

UNIVERSITE DE SAAD DAHLAB BLIDA

Faculté des sciences
Département d'informatique

MEMOIRE DE MAGISTER

Spécialité : Ingénierie des systèmes et de la connaissance

**LA CONCEPTION D'UN FRAMEWORK DE GESTION DES
CONNAISSANCES A BASE DE RAISONNEMENT A PARTIR
DE CAS POUR LA RE-MODELISATION DES PROCESSUS
METIER**

Par

BOUMAHDJ Fatima épouse RAHAL

Devant le jury composé de

A.GUessoum	Professeur, U. de Blida	Président
H.Abed	Maître de conférence, U. de Blida	Examineur
L.Admane	Maître de conférence, I.N.I, Alger	Examineur
F.Nader	Chargé de cours, I.N.I, Alger	Examineur
S.Oukid	Maître de conférence, U. de Blida	Rapporteur

Blida, juin 2006

RESUME

Le BPR (Business Process Redesign) permet de reconsidérer un processus afin d'améliorer sa performance. Les praticiens ont développé des méthodes pour supporter l'implémentation du BPR. Mais la plupart de ces méthodes ont des lacunes, ces derniers correspondent à la non réutilisation des conceptions des processus précédentes, ce qui constitue un handicap pour le succès du BPR. Dans le présent travail, nous suggérons l'utilisation de la technique de raisonnement à base de cas : CBR (Case-Based Reasoning). Dans le cas de BPR il s'agit de re-concevoir le processus métier, en s'adaptant une conception d'un processus métier similaire réussit antérieurement. Une plateforme d'implémentation de BPR et le processus cyclique de CBR sont utilisés comme un support technique de gestion de la connaissance, qui assure une réutilisation efficace des méthodes de re-conception comme les mécanismes de création et de partage de connaissance.

Dans ce mémoire, nous présentons un environnement de description de processus basée sur une représentation graphique, ce dernier, permet d'une part de générer des spécifications (workflow exécutable par le moteur) correctes de Workflow, et d'autre part, de modifier ces spécifications pour les adapter aux évolutions des besoins de l'utilisateur, et en plus, de définir un ensemble de règles de similarité entre les workflow, afin de réaliser le système BPR.

Mots clés

Raisonnement à base de cas, Gestion des connaissances, Re-modélisation de processus métier, Workflow, Extensible Markup Language (XML).

ملخص

اعادة تصميم سير العمل (BPR) يسمح باعادة النظر فيه وذلك لتطوير نتائج التطبيقين , الذين طوروا طرق لنجاح اعادة التصميم العمل.
لكن اغلبية هذه الطرق تعاني من نقائص التي تكمن في اعادة استعمال التصميمات السابقة والذي يشكل عائق من اجل نجاح السير العملي .
في هذا العمل اردنا اقتراح استعمال تقنية "التخمين عن طريق الحالة" (CBR) , في حالة BPR اعادة تصميم سير العمل , يتعلق الامر اعادة تصميم سير عمل جديد باعاطائه تصميم ناجح من قبل سير عمل اخر .
هيكل تطبيق BPR و السير الحلقي لCBR استعمالا الحامل ثقفي لتسيير المعرفة , و الذي يضمن اعادة استعمال ناجحة لطرق BPR, مثل المکانيزم تكوين اشتراك المعرفة . في هذه الرسالة , اردنا تقديم محيط لوصف سير (رسم بياني) هذا الاخير , يسمح من جهة :توريد تمييز وصف (منفذ في محرك سير العمل صحيح ل سير العمل, ومن جهة اخرى لتغيير هذه الوصفات لاجل تاقلهما مع التغيرات الحاجيات المستعمل وزيادة على هذا , لتعريف مجموعة قواعد للتشابه بين سير العمل لهدف تخفيف نظام BPR.

الكلمات الرئيسية : اعادة تصميم سير العمل (BPR) , التخمين عن طريق الحالة (CBR) , لتسيير المعرفة ,

XML .

Abstract

Process Redesign (BPR) helps rethinking a process in order to enhance its performance. Practitioners have been developing methodologies to support BPR implementation. However, most methodologies lack on deriving a process design threatening the success of BPR. In this thesis, we suggest the use of a case-based reasoning technique (CBR) to support solving new problems by adapting previously successful solutions to similar problems to support redesigning new business processes by adapting previously successful redesign to similar business process. An implementation framework for BPR and the CBR's cyclical process are used as a knowledge management technical support to serve for the effective reuses of redesign methods like a knowledge creation and sharing mechanism.

In this memory, we introduce a process description framework based on graph which contribute on a textual representation. This representation allow in a first time to generate correct specifications of workflow(a workflow which can be executed by the motor) , and in a second time to modify these specifications to be able to adapt them to the user needs(demands), and in addition to define a set of similarity rules between workflows, for the purpose to carry out the CBR system.

Keywords

Case-Based Reasoning (CBR), Knowledge Management (KM), Business Process Redesign (BPR), Workflow, Extensible Markup Language (XML).

REMERCIEMENTS

Mes plus vifs remerciements vont aux personnes qui ont contribué au bon déroulement et à l'aboutissement de ce mémoire.

Mes remerciements et profonde gratitude vont aux membres de la commission d'examen. J'exprime toute ma reconnaissance à M^{me} ABED, maître de conférence à Blida, pour avoir accepté d'examiner mon travail et de présider mon jury de soutenance. Je la remercie de l'intérêt réel qu'il a manifesté sur la teneur de ce rapport. Je suis très honoré que Monsieur ADMANE, maître de conférence à I.N.I, et M^{elle} NADER chargée de cours à I.N.I aient accepté d'être examinateur de ce travail. Un grand merci à M^{me} OUKID d'avoir accepté de rapporter ce mémoire, pour l'intérêt et pour l'enthousiasme avec lesquels elles ont évalué mon travail.

De nombreuses personnes m'ont aidé ou encouragé pendant mes études. À tous mes collègues, amis et personnel du département informatique, j'aimerais également vous offrir mon amicale reconnaissance pour votre présence, gentillesse et appui qui ont su égayer ces années de travail. Je voudrais plus particulièrement exprimer ma reconnaissance envers M^{me} OUHRANI-HARKATI, avec qui une collaboration scientifique fructueuse s'est établie.

Je renouvelle toute mon amitié et ma sympathie à ceux qui m'ont accordé du temps et m'ont témoigné un soutien constant dans ce travail de recherche. Merci tout spécialement à Rachid, Smail, Rabeh, Hamza, Fouaz, Maamar, OUCHENE, et Massaouda.

Un gros merci à mes parents pour leur soutien et pour m'avoir transmis la confiance dans mes moyens et dans la vie qui est fort utile pour aboutir un projet à long terme. Merci à mes frères et plus particulièrement Mohamed et sa femme Nacera. Un très grand merci à toutes mes sœurs, et mes nièces. Ainsi qu'à toute ma belle famille.

Un merci tout spécial à mon mari Mustapha, qui pendant ces années a su m'encourager, me guider de multiples façons. Il a été mon support au niveau intellectuel par nos échanges au sujet du mémoire, et il a été surtout mon support au niveau psychologique ce qui m'a insufflé la motivation nécessaire pour terminer ce travail.

TABLE DES MATIERES

RESUME

REMERCIEMENTS

TABLE DES MATIERES

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

INTRODUCTION	10
1. LA GESTION DES CONNAISSANCES WORKFLOW	17
1.1 Introduction	17
1.2 workflow.....	18
1.3 La modélisation	24
1.4 Business Process Redesign (BPR).....	28
1.5 La Gestion des connaissances (GC)	31
1.6 Gestion des workflows	36
1.7 Conclusion.....	37
2. RAISONNEMENT A BASE DE CAS	39
2.1 Introduction	39
2.1 Les technologies utilisées pour la GC	40
2.2 Raisonnement à base de cas	41
2.3 La correspondance entre le KM et le CBR.....	50
2.4 Re-conception d'un processus workflow par le CBR	51
2.5 Le développement de système CBR pour BPR.....	53
2.6 Conclusion.....	55
3. ACQUISITION DE CONNAISSANCE DE DOMAINE POUR LE BPR.....	56
3.1 Introduction	56
3.2 Principe de modélisation d'un workflow	57
3.3 Proposition d'un méta-modèle (structure d'un workflow)	59
3.4 La modélisation du workflow.....	63
3.5 Etude de cas	68
3.6 Conclusion.....	69

4. REPRESENTATION DE CAS	71
4.1 Introduction	71
4.2 Les objectifs de la représentation de cas	71
4.3 Le langage de représentation d'un cas.....	72
4.4 Le passage du méta-modèle au cas workflow	73
4.5 Expression de notre représentation du cas en XML.....	74
4.6 Etude de cas.....	82
4.7 Conclusion.....	86
5. IMPLEMENTATION DU SYSTEME.....	87
5.1 Introduction.....	87
5.2 Les méthodes de re-modélisation d'un workflow	88
5.3 L'architecture de la Plateforme 'BPR/CBR'	90
5.4 La sous plateforme 'Modeleur'	94
5.5 Le sous framework 'Dépôt'	99
5.6 La sous plateforme 'Gestion des connaissances'	104
5.7 Le prototype BPR/CBR.....	127
5.8 Conclusion.....	131
CONCLUSION.....	133
REFERENCES.....	

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1: Exemple de workflow	21
Figure 1.2: Le système de gestion des workflows	22
Figure 1.3: Modèle de référence d'un SGWF [21]	23
Figure 1.4: Les vagues d'amélioration de processus métier [14]	30
Figure 1.5 : Le modèle de KM [54]	32
Figure 1.6 : Le cycle de KM [35]	36
Figure 2.1: La re-mémoration et l'adaptation en RÀPC	42
Figure 2.2 : Le cycle du RBC [25]	44
Figure 2.3 : Exemple de structuration d'un cas en CBR structurel	45
Figure 2.4: Exemple de cas pour le modèle conversationnel [56]	45
Figure 2.5 : Remémoration de cas en deux étapes	47
Figure 2.6: Remémoration de cas en multi-étapes	47
Figure 2.7 : Remémoration de cas à option	48
Figure 2.8 : Adaptation de type a) transformationnel, b) dérivationnel	49
Figure 2.9 : Cycle structuré de re-modélisation du workflow	51
Figure 2.10 : Procédure de développement d'un system à base de cas [44]	54
Figure 3.1 : Cas d'utilisation « Modéliser un processus d'entreprise »	58
Figure 3.2 : Le méta-modèle de workflow proposé	61
Figure 3.3: Les types de dépendance entre les tâches	66
Figure 3.4: Modèle de workflow pour «Demande de remboursement des frais»	69
Figure 4.1: Exemple d'utilisation de la définition du modèle graphique de WF	79
Figure 4.2 : DTD de schéma de workflow	82
Figure 4.3 : Modèle de processus achat électronique	84
Figure 4.4: La spécification du workflow 'Achat électronique'	85
Figure 5.1: L'architecture de la plateforme BPR/CBR	92
Figure 5.2 : L'interface de sous plateforme 'Edition'	98
Figure 5.3 : Le diagramme de composants du système	99
Figure 5.4 : La structure de la base de cas	101

Figure 5.5 : Le schéma de la base de cas	102
Figure 5.6: Le schéma de la base de vocabulaire	104
Figure 5.7 : Le processus de re-modélisation par la plateforme BPR/CBR	105
Figure 5.8 : Séquence de la re-modélisation par ‘dépendance de domaine’.....	105
Figure 5.9: Séquence de la re-modélisation par ‘CBR’	106
Figure 5.10 : Le processus de recherche des cas par dépendance de domaine.....	107
Figure 5.11 : L’interface d’utilisateur de ‘Gestion de la base de cas’	108
Figure 5.12 : Le processus d’initialisation du modèle entré	110
Figure 5.13 : L’interface d’utilisateur pour initialiser le problème (modèle entré).....	111
Figure 5.14 : Les étapes de calcul la similarité entre les cas	113
Figure 5.15 : Exemple d’un modèle entré et un modèle trouvé.....	115
Figure 5.16 : Exemple de distance de la dépendance	119
Figure 5.17 : Un exemple de distance de dépendance entre les tâches	120
Figure 5.18 : L’interface utilisateur pour les cas candidats	123
Figure 5.19 : Un exemple d’adaptation d’un cas	124
Figure 5.20 : Exemple d’adaptation d’un ensemble des termes similaires.....	124
Figure 5.21 : L’architecture du module ‘case based reasoning’	126
Figure 5.22 : L’interface utilisateur pour la gestion des cas.....	130
Figure 5.23 : L’interface utilisateur pour un cas de workflow	130
Tableau 3.1: La notation proposée.....	68
Tableau 4.1 : Les éléments d’un cas	76
Tableau 4.2 : Les éléments d’une tâche	77
Tableau 4.3: Les éléments d’une transition	77
Tableau 4.4 : Les types de dépendance entre les tâches	78
Tableau 4.5 : Les éléments d’un rôle	79
Tableau 4.6: Les éléments d’un acteur	79
Tableau 5.1: Calculer la similarité.....	116
Tableau 5.2: Résumé de bar de menu	129

INTRODUCTION

Face à la multiplication des sources d'informations textuelles numérisées, le besoin d'outils pour traiter automatiquement ces informations s'est rapidement fait sentir, dans tous les secteurs d'activité, privés ou publics, industriels ou académiques. En effet, faute d'avoir su, pu ou voulu synthétiser les informations qu'elles détiennent pour pouvoir les réutiliser comme des connaissances à part entière [12], il n'est pas rare que des entreprises aient été obligées de reconstituer entièrement certaines expériences par manque d'informations suffisantes, ainsi le départ de quelques ouvriers peut bloquer toute la chaîne de production, c'est le cas lors d'un départ en retraite, d'une mutation ou d'un licenciement...etc. Pour les entreprises, cela signifie à la fois un gaspillage d'énergie, de temps et de ressources.

C'est dans ce contexte socio-économique de perte de connaissances due au départ de personnel (licenciement, mutation, retraite...) que l'idée de Mémoire d'Entreprise (ME) émerge dans les années 90 [47]. Parallèlement, le développement des technologies nouvelles (Gestion Électronique de Documents ou GED, Internet, workflow, etc.) et la reconnaissance de la connaissance comme valeur stratégique pour l'entreprise achèvent de mettre au devant de la scène cette notion. Comment en effet repérer les informations pertinentes qui serviront de base à la création de nouvelles connaissances devant l'ampleur et le désordre des informations diffusées par les divers outils de communication ?

Les connaissances se trouvent dans l'intelligence, le savoir-faire, la mémoire, l'expérience des personnes autant que dans les documents, bases de données, etc. Il faut donc l'extraire pour espérer la conserver et pouvoir la transmettre en vue de sa réutilisation. En effet, une simple accumulation de connaissances est loin d'être suffisante : il est nécessaire de les structurer et de mettre en place les moyens de les retrouver, en vue de prendre les meilleures décisions dans des environnements de plus en plus complexes, autrement dit de gérer ces connaissances.

