

Construire *TexteFixeCD.TexteFixeArticle2*, type "Article deuxième Capital-Décès", zone ("Liste de Conjoints", "Montant du capital-décès", "Nombre de conjoints", "Nombre de jours de mariage par conjoint", "Date de décès", "Forclusion", "Liste des enfants", "Nombre d'enfants", "Montant de la Majoration", "Genre du défunt", "Age de chaque enfant") à partir de *ListeConj, MontantCD.MtMCP, DossierCapitalDécès."Nombre de conjoints", ListeConj.nbjour, (la différence en (années, mois, jours) de CopieDécès."Date de décès" et DossierCapitalDécès."Date de dépôt"), ListEnfant, DossierCapitalDécès."Nombre d'enfants mineurs", MontCD.MtMAJ, ListeConjoint.Genre, (la différence en (années, mois, jours) de ListEnfant.DateNais et #20/06/2011#)*

Ranger *TexteFixeCD.TexteFixeArticle2* dans *TexteArticle2*

Construire *TexteArticle3*, type "Article troisième Capital-Décès" à partir de *TexteFixeCD.TexteFixeArticle3*

Construire *TexteArticle4*, type "Article quatrième Capital-Décès" à partir de *TexteFixeCD.TexteFixeArticle4*

Construire *TexteArticle5*, type "Article quatrième Capital-Décès" à partir de *TexteFixeCD.TexteFixeArticle5*

iii) Le résultat "**Résultat**"

Désignation	éléments
<i>Results: Projet d'acte de capital-décès.views</i>	$\left\{ \begin{array}{l} \text{TexteEntete}\%String, \text{TexteVisas}\%string, \\ \text{TexteArticle1}\%String, \text{TexteArctile2}\%String, \\ \text{TexteArticle3}\%String, \text{TexteArticle4}\%String, \\ \text{TexteArticle5}\%String, \text{TexteAmpliations}\%String, \\ \text{TexteLieu}\%String \end{array} \right\}$

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 8 : Nous considérons que $a = CalculerMontantPRD$ (Calcul du montant de pension de réversion, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Importance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "**Contexte**"

Keywords =

GRH, Liquidation des droits

Data =

Désignation	éléments
<i>Namespace.agregat</i>	$\left. \begin{array}{l} Grille Salariale \% \text{Something fait référence à barème de salaire,} \\ DernierActe \% \text{Something fait référence à dernier acte d'avancement,} \\ CopieDécès \% \text{Something fait référence à Copie de l'acte de décès,} \\ ActeIntégration \% \text{Somthing fait référence à Intégration,} \\ StatutGénéral \% \text{Something fait référence à Statut général de la Fonction Publique} \end{array} \right\}$

Désignation	éléments
<i>DernierActe.views</i>	$(\text{Indice Solde} \% \text{Nombre, Nom du concerné} \% \text{String fait référence à Nom du défunt dans Copie de l'acte de décès})$
<i>Grille Salariale.views</i>	$(\text{Indice} \% \text{Nombre fait référence à Indice solde dans dernier acte d'avancement, Salaire de Base} \% \text{Nombre})$
<i>CopieDécès.views</i>	$(\text{Datedécès} \% \text{Date fait référence à Date de décès dans Copie de l'acte de décès})$
<i>StatutGénéral.views</i>	$(\text{Cadre} \% \text{String fait référence à Cadre dans dernier acte d'avancement, Age de départ à la retraite} \% \text{Nombre})$
<i>ActeIntégration.views</i>	$(\text{DateRecrutement} \% \text{Date fait référence à Date de recrutement dans Integration})$

Phase 2 : Définition de la partie Description "**Description**"

i) La garde :

(ue, (la différence en(mois) de CopieDécès.Datedécès et ActeIntégration.DateRecrutement), '6')

ii) La séquence :

Ranger (la différence en (années, mois, jours) de CopieDécès.Datedécès et ActeIntégration.DateRecrutement) dans DS
Extraire dans ! NA (la partie mois) de DS

Ranger (l'extraction dans NA (la partie années) de DS) dans NA{!(lt, NA,'3')}

Ranger (l'ajout de (l'extraction dans NA (la partie années) de DS) à '0,5') dans NA{!(^,(ue, NA,'3'),(le, NA,'8'))}

Ranger (l'ajout de (l'extraction dans NA (la partie années) de DS) à '1') dans NA{!(ue, NA,'9')}

Ranger (la division de (le produit de (le produit de Grille Salariale.Indice par NA) par '2') par '100') dans MP{(ut, Grille Salariale.Indice,'100')}

Ranger (la division de (le produit de (le produit de Grille Salariale.Indice par NA) par '4') par '100') dans MP{(le, Grille Salariale.Indice,'100')}

iii) Le résultat "**Résultat**"

Désignation	éléments
Results:MontantPRD.views	{ DS%Nombre, NA%Nombre, MP%Nombre }

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 9 : Nous considérons que $a = RepartPRD$ (Répartition de la pension de réversion aux ayant-droits, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Inportance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "**Contexte**"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
Namespace.agregat	{ DossierCapitalDécès%Something fait référence à de Capital – Décès , CopieDécès%Something fait référence à Copie de l'acte de décès, ListeConj%Something :artifact , MontCD%Something : }

	<i>artifact, Listenf%Something :artifact, MtPRD%Something :artifact</i>
Désignation	éléments
<i>CopieDécès.views</i>	(<i>Date de décès%Date</i>)
<i>DossierCapitalDécès.views</i>	(<i>Date de dépôt %Date, Nombre de conjoints%Nombre, Nombre d'enfants mineurs%Nombre</i>)
<i>MtPRD.views</i>	(<i>NA%Nombre fait référence à NA dans MontantPRD, MP%Nombre fait référence à MP dans MontantPRD</i>)

Phase 2 : Définition de la partie Description "**Description**"

i) La garde :

Condition 1 :

(lt, (la différence en (années, mois, jours) de CopieDécès."Date de décès" et DossierCapitalDécès."Date de dépôt"), '4 ans 3 mois')

Condition 2 :

(ue, (la différence en (années, mois, jours) de CopieDécès."Date de décès" et DossierCapitalDécès."Date de dépôt"), '4 ans 3 mois')

ii) La séquence :

Cas 1 : Condition 1 satisfaite

Ranger (la division de (le produit de MtPRD.MP par '50') par (le produit de DossierCapitalDécès."Nombre de conjoints" par '100')) dans PCC{(ut, DossierCapitalDécès."Nombre de conjoints", '0')}

Ranger (la division de (le produit de MtPRD.MP par '50') par (le produit de DossierCapitalDécès."Nombre d'enfants mineurs" par '100')) dans PCEL {(^, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (ut, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger (la division de (le produit de MtPRD.MP par '100') par (le produit de DossierCapitalDécès."Nombre d'enfants mineurs" par '100')) dans PCEL { (^, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (ut, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger '0' dans PCEL {(ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0')}

Ranger '0' dans NAP

Ranger CopieDécès."Date de décès" dans 'Date ouverture de droits'

Cas 2 : le cas échéant

Ranger '0' dans PCC

Ranger (la division de (la différence en (années, mois, jours) de CopieDécès."Date de décès" et DossierCapitalDécès."Date de dépôt") par '4 ans 3 mois') dans NAP

Ranger (l'ajout de CopieDécès."Date de décès" à (le produit de '4 ans 3 mois' par NAP)) dans 'Date ouverture de droits'

Ranger (la division de (le produit de MtPRD.MP par '50') par (le produit de DossierCapitalDécès."Nombre de conjoints" par '100')) dans PCC{(ut, DossierCapitalDécès."Nombre de conjoints", '0')}

Ranger (la division de (le produit de MtPRD.MP par '50') par (le produit de DossierCapitalDécès."Nombre d'enfants mineurs" par '100')) dans PCEL { (∧, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (ut, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger (la division de (le produit de MtPRD.MP par '100') par (le produit de DossierCapitalDécès."Nombre d'enfants mineurs" par '100')) dans PCEL { (∧, (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'), (≡, DossierCapitalDécès."Nombre de conjoints", '0'))}

Ranger '0' dans PCEL { (≡, DossierCapitalDécès."Nombre d'enfants mineurs", '0')}

iii) Le résultat "**Résultat**"

Désignation	éléments
Results: MontantRepartitionCD.views	{ PCEL%Nombre, PCC%Nombre, NAP%Nombre, Date ouverture de droits%Date }

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 10 : Nous considérons que $a = ProdActeDossPRD$ (production du projet d'acte d'attribution d'une pension de réversion aux ayant-droits, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Inportance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "**Contexte**"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
Namespace.agregat	{ ListeConj%Something :artifact , MontCD%Something : artifact, Listeenf%Something :artifact, ActeIntégration%Something :artifact, TexteFixePRD%Something fait référence à template des parties fixes du projet d'acte de pension de reversion aux ayant-droits, DossierCapitalDécès%Something fait référence à de Capital – Décès , CopieDécès%Something fait référence à Copie de l'acte de décès, }

	<i>DernierActe%Something fait référence à Dernier acte d'avancement, StatutGénéral%Something fait référence à Statut général de la Fonction Publique, MtPRD%Something : artifact, MtRepartitionCD%Something :artifact</i>
--	---

Désignation	éléments
<i>CopieDécès.views</i>	(<i>Date de décès%Date, LieuDécès%string fait référence à Lieu de décès dans Copie de l'acte de décès</i>)
<i>DossierCapitalDécès.views</i>	(<i>Date de dépôt %Date, Nombre de conjoints%Nombre, Nombre d'enfants mineurs%Nombre</i>)
<i>ListeConj.views</i>	(<i>Genre%String fait référence à Genre du conjoint dans ListeConjoints nbjour%Nombre fait référence à Nombre de jours de mariage dans ListeConjoints</i>)
<i>ListEnfant.views</i>	(<i>DateNaiss%Date fait référence à Date de naissance dans ListeEnfants</i>)
<i>DernierActe.views</i>	(<i>Indice Solde%Nombre, Nom du concerné%String fait référence à Nom du défunt dans Copie de l'acte de décès, SituationAdmin fait référence à Situation administrative dans Dernier acte d'avancement, Grade%String fait référence à grade dans Dernier acte d'avancement</i>)
<i>ActeIntégration.views</i>	(<i>SituationCivile%String fait référence à Situation civile dans Intégration, Nom du concerné%String fait référence à Nom du défunt dans Copie de l'acte de décès</i>)
<i>TexteFixePRD.views</i>	(<i>TexteFixeentete%String, TexteFixeVisas%string, TexteFixeArticle1%String, TexteFixeArticle2%String, TexteFixeArticle3%String, TexteFixeArticle4%String, TexteFixeArticle5%String, TexteFixeArticle6%String, TexteFixeampliations%String, TexteFixelieu%String</i>)
<i>StatutGénéral.views</i>	(<i>Corps%String fait référence à Corps dans Dernier acte d'avancement, Réfstatutparticulier%string</i>)
<i>MtPRD.views</i>	(<i>NA%Nombre fait référence à NA dans MontantPRD, MP%Nombre fait référence à MP dans MontantPRD, DS%Nombre fait référence dans MontantPRD</i>)
<i>MtRepartitionCD.views</i>	(<i>MtPCEL%Nombre fait référence à PCEL dans MontantRepartitionCD, MtPCC%Nombre fait référence à PCC dans MontantRepartitionCD, MtNAP%Nombre, DateOuvdroits%Date fait référence à Date ouverture de droits dans MontantRepartitionCD</i>)

Phase 2 : Définition de la partie Description "**Description**"

i) La garde :

(*v*, (*ut*, *DossierCapitalDécès."Nombre de Conjoints"*, '0'), (*ut*, *DossierCapitalDécès."Nombre d'enfants mineurs"*, '0'))

ii) La séquence :

Construire *TexteFixePRD.TexteFixeEntete*, type "*Entete Pension de réversion aux ayants-droits*", zone (*Nom du concerné, grade*) à partir de *ActeIntégration."nom du concerné"*, *DernierActe.Grade*

Construire *TexteFixePRD.TexteFixeVisas*, type "*Visas*", zone ("*statut particulier du corps*") à partir de *StatutGénéral.Réfstatutparticulier*

Construire *TexteAmpliation*, type "*Ampliation Capital-Décès*" à partir de

TexteFixePRD.TexteFixeAmpliation

Construire *TexteLieu*, type "Lieu" à partir de *TexteFixePRD.TexteFixeLieu*

Construire *TexteFixePRD.TexteFixeArticle1*, type "Article premier Pension de réversion aux ayant-droits", zone ("Situation Administrative", "Situation Civile", "Date de décès", "Durée de service", "Nombre d'annuités") à partir de *DernierActe.SituationAdmin*, *ActeIntégration.SituationCivile*, *ActeDécès.Date de décès*, *MtPRD.DS*, *MtPRD.NA*

Ranger *TexteFixeCD.TexteFixeArticle1* dans *TexteArticle1*

Construire *TexteArticle2*, type "Article deuxième pension de réversion aux ayant-droits" à partir de *TexteFixePRD.TexteFixeArticle2*

Construire *TexteFixePRD.TexteFixeArticle3*, type "Article deuxième Capital-Décès", zone ("Liste de Conjoints", "Nombre de conjoints", "Nombre de jours de mariage par conjoint", "Date de décès", "Prescription", "Liste des enfants", "Nombre d'enfants", "Genre du défunt", "Age de chaque enfant", "Date d'effet de la pension", "Part de chaque enfant", "Montant Pension de réversion", "part de chaque conjoint") à partir de *ListeConj*, *DossierCapitalDécès.Nombre de conjoints*, *ListeConj.nbjour*, (la différence en (années, mois, jours) de *CopieDécès.Date de décès* et *DossierCapitalDécès.Date de dépôt*), *ListEnfant*, *DossierCapitalDécès.Nombre d'enfants mineurs*, *ListeConjoint.Genre*, (la différence en (années, mois, jours) de *ListEnfant.DateNais* et #20/06/2011#), *MtRepartitionCD.DateOuvDroits*, *MtRepartitionCD.MtPCEL*, *MtPRD.MP*, *MtPRD.MtPCC*

Ranger *TexteFixeCD.TexteFixeArticle3* dans *TexteArticle3*

Construire *TexteArticle3*, type "Article troisième Capital-Décès" à partir de *TexteFixePRD.TexteFixeArticle3*

Construire *TexteArticle4*, type "Article quatrième Capital-Décès" à partir de *TexteFixePRD.TexteFixeArticle4*

Construire *TexteArticle5*, type "Article quatrième Capital-Décès" à partir de *TexteFixePRD.TexteFixeArticle5*

Construire *TexteArticle6*, type "Article quatrième Capital-Décès" à partir de *TexteFixePRD.TexteFixeArticle6*

iii) Le résultat "Résultat"

Désignation	éléments
<i>Results: Projet d'acte de Pension de réversion.views</i>	$\left\{ \begin{array}{l} \text{TexteEntete}\%String, \text{TexteVisas}\%string, \\ \text{TexteArticle1}\%String, \text{TexteArticle2}\%String, \\ \text{TexteArticle3}\%String, \text{TexteArticle4}\%String, \\ \text{TexteArticle5}\%String, \text{TexteArticle6}\%String, \\ \text{TexteAmpliations}\%String, \text{TexteLieu}\%String \end{array} \right\}$

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;

- C_a . *Contraintes* = a . *Contraintes* ;
- C_a . *Importance* = a . "Priorité de l'attente".

Cas 11 : Nous considérons que $a = ElaFichTraitDossCD$ (élaboration de la fiche de traitement du dossier d'attribution d'un capital-décès et une pension de réversion aux ayant-droits, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Importance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
<i>Namespace.agregat</i>	$\left\{ \begin{array}{l} ActeIntégration\%Something :artifact, \\ TexteFixeFichTraitCD\%Something \text{ fait référence à} \\ \text{template des parties fixes de la fiche de traitement,} \\ DernierActe\%Something \text{ fait référence à Dernier acte} \\ \text{d'avancement} \end{array} \right\}$

Désignation	éléments
<i>DernierActe.views</i>	$\left(\begin{array}{l} Indice Solde\%Nombre, Nom du concerné\%String \text{ fait} \\ \text{référence à Nom du défunt dans Copie de l'acte de décès,} \\ SituationAdmin \text{ fait référence à Situation administrative} \\ \text{dans Dernier acte d'avancement, Grade\%String fait} \\ \text{référence à grade dans Dernier acte d'avancement} \end{array} \right)$
<i>ActeIntégration.views</i>	$\left(\begin{array}{l} SituationCivile\%String \text{ fait référence à Situation civile dans} \\ \text{Intégration, Nom du concerné\%String fait référence à Nom} \\ \text{du défunt dans Copie de l'acte de décès} \end{array} \right)$
<i>TexteFixeFichTraitCD.views</i>	$\left(\begin{array}{l} TimbreFixe\%String, TitreFixe\%string, LibelleFixe\%String, \\ ServiceFixe\%String, TableauFixe\%String, \\ MiseEngarde\%String \end{array} \right)$

Phase 2 : Définition de la partie Description "*Description*"

i) La garde :

$(\vee, (ut, DossierCapitalDécès."Nombre de Conjoints", '0'), (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0'))$

ii) La séquence :

Construire *Timbre*, type "*Timbre Fiche de Traitement Capital-décès*" à partir de *TexteFixeFichTraitCD.TimbreFixe*

Construire *Titre*, type "*Titre Fiche de Traitement Capital-décès*" à partir de *TexteFixeFichTraitCD.TitreFixe*

Construire *TexteFixeFichTraitCD.LibelleFixe*, type "*Libellé Fiche de traitement Capital-décès*", zone (*Nom du concerné, grade, numéros*) à partir de *ActeIntégration."nom du concerné", DernierActe.Grade, ":"*

Ranger TexteFixeFichTraitCD.LibelleFixe dans Libelle

Construire *Service*, type "*Service Fiche de Traitement Capital-décès*" à partir de *TexteFixeFichTraitCD.ServiceFixe*

Construire *Tableau*, type "*Tableau Fiche de Traitement Capital-décès*" à partir de *TexteFixeFichTraitCD.TableauFixe*

Construire *MiseEngarde*, type "*Mises en garde Fiche de Traitement Capital-décès*" à partir de *TexteFixeFichTraitCD.MiseEngarde*

iii) Le résultat "*Résultat*"

Désignation	éléments
<i>Results: Fiche de traitement.views</i>	{ <i>Timbre %String, Titre%string, Libelle%String, Service%String, Tableau%String, MiseEngarde%String</i> }

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 12 : Nous considérons que $a = ElaFichVisaDossCD$ (élaboration de la fiche des visas du dossier d'attribution d'un capital-décès et une pension de réversion aux ayant-droits, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Inportance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
<i>NameSpace.agregat</i>	{ <i>ActeIntégration%Something :artifact, TexteFixeFichVisaCD%Something fait référence à template des parties fixes de la fiche de traitement, DernierActe%Something fait référence à Dernier acte d'avancement</i> }

Désignation	éléments
-------------	----------

<i>DernierActe.views</i>	(<i>Indice Solde%Nombre, Nom du concerné%String fait référence à Nom du défunt dans Copie de l'acte de décès, SituationAdmin fait référence à Situation administrative dans Dernier acte d'avancement, Grade%String fait référence à grade dans Dernier acte d'avancement</i>)
<i>ActeIntégration.views</i>	(<i>SituationCivile%String fait référence à Situation civile dans Intégration, Nom du concerné%String fait référence à Nom du défunt dans Copie de l'acte de décès</i>)
<i>TexteFixeFichVisaCD.views</i>	(<i>TimbreFixe%String, TitreFixe%string, LibelleFixe%String, DateEntréesSortiesFixe%String, TableauFixe%String, TexteFixeUnité%String, TexteFixeTitreSignataire%String, TexteFixePiecesJointes%String</i>)

Phase 2 : Définition de la partie Description "**Description**"

i) La garde :

(*v, (ut, DossierCapitalDécès."Nombre de Conjoints", '0'), (ut, DossierCapitalDécès."Nombre d'enfants mineurs", '0')*)

ii) La séquence :

Construire *Timbre*, type "*Timbre Fiche de Traitement Capital-décès*" à partir de *TexteFixeFichVisaCD.TimbreFixe*

Construire *Titre*, type "*Titre Fiche des visas Capital-décès*" à partir de *TexteFixeFichVisaCD.TitreFixe*

Construire *TexteFixeFichVisaCD.LibelleFixe*, type "*Libellé Fiche de traitement Capital-décès*", zone (*Nom du concerné, grade, numéros*) à partir de *ActeIntégration."nom du concerné", DernierActe.Grade, ".:"*

Ranger *TexteFixeFichVisaCD.LibelleFixe* dans *Libelle*

Construire *DateEntréesSorties*, type "*date d'entrée et de sortie dans les services du CF de la Fiche de visas Capital-décès*" à partir de *TexteFixeFichVisaCD.DateEntréesSortiesFixe*

Construire *Tableau*, type "*Tableau de la Fiche des visas Capital-décès*" à partir de *TexteFixeFichVisaCD.TableauFixe*

Construire *TexteUnité*, type "*Unité administrative de la Fiche de visas Capital-décès*" à partir de *TexteFixeFichVisaCD.TexteFixeUnité*

Construire *TitreSignataire*, type "*Titre du signataire Fiche de visas Capital-décès*" à partir de *TexteFixeFichVisaCD.TexteFixeTitreSignataire*

Construire *TextePiecesJointes*, type "*Pièces jointes de la Fiche de Traitement Capital-décès*" à partir de *TexteFixeFichVisaCD.TexteFixePiecesJointes*

iii) Le résultat "**Résultat**"