La gestion des connaissances est un moyen permettant autant que possible, de valoriser les capacités et l'expérience de chacun à la place qui lui convient le mieux, de faire circuler l'information utile et d'aider à trouver au bon moment celle dont on a réellement besoin dans l'action [31].

La gestion des connaissances (Knowledge Management) s'occupe ainsi d'identifier, collecter, capitaliser et diffuser les connaissances d'une entreprise [10]. Ce travail est loin d'être trivial, car au-delà des questions de base qu'il soulève (qu'est-ce qu'une connaissance, comment la représenter et comment structurer une collection de connaissances), il implique souvent d'organiser les connaissances selon une forme réutilisable dans des conditions inconnues au moment de cette organisation. En effet, l'un des problèmes de la capitalisation des connaissances est de définir des critères pour le choix des connaissances à conserver, dans le but de prévoir une réutilisation de ces connaissances. La mémoire d'entreprise, support privilégié de la gestion, se situe en effet dans un contexte de très long terme puisqu'elle vise à capitaliser des connaissances dans le but de pouvoir les réutiliser en dehors de leur contexte d'origine et pour des objectifs différents[12].

La modélisation du processus d'entreprise (métier) dans divers domaines fait apparaître rapidement le besoin de pouvoir réutiliser des processus construits auparavant, c'est à dire, on peut se retrouver en face d'un problème qui a déjà été résolu dans un domaine d'application différent. Par exemple, on voudra réutiliser la stratégie mise en œuvre pour résoudre un problème proche ou une partie d'un modèle construit précédemment qui correspond à un sous-problème et à des conditions similaires.

Mais les connaissances apparaissent comme un patrimoine fragile de l'entreprise [12]. Elles circulent, s'enrichissent, se périment et peuvent également disparaître. Si rien n'est prévu pour conserver ces connaissances, l'entreprise perd une partie de sa mémoire, de son savoir-faire et donc, une part de sa compétitivité. Le fait que la connaissance existe dans le rapport personnel d'un acteur ne veut pas dire qu'elle n'est pas partageable et partagée. Les enjeux de notre travail s'expriment bien en terme de partage de connaissances au sein des entreprises.

Dans le contexte de la re-modélisation de processus métier, la gestion des connaissances consiste à réutiliser l'expérience et les bonnes pratiques connues pour l'élaboration d'un nouveau processus métier. Mais leur mise en place demeure un obstacle, en particulier au niveau de l'extraction des connaissances :

- ◆ Nécessité d'employer de nouveaux acteurs [25] comme le « cogniticien », dont le rôle est d'aider les experts à décrire leurs connaissances et qui implique des investissements supplémentaires,
- ◆ Implication intellectuelle importante des experts qui doivent faire des efforts d'organisation et de structuration de leurs connaissances, en faisant abstraction du contexte d'utilisation,
- ◆ Investissement en temps consacré à cette extraction, qui représente souvent une surcharge.

Pour élaborer un système de re-modélisation par la gestion des connaissances, il existe plusieurs techniques utilisées, nous avons choisi le raisonnement à Partir de cas (RàPC). Le RàPC est l'un des paradigmes proposés par l'IA pour la résolution de problème [22], il est considéré comme une technique intéressante pour la gestion des connaissances, le CBR est en effet adopté pour résoudre des problèmes difficiles à modéliser et même à l'expertise est difficile à rendre explicite. Il a pour particularité de faire intervenir des cas déjà résolus pour résoudre un nouveau problème. Il intègre donc en lui-même un processus d'apprentissage. Il utilise comme base de connaissance une base d'expériences passées ou base de cas. Son principe est de chercher dans la base de cas le cas qui soit le plus proche possible du problème courant, d'adapter la solution du cas trouvé pour résoudre le problème cible, et mémoriser ce nouveau problème avec la solution produite dans la base de cas comme un nouveau cas [55].

Notre travail peut se concentrer sur la re-modélisation des processus d'entreprises et la réutilisation des modèles construits. Ce travail décrit la démarche que nous proposons pour aborder le problème de la gestion des connaissances à l'aide de la technique de raisonnement à base de cas. L'objectif de ce travail est d'élaborer une architecture d'une plateforme d'un système de re-modélisation dans le domaine de la gestion des connaissances en s'appuyant sur le paradigme de RàPC.

L'objectif de notre travail est d'étudier l'intégration de la gestion des connaissances dans le système que représente l'entreprise : processus d'entreprise. C'est un système de connaissance dans lequel des acteurs œuvrent, de manière coordonnée, à l'obtention du résultat que l'entreprise veut atteindre.

L'objectif primaire de ce travail est de proposer une plateforme intelligente pour remodeler les modèles de workflow appelé BPR/CBR (Business Process Redesign with Case Based Reasoning). Dans ce mémoire, nous présentons les méthodes de remodelisation, nous proposons une architecture de la plateforme BPR/CBR, et la mémoire d'entreprise qui inclut la base de cas et la base de vocabulaire. Pour développer la base de cas, la représentation d'un cas et la structure de la base de cas sont représentées. Et pour la base de vocabulaire, une architecture de la base de vocabulaire est également proposée. Les travaux entrepris se sont focalisés sur les points suivants:

- ◆ Proposer un méta-modèle et un modèle de workflow : mettre au point un méta-modèle de processus totalement et facilement extensible : générique afin de le spécialiser aux contextes d'applications de nos clients, extensible afin d'y ajouter de nouveaux concepts, de nouvelles classes d'objets et de nouveaux attributs. La remodelisation des processus métiers s'appuie sur la définition d'un méta-modèle qui comporte tous les éléments d'un workflow, et un modèle qui définit un processus métier.

- ◆ Le passage : nous décrivons une technique, qui permet le passage, de la connaissance tacite (modèle de workflow) vers la connaissance explicite (cas), en utilisant le langage XML. XML fournit un moyen de vérifier la syntaxe d'un document grâce aux DTD (Document Type Definition). Il s'agit d'un fichier décrivant la structure des documents y faisant référence grâce à un langage adapté.

- ◆ Proposer une architecture du framework BPR/CBR : l'architecture du système est centrée autour de CBR. Cette architecture est composée de trois parties : 'Edition', 'Dépôt' et 'Gestion des connaissances'. La partie 'Gestion des connaissances' comporte trois couches : 'Gestion de la base de cas', 'CBR' et 'Gestion de la base de vocabulaire'. Le dépôt inclut 'la base de cas' et 'la base de vocabulaire'.

◆ Re-modélisation des processus par BPR/CBR : BPR/CBR emploie trois stratégies de re-modélisation : la première, re-modélisation par ‘feuille propre ’, la deuxième, par ‘dépendance de domaine’ et la dernière par la technique ‘CBR’. Ces méthodes de re-modélisation sont présentées en utilisant un exemple pour démontrer l’utilité de notre plateforme.

Pour re-modéliser par la technique CBR, nous proposons des mécanismes pour récupérer les cas similaires à partir de la base de cas, adapter le cas similaire, et finalement stocker le nouveau cas dans la base de cas pour future utilisation. L’étape de récupération est un problème de recherche essentielle, elle englobe la recherche dans la base de cas les cas similaires et la sélection du meilleur cas. L’adaptation d’un cas s’effectue par modifier le cas sélectionné afin d’être introduit dans la base de cas.

◆ Suggérer une structure pour la base de cas et un format pour la représentation d’un cas : pour montrer comment les cas sont indexés, un schéma pour la base de cas et un format pour la représentation d’un cas sont proposés.

◆ Proposer une structure de la base de vocabulaire : BPR/CBR emploie une mesure de similarité sémantique entre les termes pour récupérer le cas similaire à partir de la base de cas. Pour cette raison, nous avons construit la base de vocabulaire où les termes sont liés. Pour montrer la structure interne de la base de vocabulaire, nous suggérons une architecture de la base de vocabulaire

◆ Développer un prototype de BPR/CBR : un prototype de BPR/CBR est développé pour valider la faisabilité technique de la plateforme BPR/CBR. Ce mémoire contient un cas réel ‘Commerce Electronique’, et illustre l’usage et les bénéfices de BPR/CBR par leur utilisation.

Dans le premier chapitre, nous exposons le contexte et la problématique de nos travaux. Nous présentons une étude sur la re-modélisation d’un workflow afin de montrer la nécessité de BPR pour les entreprises, nous exposons précisément les solutions proposées pour aborder les problèmes liés au re-conception des processus, pour argumenter le choix de KM comme une solution de BPR. Le bilan de ces premiers éléments nous

permet d'établir la problématique de notre recherche et de proposer la gestion des connaissances comme une solution possible.

Le deuxième chapitre décrit la parenté et la synergie entre KM et le raisonnement à base de cas, ainsi qu'un état d'art sur la technique CBR. Ce chapitre est entièrement consacré au RàPC. Nous montrons en quoi il peut améliorer l'aide apportée à l'utilisateur par notre système et nous décrivons les modèles et techniques utilisés dans ce domaine, en introduisant notamment les étapes du raisonnement, ses composants et les conditions de son application. Nous présentons ainsi, l'approche adoptée pour aborder un problème de BPR en utilisant le CBR, et nous illustrons cette approche par une architecture qui projette les problèmes de BPR sur les étapes du CBR. Ce chapitre nous permet de positionner la relation entre le CBR et le BPR. Nous présentons aussi une architecture générique de conception d'un système CBR.

A partir de l'étude présentée dans le premier chapitre, nous abordons dans le troisième chapitre nos propositions et contributions pour permettre la re-modélisation d'un workflow par le CBR. Nous abordons dans ce chapitre la première étape de conception d'un système de CBR 'Acquisition des connaissances de domaine'. Après une étude sur plusieurs méta-modèles de workflow existants, nous proposons un méta-modèle qui puisse servir compte des manipulations (raisonnement) qui sera effectués sur les processus entrants qu'entité (cas), et montrons notre contribution qu'ils apportent à notre étude. Nous définissons alors notre modèle de workflow.

Dans le deuxième chapitre nous présentons la deuxième étape de conception du système CBR 'Représentation de cas'. Dans cette étape, nous appliquons les résultats de nos développements de la première étape, afin de définir un langage de définition d'un cas. Pour cela, nous spécifions le langage choisi pour la représentation des cas, puis nous construisons la représentation des cas par le langage XML. Nous aboutissons alors à une description des modèles de workflow qui représente l'instanciation de notre méta-modèle. A la fin de ce chapitre nous donnons un exemple pour montrer l'utilisation de notre méta-modèle, le modèle proposé, et la représentation développée.

Le chapitre 5 est un chapitre central. C'est lui qui décrit l'architecture de la plateforme BPR/CBR. Nous donnons d'abord une vision globale du framework CBR/BPR

en respectant toutes les méthodes de re-modélisation présentées dans le chapitre, puis son architecture basée sur le CBR. Nous discutons après les sous plateformes que nous avons proposées pour réaliser notre framework. Les différentes sous plateformes sont développées, en particulier celui de «Gestion des connaissances» qui implémente le raisonnement à base de cas. Un exemple d'application a été choisi dans le domaine de « commerce électronique », afin d'évaluer la conception du framework et de démontrer les résultats du fonctionnement du système élaboré.

Le dernier chapitre, comme il se doit, présente les conclusions et les perspectives. Dans la conclusion générale, nous rappelons l'ensemble des développements réalisés au cours de nos travaux. Nous proposons des perspectives en présentant plusieurs pistes.

CHAPITRE 1

LA GESTION DES CONNAISSANCES WORKFLOW

1.1 Introduction

Si beaucoup d'organisations traditionnelles sont encore bâties sur les principes Tayloriens (décomposition des tâches et hiérarchie) [52], elles ont cependant commencé à évoluer suivant une approche processus orienté client afin de faire face non seulement à l'évolution des marchés (concurrence mondiale, diversification,...) mais aussi à celle des technologies.

Cette nouvelle approche qui apporte comme avantages la réduction des coûts, la flexibilité et l'amélioration de la qualité de vie productive, place la satisfaction du client au cœur du nouveau modèle de gestion des processus [3]. Elle exige par contre une évolution dans la recherche de la Qualité qui doit aller au delà de la qualité du produit en tenant compte notamment des qualités organisationnelle et sociale [21].

Cependant, faute de mesure fiable, les organisations conventionnelles basées sur les procédures ne peuvent pas véritablement intégrer la satisfaction du client. C'est pourquoi l'approche procédurale inadaptée au concept de processus doit laisser sa place au concept de "workflow" qui est présenté comme une unité de travail se répétant et finissant avec la satisfaction du client.

Pour parvenir à une organisation orientée processus, il faut identifier les processus primordiaux qui répondent aux objectifs majeurs de l'entreprise. Ces processus transfonctionnels et composés de processus opérationnels incluent l'ensemble des processus métiers, de contrôle et de support.

Beaucoup d'organismes ont besoin de changer plus qu'augmenter les processus existants, pour réaliser les résultats nécessaires. En effet, l'utilisation d'un système de re-

modélisation de workflow (BPR) dans l'entreprise va permettre la modélisation des procédures de travail, et de modifier son organisation et de reconfigurer ses processus pour les orienter vers les clients.

Pour apercevoir l'utilité de la technologie GC pour le BPR, nous avons organisé ce chapitre comme suit: nous exposons d'abord, la technologie workflow, ces types, le système de gestion de workflow, et les méthodes de modélisation de workflow existents. La deuxième section consiste à présenter la technologie BPR, et les raisons de re-modéliser le workflow, nous montrons par la suite, l'évolution des systèmes BPR, et quelle sont les technologies utilisées ? Pour choisir la technologie que nous devons utiliser, en présentant également le KM et l'utilité de cette technologie. La dernière section de ce chapitre présente les activités de GC.

1.2 workflow

Le Workflow management coalition (WfMC) définit un workflow comme étant l'automatisation partielle ou entière, ou la facilitation par informatisation d'un processus d'affaires [41]. On appelle workflow la modélisation et la gestion informatique de l'ensemble des tâches à accomplir et des différents acteurs impliqués dans la réalisation d'un processus métier [36]. Un workflow peut donc être défini comme un modèle informatique pour représenter un processus de travail.

1.2.1 Typologie du workflow

Il existe plusieurs types de workflow. Ces différents types dépendent des objectifs et des besoins des organisations. Tous les workflow ne mettent pas en œuvre des fonctionnalités identiques. Il est donc nécessaire d'établir une typologie des applications workflow en fonction des différents types de processus [30]. Dans les applications de workflow, on distingue quatre catégories :

1.2.1.1 Workflow de production

Correspond aux processus de base de l'entreprise. Les activités quotidiennes qui s'y rattachent subissent un peu de changement dans le temps et les transactions sont totalement répétitives [5]. Citons par exemple les processus de production de contrats d'assurance, la

gestion de réclamations clients, ...etc.