Désignation	éléments
<i>Results: Fiche des visas.views</i>	{ <i>Timbre %String, Titre%string, Libelle%String, DateEntréesSorties%String, Tableau%String, TexteUnité%String, TitreSignataire%String, TextePiecesJointes%String</i> }

Etape 2 : déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

Cas 13 : Nous considérons que $a = AppretDossCD$ (apprête le dossier le dossier d'attribution d'un capital-décès et une pension de réversion aux ayant-droits, cf. : Annexe 4), et $C_a = (Contexte, but, Règle, contraintes, Inportance)$, un fragment de connaissance pris dans les mêmes conditions que dans le cas 1. La Construction de C_a à partir de a se fait ainsi qu'il suit :

Etape 1 : Formalisation de la règle métier

Phase 1 : définition du contexte "*Contexte*"

Keywords =

GRH, Liquidation des droits, solde

Data =

Désignation	éléments
<i>Namespace.agregat</i>	$\left\{ \begin{array}{l} FichTrait\%Something :artifact, \\ FichVisa\%Something :artifact, ProjActPRD\%Something \\ \text{fait référence à Projet d'acte de Pension de réversion,} \\ ProjActCD\%Something \text{ fait référence à Projet d'acte de} \\ \text{capital-décès, Récép}\%Something :artifact, \\ Récépissé\%Something :artifact, EGS\%Something : \\ \text{control} \end{array} \right\}$

Désignation	éléments
<i>FichTrait.views</i>	(<i>Libelle%String</i> fait référence à <i>Libelle</i> dans <i>Fiche de traitement</i>)
<i>FichVisa.views</i>	(<i>Libelle%String</i> fait référence à <i>Libelle</i> dans <i>Fiche des visas</i>)
<i>Récép.views</i>	(<i>NumRecept%string</i> fait référence à <i>Numéro</i> dans <i>Récépissé</i>)

Phase 2 : Définition de la partie Description "*Description*"

i) La garde :

(\vee , (*ut*, *DossierCapitalDécès."Nombre de Conjoints"*, '0'), (*ut*, *DossierCapitalDécès."Nombre d'enfants mineurs"*, '0'))

ii) La séquence :

Construire *FichTrait.Libelle*, type "*Libellé Fiche de Traitement Capital-décès*", zone ("*Numéro*") à partir de *FichTrait.Libelle*, *Récép.NumRecept*

Ranger *FichTrait.Libelle* dans *FichVisa.Libelle*

Editer *FichTrait*

Editer *FichVisa*

Editer *ProjtActCD*

Editer *ProjtActPRD*

Ranger "dossier prêt" dans Etat du dossier

iii) Le résultat "*Résultat*"

Désignation	éléments
<i>Results: Etat.views</i>	{ <i>Etat du dossier%Nombre</i> }

Etape 2: déduction du fragment de connaissance C_a

Le fragment de connaissance C_a est obtenu ainsi :

- $C_a.Contexte = "Contexte"$;
- $C_a.But = a.But$;
- $C_a.Règle = "Règle"$;
- $C_a.Contraintes = a.Contraintes$;
- $C_a.Importance = a."Priorité de l'attente"$.

IV.1.2 ELABORATION DU MODELE DE REFERENCE

L'élaboration du modèle est triviale, il suffit d'utiliser les axiomes 3.19 et 3.20. Une représentation graphique du modèle des besoins obtenus est illustrée par les figures 19 et 20. La différence entre ces deux figures ne réside qu'au niveau des nœuds. La *Figure 19* a des nœuds formés par les buts alors la figure 20 a des nœuds formés des noms de fragments de connaissances.

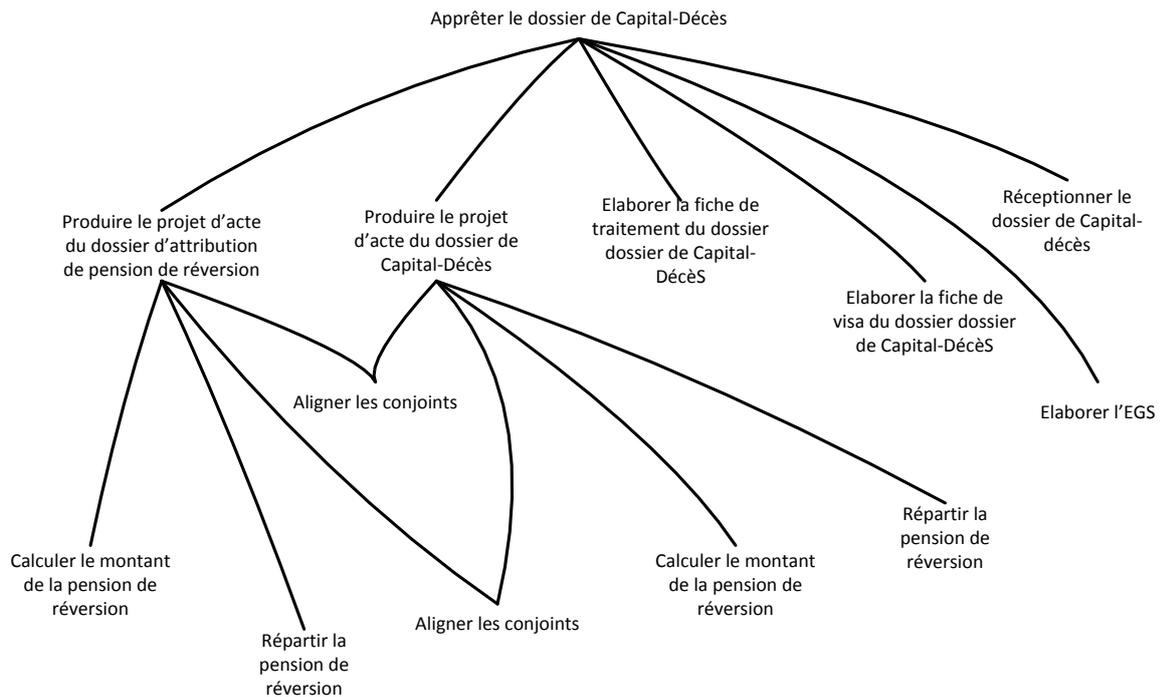


Figure 22 : Illustration à base de buts du modèle de référence obtenus pour un dossier d'attribution d'un capital-décès et d'une pension de réversion

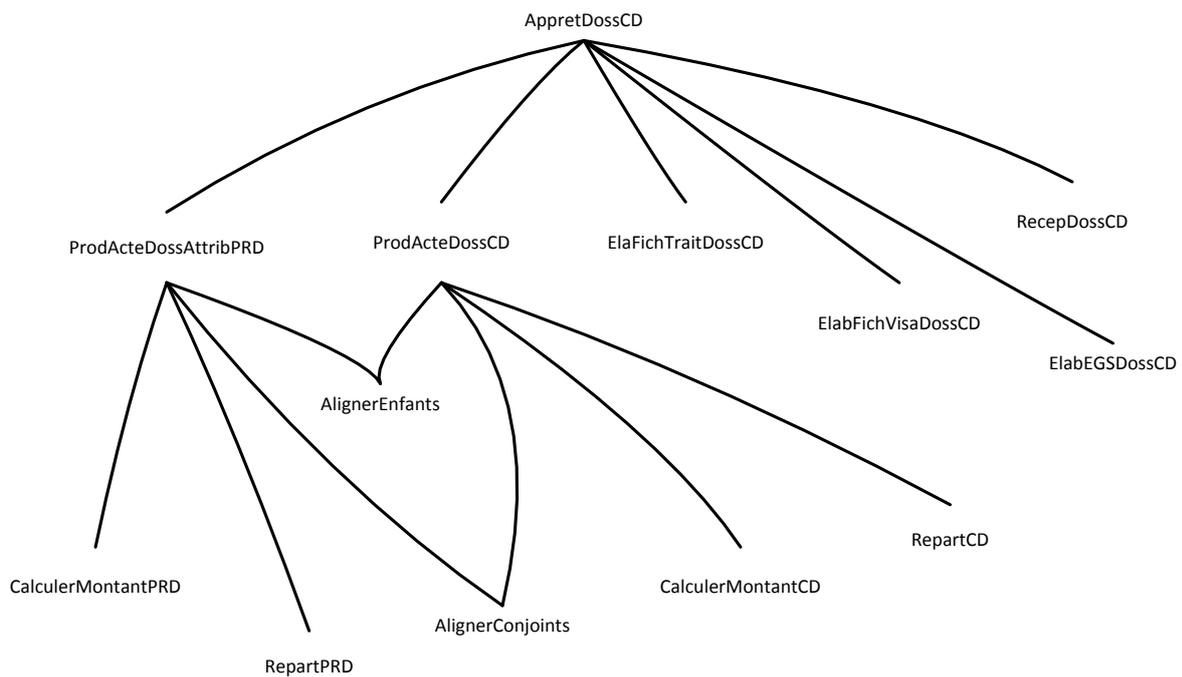


Figure 23 : Illustration à base de fragments de connaissances du modèle de référence obtenus pour un dossier d'attribution d'un capital-décès et d'une pension de réversion

IV.2 CONSTRUCTIONS DES INDICATEURS DE PERFORMANCES.

La construction des indicateurs de performances nécessite la construction au préalable du modèle métier. Le modèle métier permet d'obtenir une structuration hiérarchique des fragments de connaissances. Cette structuration nous permet de déduire l'ensemble des tâches nécessaires à la réalisation du but global ou l'ensemble de tâches du processus d'affaire d'une part, les tâches associées à chaque fragment de connaissance d'autre part. C'est sur ces tâches que sont rattachés les observateurs de performances.

IV.2.1 TRANSFORMATION DU FRAGMENT DE CONNAISSANCES EN UNE TACHE.

Cette activité vise la transformation d'un fragment de connaissance en tâche compatible formalisme présenté par Atsa Etoundi Roger dans [1]. La figure 21 illustre la structuration de l'ensemble des tâches obtenues, après application de la règle de transformation des fragments de connaissances en tâches.

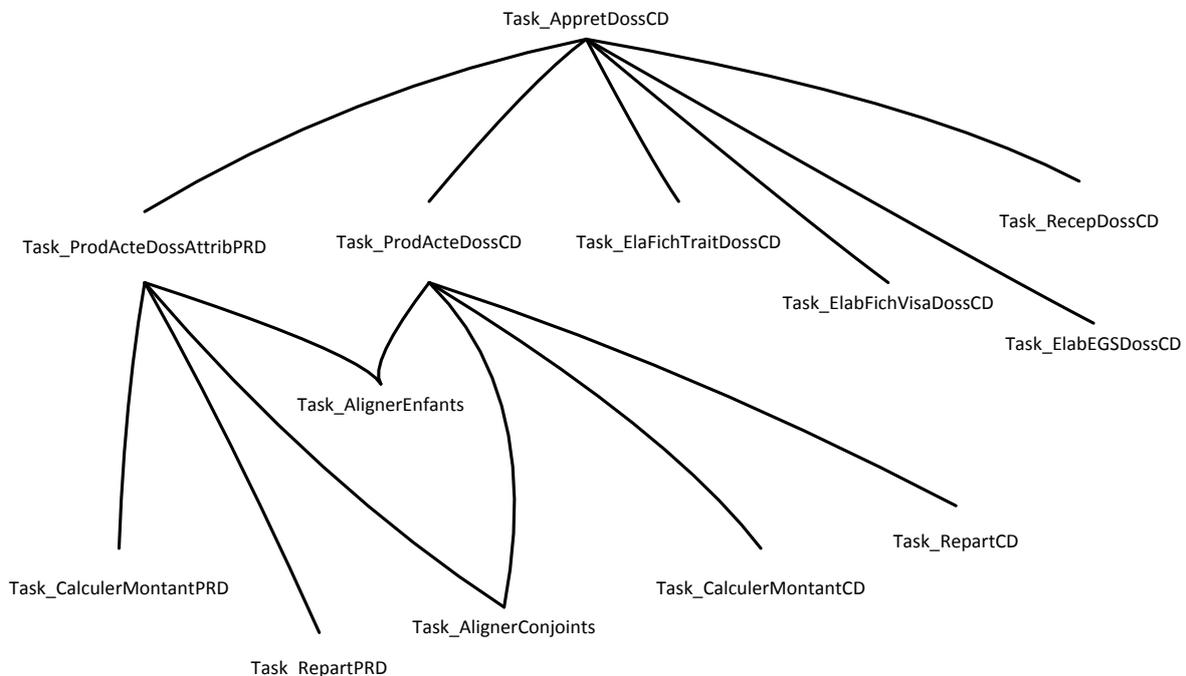


Figure 24 : Illustration à base de tâches du modèle métier obtenus.

IV.2.2 DEMARCHE

IV.2.2.1 CONSTRUCTION DES OBSERVATEURS

Dans cette partie, il est simplement question d'associer à chaque tâche un ensemble de variables appelées observateurs. Nous avons, dans la section II, défini l'ensemble de facteurs de qualités. Il est important de souligner qu'un facteur de qualité peut être aussi appelé critères de qualité. Ainsi, en lieu et place de facteurs de qualités, nous utiliserons de temps à autres la terminologie critère de qualité. La construction des observateurs associés à chaque tâche se fait ainsi qu'il suit :

Etape 1 : identification des tâches associées aux buts opérationnels ou tâches

terminales

Cette identification se fait sur la base du modèle illustré par la figure 21. Dans la figure 22, les tâches terminales sont celles qui se trouvent dans les cases de couleur verte.

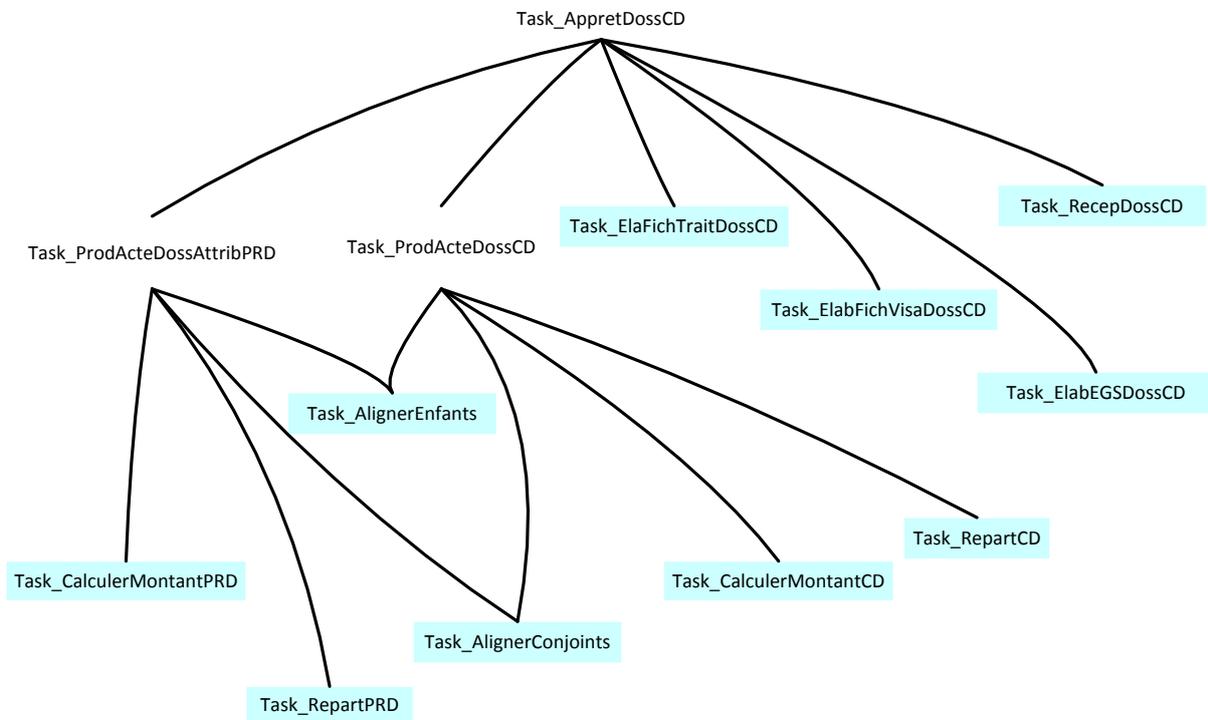


Figure 25 : Illustration de l'identification des tâches terminales.

Etape 2 : association des observateurs aux tâches

Dans cette étape, on choisit arbitrairement des variables et on les associe à des tâches. L'attribution des noms d'observateurs est arbitraire. La figure 23 ci-dessous illustre les observateurs définis pour chaque tâche.

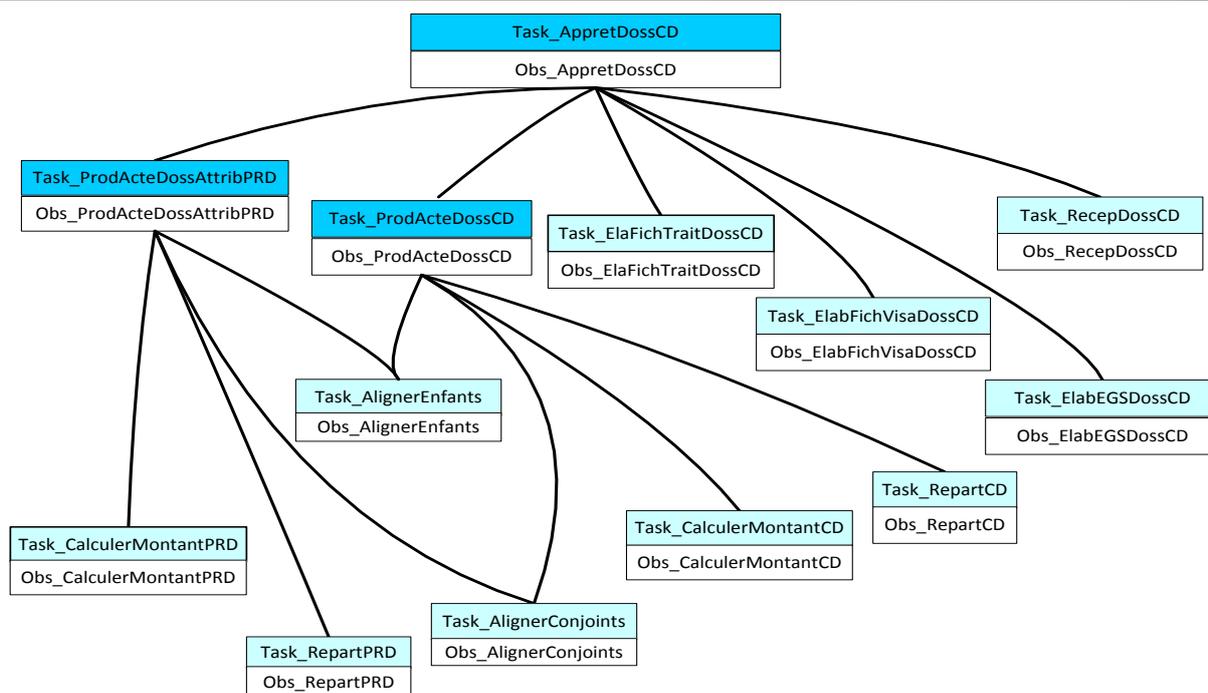


Figure 26 : Illustration à base de tâches du modèle de métier obtenu enrichi d'observateurs.

Dans la figure ci-dessus, les tâches sont représentées par des cases. Chacune de ces cases comporte deux parties :

- pour les tâches terminales : la partie de la case coloriée en turquoise clair contient le nom de la tâche ; celle coloriée en blanc, contient le nom de l'observateur ;
- pour les autres tâches: la partie de la case coloriée en bleu ciel contient le nom de la tâche ; celle coloriée en blanc, contient le nom de l'observateur

Etape 3 : définition des fonctions d'évaluations pour les tâches terminales

Dans la feuille de route du Ministre de la fonction Publique et de la Réforme Administrative, il avait été demandé qu'un dossier de liquidation des droits soit traité en trente (30) jours au plus. Dans les Manuels de Procédures administratives(MPA), les phases de préparation des projets d'actes de liquidation des droits couvrent au plus cinq (05) jours. Le rendement d'un agent à la sous direction de la liquidation des droits se mesure en fonction du nombre de dossiers traités par jours. En moyenne, dans cette sous direction, un cadre métier doit apprêter 30 dossiers complets⁸ par jours. L'élaboration des états généraux de service ne relève pas de la sous direction de la liquidation des droits, mais plutôt de la sous direction du fichier central. Dans cette sous direction, un agent élabore en moyenne 15 états généraux de services. De plus, le temps mis pour qu'un lot de dossiers⁹ (ensemble des dossiers de la sous direction de la liquidation des droits pour lesquels les états généraux de services sont attendus) parte de la sous direction de la liquidation des droits à la sous direction du fichier central est de trois (03) jours. De plus deux personnels sont affectés à la

⁸ Un dossier complet est un dossier dans lequel l'utilisateur a joint tous les documents requis.

⁹ Le lot de dossiers peut comporter autant de dossiers que possible

confection des actes de liquidations des droits des personnels fonctionnaires et cinq personnels de la sous direction du fichier central sont affectés à la confection des états généraux de service y afférant.

Compte tenu de toutes ces spécificités, nous définissons les fonctions d'évaluation associées aux observateurs des tâches terminales ainsi que les seuils correspondants tel que présenté dans le tableau ci-dessus.