Les workflows de production s'appliquent aux processus opérationnels répétitifs et critiques, en fait aux processus métiers de l'entreprise ou du groupe qui les utilise, et qui requiert une forte productivité et une réponse rapide du système.

1.2.1.2 Workflow coopératif (collaboratif)

C'est un workflow "intelligent" qui permet à une équipe de modéliser un processus de travail, d'en fixer les règles, d'exploiter directement l'application et, de façon itérative, de faire évoluer le processus et ses règles de gestion en fonction des transformations des modes opératoires, donc il gère des procédures évoluant assez fréquemment, et liées à un groupe de travail restreint dans l'entreprise [15].

1.2.1.3 Workflow ad hoc

Ils s'appliquent aux procédures d'exception, qui n'arrivent qu'occasionnellement ou même qu'une seule fois dans la vie de l'entreprise (par exemple workflow mis en place pour la fusion de deux entreprises).

Ces processus peuvent toutefois représenter des enjeux critiques pour l'entreprise mais, généralement, ils sont liés à des routines administratives. Les workflow ad hoc s'appliquent surtout à des processus qui se rapprochent des projets [55].

1.2.1.4 Workflow administratif

Le workflow administratif correspond à des processus de routages de formulaires simples. Il s'agit alors d'automatiser la manipulation de formulaires électroniques, en remplacement des imprimés traditionnels, entre plusieurs personnes intervenant sur leur traitement et suivant des procédures pré définies. Par exemple la gestion d'autorisation de dépenses, la gestion de frais de mission [5].

Le tableau suivant résume les différences entre les quatre différents types d'applications workflow :

Production	Administratif	Coopératif	Ad-hoc
Haute capacité de traitement	Capacité de traitement inférieure moins que pour le Workflow de production.	La capacité de changer la définition d'un processus est essentielle.	Facilité d'utilisation et d'apprentissage sont très importantes.
Employés travaillant à plein temps sur des activités courtes	Un grand nombre d'employés peut être impliqué.	Fournir une voie structurée pour travailler ensemble.	La modification dynamique et rapide des processus est essentielle.
Processus formels avec peu de variation.	Une variété de processus peut exister dans le même système	Les processus sont moins rigides.	Facilité de déploiement.

Tableau 1.1: Les différences entre les différents types d'applications workflow [55]

1.2.2 Les trois R du workflow

Trois éléments caractérisent fondamentalement tout workflow. Ce sont les routes, les règles et les rôles. L'image des "3R" est énoncée par Ronni Marshak [9], il résume bien les principales caractéristiques d'une application workflow.

- Routes: les routes déterminent des chaînes d'activités plus ou moins complexes (parallélisme, séquençement...) qui forment des cheminements le long desquels se déplacent les objets de gestion [8]. Les routes définissent les itinéraires du workflow, c'est-à-dire la synchronisation des activités et les chemins des flux informationnels qui s'appliquent en fonction des règles plus ou moins prédéfinies.

- Règles: les règles regroupent les informations concernant les tâches à réaliser pour accomplir une activité (règles de gestion, formulaire, données, applications) [8], c'est-à-dire les tâches et les éventuelles nécessités à leur réalisation. Ces règles permettent de fixer les principes de gestion, les données et les applications externes de

workflow qui seront souhaitées.

- Rôles: les rôles définissent les compétences nécessaires pour assurer la responsabilité d'activités à accomplir et de résultats à obtenir.

1.2.3 Exemple de workflow

Des articles sont achetés dans un grand magasin par un client qui n'est pas satisfait de la marchandise et retourne le produit pour obtenir un remboursement. Ce retour affecte plusieurs départements du magasin. La figure suivante montre la modélisation de ce workflow.

La réception (ou service à la clientèle) reçoit les biens et rembourse le client

- Des employés ramènent les articles dans l'entrepôt et les remettent dans le stock
- Les gestionnaires d'inventaire modifient les bases de données pour noter la transaction de retour de marchandise
- Les responsables des promotions (ou marketing) déterminent le prix auquel l'article était vendu pour le déduire des revenus. En effet, le prix du produit peut avoir changé depuis le moment de la vente.
- Le responsable des ventes réduit la commission du vendeur qui a vendu le produit selon le cas, et consigne le remboursement du client.

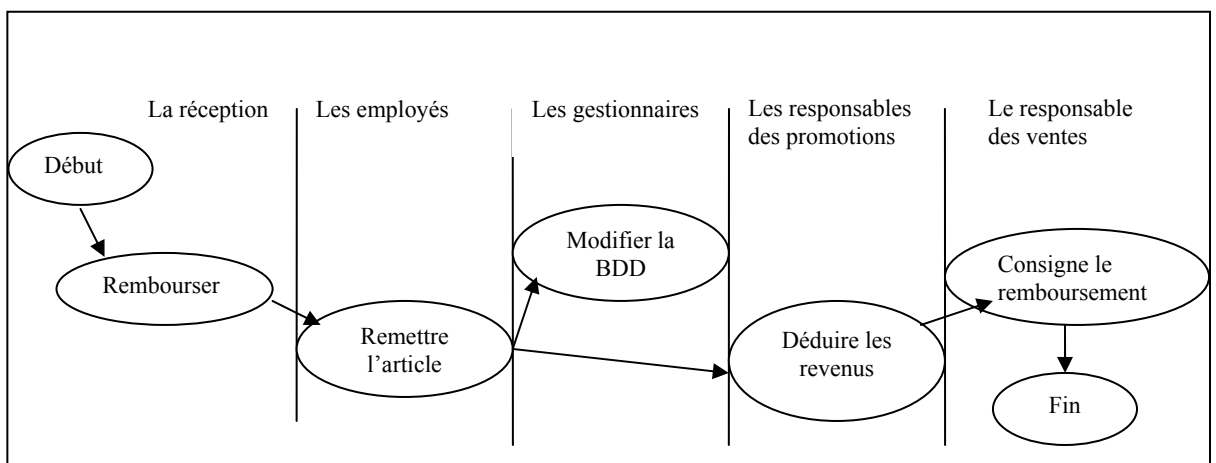


Figure 1.1: Exemple de workflow

1.2.4 Système de gestion de workflow (SGWF)

Le système de gestion de workflow est un système complet qui sert à définir, gérer et exécuter des procédures (workflow) en exécutant des programmes dont l'ordre d'exécution est prédéfini dans une présentation informatique de la logique de ces procédures [Levan, 99]. Cette exécution passe par l'assignation, pour chaque activité, des tâches aux personnes compétentes. La figure suivante illustre l'utilisation d'un tel système dans le cas d'un processus d'approbation de crédit dans une banque. Le processus requiert trois interventions humaines. La première consiste à saisir les informations relatives à la demande. La seconde consiste pour un responsable de crédit à approuver le prêt et enfin la dernière activité consiste pour le gérant de la banque à donner son approbation finale.

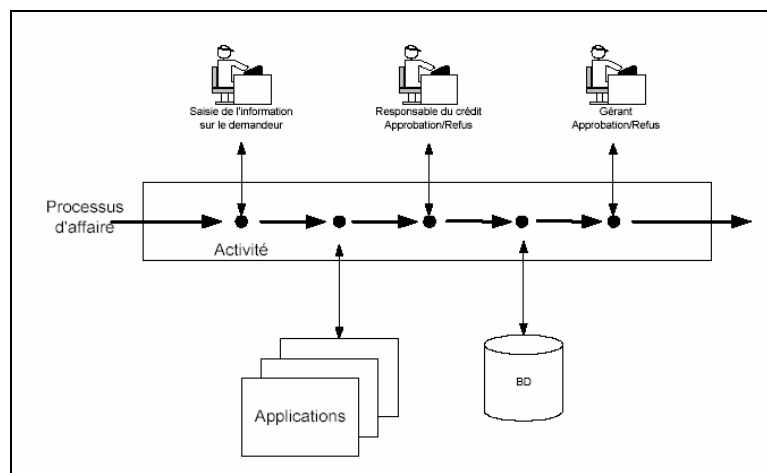


Figure 11.2: Le système de gestion des workflows

Le moteur de workflow contrôle la réalisation de processus alimenté par les règles définies dans le modèle de processus il calcule les tâches à réaliser en fonction des tâches déjà effectuées et des données qui y ont été enregistrées. Il informe en temps réel les différents acteurs en leur faisant parvenir le travail qui leur est affecté, un peu comme un serveur de messagerie distribue le courrier électronique entre les utilisateurs. La figure suivante présente les composants de base d'un SGWF ainsi que les interfaces entre ces composants.

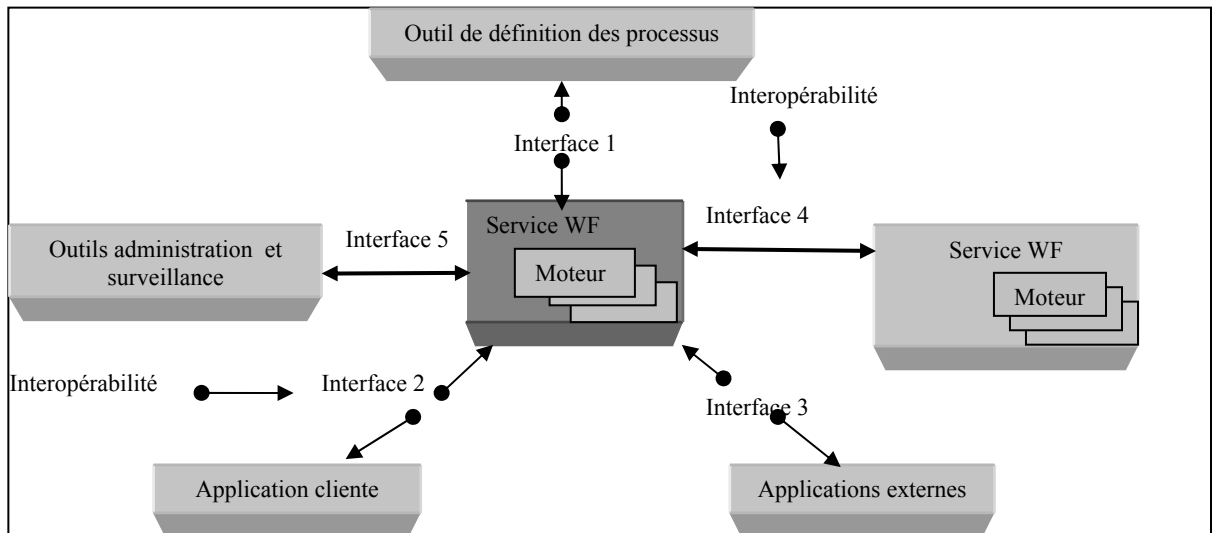


Figure 1.3: Modèle de référence d'un SGWF [21]

1) Outils de construction/définition des processus : ce sont des outils utilisés pour analyser, modéliser et décrire un processus dans une notation abstraite la logique de fonctionnement des processus. Le résultat de ces outils est une définition des processus qui sera interprétée par le moteur workflow.

2) Service workflow centre nerveux du système workflow : ce service consiste en un ou plusieurs moteurs de workflow, il est capable d'exécuter un ou plusieurs workflow [55]. Le serveur workflow a deux interfaces (2 et 3) qui représentent d'un côté le lien vers les applications clientes du workflow et d'un autre côté les applications appelées par le workflow.

3) Application cliente : l'application cliente est chargée d'organiser le travail pour le compte d'un utilisateur. Cette application cliente peut appeler des applications et des outils logiciels nécessaires à l'accomplissement des tâches.

4) Applications externes : Les systèmes de gestion de workflow doivent communiquer avec toutes les applications externes nécessaires à l'accomplissement des tâches, pour cela des ressources sont utilisées par le serveur workflow pour compléter les activités des processus.

5) Outils d'administration et de surveillance : ce sont des outils utilisés pour suivre l'exécution des processus. Ils permettent d'observer l'état d'avancement du processus.

Le modèle de référence d'un workflow dispose de cinq interfaces reliant les composants précédents entre eux :

1)Interface 1(serveur–concepteur) : elle permettant d'importer/exporter des définitions de processus. Elle définit un format commun pour l'échange des définitions (spécifications) des processus entre l'outil de définition des processus et le serveur workflow.

2)Interface 2(client-serveur): Cette interface est celle du client à travers laquelle le moteur de workflow interagit. Elle supporte les interactions entre l 'application client du workflow et le serveur workflow.

3)Interface 3(invocation d'application) : cette interface permet au moteur de workflow d'appeler une application spécifique d'une activité donnée pour exécuter une certaine tâche [55].

4)Interface 4(serveur –serveur) : (interopérabilité workflow / workflow) : il existe des produits de workflow qui sont de simples outils de routage de données, et il en existe d'autres qui font la gestion de processus complexes et évolutifs. Tous ces produits satisfont des besoins propres aux utilisateurs et tirent des avantages dans leur domaine. Il faut donc que ces différents produits puissent interagir [55].

5)Interface 5 (surveillant-serveur) : Il s'agit de définir un standard d'interface permettant à un outil d'administration et de pilotage de travailler avec n'importe quel moteur de services workflow [35].

1.3 La modélisation

Dans cette section, nous proposons de définir et positionner certaines notions de modélisation, afin de lever toute ambiguïté terminologique dans la présentation que nous faisons des techniques de modélisation de donnée dans ce chapitre où dans les chapitres suivants.

Méta modèle : décrit de manière formelle les éléments de modélisation, la syntaxe et la sémantique de la notation qui permet de le manipuler [58].

Modèle : un modèle tel que défini par N.Kettani [58] est une abstraction de la réalité qui, pour un domaine, est prise comme une représentation d'une classe de phénomènes plus ou moins habilement dégagés de leur contexte par un observateur, pour servir de support à l'investigation et/ou à la communication. Un modèle est toujours construit sur la base d'un langage que celui-ci soit informel (langage naturel, par exemple), semi-formel (s'appuie sur un vocabulaire et une syntaxe formels par exemple graphique) ou formel (ayant une syntaxe et une sémantique bien définies).

La cartographie des processus : C'est la première étape de la mise en oeuvre de la re-modélisation des processus [21]. Avant de re-modéliser, il faut connaître et il faut comprendre, ce qui implique 'modélise le workflow'. Une cartographie est un plan qui identifie les processus (les activités, les rôles...) afin de montrer les liens opérationnels entre les données d'entrée et les données de sortie.

1.3.1 La modélisation des processus

La modélisation des processus vise tout d'abord à représenter sous forme informatique (souvent graphique), le fonctionnement de l'entreprise en utilisant un langage spécifique [54]. Il est important d'arriver à une modélisation qui est suffisamment pertinente pour qu'on puisse se baser sur elle dans des buts d'amélioration de processus.

Un workflow est représenté comme un ensemble de tâches (composites ou élémentaires) liées par des relations de composition. Cette représentation induit un arbre dont les feuilles, les tâches atomiques (élémentaires), sont liées par des relations de composition.

Une représentation du workflow doit tenir compte des dimensions logique et temporelle de la problématique du workflow [26]. On peut spécifier un workflow par un ensemble d'objets (les tâches) et de règles (les relations entre les tâches). Il existe différents éléments de base d'un processus devant être modélisés :

- ❖ L'activité symbolisant une étape du processus.
- ❖ Le rôle accomplissant une activité.
- ❖ La route représentant la transition entre les activités.

Une activité de modélisation commence donc par la description de l'existant. Comment l'entreprise fonctionne ? Qui fait quoi ? Comment est-ce que les choses sont faites ?

1.3.2 Les éléments de Modélisation d'un workflow

Nous donnons dans cette section la définition des principaux concepts utilisés dans la modélisation du workflow. Nous utilisons ensuite ces concepts pour définir notre modélisation. Les principaux concepts d'un processus workflow sont:

- **Activité** : est un ensemble partiellement ordonné de tâches. L'activité est l'objet élémentaire de gestion qui permet de maîtriser le management d'un processus [Saadoun, 02]. Une activité est un enchaînement de tâches élémentaires, déclenchées en amont par une demande et aboutissant à un résultat attendu. L'activité est le niveau de décomposition d'un processus.

- **Tâche** : est une unité de travail, accomplie par un rôle pendant une période bien déterminée. La tâche (ou opération) correspond à un niveau de décomposition de l'activité. Comme nous l'avons précisé, une activité est un enchaînement de tâches élémentaires. Dans un workflow, on peut distinguer des tâches élémentaires et des tâches composées. Ces deux catégories de tâches sont prises en compte dans la conception du workflow dans le chapitre 3.

- **Rôle** : un rôle est une collection d'acteurs, collaborant afin d'atteindre un objectif commun. Lors de la conception d'un workflow, chaque tâche doit être associée à un rôle qui en a la responsabilité.

- **Acteur** : est une personne, ou un outil.

Le workflow est un ensemble de tâches. Les informations sont échangées entre les participants selon des règles prédéfinies pour accomplir ou attribuer à un objectif global.

1.3.3 Les méthodes de modélisation de workflow

Il n’existait pas une méthode standard permettant de modéliser un processus métier, et d’en dériver une implémentation pouvant être exécutée, mais diverses méthodes sont utilisées pour la modélisation des processus métier, nous citons :

- ◆ SADT (Structured Analysis and Design Technique) proposée par Ross [48] à la fin des années 1970 pour permettre une analyse structurée des systèmes, a ouvert la voie à la modélisation par représentation graphique des activités et des chaînes d’activités. La méthode SADT introduit le principe de décomposition fonctionnelle et formalise le concept d’activité. Elle se présente comme un langage graphique et un ensemble limité de primitives, des «boîtes» et des «flèches», pour la représentation des composants des systèmes et des interfaces.
- ◆ D’autres méthodes, plus élaborées mais toujours issues du génie logiciel proposent des supports d’analyse statique ou dynamique en se basant sur des approches fonctionnelles, relationnelles ou objet (MERISE et ses modèles de traitement, OMT [8], OOD [25], OOSE [17], UML, etc.).
- ◆ UML – Unified Modeling Language – propose des diagrammes spécialisés (dont les diagrammes d’activité, de séquence, de classe, etc.) ayant chacun une fonction précise. Il n’existe pas un diagramme UML spécialisé pour la modélisation des processus métiers. Cependant, UML propose un mécanisme d’extensibilité permettant de spécialiser chaque diagramme pour une utilisation particulière. Il est par exemple possible de spécialiser les diagrammes d’activité pour la modélisation des processus métiers.

La limitation de ces méthodes est tout de même de creuser un gap entre la modélisation des processus et leur exécution. Le langage d’exécution de processus est directement interprétable par un moteur. Il est donc nécessaire de créer une correspondance entre la représentation graphique des processus (UML ou autre) et leur langage d’exécution, afin de permettre un déploiement direct des processus modélisés. Ce besoin n’est pas couvert par UML, qui adresse uniquement la modélisation. A cause de cette

inconvenient, nous développons notre propre méthode de modélisation (chapitre 3), pour représenter un workflow, mais en utilisant ces méthodes connues.

1.4 Business Process Redesign (BPR)

La définition fournie par Hammer et Champy [14], pour la re_conception des processus métier correspond au "ré examen fondamental, re-définition radicale des processus de l'entreprise pour avoir des améliorations substantielles en terme de coût, qualité, service et de temps". La re-modélisation des processus d'affaires doit améliorer l'efficacité des processus dans une organisation.

1.4.1 L'approche BPR

L'approche de la re-modélisation des processus d'entreprises est basée sur les étapes suivantes [26] :

- Il faut interroger toutes les personnes impliquées, pour définir quel est leur rôle dans un processus de travail, quels sont les facteurs intervenant dans l'exécution de leurs tâches, quel est le temps qu'il leur faut pour les accomplir, avec quelles autres personnes sont elles en communication directe pour les besoins du travail. Cela permet de dégager la structure générale du processus de travail.
- On inventorie les ressources (humains, machines) et les tâches effectuées, ainsi que les contraintes qui portent sur ces tâches (temps, facteurs externes).
- On effectue un découpage des tâches impliqués dans le processus global (Hiérarchisation).
- On modélise le workflow à l'aide d'un outil d'édition ; une vue graphique du workflow peut alors mettre en évidence certains dysfonctionnements simples. On peut alors mettre en place le workflow, application qui s'exécutera sur une machine serveur reliée à toutes les cellules de l'entreprise.
- Après l'exécution d'un certain nombre de cycles du processus de travail, on peut observer un échantillon caractéristique de résultats réels (les temps effectifs, les blocages éventuels).
- Une étude poussée effectuée par des personnes compétentes en organisation à partir des résultats permet alors d'identifier les problèmes, pour résoudre ces problèmes on fait

appelle au BPR, on est en mesure d'apporter des solutions viables aux problèmes d'organisation.

Le BPR est alors un outil mettant en place l'organisation définie. A chaque fois qu'un problème se présente, nous sommes obligés d'utiliser le BPR.

1.4.2 Pour quoi re-modéliser un processus métier?

La plupart des points sensibles dans les processus métier se trouvent à l'intersection entre les personnes, les systèmes informatiques et les processus eux-mêmes. Les causes d'inefficacités principales sont [26], [3], [29]:

a) Applications et processus cloisonnés

Des processus business élaborés passent par plusieurs applications qui sont rarement reliées ou intégrées. De ce fait, les employés passent par des procédures manuelles pour faire le pont entre les applications. Ces manipulations sont bien entendu peu efficaces et surtout induisent de nombreuses erreurs, que ce soit lors de la collecte d'informations depuis un système, dans le traitement manuel de l'information ou dans la retranscription de ces informations traitées dans un autre système (où de plus, il manquera le contexte de l'opération). Par exemple, dans le cas d'une société qui regroupe plusieurs nouvelles entités par acquisition où les commandes sont centralisées mais pas les systèmes d'informations disparates des différentes entités.

b) Changements dans les relations commerciales

L'évolution ou le changement radical des manières de faire du commerce entre l'entreprise et ses partenaires conduit à devoir ajouter de nouveaux processus business ou de prendre en compte de nouvelles exceptions. Par exemple, dans le cas de multinationales, de nombreux clients exigent une facturation unique pour tous les produits du groupe, ce qui requiert un grand travail, souvent manuel, d'agrégation et de contrôle de la part d'employés qualifiés sur de nombreux systèmes.

c) Gestion des exceptions dans les processus métier

La plupart des applications automatisent les parties des processus business qui se déroulent normalement mais ne sont pas du tout prévues pour gérer les exceptions qui

peuvent se produire, si ce n'est l'envoi d'une notification d'erreur à un employé particulier. Celui-ci se retrouve la plupart du temps à devoir résoudre manuellement l'exception ce qui passe souvent par une étape de recherche manuelle des informations nécessaires. Un exemple souvent rencontré concerne le bouclage périodique des comptes qui requière une recherche fastidieuse en cas de balance inexacte.

d) Processus non automatisés ou non optimisés

De nombreuses étapes à l'intérieur de processus business ne sont pas supportées par les applications de l'entreprise, ce qui cause de nombreuses pertes de temps ainsi que des erreurs. Par exemple, même si une grande partie du processus de commande se fait de manière électronique, la commande finale et souvent transmise par fax avant d'être introduite dans le système (au lieu de prévoir un système de commande électronique complet pour des acheteurs authentifiés).

1.4.3 Les phases d'évolution du BPR

La phase « considérer le processus comme source pour améliorer l'exécution de l'organisation », est passé par un certain nombre de phases dans les 20 dernières années [14] (la figure suivante). D'abord, il y avait la phase de gestion de la qualité totale, qui est très utilisée dans le début des années 80 [14] ; cette phase est basée sur l'amélioration continue, en conséquent, le BPR est apparu dans le début des années 90 [14]. En milieu des années 90, la première vague BPR est apparue [14].

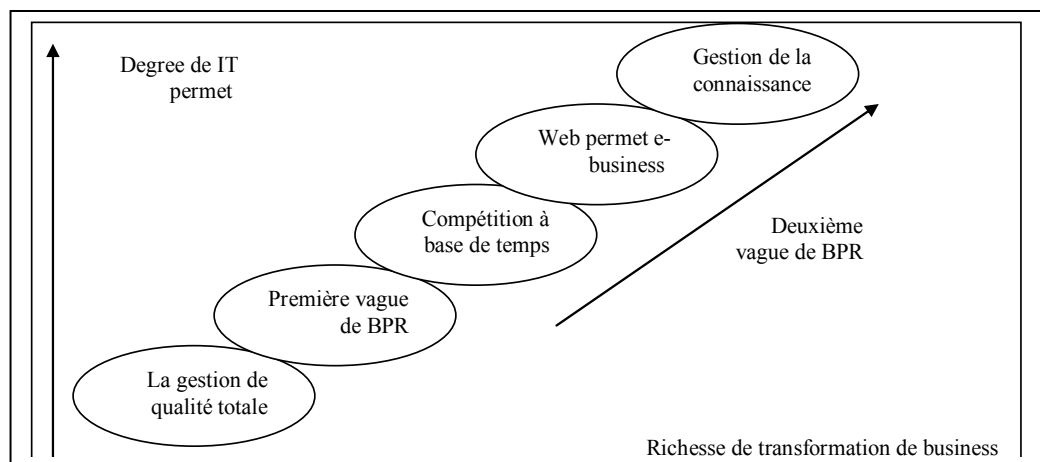


Figure 1.4: Les vagues d'amélioration de processus métier [14]

Ensuite, l'Internet et le phénomène de World Wide Web (WWW) sont réponsus depuis 1995 [14], ont permis l'apparition du commerce électronique et des nouvelles formes de Web basés sur le processus métier. En 1998 [13] les entreprises ont commencé à re-modéliser leurs processus métier pour tirer profit de l'Internet.

Les processus métiers ont été re-modélisé par les entreprises pour profiter les possibilités croissantes de l'Internet. Une autre approche de BPR est proposée récemment (l'année 2000) par Edward.W [13] pour améliorer le processus métier, basée sur la gestion de la connaissance efficace autour du processus métier

L'ensemble de toutes les approches d'amélioration de processus métier est appelé la deuxième vague BPR [14], cette dernière a des dimensions plus riches que la première vague BPR. Elles profitent de l'acquisition et de la création des connaissances autour de processus métier; et elles tirent les avantages de l'Internet.

Le meilleur BPR celui qui prend la forme de deuxième vague de BPR [14], et le BPR le plus efficace utilise la dernière technologie employée pour le BPR, c'est la gestion de la connaissance, donc notre choix est porté sur cette technologie, pour aboutir à un système BPR plus efficace. Nous présentons cette technologie dans la section suivante.

1.5 La Gestion des connaissances (GC)

La gestion des connaissances est « un mode de gestion systématique des savoir-faire et des connaissances dans les organisations, dont la finalité est de leur permettre d'obtenir un avantage compétitif » [04]. La gestion des connaissances a donc pour mission d'améliorer la performance de l'entreprise. En outre, la GC permet d'obtenir une vision d'ensemble des compétences et des savoirs de l'entreprise [16].

1.5.1 Les principes de la gestion des connaissances

La gestion des connaissances est l'ensemble des méthodes et des techniques permettant d'identifier, d'analyser, d'organiser, de stocker, de fournir et de partager des

connaissances aux membres de l'organisation [Karl, 04], en particulier les savoirs créés par l'entreprise elle-même (recherche et développement) ou acquis de l'extérieur (intelligence économique). Les acteurs de l'organisation ne doivent pas se limiter à la consommation d'informations, il faut qu'ils en soient également producteurs pour favoriser les échanges et pérenniser la capitalisation des connaissances.

Rao Madanmohan [33] caractérise la GC à l'aide d'un modèle, que nous présentons sur le schéma de la figure suivante.