Tableau 4.2 : définition des fonctions d'évaluation et des seuils

Tâches terminales	Observateur	Seuil de satisfaction (en dossiers /jour)	Fonction d'évaluation
Task_CalculerMontantPRD	Obs_CalculerMontantPRD	30	<i>Considérons un observateur o, associé à une tâche terminale X. $f_{eval}(o)$ retourne le nombre de fois que la tâche X a été exécutée avec succès en une journée.</i>
Task_RepartPRD	Obs_RepartPRD	30	
Task_AlignerEnfants	Obs_AlignerEnfants	60	
Task_AlignerConjoints	Obs_AlignerConjoints	60	
Task_CalculerMontantCD	Obs_CalculerMontantCD	30	
Task_RepartCD	Obs_RepartCD	30	
Task_ElaFichTraitDossCD	Obs_ElaFichTraitDossCD	30	
Task_ElabFichVisaDossCD	Obs_ElabFichVisaDossCD	30	
Task_ElabEGSDossCD	Obs_ElabEGSDossCD	15	
Task_RecepDossCD	Obs_RecepDossCD	30	

Pour ce qui est des tâches intermédiaires ou la tâche racine, l'axiome 3.3, défini comment évaluer ou déduire la valeur de l'observateur à elles associées. Dans la suite, cette règle de calcul sera dénotée par f_{cal} . Ainsi, pour un observateur quelconque o , d'une tâche intermédiaire ou racine, $f_{cal}(o)$, dénote la valeur de l'indicateur associé à cet observateur.

Etape 4 : association des observateurs aux Objets métiers

Nous avons dans l'étape précédente, défini un ensemble d'observateurs associés aux différentes tâches de notre processus d'affaire. Ces observateurs sont destinés à l'évaluation du rendement de chaque cadre métier intervenant dans la confection des projets d'actes de liquidation des droits. Or, il peut arriver qu'un dossier soit réceptionné sans toutefois avoir la suite conséquente à lui réservée. Pour détecter ces cas, il est recommandé d'adjoindre un observateur de performance audit objet. Dans le cas d'espèce, l'objet tracé est : "Dossier de capital-décès" L'adjonction d'un observateur de performance op au "Dossier de capital-décès" se fait de la manière suivante :

$$op = \langle GRH, "Dossier de capital - décès", \{Obs_temps_mis\} \rangle$$

Où : Obs_temps_mis est l'observateur chargé de recueillir les informations sur le temps que le dossier a mis au sein de la sous-direction de la liquidation des droits.

L'indicateur de performance $f_{ind}(\{Obs_temps_mis\})$ associé à op représente le temps

que le dossier de capital-décès a mis dans ladite sous-direction. Sa valeur s'exprime en terme de jours. C'est un indicateur d'alerte. Le seuil de satisfaction qui lui est associé est de cinq (05) jours. Ce seuil de satisfaction peut être réajusté en fonction de ce que l'on aimerait faire.

IV.2.2.2 CONSTRUCTION DES FACTEURS DE PERFORMANCES

La figure 24 ci-dessous, illustre à merveille les facteurs de qualités associés aux tâches terminales. Les cases représentent les facteurs de performances. Chacune de ces cases comporte trois rubriques :

- la première rubrique en bleu clair ou en turquoise clair, représente l'indicateur de performance,
- la deuxième rubrique matérialise le poids du facteur de performance,
- la troisième rubrique, dénote la tâche à laquelle est associé ce facteur de performance.

Les cases colorées en bleu clair représentent les tâches intermédiaires ou racine tandis que celles colorées en turquoise clair représentent les tâches terminales.

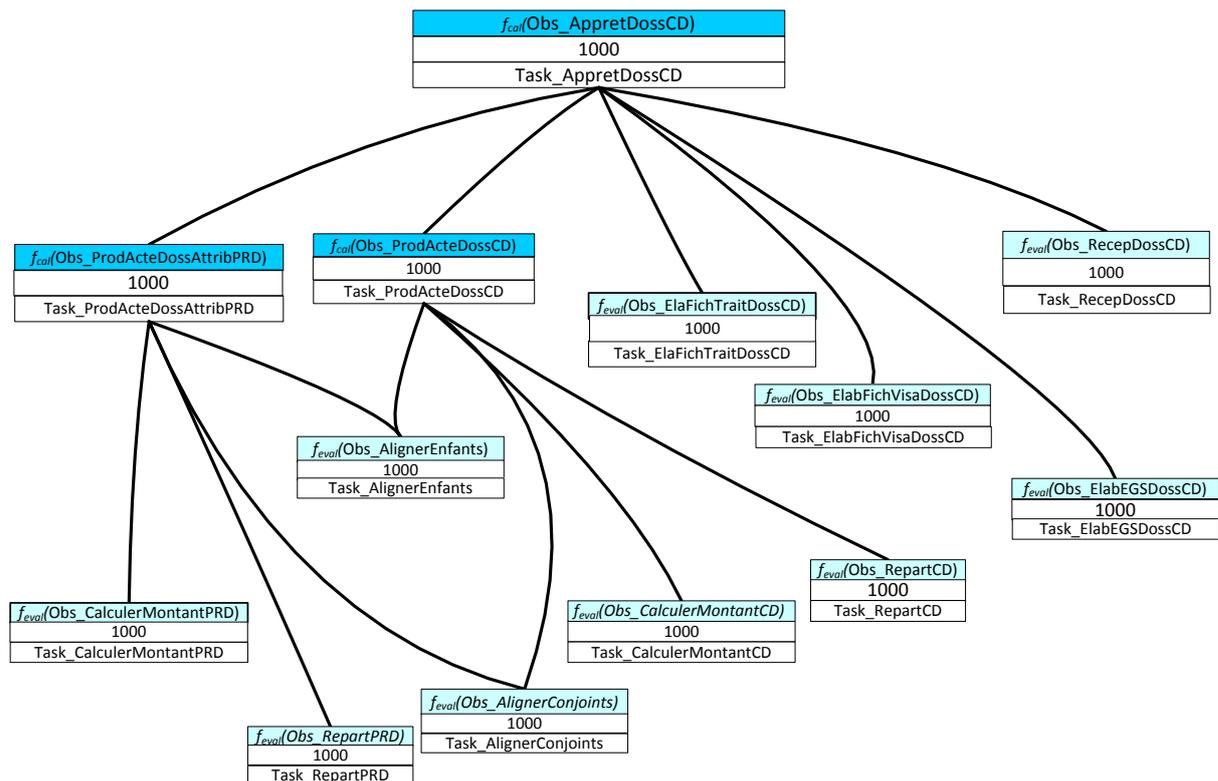


Figure 27 : Illustration des facteurs de performance associés aux tâches terminales construites.

Si on avait plus d'un facteur de qualité ayant des poids différents, nous devions aussi avoir plus d'un observateur par tâche et le poids associé à un facteur de performance allait varier selon que le facteur de qualité est associé ou non à un facteur de qualité.

Par ailleurs, il est important de souligner que tous les indicateurs sont de même

type dans la mesure où il a été considéré un seul facteur/critère de qualité.

IV.3 JONCTION DE PRÉOCCUPATIONS.

Dans cette partie il sera question d'enrichir les règles métiers d'observateurs permettant l'évaluation la performance de satisfaction du but de la règle métier. Cette jonction ne modifie pas la hiérarchie des fragments de connaissance. Elle permet d'enrichir les fragments de connaissances des observateurs de performance et les axiomes permettant de mettre à jour la valeur de l'observateur. Dans le cadre de l'élaboration du modèle des besoins d'un processus d'affaires, on se limite à la déclaration de ces observateurs. Cette déclaration se passe dans le contexte de la règle métier et plus spécifiquement entre les suffixes « agrégat » et « views ». la déduction de la description de cet observateur est triviale.

Pour les observateurs associés aux objets métiers, ils sont déclarés dans la règle métier dans laquelle est déclaré ledit objet métier. Les observateurs de performance sont déclarés à l'intérieur du suffixe « views » de l'objet métier.

V. VALIDATION DU MODELE

La validation du modèle vise la vérification de la cohérence entre les différents concepts intervenant dans l'élaboration du modèle des besoins après son enrichissement par les concepts de la satisfaction des bénéficiaires de service. Dans le cas d'espèce, les fragments de connaissance *AlignerEnfants* et *AlignerConjoints* ont le même niveau d'importance que les autres fragments de connaissance du même père. Cette valeur du niveau d'importance est incohérente avec le seuil de satisfaction. Ainsi le niveau d'importance de *AlignerEnfants* et *AlignerConjoints* pas à *très très important* ou en valeur numérique 60.

VI. CONCLUSION

Notre travail, dans ce chapitre a porté essentiellement sur une trace d'exécution de notre processus d'ingénierie des besoins visant à construire un modèle de référence compatible au formalisme de Farida Semmak et Joël Brunet et à traduire le modèle obtenu au formalisme de modélisation des processus d'affaire développé par Dr. Roger Atsa Etoundi. Durant cette phase, nous avons revisité tous les concepts utilisés dans le cadre du processus d'ingénierie sus évoqué. Au terme de cet exercice, nous nous sommes rendu compte que le processus d'ingénierie proposé permettait de décrire aisément le métier d'une organisation. Ce travail ci-dessus abattu nous a permis au sein de la Division Informatique de déceler des problèmes dans l'implémentation du système informatique qu'utilise l'état camerounais pour gérer la carrière des fonctionnaires.

**CHAPITRE VI:
CONCLUSION GENERALE
ET PERSPECTIVES**

Concilier l'expression des besoins des organisations à la production des modèles de références des systèmes informatiques c'est-à-dire définir les règles de passage du formalisme de Farida Semmak et Joël Brunet au formalisme d'Atsa Etoundi Roger et Marcel Fouda d'une part, et d'enrichir le formalisme de Farida Semmak et Joël Brunet d'un processus d'ingénierie des besoins qui permet de partir des besoins des organisations (attentes des cadres métiers de l'organisation et critères/facteurs de qualité des bénéficiaires des services offerts) pour produire un modèle de référence d'autre part. C'est l'activité qui a été l'objet de cette thèse. Nous avons par conséquent défini un processus d'ingénierie de besoins approprié. Compte tenu du fait que les travaux relatés dans [3], étaient axés ingénierie des lignes des produits, nous les avons adaptés aux contextes du développement sur mesure qui sied bien avec la modélisation des processus d'affaire. De plus, la satisfaction des bénéficiaires des services n'étant pas prise en compte dans [3], nous l'avons introduit. Cette façon de procéder remet le client/utilisateur et bénéficiaires des services au centre du processus d'ingénierie. L'approche proposée dans cette thèse comporte quatre (04) étapes fondamentales : le recensement, la sélection, l'élaboration du modèle, et la validation.

L'étape de recensement permet de collecter les besoins des cadres métiers de l'organisation et les critères de qualités des clients de cette organisation. Elle offre comme résultats l'ensemble des documents de recensement des besoins et celui des facteurs/critères de qualité.

L'étape de sélection permet de retenir ceux des besoins des cadres métiers de l'organisation qui sont consistants. Elle fournit comme résultats l'ensemble des besoins consistants de l'organisation.