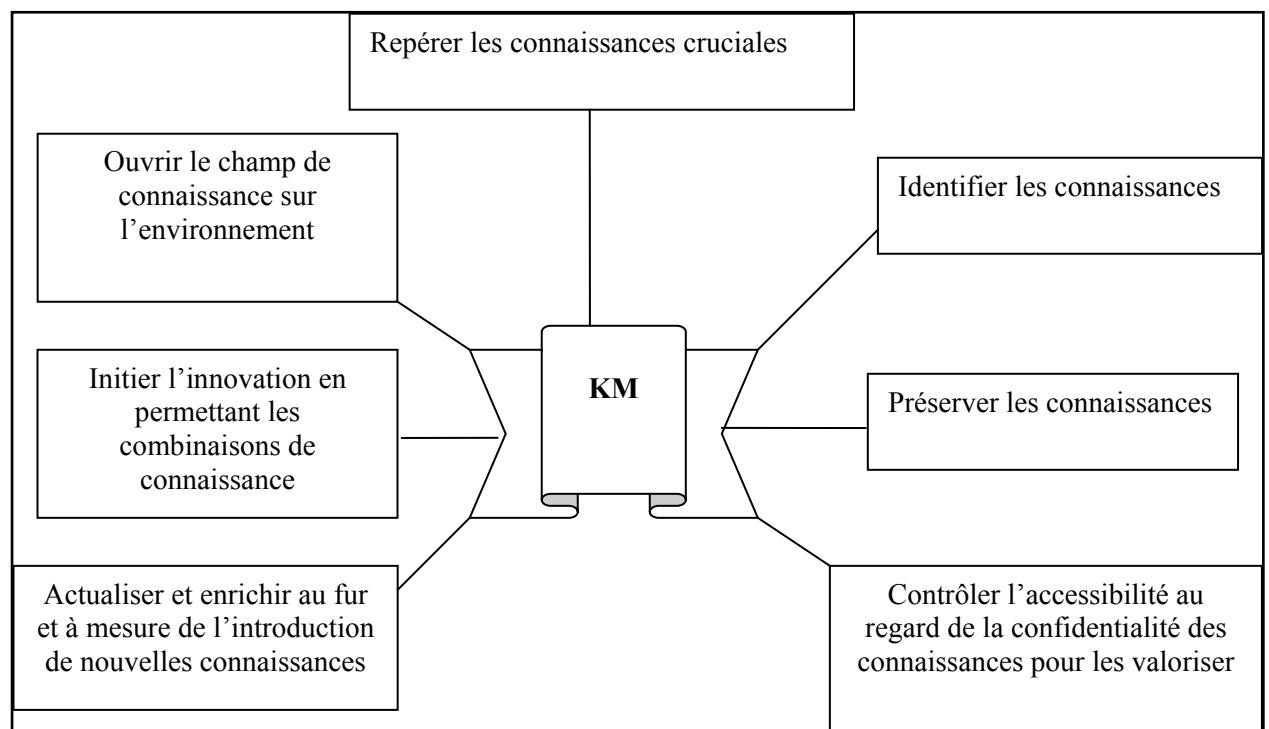


Figure 1.5 : Le modèle de KM [54]

Une des premières tâches est de repérer les connaissances cruciales, c'est-à-dire les savoirs et les savoir-faire nécessaires au déroulement des processus essentiels constituant le cœur des activités de l'entreprise. Il faut les identifier, les localiser, et les caractériser. Ensuite, il faut les préserver, c'est-à-dire les modéliser, les formaliser et les conserver. Puis, il faut valoriser et mettre au service du développement et de l'expansion de l'entreprise ces savoirs et savoir-faire, c'est-à-dire les rendre accessibles selon certaines règles de confidentialité et de sécurité, les diffuser, les exploiter, les combiner et créer des connaissances nouvelles. Enfin, il faut pouvoir les actualiser, c'est-à-dire les évaluer, les

mettre à jour et les enrichir au fur et à mesure des retours d'expérience et de la création de connaissances nouvelles. La gestion des connaissances consiste :

- ❖ à collecter le savoir des personnes de la structure/firme par des entretiens et à le rassembler dans des ouvrages de référence et/ou bases de connaissance,
- ❖ éventuellement à mettre en forme et échanger ce savoir par le biais du système de la Communauté de pratique
- ❖ à diffuser ce savoir au sein de la structure/firme, notamment (mais pas seulement) par des moyens informatiques.

La vision stratégique, que peuvent avoir les dirigeants d'entreprise sur leur patrimoine de connaissances, les amène à définir des objectifs globaux pour gérer au mieux les ressources. Ces objectifs s'articulent toujours autour de trois points-clés :

Capitaliser : ce principe est résumé par " Savoir d'où l'on vient, savoir où l'on est, pour mieux savoir où l'on va " [10]. Il procède d'une logique " évolutionniste " de la connaissance, à savoir que l'évolution des connaissances (et donc de l'entreprise) se fait toujours à partir d'un patrimoine existant.

Partager : il est résumé par " Passer de l'intelligence individuelle à l'intelligence collective " [10]. Il est l'aboutissement du constat que, actuellement, dans la gestion de la complexité de leur processus, les entreprises ne peuvent plus se satisfaire des actions individuelles des acteurs, mais doivent les intégrer dans des ensembles collaboratifs harmonieux et créatifs.

Créer : ce principe est résumé par " Créer, innover pour survivre " [10]. Dans le monde économique actuel, la survie des entreprises passe par une innovation constante et soutenue. Le processus de créativité et d'innovation des entreprises passe désormais par une interaction bien comprise et bien gérée entre ses ressources internes de savoir et son environnement économique et concurrentiel.

1.5.2 La nécessité de la gestion de la connaissance pour les entreprises

La gestion de la connaissance se concentre sur 'faire la bonne chose 'au lieu de 'faire des choses bien' [26]. La gestion de la connaissance est une plateforme dans laquelle

l'organisation regarde tous ses processus comme un processus de manipulation de la connaissance. Dans cette vue, tous les processus métiers impliquent la création, la diffusion, le renouvellement, et l'application de la connaissance vers l'alimentation et la survie d'organisation [39]. La gestion des connaissances a été créée à la suite de plusieurs constats :

- ❖ L'information n'est pas pérenne dans l'organisation :
 - L'information est la principale richesse des organisations modernes, sous forme de technologies, de savoir, de savoir-faire, de brevet, de compétence métier, de stratégies ou autres ;
 - Dans le fonctionnement traditionnel d'une entreprise, l'information utile (organisée) est essentiellement détenue par les employés de l'organisation, avec tous les aléas que cela peut comporter (mutation, départ à la retraite, démission, licenciement ou autres causes d'indisponibilité) ;
- ❖ La quantité d'informations disponibles pour les membres d'une organisation est trop importante pour qu'ils puissent rapidement trouver les informations pertinentes et utiles. Par exemple, certains managers sont tellement envahis par les courriers électroniques qu'ils ne les lisent plus du tout, faute de pouvoir trier efficacement entre messages utiles et inutiles.

1.5.3 Les objectifs

Les objectifs de la gestion des connaissances sont résumés en [15]:

- ❖ Le premier objectif vise à repérer les connaissances cruciales : ces connaissances fondent ce que certains pourraient appeler le capital immatériel de l'entreprise.
- ❖ Un second objectif apparaît dans l'identification, la localisation puis la caractérisation des connaissances, dans le but d'en estimer la valeur. Cet objectif est essentiel au sein des structures, car cette identification permet de limiter le nombre de sources de connaissances nécessaires.

- ❖ Le troisième objectif est celui de la mémorisation, de la préservation des connaissances dans le temps. Cette mémorisation implique un ensemble d'étapes successives d'acquisition, de modélisation, de formalisation et de conservation.
- ❖ Le quatrième objectif est celui de la sécurité liée à l'expression explicite de connaissances sous forme d'informations valorisables au sein ou hors de l'organisation. Cette sécurité passe par la mise en œuvre de clauses de confidentialité dans l'accès à l'information, elle s'agit d'assurer la fiabilité et la sûreté des connaissances formalisées.
- ❖ Le cinquième objectif consiste à permettre l'actualisation des connaissances capitalisées en autorisant l'introduction de nouvelles connaissances et en assurant la suppression des connaissances caduques.
- ❖ Le sixième objectif de la gestion des connaissances est de constituer le moteur d'une création de la connaissance. Ce principe découle de l'analyse de I. Nonaka [9] sur la combinaison créatrice des connaissances dans une organisation. Les travaux de Nonaka décrivent l'importance de l'existence concomitante des connaissances explicites et implicites dans une organisation et de leur degré de diffusion, dans le processus de création de connaissances. Nonaka explique la création de nouvelles connaissances comme la résultante d'un cycle de transformation des connaissances du niveau individuel au niveau organisationnel au travers des processus d'explicitation et d'internalisation.
- ❖ Enfin, le septième objectif concerne la prise en compte par ces méthodes des connaissances environnementales proches du domaine de connaissances. L'ouverture des modèles de connaissances permet de stimuler une dynamique de la connaissance et un enrichissement des modèles.

1.5.4 Le cycle de la GC

Un système de gestion de la connaissance est caractérisé par les quatre activités suivantes [44]:

- Acquisition de la connaissance (apprendre, créer, et identifier) ;

- Analyse de la connaissance (évaluer, valider, et localiser) ;
- Conservation de la connaissance (organiser, représenter, et maintenir) ; et
- L'utilisation de la connaissance (appliquer, transférer, et partager).

Pour contrôler la connaissance nous devons d'abord avoir la connaissance à contrôler, puis nous analysons la connaissance que nous possédons, ensuite, nous devons stocker la connaissance dans une base de connaissance, à la fin, nous pouvons accéder et employer la connaissance à l'avenir. Ces activités n'existent pas en isolation, nous pouvons les penser comme un cycle suivant les indications du schéma suivante [41].

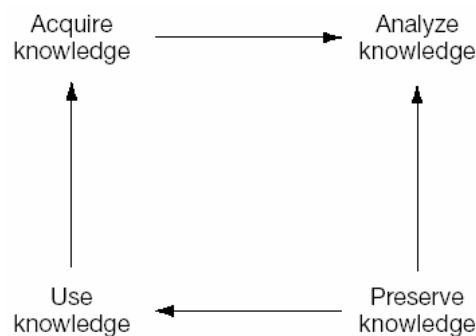


Figure 1.6 : Le cycle de KM [35]

L'élément qui lie le cycle de KM est l'utilisation de la connaissance, puisque, quand la connaissance est employée, un nouvel aperçu de la connaissance est créé [37]. Ces nouvelles connaissances exigent leurs acquisitions, analyse, et leur préservation pour l'utilisation future; le système de gestion de la connaissance apprend et évolue continuellement.

1.6 Gestion des workflows

La réutilisation de modèle workflow ou de parties de workflow pour re-modéliser un workflow a un double intérêt pour l'expert de modélisation [34]:

- Le gain de temps : la re-modélisation d'un processus métier est un travail long et complexe pour lequel la réutilisation permet de réduire certaines étapes (que ce soit pour le choix de la stratégie en générale ou de la technique appliquée pour un point particulier).

- Le partage des connaissances et des expériences : un utilisateur peut réutiliser une application créée par un autre utilisateur, ou utiliser les connaissances et l'expérience modélisées pour un domaine d'application dans un autre domaine.

Pour mettre en œuvre ce type de réutilisation, nous avons choisi d'employer la technique gestion des connaissances, cette dernière, recouvre un ensemble de modèles ou méthodes pouvant mettre en œuvre des outils de traitement de l'information et de communication visant à structurer, valoriser et permettre un accès par toute l'organisation aux connaissances qui y ont été développées et qui y ont été ou sont encore mises en pratique en son sein [54].

1.7 Conclusion

Le workflow constitue le comment du processus, de l'enchaînement d'activités. Il définit la suite de tâches manuelles ou automatisées exécutées sur plusieurs postes de travail, selon un ordre. Un système de gestion de workflows exécute de façon dynamique chaque activité en assignant à chaque individu les tâches qu'il doit compléter.

Dans ce chapitre nous avons présenté la relation entre la re-modélisation des processus métier et la gestion des connaissances. Le développement des systèmes BPR est passé par plusieurs étapes. Dans chaque étape les chercheurs utilisent une technologie soutient les principes et les techniques de BPR. Dans la présente étude, nous sommes intéressés à la dernière technologie utilisée pour le BPR c'est la gestion des connaissances, car les autres technologies ont des inconvénients, ces derniers dirigent les chercheurs à utiliser le KM pour le BPR [14].

Afin de surmonter les difficultés présentées par la mise en œuvre de la gestion des connaissances des workflows, nous proposons d'utiliser le raisonnement à base de cas comme une technique privilégiée de gestion des connaissances, pour maintenir les mémoires d'organisation, de sorte que la connaissance puisse être stockée et employée à l'avenir.

Nous tentons de montrer les raisons pour lesquelles le raisonnement à base de cas est adopté pour résoudre le problème de gestion de la connaissance dans les systèmes BPR, par l'extraction de la synergie qui existe entre le KM et le CBR, et la relation entre le BPR et le CBR, dans le chapitre suivant.

CHAPITRE 2

RAISONNEMENT A BASE DE CAS

2.1 Introduction

Un système de KM peut-être mieux regardés comme un system intelligent [48], que le considéré comme une technologie d'un système. Ainsi, un système de KM doit soutenir toutes les activités qui encouragent la culture de la connaissance partagée avec l'apprentissage de organisation. Le CBR fournit l'appui méthodologique pour ces activités [43]. En conséquence, notre système de gestion de la connaissance devrait employer CBR en tant que leur méthodologie noyau.

À ce jour, nous tous employons le CBR tout le temps pour résoudre les problèmes dans notre vie. Nous quotidiennement recherchent les expériences semblables du notre mémoires, nous réutilisons et mettons à jour la connaissance qu'ils contiennent, nous passons en revue les leçons apprises de nouvelles expériences et maintiennent elles, et l'excédent temps, nous raffinons notre connaissance, constamment filtrant et mettant la à jour.

Ce chapitre nous montre conceptuellement comment la méthode raisonnement à base de cas fournit des mécanismes pour traiter chacun des quatre activités de gestion de la connaissance et comment il correspond au cycle de KM, qui argumente notre choix de cette méthode, pour le faire nous réalisons une correspondance entre la technologie KM et la méthode CBR, puis nous étudions le paradigme de CBR, afin de mettre en évidence les tâches principales et les besoins de ce système en termes de connaissances et de mécanismes de raisonnement. Le CBR est d'abord arboré par une définition et une présentation de ces déférents modèles et étapes, en abordant les étapes de résolution d'un problème. La section 2.4 montre la relation qui existe entre le BPR et le CBR, en assimilons les étapes de BPR avec celle de CBR. La dernière section de ce chapitre

présente les étapes de développement d'un système de BPR en utilisant le KM et en suivant le CBR, et qui sont traitées dans les chapitres suivants.

2.1 Les technologies utilisées pour la GC

Pour réaliser les activités du KM des méthode de l'IA proposées, comme les systèmes experts (es), le raisonnement à base de cas, et les ontologies sont des méthodologies basées sur la connaissance, qui ont beaucoup contribué au systèmes KM parce qu'ils manoeuvrent la connaissance pour mettre en application les déférentes tâches [35]. Les références [40], [2], [39], et [31] présentent des applications de gestion des connaissances par ces technologies. Dans notre travail nous nous intéressons au CBR.

La gestion des connaissances consiste à acquérir et représenter les connaissances utiles à un domaine dans le but d'en favoriser l'accès, la réutilisation et l'évolution [53]. Il s'agit ensuite de fournir un accès à ces connaissances, c'est-à-dire de les diffuser dans le but d'en permettre une utilisation efficace. Nous nous intéressons plus particulièrement ici à la mise en œuvre des connaissances en workflow afin d'en fournir une re-modélisation intelligente au travers la plateforme BPR/CBR. Parmi les méthodes de KM existent nous utilisons quelle méthode?

À un atelier de l'université de Cambridge un groupe de chercheurs travaille en KM et l'intelligence Artificielle (IA) a identifié les activités principales utilisées par un système de KM [43], et qui sont correspond aux méthodes ou techniques d'IA. Les activités principales de KM ont été identifiées en tant que : acquisition, analyse, conservation et utilisation de la connaissance. Les avantages apportés par le CBR à notre système se rapprochent de ceux exposés par Kolodner [50] sur « l'aide à la décision basée sur les cas ». Elle rappelle que le RàPC est un raisonnement fréquemment utilisé par l'humain pour résoudre les problèmes, mais ce raisonnement est rarement exploité pleinement, pour plusieurs raisons :

- Il lui est impossible de conserver tous les cas en mémoire,
- Parmi ceux qu'il connaît, il ne se souvient pas toujours du cas le mieux approprié au moment voulu,
- S'il est un novice du domaine, il lui est difficile de réaliser la phase d'adaptation.

Kolodner propose donc un système qui permette à l'humain (le novice et l'expert) de mieux raisonner par analogie, en laissant au système la charge de la mémoire et de la sélection des meilleurs cas, et en laissant à l'utilisateur la charge de la décision finale. La section suivante présente la méthode CBR, puis nous montrons comment le CBR peut répondre à chacun des besoins de KM.

2.2 Raisonnement à base de cas

Après avoir défini la technique adoptée pour la réalisation de notre approche qui est le raisonnement à base de cas, nous présentons maintenant cette méthode et ses étapes.

L'ensemble des expériences passées forme une base de cas. La solution est construite en calculant la similarité entre le nouveau cas et les cas qui sont contenus dans la base de cas. Pour cette raison, on utilise une fonction de similarité dont le but est de déterminer la similarité de deux cas dans la base en se basant sur leurs attributs. On peut donc, à partir de la description d'un cas, obtenir une liste ordonnée (en fonction de la valeur donnée par la fonction de similarité) des cas similaires. La métrique de similarité est une fonction qui prend deux entités en paramètres et qui retourne une valeur reflétant la similarité entre ces deux entités selon un certain but.

2.2.1 Les principes du RÀPC

Nous considérons un cas, noté $(pb, sol(pb))$, comme la description d'un problème pb associé à sa solution $sol(pb)$ [1]. L'objectif principal du processus de RÀPC est d'établir une solution $sol(cible)$ d'un problème cible, noté $cible$, en réutilisant la solution $sol(source)$ contenue dans un cas source $(source, sol(source))$ connu. Les deux étapes fondamentales pour cela sont la remémoration et l'adaptation (voir Figure 22.1). La remémoration consiste à retrouver, parmi un ensemble de cas sources stockés dans une base de cas, un problème source jugé similaire au problème cible à résoudre. La solution du problème $sol(source)$ est ensuite construite par adaptation de la solution du problème source remémoré.

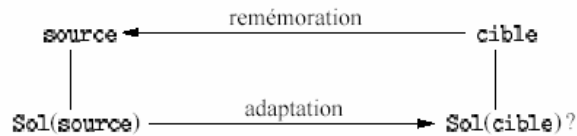


Figure 22.1: La re-mémoration et l'adaptation en RÀPC

2.2.2 Les étapes de CBR

Outre les deux étapes fondamentales de remémoration et d'adaptation qui caractérisent le RÀPC, d'autres opérations peuvent être considérées, en fonction des besoins du domaine d'application. En effet, Aamodt [9] définit un cycle du RÀPC composé de quatre étapes. À la suite de la remémoration et de l'adaptation (retrieve et reuse) s'ajoutent les étapes de révision de la solution (revise) et de mémorisation du cas (retain). La révision suppose d'être en mesure de tester la validité de la solution $\text{sol}(\text{cible})$ construite et éventuellement de la modifier pour la rendre valide. La mémorisation correspond à une étape d'apprentissage au sein du processus de RÀPC. Il s'agit, si cela est jugé utile et une fois la solution $\text{sol}(\text{cible})$ validée, d'ajouter le cas (cible, $\text{sol}(\text{cible})$) à la base de cas pour une éventuelle utilisation future.

Les différentes étapes du cycle du RÀPC sont détaillées dans la Figure 2.2 , et qui sont [1]:

L'élaboration : L'élaboration est l'étape où la représentation du problème cible est construite. Il s'agit ici d'aider l'utilisateur à formuler le problème de façon à ce qu'il soit manipulable par le système. Le modèle des connaissances du domaine fournit ici un cadre, un vocabulaire, pour cette description. Par ailleurs, des informations supplémentaires, venant s'ajouter à la description éventuellement incomplète du problème, pourront être inférées lors de cette étape, à partir des connaissances du domaine [1].

La remémoration : consiste à chercher dans la mémoire un cas passé résolu. Il s'agit d'identifier les caractéristiques les plus pertinentes, les indices, et de les utiliser pour rechercher des cas passés potentiellement réutilisables. Les cas de la mémoire sont ensuite appariés avec le problème courant afin d'évaluer leur similarité avec le problème courant,

et de les ordonner par similarité croissante. Le critère de similarité est défini de manière à sélectionner le cas qui aura des chances d'être adapté le plus facilement.

La réutilisation (Adaptation) : L'adaptation est souvent considérée comme l'étape la plus importante et la plus complexe du RÀPC [39]. C'est en effet lors de cette opération qu'est construite la solution du problème cible. Il s'agit de réutiliser la solution du problème source remémoré et de l'adapter afin qu'elle corresponde à une solution du problème cible. Les connaissances d'adaptation sont dans ce cadre essentielles pour déterminer les adaptations nécessaires, en fonction du cas source, du problème cible et des éléments de similarité entre source et cible utilisés lors de la remémoration. Certains systèmes s'appuient par ailleurs sur plusieurs cas sources et nécessitent ainsi l'adaptation et la combinaison de plusieurs solutions sol (source).

La révision : La révision consiste à tester la validité de la solution sol (cible) construite par adaptation et éventuellement à la réparer si elle n'est pas valide [36]. La révision est une opération essentielle, elle repose principalement sur les connaissances du domaine, qui contiennent les éléments permettant de reconnaître une solution comme valide. Néanmoins, les connaissances nécessaires pour cela n'étant que rarement formalisées, la révision est dans la plupart des cas réalisée par l'utilisateur [40].

La mémorisation: le nouveau cas est intégré dans la mémoire des cas passés. Cette intégration doit tenir compte des caractéristiques pertinentes par mise en place d'index appropriés [38]. La mémoire des cas est ainsi enrichie. L'apprentissage consiste à mémoriser le cas résolu et à réaliser des généralisations à partir des cas pour générer de nouvelles connaissances.

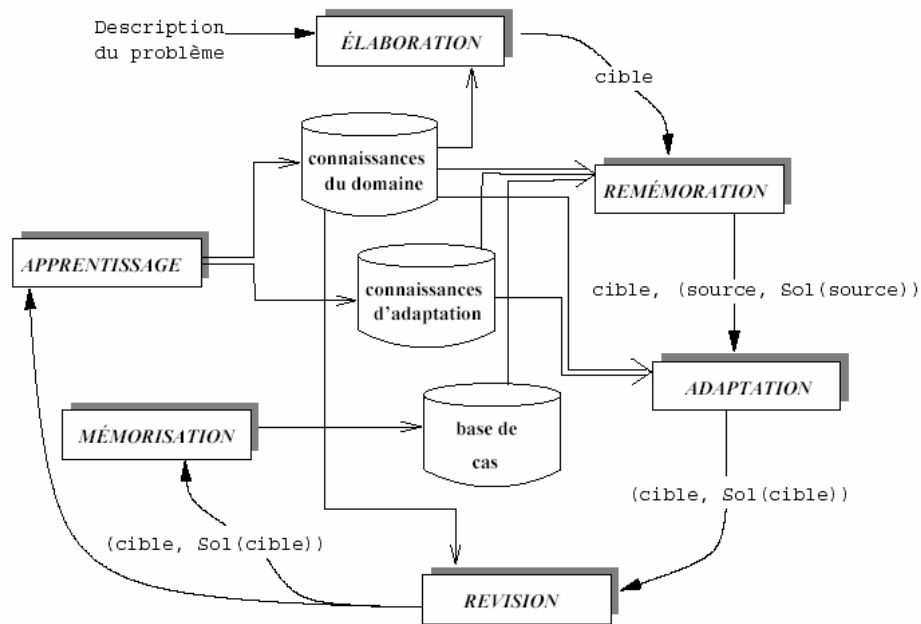


Figure 2.2 : Le cycle du RBC [25]

2.2.3 Les modèles de représentation d'un CBR

Il existe plusieurs modèles de représentation des cas pour le raisonnement à base de cas. Ces modèles sont regroupés en trois grandes familles: structurelle, conversationnelle et textuelle. Puisque ces modèles reposent sur des principes variés, nous décrivons dans les sections suivantes les plus importantes.

2.2.3.1 Modèle structurel

Le modèle structurel a émergé des premières vagues de systèmes CBR [16]. Dans ce modèle, toutes les caractéristiques importantes pour décrire un cas sont déterminées à l'avance par le concepteur du système (Figure 2.3). Les cas sont complètement structurés et sont représentés par paires <attribut, valeur>. D'un point de vue applicatif, un attribut représente une caractéristique importante du domaine d'application. Les échelles de valeurs les plus fréquemment utilisées pour structurer les attributs sont les entiers/réels, les booléens et les symboles.

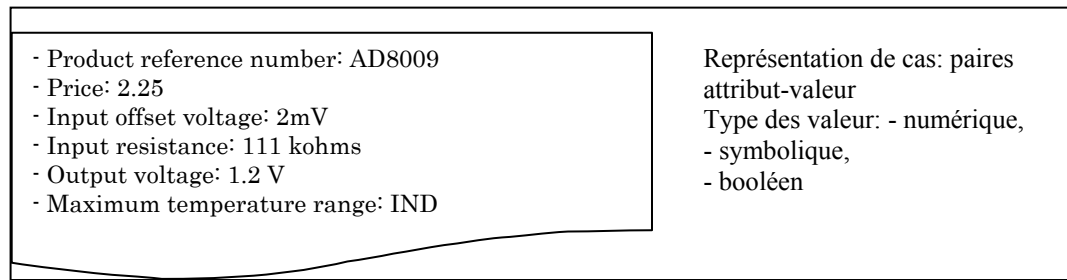


Figure 2.3 : Exemple de structuration d'un cas en CBR structurel

2.2.3.2 Modèle conversationnel

Dans le modèle structure, un problème doit être complètement décrit avant que ne débute la recherche dans la base de cas. Pour obtenir des solutions satisfaisantes, l'utilisateur doit avoir a priori une bonne idée de tous les facteurs pouvant influencer la résolution de son problème [28]. Toutefois pour certains problèmes, il est difficile de déterminer à l'avance les aspects de la situation importants. Par ailleurs, cette exigence présuppose une expertise du domaine d'application, ce qui n'est pas le cas chez tous les usagers de systèmes CBR. Des usagers novices éprouvent parfois des difficultés à bien caractériser une situation à l'aide de valeurs numériques ou symboliques.

Le modèle conversationnel a donc été proposé pour surmonter ces difficultés. Comme son nom l'indique, le modèle CBR conversationnel mise sur l'interaction entre l'utilisateur et le système pour définir progressivement le problème à résoudre et pour sélectionner les solutions les plus appropriées [8].

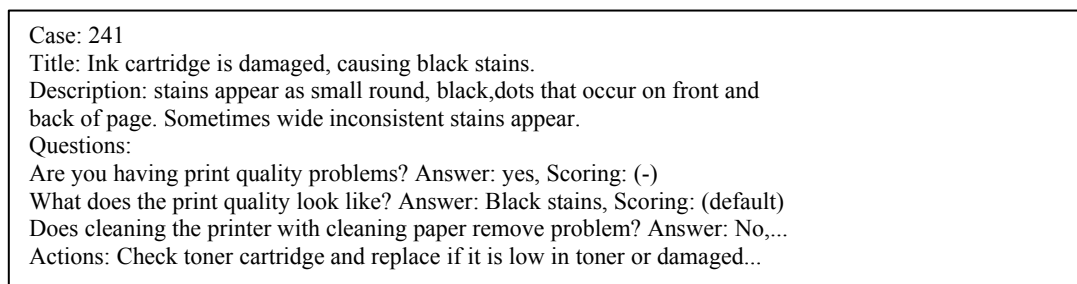


Figure 2.4: Exemple de cas pour le modèle conversationnel [56]

2.2.3.3 Modèle textuel

Les travaux sur le raisonnement à base de cas textuel portent sur la résolution de problème à partir d'expériences dont la description est contenue dans des documents textuels. Dans cette approche, les cas textuels sont soit non-structurés ou semi-structurés. Ils sont non-structurés si leur description est complètement en "free-text". Ils sont semi-structurés lorsque le texte est découpé en plusieurs portions étiquetées par des descripteurs tels que "problème", "solution", etc. Un cas textuel non-structuré est un cas qui a un seul attribut dont la valeur est textuelle tandis qu'un cas textuel semi-structuré est un cas dont un sous-ensemble des attributs est textuel.

2.2.4 La phase de remémoration

Le but de cette phase est de sélectionner un ou plusieurs cas pour proposer, soit directement une solution, soit des moyens de progresser vers la solution (modèles, méthodes, solution partielle, ...) [47].

Pour la réaliser, il est nécessaire de définir des techniques de remémoration et d'indexation pour retrouver les cas les plus pertinents et ce, le plus rapidement possible. Le système doit également posséder une (ou des) mesure(s) de similarité pour effectuer une sélection fine des cas.

Nous présentons ci-après quelques techniques de remémoration des systèmes étudiés en fonction du nombre d'étapes qu'ils nécessitent pour réaliser cette phase.

2.2.4.1 Remémoration en deux étapes

Certains systèmes séparent la phase de remémoration en deux étapes (Figure 2.5), la première ayant pour but de réduire l'espace de recherche et la deuxième de sélectionner les cas les plus proches du cas courant dans cet espace réduit. C'est le principe utilisé par les systèmes EADOCS, ISAC et DESIGnER [17].

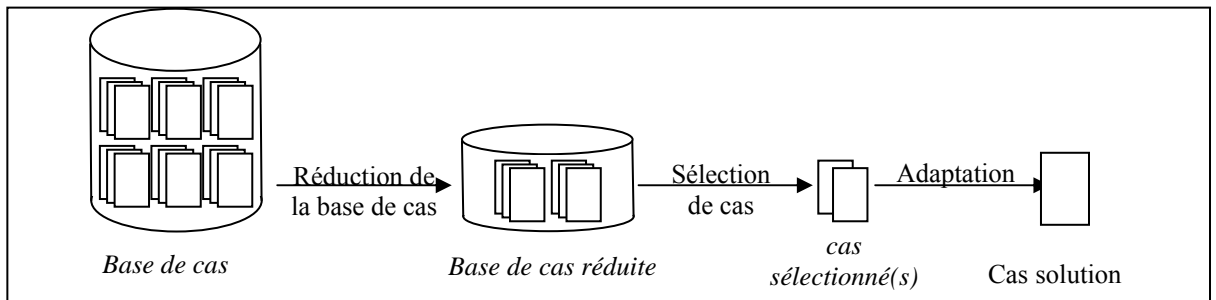


Figure 2.5 : Remémoration de cas en deux étapes

2.2.4.2 Remémoration multi-étapes

Les systèmes modélisant les cas sur plusieurs niveaux d'abstraction, en particulier les planificateurs à base de cas, utilisent généralement un cycle remémoration/adaptation (Figure 2.6) qu'ils exécutent, soit un nombre prédéfini de fois, soit jusqu'à respect une certaine condition. Nous présentons ici la phase de remémoration multi-étapes du système de Chiu, de Déjà-Vu et du système de Prasad [11][17].