L'étape d'élaboration du modèle permet de construire le modèle de référence à partir de l'ensemble des besoins consistants. Cette élaboration en elle-même est un processus à six phases regroupées sous deux catégories : la modélisation du métier d'une organisation et la modélisation de la satisfaction des bénéficiaires des services de cette organisation. Les deux catégories constituent deux axes de préoccupations

que nous avons dénommés axes de préoccupations relatives au métier pour la modélisation du métier et axes de préoccupations relatives à la satisfaction des bénéficiaire de service pour la modélisation de la satisfaction des bénéficiaires de service. La première catégorie couvre la première et la deuxième phase tandis que la deuxième catégorie, couvre la troisième, la quatrième et la cinquième phase. La première phase est réservée à la transformation des besoins retenus lors de l'étape de sélection en fragments de connaissance. Elle vise à formaliser la règle métier d'un besoin retenu dans le modèle de description des règles métiers élaboré à cet effet. La deuxième phase est dédiée à l'élaboration d'un modèle métier. Ce dernier est construit à partir de ces fragments de connaissance obtenus à la phase précédente et n'intègre pas les aspects de modélisation de la satisfaction des clients de l'organisation. Le modèle métier ainsi obtenu constitue ce que nous appelons un axe de préoccupations métiers. La troisième phase est destinée à la déduction du graphe de dépendance entre les tâches du processus d'affaire à partir du modèle métier produit à la phase précédente. Elle marque le début de la modélisation de la satisfaction de la clientèle. La quatrième phase vise l'adjonction des observateurs et des seuils de satisfactions aux tâches constituant le graphe de dépendance résultat de la phase précédente ou aux objets métiers d'une part, la définition des fonctions d'évaluation, par critère de qualité, associées auxdits observateurs, d'autre part. La cinquième phase vise la construction des observateurs de performances associés à chaque tâche, à partir des résultats de la phase précédente. Le résultat obtenu constitue un axe de préoccupations relatives à la modélisation de la satisfaction des bénéficiaires de service. Elle met fin à la modélisation de la satisfaction de la clientèle. La dernière phase est axée sur la jonction de ces préoccupations c'est-à-dire, l'intégration des deux axes de préoccupations. Dans cette optique, il est défini des points d'échange entre les deux axes de préoccupations. Ce n'est qu'après cette étape qu'on obtient le modèle de référence.

L'étape de validation quant à elle permet la validation du modèle ainsi obtenu. Dans cette étape, il est question de se rassurer de la cohérence des différents concepts. Pour cela, il a été défini un éventail d'axiomes à partir desquels s'appuie cette validation.

Ce travail couvre trois activités de l'ingénierie des besoins que sont : l'explicitation, la sélection et la validation des besoins étant entendu que la spécification desdits besoins a été définie dans la littérature. Naturellement, pour arriver au résultat escompté, des enrichissements ont été apportés à notre modèle de spécification de base proposé dans la littérature par Farida Semmak et Joël Brunet. Ces enrichissements concernent le fragment de connaissance. En effet, nous avons substitué le fragment conceptuel par trois composantes : «le contexte », le « niveau d'importance », les « contraintes ». Le « contexte », première composante du fragment de connaissance enrichi, décrit l'environnement d'utilisation du fragment de connaissance ; le « niveau d'importance », dernière composante du fragment de connaissance enrichi, traduit l'importance que revêt cette connaissance dans la chaîne de production du processus d'affaire ; et les « contraintes », quatrième composante du fragment de connaissance enrichi, représentent les besoins non fonctionnels associés à l'objectif visé ou but. Nous avons au niveau de la modélisation de la satisfaction des clients d'une organisation effectué un travail ontologique, les concepts de points de vue, facteurs de qualités, Observateur, Observation, Indicateurs de performances, Observateur de performance ont été élucidés et une représentation formelle a été proposée. Des résultats connexes ont été obtenus notamment, une approche de spécification formelle des règles métiers qui permet de manipuler les règles métiers avec la même aisance que pour les types numériques ; une ontologie permettant de construire les composants métiers orientés aspects. Un composant métier orienté aspect est une unité formée par la composition de plusieurs aspects métiers. Un aspect métier est un aspect obtenu après prise en compte de toutes les préoccupations encapsulées par les différentes catégories de prédicats d'une règle métier.

Cependant, nous n'avons pas mis un accent particulier sur une représentation formelle contraintes non fonctionnelles car cela fait l'objet des études en cours dans notre groupe de travail. Nous nous sommes limités à définir un ensemble de concepts permettant d'introduire les métriques de qualité dans la modélisation des organisations. Ce travail nous a permis d'ouvrir de nouveaux pans de recherches

notamment sur :

- L'approche de modélisation par triangulaire. C'est une approche de modélisation qui stipule, que la modélisation des processus d'affaire, pour qu'elle soit complète, doit prendre en compte la description des tâches, les objectifs du système et la satisfaction des clients. Chacune de ces préoccupations doit être régie par un processus d'ingénierie. Seul le processus d'ingénierie relatif aux besoins du monde réel a été étudié. Les autres feront l'objet des recherches ultérieures.

A l'issue de ces travaux, nous avons obtenu comme résultats, la publication de sept articles permettant de faire valider par la communauté scientifique, les différentes idées qui sont à la base de ce travail. Les résultats de ces travaux peuvent être utilisés aussi bien dans le domaine d'ingénierie des besoins, que dans l'ingénierie dirigée par les modèles ; ou alors dans le développement d'applications à base de composants logiciels. Ce qui explique cet état de chose, c'est que les travaux qui ont été effectués dans ce cadre sont en amont de tous ces domaines de recherche sur le développement d'applications.

En perspectives, ce travail pourra être :

- Enrichi d'une méthode permettant l'extraction de connaissance, dans une base de connaissances, permettant de satisfaire un besoin spécifique.
- Enrichi par le développement d'une plateforme de modélisation des processus d'affaires;
- Etendu pour développer les systèmes auto-adaptatifs dans la mesure où les besoins auront été bien captés;
- Adapté à la sélection des composants logiciels.

RÉFÉRENCES

- [1]. Atsa Etoundi Roger, thèse, A domain approach for a multi-perspectives workflows modeling, Université de Yaoundé I, 2004
- [2]. R. Atsa Etoundi, M. Fouda Ndjodo, « Feature-Oriented Business Process and Workflow », IEEE SITIS pp 114-121, 2005.
- [3]. Farida Semmak, Joël Brunet, « Un métamodèle orienté buts pour spécifier les besoins d'un domaine », 23e Congrès INFORSID, pp 115-132, mai 2005.
- [4]. Ken Decreus and Geert Poels, "A Goal-Oriented Requirements Engineering Method for Business Processes", INFORMATION SYSTEMS EVOLUTION, Lecture Notes in Business Information Processing, Volume 72, pp: 29-43, 2010
- [5]. Valérie Emin-Martinez, Modélisation dirigée par les intentions pour la conception, le partage et la réutilisation de scénarios pédagogiques, thèse, Université de Joseph Fourier, 2010
- [6]. Emmanuel Letier, Jeff Kramer, Jeff Magee , Sebastian Uchitel, "Deriving Event-Based Transition Systems from Goal-Oriented Requirements Models", Journal of Automated Software Engineering, volume 15, pp 175-206, 2008
- [7]. Anis Ferchichi, thèse, « contribution a l'intégration des processus métiers : application a la mise en place d'un référentiel qualité multi-vues», Université Bordeaux I, 2008
- [8]. Jacqueline Konaté, thèse, « approche système pour la conception d'une méthodologie pour l'élicitation collaborative des exigences », Université Toulouse 3 Paul Sabatier, 2009
- [9]. Atsa Etoundi Roger, Fouda Ndjodo Marcel, Atouba Christian Lopez, A Goal Oriented Approach for the Definition of a Business Process Requirement Model. International Journal of Computer Applications 9(7):1-7, 2010.
- [10]. Roger Atsa Etoundi, Marcel Fouda Ndjodo, Atouba Christian Lopez, A Goal Based Approach for QFD Refinement in Systematizing and Identifying Business Process Requirements, International Journal of Computer Science Issues, Vol. 7, Issue 6, pp 343-350, 2010.

RÉFÉRENCES

- [11]. Atsa Etoundi Roger, Fouda Ndjodo Marcel and Atouba Christian Lopez, A Model based Business Process Requirement Rule Specification. *International Journal of Computer Applications* 11(9):17–24, 2010.
- [12]. Atsa Etoundi Roger, Fouda Ndjodo Marcel, Atouba Christian Lopez, Business Process Requirement Engineering. *International Journal on Computer Science and Engineering*, volume 2, n° 9, 2010.
- [13]. Mouhamed Diouf, « Spécification Et Mise En Œuvre D'un Formalisme De Règles Métier » , thèse n°3507, Université Bordeaux I, décembre 2007.
- [14]. Atsa Etoundi Roger, Fouda Ndjodo Marcel, Atouba Christian Lopez, A Formal Approach for the Inclusion of Key Performance Indicators in a Business Process. *International Journal on Computer Science and Engineering*, volume 3, n° 9, 2011.
- [15]. M. Hammer et J. CHAMPY. *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business Essentials, 2003.
- [16]. Hafedh Mili, Guitta Bou Jaoude, Eric Iefebvre, Guy Tremblay, and Alex Petrenko. *Business process modeling languages : Sorting through the alphabet soup*. Rapport de recherche, D'épartement d'Informatique, UQAM, Janvier 2004.
- [17]. Ismael Ghalimi and David McGoveran. *Standards and BPM*. *Business Integration Journal - Business Process Management Journal*, 2004.
- [18]. Francois Vernadat and L. Hamaidi. *La modélisation en entreprise : Méthodes descriptives des processus opérationnels*. Economica, Paris, 1998.
- [19]. Bernard Debauche and Patrick Megard. *BPM Business Process Management : Pilotage métier de l'entreprise*. Hermes Science Publications, 2004.
- [20]. Mathieu Luras, Hervé Pingaud, and Jacques Lamothe. *Méthodes de diagnostic et d'évaluation de performance pour la gestion de chaînes logistiques : application a la coopération maison-mère filiales internationales dans un groupe pharmaceutique et cosmétique*. These préparée au Centre de Genie Industriel de l'Ecole des Mines d'Albi-Carmaux, 2004.
- [21]. Setrag Khoshafian and Marek Buckiewicz. *Introduction to Groupware, Workflow, and Workgroup Computing*. Wiley, 1995.