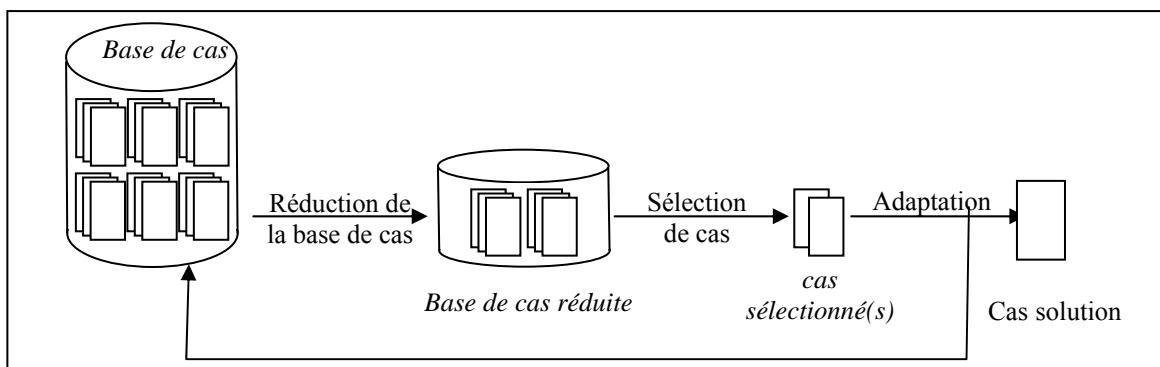


Figure 2.6: Remémoration de cas en multi-étapes

2.2.4.3 Remémoration à options

D'autres systèmes proposent plusieurs options pour réaliser la remémoration (Figure 2.7). Cela permet de choisir dynamiquement la technique à mettre en oeuvre en fonction de ce que recherche l'utilisateur ou de ce que le système peut lui proposer comme solution. Cette technique est utilisée par : le système Resyn/RàPC qui offre deux options différentes suivant qu'il existe ou non un cas s'appariant exactement au nouveau problème [23].

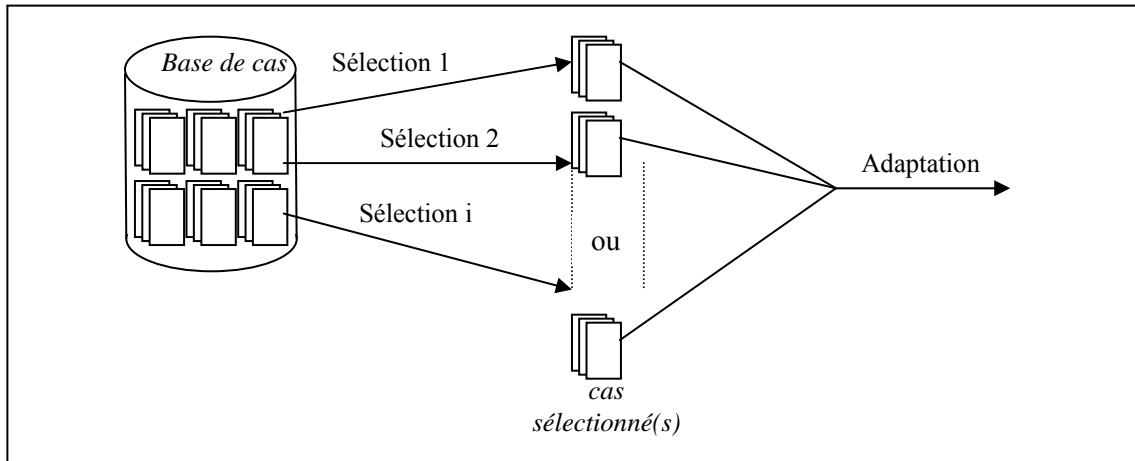


Figure2.7 : Remémoration de cas à option

2.2.5 Adaptation d'un cas

L'adaptation revient à ajuster un cas sélectionné pour le faire coïncider au problème cible à traiter [44]. La plus simple adaptation est de prendre la solution du cas telle quelle. En pratique n'existent pas des méthodes spécifiées pour l'étape d'adaptation, mais on peut utiliser les principes du raisonnement par analogie (analogical reasoning). L'existence d'une analogie entre le problème source et le problème cible est alors utilisée pour le transfert du principe de la solution source, vers le problème cible, afin de produire une solution cible. C'est pourquoi nous étudions, dans la suite, les approches de l'analogie.

Un problème d'analogie s'exprime selon l'expression : « D est à C ce que B est à A. Connaissant A, B, et C que vaut D » [39]. La résolution de problèmes par analogie consiste à transférer de la connaissance à partir d'épisodes précédents de résolution aux nouveaux problèmes, qui partagent des aspects significatifs de l'expérience précédente et à utiliser les connaissances transférées pour construire des solutions pour les nouveaux problèmes [34]. Dans la figure suivante A et B définissent un épisode de résolution, par un problème (A) et sa solution (B). C représente le problème à résoudre et D la solution recherchée Carbonell propose deux approches de résolution de problème par analogie : analogie par transformation et analogie par dérivation [33].

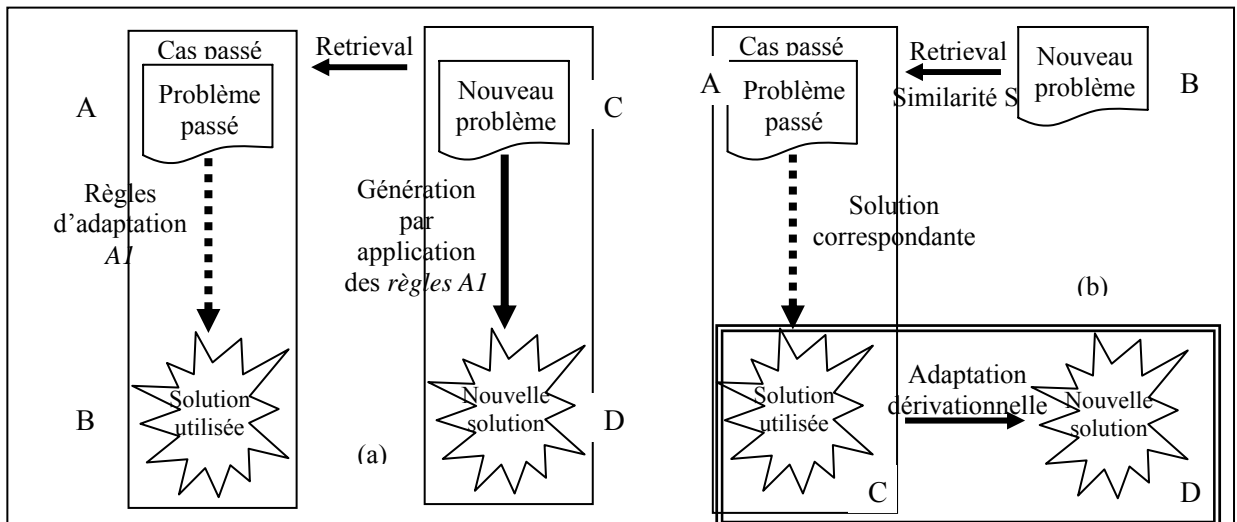


Figure 2.8 : Adaptation de type a) transformationnel, b) dérivational

a. L'analogie par transformation : tente de réutiliser avec des modifications mineures une solution précédemment trouvée pour un problème similaire. Si δ_1 représente la différence entre A et C, il s'agit de déterminer δ_2 la différence entre B et D afin d'en déduire D en propageant δ_2 dans B [29].

b. L'analogie par dérivation : il s'agit d'adapter le processus (ou dérivation) ayant abouti à la solution d'un problème précédent pour construire la solution d'un problème similaire. Si s est la similarité entre A et C et P est la méthode de construction de la solution B à partir du problème A, la résolution du problème C se fait alors par le calcul d'une méthode de dérivation P_1 à partir de P en utilisant s (s sert à reconnaître les éléments de P qui sont encore valables dans la nouvelle situation C) [26].

2.2.6 Avantages et inconvénient du CBR

Le CBR présente des avantages parmi lesquels: la réduction des efforts dans l'acquisition de la connaissance, la facilité de maintenance, l'amélioration de la performance par association des solutions existantes, l'utilisation des solutions existantes, la pro-activité (amélioration avec le temps), l'élimination des connaissances superflues, la structuration de la connaissance (cas), la flexibilité et la souplesse.

Les principaux inconvénients de CBR résident dans sa mise en œuvre. Il est en effet très difficile [37][38] de définir et représenter un cas, d'organiser la base de cas, d'indexer

les cas, de définir les « bonnes » mesures de similarités, de trouver une « bonne » stratégie d'adaptation.

2.3 La correspondance entre le KM et le CBR

Les étapes du cycle de CBR peuvent être mis en correspondance avec les activités d'un système de gestion de la connaissance. Sankar.K [37] et autres expliquent pourquoi le CBR a été employé avec succès dans beaucoup de systèmes de KM, et c'est la meilleure méthode utilisée pour la gestion des connaissances. Nous résumons dans cette section les travaux de [44][37][43] dans le domaine de l'extraction de la synergie entre le KM et le CBR. Revisitons la Figure 1.6 (chapitre I) et superposons les étapes du cycle CBR sur lui.

Dans cette figure nous pouvons voir les événements est les données qui comportent un cas. Un cas est mémorisé dans la base de cas, il subi typiquement du prétraitement (indexation). L'agent ou le raisonneur recherche le cas pour résoudre un certain problème et essaie de le réutiliser. Ceci peut avoir comme conséquence une certaine révision ou adaptation de la solution du cas et par conséquent un nouveau cas est créé. Le nouveau cas est révisé, et s'il est jugé utile le cas est maintenu. En outre, l'utilisation du cas (avec succès ou autrement) peut avoir comme conséquence un changement de la base de cas 'raffinés'. Ceci accomplit le cycle de gestion de la connaissance.

Les étapes du cycle CBR peut être mises en correspondance les activités requises par le cycle de KM représenté dans le chapitre précédent, comme suit :

1. Les processus de la récupération, réutilisation, et la révision support l'acquisition de la connaissance.
2. Les processus de la révision et raffinement soutiennent l'analyse de la connaissance.
3. La mémoire elle-même (avec la récupération et la mémorisation) support la conservation de la connaissance.
4. A la fin, la récupération, la réutilisation, et la révision support l'utilisation de la connaissance.

2.4 Re-conception d'un processus workflow par le CBR

Le processus de re-modélisation de workflow est basé sur l'amélioration incrémentale des spécifications de workflow (les modèles de workflow exécutables), il est divisé dans les quatre phases de cycle de CBR, à savoir récupération (re-mémorisation), réutilisation, révision et la mémorisation. La base de cas stocke les schémas de workflow. Le processus de re-conception de workflow, proposé par notre système est montré dans la Figure suivante.

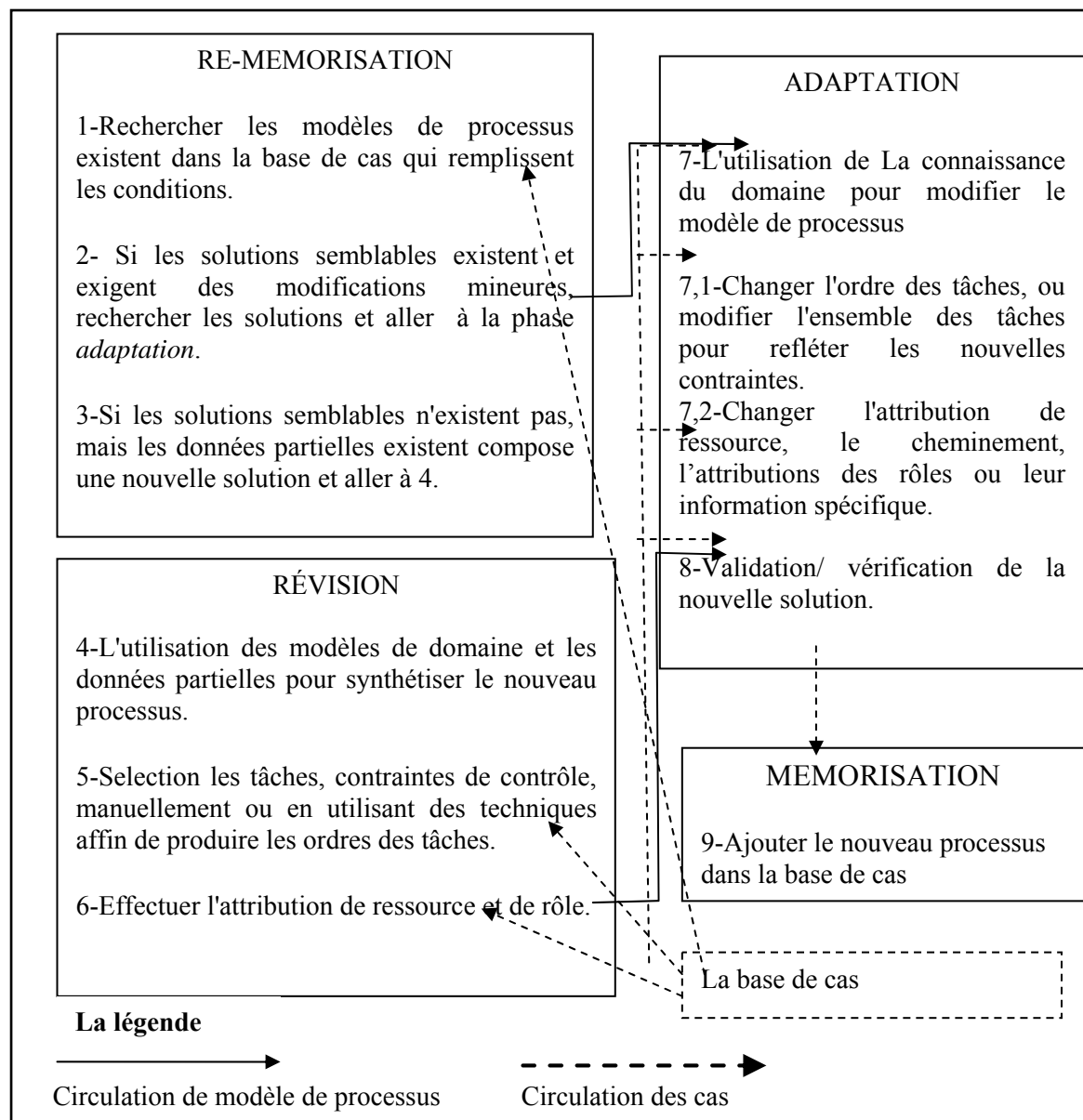


Figure 2.9 : Cycle structuré de re-modélisation du workflow

Le processus de re-conception montré dans la Figure 2.9 (l'enchaînement est

montrée par les flèches pleines, et les nombres indiquent les étapes) commence quand il y'a des nouvelles conditions de travail fournies, et qui sont utilisées pour lancer une recherche dans la base de cas (étape 1).