RÉFÉRENCES

- [22]. Tom Kimbrough and Linda Levine. The IDEALSM Transition Framework - Speeding Managed Change. Software Engineering Institute (SEI), 1997.
- [23]. M.A. Ould. Business Processes: Modeling and Analysis for re-engineering and improvement. John Wiley & Sons, 1995.
- [24]. David McGoveran. An Introduction to BPM BPMS. Business Integration Journal - Business Process Management Journal, 2004.
- [25]. B. Curtis, M.I. Kellner, and J. Over. Process modeling. Communication ACM, 1992.
- [26]. BPMI Business Process Management Initiative. Business process modeling language (bpml). Novembre 2002.
- [27]. BPMI Business Process Management Initiative. Business process modeling notation (bpmn), specification. Mai 2004.
- [28]. OMG Object Management Group and BPMI Business Process Management Initiative. Business process modeling notation (bpmn), version 1.0. Février 2006.
- [29]. M. Zur Muehlen. Workflow-Based Process Controlling: Foundation, Design, and application of Workflow - driven Process information system. 2002.
- [30]. WFMC. Workflow management coalition terminology and glossary. Technical Report WFMC-TC-1011, Workflow Management Coalition, Brussels, 1999.
- [31]. Dictionnaire en ligne et moteur de recherche des définitions reliées à l'informatique et à l'internet. www.webopedia.com, Mars 2010.
- [32]. Thornton Gale and James Eldred. Getting Results with the Object-Oriented Enterprise Model. SIGS Books, Janvier 1996.
- [33]. Francois Vernadat and L. Hamaidi. La modélisation en entreprise : Méthodes descriptives des processus opérationnels. Economica, Paris, 1998.
- [34]. H. Tardieu, A. Rochfeld, and C. Colett. La méthode merise. 2000.
- [35]. G.W. Brams. Réseaux de Petri : théorie et pratique. Masson, 1983.
- [36]. Michel Diaz. Les réseaux de Petri : modèles fondamentaux. Hermes Science, 2001.
- [37]. G. Doumeingts. La méthode GRAI. Thèse de doctorat d'état, Université Bordeaux I, 1984.

RÉFÉRENCES

- [38]. Wil van der Aalst and Kees Max van Hee. Workflow Management Models, Methods, and Systems. MIT Press, 2002.
- [39]. M. A. Jackson and Graham Twaddle. Business Process Implementation Building Work-flow Systems. Addison Wesley, 1997.
- [40]. Daniel Boeri. MaItriser la qualité: Tout sur la certification et la qualité. Maxima, 2003.
- [41]. Jean-Louis Tomas. ERP et Progiciels intégrés : la mutation des systèmes d'information, Edition. Dunod, 2000.
- [42]. Jean-Noel Gillot. Thèse « La gestion des processus métiers : Aligner les objectifs stratégiques »
- [43]. Zave, P., "Classification of Research Efforts in Requirements Engineering", ACM Computing Surveys, Vol. 29 No. 4, 1997, 3 15-321
- [44]. Rolland, C., Capturing System Intentionality with Maps. In Conceptual Modelling in Information Systems Engineering. Krogstie, J. Opdahl, A.L., Brinkkemper, S. (Eds). Springer, 2007
- [45]. Mylopoulos, J., Chung, L. & Yu, E., "From Object-Oriented to Goal-Oriented Requirements Analysis", Communications of the ACM, Vol. 42 No. 1, January 1999, 31-37.
- [46]. Rolland, C. and Prakash, N. (2000). "From conceptual modelling to requirements engineering". Ann. Software Eng., 10 :151-176.
- [47]. European Software Institute, European User Survey Analysis, Report USV_EUR 2.1, ESPITI Project. (1996)
- [48]. The Standish Group International, Inc. The CHAOS report, 1995.
- [49]. Rolland C., Grosz G., <<De la modélisation conceptuelle a l'ingénierie des besoins>> , chapter 4 in Cauvet C., Rosenthal-sabroux C., Ingénierie des systèmes d'information, editions Hermes, ISBN 2-7462-0219-0, (02/200 1).
- [50]. Leishman, T., R., Cook D. A., Requirements Risks Can Drown Software Projects, Journal of Defense Software Engineering Apr 2002.
- [51]. MARTLNEZ-EMIN Valerie, « Modélisation dirigée par les intentions pour la conception, le partage et la réutilisation de scenarios pédagogiques », Université de Grenoble, 2006

RÉFÉRENCES

- [52]. Van Lamsweerde, A., Requirements Engineering in the year 2000 : A research perspective. 22nd International Conference on Software Engineering, Limerick, Ireland, 2000.
- [53]. [Robertson, S. et Robertson, J. (1999), Mastering the Requirements Process, Addison-Wesley Professional edition.
- [54]. Chung, L., Nixon B.A., Yu E., Mylopoulos J., Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers, Boston (2000)
- [55]. Dardenne, A., Fickas, S., Van Lamsweerde, A., Goal-directed concept acquisition in requirements elicitation, Proc. 6th IEEE Workshop System Specification and DesignO, Como, Italy, 1991, 14-21.
- [56]. A. Van Lamsweerde, R. Dairmont, P. Massonet. Goal Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt, in Proc. Of RE'95 - 2nd Int Symp. On Requirements Engineering, York, IEEE, 1995, pp 194 -204.
- [57]. Yu, E. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97), Washington D.C., USA. pp. 226-235, 1997
- [58]. Rolland C., Prakash N. et Benjamin A., "A Multi-model View of Process Modelling", Requirements Engineering Journal, 1999, p. 169-187
- [59]. Van Lamsweerde, A., "Goal-Oriented Requirements Engineering: A Roundtrip from Research to Practice", Keynote paper, Proc. RE'04, 12th IEEE Joint Intl. Requirements Engineering Conf., Kyoto, Sept. 2004, 4-8.
- [60]. M. Jackson. Software Requirements and Specifications - A Lexicon of Practice, Principles and Prejudices. ACM Press, Addison-Wesley, 1995.
- [61]. Rolland C., Prakash N. et Benjamin A., "A Multi-model View of Process Modelling", Requirements Engineering Journal, 1999, p. 169-187
- [62]. Yu, E., Mylopoulos J. Why Goal-Oriented Requirements Engineering, <http://www.cs.toronto.edu/pub/eric/REFSQ98.htm> 1998
- [63]. Hammer M., and Champy J., Re-engineering the Corporation: A Manifesto for Business Revolution, Harper Collins Publishers, New York. (1993)

RÉFÉRENCES

- [64]. Yu, E., Mylopoulos J. Why Goal-Oriented Requirements Engineering, <http://www.cs.toronto.edu/pub/eric/REFSQ98.htm> 1998
- [65]. J. A. Bubenko, Extending the Scope of Information Modeling, Proc. 4th mt. Workshop on the Deductive Approach to Information Systems and Databases, Lloret-Costa Brava, Catalonia, Sept. 20-22, 1993, pp. 73-98.
- [66]. E. Yu and J. Mylopoulos, "From E-R to 'A-R' -- Modelling Strategic Actor Relationships for Business Process Reengineering," Entity-Relationship Approach (ER '94) -- Business Modelling and Re-Engineering (Proceedings of 13th mt. Conf on the Entity-Relationship Approach, Manchester, UK, December 1994), P. Loucopoulos (Ed.), Lecture Notes in Computer Science no. 881, Springer-Verlag. pp. 548-565.
- [67]. Jacobs and R. Holten, "Goal-Driven Business Modelling - Supporting Decision Making within Information Systems Development," Proc. Conf On Organizational Computing Systems, Milpitas, Calif. 1995, pp. 96-105.
- [68]. Pohl, K. (1996). Process-Centered Requirements Engineering. Taunton, Somerset, England, Research Studies Press Ltd.
- [69]. Kavakli, E. and Loucopoulos P. (2003) Goal Driven Requirements Engineering: Evaluation of Current Methods, the 8th CAiSE/IFIP8. 1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD '03), Velden, Austria, 16-17 June 2003
- [70]. Yu, E. Modelling Strategic Relationships for Process Reengineering Ph.D. Thesis. Dept of Computer Science, University of Toronto. 1995.
- [71]. A. Fuxman, M. Pistore, J. Mylopoulos, P. Traverso. Model Checking Early Requirements Specifications in Tropos. Proc. 5th International Symposium on Requirements Engineering (RE'01), Toronto, Canada, August 2001.
- [72]. Dardenne, A., Van Lamsweerde, A., Fickas, S., Goal-directed requirements acquisition, Science of Computer Programming, 20 (1-2), avril 1993
- [73]. Letier, E. Reasoning about Agents in Goal-Oriented Requirements Engineering. Unpublished PhD Thesis, Université Catholique de Louvain, 2001.
- [74]. Chung, L., Nixon B.A., Yu E., Mylopoulos J., Non-Functional Requirements in Software Engineering, KLUWER ACADEMIC PUBLISHERS, BOSTON (2000)

RÉFÉRENCES

- [75]. Annis Ferchichi, contribution a l'integration des pro cessus métier : application à la mise en place d'un référentiel qualité multi-vues, école centrale de Lille école centrale de paris, 2008
- [76]. Shi Kuo Chang. Handbook of Software Engineering and Knowledge Engineering : Fundamentals. World Scientific Publishing Company, 2002.
- [77]. DIN. Systemes de management de la qualité; Exigences; (ISO 9001 .2000), Version trilingue EN ISO 9001 .2000. DIN, 2000.
- [78]. John Cunningham Wood and Michael C. Wood. W. Edwards Deming: Critical Evaluations in Business and Management. Routledge, 2005.
- [79]. François Ott, Pierre Vaast, « livre de grammaire : sixième en troisième », Hatier, 1991
- [80]. François Ott, Pierre Vaast, « livre de grammaire : Première en Terminale », Hatier, 1991
- [81]. Mehdi Yousfi-Monod, Violaine Prince, « Compression de phrases par élagage de leur arbre morpho-syntaxique : Une première application sur les phrases narratives », RSTI-TSI-25/2006. Document numérique, pp 437-468, 2006
- [82]. Minel J.-L., « Le résumé automatique de textes : solutions et perspectives », in Proceedings of TAL, Résumé automatique de textes, vol. 45/1, p. 7-13, 2004.
- [83]. Zave P., Jackson M., « Four Dark Corners of requirements Engineering », ACM Transactions on Software Engineering and Methodology, 1997
- [84]. Carman James M., Consumer Perceptions of Service Quality: An Assessment of the SERVQUAL Dimensions, Journal of Retailing, 66, printemps, pp : 33-55, 1990.
- [85]. Mouhamed Diouf, « Spécification Et Mise En Œuvre D'un Formalisme De Règles Métier », thèse n°3507, Université Bordeaux I, décembre 2007.
- [86]. Vincent Legendre, Gérald Petitjean , Thierry Lapatre, « Gestion des règles métier », Génie Logiciel no 92 mars 2010, pp 43-52, 2010
- [87]. Business Semantics of Business Rules, Business Rules Team Response to RFP: SBVR Submission bei/2005-08-01 Version 8. <http://www.omg.org/docs/bei/05-08-01.pdf> last accessed on 2007/05/02

RÉFÉRENCES

- [88]. Semantics of Business Vocabulary and Business Rules Specification, Document de specification de l'OMG, 2006. <http://www.omg.org/docs/dtc/06-03-02.pdf> last accessed on 2007/05/02
- [89]. Ronald G. Ross. Principles of the Business Rule Approach. Addison-Wesley, Boston, USA, 2003.
- [90]. Business Rule Group Community BRG. <http://www.brcommunity.com>. Web page.
- [91]. D. Hay and K. A Healy. GUIDE Business Rule Project Final Report. Technical report, October 1997.
- [92]. Object Management Group, Production Rule Representation: Request for Proposal, br/2003-09-03, Sept. 2003. <http://www.omg.org/docs/br/03-09-03.pdf>.
- [93]. The Object Management Group OMG, UML 2.0 OCL Specification. OMG Specification, October 2003.
- [94]. Michael Jackson. «Problem Frames. Analyzing and structuring software development problems». Addison-Wesley, 2001.
- [95]. Atsa Etoundi Roger, Fouda Ndjodo Marcel, Atouba Christian Lopez, Customer Satisfaction within an Organization driven Aspect Oriented Business Component Model, International Journal of Advanced Research in Computer Science , pp215 à 225, vol 2, n°1, 2011.
- [96]. Daniel CHEUNG-FOO-WO, Adaptation Dynamique par Tissage d'Aspects d'assemblage, Thèse, Université de Nice-Sophia Antipolis, Février 2010.
- [97]. Ouafa Hachani, Patrons de conception à base d'aspects pour l'ingénierie des systèmes d'information par réutilisation, thèse, Université Joseph Fourier-Grenoble I, Novembre 2006.
- [98]. Laurent QUINTIAN, JADAPT : Un modèle pour améliorer la réutilisation des préoccupations dans le paradigme objet, thèse, Université de Nice Sophia-Antipolis, Juillet 2004.
- [99]. Atsa Etoundi Roger, Fouda Ndjodo Marcel, Atouba Christian Lopez, Abessolo Alo'o G., Knowledge Management Driven Business Process and Workflow Modeling within an Organization for Customer Satisfaction, International Journal of Engineering Science and Technology, pp 7350-7362, 2010

RÉFÉRENCES

- [100]. Herzum P., Sims O., "Business Component Factory: a Comprehensive Overview of Component-Based Development for the Enterprise", Wiley Computer Publishing, 1999.
- [101]. Hassine I., "Specification et formalisation des démarches de développement à base de composants métier : la démarche Symphony", Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, 2005.
- [102]. Cauvet C., Ramadour P., « Les composants métier dans l'ingénierie des systèmes d'information », Vuibert (Ed.), Composants : concepts, techniques et outils, 2005.
- [103]. Barbier F., Atkinson C., "Business Components", Business Component-Based Software Engineering, Kluwer, vol. 705, Chap. 1, pp. 1-26, 2002.
- [104]. Bass L., Buhman C., Comella-Dorda S., Long F., Robert J., Seacord R., Wallnau K., "Volume I: Market Assessment of Component- Based Software Engineering", Carnegie Mellon University, Software Engineering Institute, TECHNICAL REPORT CMU/SEI- 2000-TR-008, ESC-TR-2000-007, Mai, 2000.
- [105]. Heineman G., Councill W., "Component-Based Software Engineering: Putting the Pieces Together", Addison-Wesley, 2001.
- [106]. Andro T., Chauvet J.-M., « Objets métier », Eyrolles, Paris, 232 pp., 1998.
- [107]. Ambler S.W., "Process Patterns: building Large Scale Systems using Object Technology", SIGS Books, Cambridge University Press, Décembre, 1998.
- [108]. Cherbakov L., Galambos G., Harishankar R., Kalyana S., Rackham G., "Impact of service orientation at the business level", IBM Systems Journal 44(4): 653-668, 2005.
- [109]. Herzum P. and Sims O., "The Business Component Approach", OOPSLA'98 Business Object Workshop, Octobre, Canada, 1998.
- [110]. Rajaa SAIDI, Conception et Usage des Composants Métier Processus pour les Systèmes d'Information, thèse, Institut Polytechnique de Grenoble, Septembre 2009.
- [111]. Schmid H.A., "Business entity and process components", Springer (Editor) OOPSLA'99, Business Object Design and Implementation, Denver, USA, 2 Novembre, 1999.

RÉFÉRENCES

- [112]. Lopes C.V., Hirsch W., Separation of Concerns, Technical Report NU-CCS-95-03, Northeastern University, Boston, February 1995. (disponible à <http://citeseer.nj.nec.com/lopes95separation.html>).
- [113]. Carver L., Griswold W.G., « Sorting out Concerns », Workshop on Multi-Dimensional Separation of Concerns, OOPSLA'99, 1999.
- [114]. Tarr P., Harrison W., Ossher H., Finkelstein A., Nuseibeh B., Perry D., Summary of Workshop on MultiDimensional Separation of Concerns in Software Engineering, ISCE 2000.
- [115]. Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes C.V., Loingtier J.M., Irwin J., « Aspect Oriented Programming », ECOOP 97,1997.
- [116]. Fradet P., Südholt M, « AOP: towards a generic framework using program transformation and analysis », Workshop Aspect Oriented Programming, ECOOP 98, 1998.
- [117]. David P.C., Ledoux T., Bouraqadi-Saâdani N.M.N. « Two-Step Weaving with Reflexion using AspectJ », Workshop Advanced Separation of Concern, OOPSLA 01, 2001.
- [118]. Ségura-Devillechaise M., Menaud J.M., Muller G., Lawall J.L., « Web Cache Prefetching as an Aspect : Towards a Dynamic-Weaving Based Solution», AOSD 03, 2003.
- [119]. H. Ossher and P.L. Tarr — Hyper/JTM: Multi-dimensional separation of concerns for JavaTM. In Proceedings of the ICSE 2000, International Conference on Software Engineering, Limerick, Ireland, June 2000.
- [120]. Harker S.D.P., Eason K.D. & Dobson J.E., "The Change and Evolution of Requirements as a Challenge to the Practice of Software Engineering", IEEE Symposium on Requirements Engineering, RE'93, San Diego, CA, Jan. 4-6, 1993, 266-272.

LISTE DES PUBLICATIONS ISSUES DE LA THESE

- [1]. A Goal Oriented Approach for the Definition of a Business Process Requirement Model. International Journal of Computer Applications 9(7):1–7, 2010.
- [2]. Atouba Christian Lopez, A Goal Based Approach for QFD Refinement in Systematizing and Identifying Business Process Requirements, International Journal of Computer Science Issues, Vol. 7, Issue 6, pp 343-350, 2010.
- [3]. A Model based Business Process Requirement Rule Specification. International Journal of Computer Applications 11(9):17–24, 2010.
- [4]. Business Process Requirement Engineering. International Journal on Computer Science and Engineering, volume 2, n° 9, 2010.
- [5]. Knowledge Management Driven Business Process and Workflow Modeling within an Organization for Customer Satisfaction, International Journal of Engineering Science and Technology, pp 7350-7362, 2010.
- [6]. A Formal Approach for the Inclusion of Key Performance Indicators in a Business Process. International Journal on Computer Science and Engineering, volume 3, n° 9, 2011.
- [7]. Customer Satisfaction within an Organization driven Aspect Oriented Business Component Model, International Journal of Advanced Research in Computer Science , pp215 à 225, vol 2, n°1, 2011.

INDEX

A

activités · 16
adaptables · 5
agrégat · 118
aspect · 9
Aspect · 130, 194, 196
aspect métier · 141
Aspects · 194
attentes · 5
attentes des utilisateurs · 6

B

Besoin · 119
besoin d'un cadre métier · 71
besoin exprimé · 3
besoins · 3
Besoins Critiques · 120
besoins d'un domaine · 4
besoins réels · 3
But · 71, 73
but fonctionnel · lxxii, 72
but inconsistant · 73
buts · 2

C

cadre métier · 71
capter · 12
clients · 5
composant métier orienté aspect · 9
composants · 9
composants métiers · 10
conciliation · 7
Connaissance · xxi, 80
connaissance de la satisfaction · 12
connaissances · 6
Construction des facteurs de performances · 117, 180
Construction des observateurs · 176
construction des observateurs de performances · 67
Contexte · 91
contraintes · 7
critère de qualité · 67

D

démarche · 12
Dépendance entre les Indicateurs de Performances · 82

INDEX

développement · 5
domaine · 3
dynamique · 12
dynamique du besoin · 13

E

élaboration du modèle · 12
enrichir · 6
ensemble des observateurs de performance · 80
ensemble hiérarchisé · 5
environnement d'exécution
 environnement · 9
environnements · 4
évolution des besoins · 4
évolutions · 8
exigences de qualité · 5
expression des besoins · 5
expression des besoins des organisations · 5
extension · 8

F

facteur de performance · 83
Fonction d'Évaluation · 67
Fonction d'évaluation · 179
Fonction d'Observation · 77
Fonction de Satisfaction · 79
fonctions d'évaluation · 67
formalisme · 3
formalisme de modélisation · 2
fragment conceptuel · 5
fragment de connaissance · 3
Fragment de connaissance · 176

G

Garde · 94
gestion de processus d'affaire · 21

I

identification des indicateurs · 9
Indicateur de performance · x
Indicateur de Performance · 81
indicateurs de performance · 9
indicateurs de performances · 9
INGENIERIE · i, xi, 190
ingénierie des besoins · 4
intention d'usage · 8

L

liquidation des droits · 13

INDEX

M

méta-modèle · 6
modèle · 6
MODELE · i, 9, 46, 47, 139
modèle de référence · 3
modèle mathématique · 17
modélisation · 3
Modélisation · xi, 9, 23, 150, 187, 190
modélisation des besoins d'un processus d'affaires · 11
modélisation de la satisfaction · 67
modélisation des organisations · 3
modélisation des processus d'affaire · 3
Modélisation des processus d'affaire · 150

N

niveau d'importance · 7

O

objet métier · 80
observateur · 35
Observateur · x, xxi, 68, 76, 80, 179, 185
observateur de performance · 80
Observateur de performance · x, 68, 185
observation · 35
Observation · x, xxi, 68, 76, 77, 185
Observation d'un Indicateur · 77
organisation · 3

P

performances · 5
point de vu · 79
Point de vue · xxi, 78
points fixes · 13
pont semi-formel · 7
préoccupations · 3
processus d'affaire · 2
Processus d'affaire · xi
processus d'affaires de production · 5
processus d'ingénierie · 6
processus métier · 18
processus noyau · 22
processus support · 22

Q

qualité de service · 7
Qualité de service · 5
qualité de service externe · 7
qualité de service interne · 7

INDEX

R

raffinement · 8
recensement · 11
Recensement des attentes de qualité · 106
recensement des besoins · 71
Recensement des besoins · 105, 146
règle · 5
règle de gestion · 7
règles métiers · 8
rendement · 10
représentation · 4
représenter les besoins · 3
réutilisation · 6
réutilisation des connaissances · 6

S

satisfaction · 7
Satisfaction · xxi, 79, 80, 194
satisfaction des bénéficiaires · 7
satisfaction des clients · 9
sélection · 11
Séquence · 97
service fourni · 5
Seuil · 119, 179
Seuil de satisfaction · 179
seuils de satisfactions · 67
spécification · 8
suffixe · 118
systèmes informatiques · 6

T

tâche · 2
Transformation des besoins · 110, 150
type · 89
Type · 95, 150

U

utilisateurs · 71

V

views · 118

W

Workflows · 2