La phase de re-mémorisation (récupération) assure la disponibilité de la connaissance, qui peut être utilisée pour re-modéliser un workflow. Les cas recherchés sont analysés pour décider si quel de la connaissance recherchée (en particulier modèles de processus) est approprié pour des modifications ou une nouvelle solution doit être produite à partir de zéro (étapes 2 et 3). Pendant la phase de adaptation (réutilisation), un modèle de processus est modifié (choisit parmi les modèles sélectionnés) (étape 7).

La connaissance de domaine peut-être utiliser pour changer l'ordre des tâches du processus, ajouter des nouvelles tâches, les supprimer ou bien modifier les contraintes (étape 7.1). Le cheminement des tâches, détermine une nouvelle allocation de ressource (tel que les acteurs, les données utilisées, ...etc.) qui permet l'exécution parallèle (étape 7.2).

Ce nouveau modèle est porté alors à la validation et à la vérification (étape 8). La validation réussie peut déclencher le déploiement du workflow dans le système de gestion de workflow (WFMS). En plus, cette solution développée récemment est stockée dans la base de cas pendant la phase de mémorisation du cycle de re-modélisation (étape 9).

Alternativement, la composition d'une nouvelle solution peut être choisie et lancée dans l'étape d'analyse de re-mémorisation (étape 4). Deux scénarios alternatifs sont possibles dans la phase de révision de cycle de CBR. Dans le premier scénario, si un modèle de processus (proposé par l'étape de re-mémorisation) est sélectionné par l'utilisateur, c-à-d la similarité entre ce modèle et le modèle problème est supérieur à un seuil (défini par l'utilisateur), alors le modèle sélectionné peut être adapté pour développer un nouveau workflow.

Dans le deuxième scénario, le système ne trouve pas un modèle similaire au workflow problème, dans ce cas des données partielles peuvent-être retourner de la phase de re-mémorisation, donc un nouveau modèle de workflow peut être développé à partir de zéro, basé sur la connaissance de domaine de travail.

Chaque schéma de workflow (un cas) est modélisé comme une tâche composée,

cette dernière est composée par des tâches primitives. Dans le premier scénario, un nouveau workflow peut se composer des tâches composées, tandis que dans le deuxième scénario, un nouveau modèle est développé par la composition des tâches primitives.

Les étapes 4 et 5 peuvent mener à un modèle de workflow, qui est alors détaillé avec l'information des ressources (rôle, ressources utilisées, ...etc.) (étape 6). Le nouveau modèle de workflow est vérifié et puis validé (étape 8).

Il est important de noter que chacune de ces étapes de re-conception utilise la base de cas (dont l'utilisation est illustrée par les flèches à tiret). Le résultat de la re-conception est un schéma de workflow qui remplit les conditions de l'utilisateur, ce schéma peut être exécuté par un système de gestion de workflow.

L'automatisation de ses étapes (ci-dessus) exige : (1) développement d'une représentation de cas pour le workflow et l'initialisation de la base de cas (en basant sur un méta modèle de workflow) (2) développement d'un algorithme (des règles) de récupération de cas basés sur la similarité, et (3) développement d'une technique pour la composition des cas, (4) développement d'une technique d'adaptation et à la fin développement d'une technique de mémorisation.

2.5 Le développement de système CBR pour BPR

Pour entamer un projet de CBR il est important d'utiliser un processus clair de développement. Les étapes pour développer un système de BPR-CBR sont habituellement comme suit [Selma &all, 03] et sont représentés dedans la figure suivante :

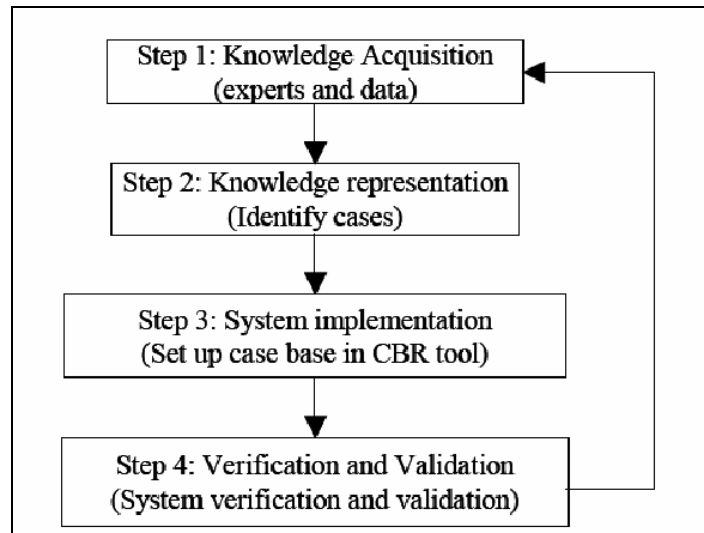


Figure 2.10 : Procédure de développement d'un system à base de cas [44].

Étape 1 : Acquisition (extraction) de connaissance de domaine : dans cette étape, tout effort est faite pour comprendre le domaine de problème et les symptômes. Les informations sur diagnostic du problème et des solutions adoptées sont également rassemblées dedans cette étape. Pour notre système, ceci moyens (a) définissant conceptuellement le processus métier qui doit être re-modélisé, il s'agit de définir un méta-modèle qui représente un workflow (b) définir un modèle à suivre pour modéliser un workflow, ce modèle sera l'instanciation du méta-modèle créée auparavant. Pour entreprendre cette étape nous avons basé notre recherche dessus les méthodologies précédentes utilisées pour le BPR.

Étape 2 : Représentation de cas : dans cette étape, le langage à employer pour la représentation des connaissances devrait être choisie. Afin de décrire le cas (décrire les concepts de workflow dans le langage choisi).

Étape 3 : Implémentation du système : elle décrit le système final comprenant la base de cas, l'organisation de la mémoire, le processus de récupération (re-mémoration), une stratégie d'adaptation et de mémorisation dans le logiciel. Le chapitre VI de la partie II du mémoire montre comment nous avons réalisé cette étape.

Étape 4 : Vérification et validation : en cette étape, la vérification et la validation devrait être faite. La validation est " la détermination de l'exactitude du programme ou logiciel final produit à partir du projet de développement en ce qui concerne les besoins et

exigences d'utilisateur ". D'où la nécessité de montrer le système aux praticiens non impliqués dans le développement du système, afin de juger les résultats de l'outil. Chaque chapitre expose les travaux et les résultats d'une étape de la conception du système CBR.

2.6 Conclusion

Au cours du présent chapitre nous avons essayé de montrer comment le KM est abordé par le mécanisme de CBR, où les différentes étapes de CBR sont facilement mises en correspondance dans un processus de KM.

Le CBR correspond étroitement aux activités du KM pour l'acquisition (révision et conservation), l'analyse (amélioration), la conservation (conservation) et l'usage de la connaissance (récupération, réutilisation et révision).

Le CBR utilisé dans ce domaine prend plusieurs formes, c.-à-d. approches conversationnelles, textuelles et structurales, puisque la connaissance varie. Quand l'utilisateur a besoin de l'expérience antérieure pour résoudre le problème, il peut employer le CBR pour obtenir la solution la plus susceptible.

Notre objectif est de concevoir une plateforme de gestion des connaissances workflow. Nous montrons dans les chapitres suivants comment nous avons pu répondre à cet objectif tout en essayant de prendre en compte les relations établies, afin de réaliser les étapes soulignées ci-dessus. Après avoir choisi la technologie et la méthode de conception du système BPR, nous entamons la conception de notre plateforme, par la réalisation de la première (Acquisition), deuxième (Représentation de cas) et la troisième (Implémentation du système) étapes dans la partie suivante.

CHAPITRE 3

ACQUISITION DE CONNAISSANCE DE DOMAINE POUR LE BPR

3.1 Introduction

Nous avons montré dans la première partie la relation entre le BPR, KM et le CBR, afin d'aboutir aux différentes étapes de conception de notre plateforme. Notre objectif dans ce chapitre est d'élaborer des modèles de processus métier, à partir des objectifs de l'entreprise. Ceci afin d'aider l'entreprise à optimiser son fonctionnement et à capitaliser une partie de son savoir-faire sous forme de modèles réutilisables et partageables par les différents acteurs de la chaîne d'entreprises. Dans cette étape, nos travaux portent essentiellement sur deux aspects :

- Le premier aspect concerne l'identification et la définition des concepts nécessaires à la modélisation d'un workflow. Ces concepts sont définis dans un méta-modèle.

- Le deuxième aspect propose une méthodologie d'utilisation des concepts définis dans l'aspect précédent pour permettre le développement des modèles de processus métier. Il s'agit de fournir une notation graphique complète permettant de représenter un processus métier, qui fournit un cadre de travail commun aux utilisateurs.

L'objectif de ce chapitre est de présenter le cadre de modélisation que nous proposons pour formaliser le workflow, afin de réaliser la première étape de conception de la plateforme BPR/CBR 'Acquisition de connaissance de domaine'. Nous proposons un méta-modèle à la base des travaux réalisés dans ce domaine, tout en prenant en considération les propriétés utilisées pour calculer la similarité entre deux workflows. La section 4 présente notre méthode de représentation graphique du workflow, la notation proposée est liée à : types de dépendance inter-tâche pour exprimer en détail la relation entre les tâches, aux transitions pour décrire les contraintes, aux tâches, aux couloirs d'activité qui sont

associés au rôles, aux ressources utilisées par les tâches,...etc.

3.2 Principe de modélisation d'un workflow

Comme nous avons défini dans le chapitre précédent, la base de cas contient des informations qui peuvent être partagées et réutilisées par plusieurs acteurs connectés à la base de cas. En effet, chaque acteur peut récupérer des informations ou des modèles dont il a besoin pour les utiliser dans son environnement, afin de permettre à l'entreprise de développer leurs propres modèles de processus d'entreprise et les placer dans la base de cas.

Les modèles développés sont évalués et utilisés par des utilisateurs tels que des concepteurs, des techniciens, du personnel administratif, etc. Nous définissons dans cette section les acteurs métier qui contribuent au développement des modèles workflow. Ces acteurs sont les groupes d'administrateurs et les groupes de modélisateurs. En effet, les responsables métier, les experts et les techniciens forment les groupes de modélisateurs. Alors que les responsables d'organisation, sont les groupes d'administrateurs.

Les acteurs métier collaborent pour modéliser un processus d'entreprise (Figure 3.1), en se basant sur une méthodologie qui permet de les orienter et aussi de développer des modèles cohérents et efficaces. La modélisation d'un processus d'entreprise est réalisée en se basant sur le savoir et le savoir-faire des experts en modélisation des processus métier. La Figure suivante représente le cas d'utilisation « Modéliser un processus d'entreprise ».

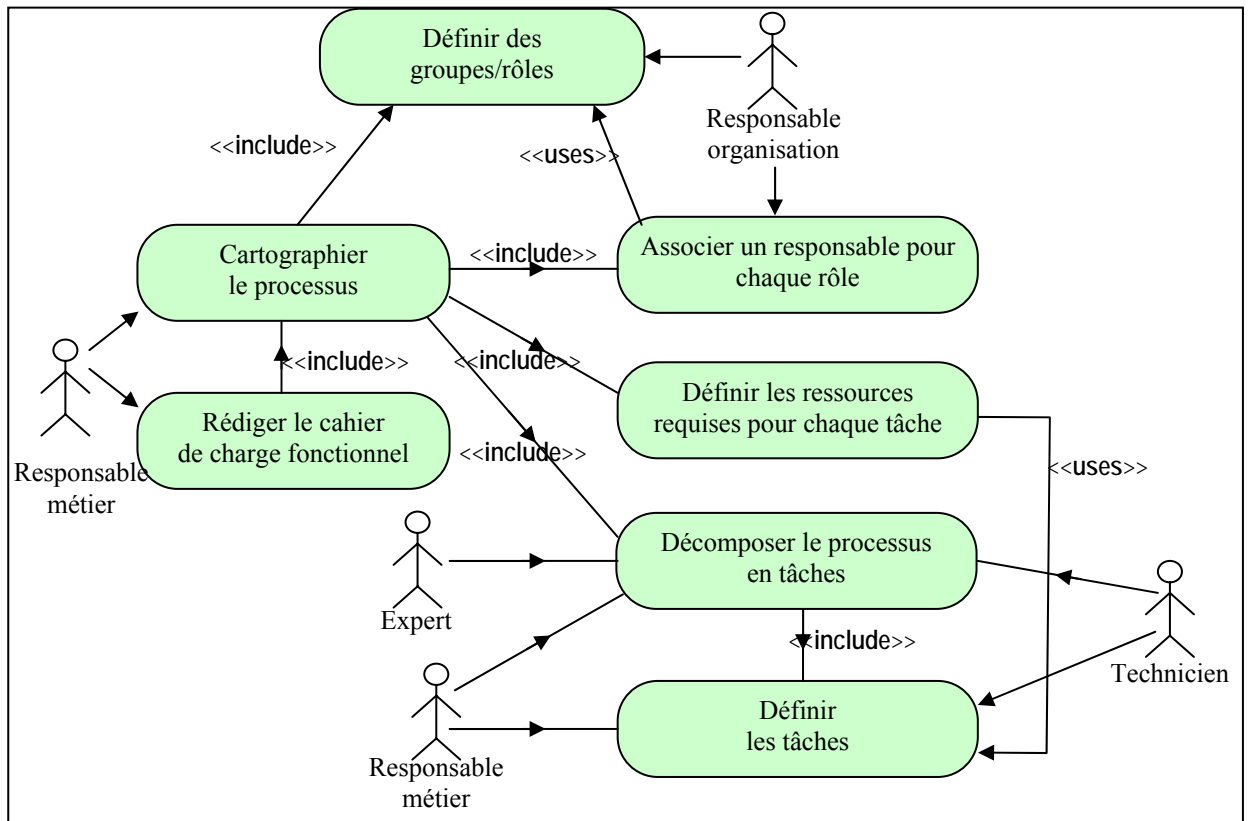


Figure 3.1 : Cas d'utilisation « Modéliser un processus d'entreprise »

En effet, pour modéliser un workflow, il faut définir le domaine correspond aux objectifs de celui-ci dans le but de diminuer la complexité de modélisation et de diviser un tel système en parties fonctionnellement autonomes. Ensuite, nous établissons la liste des processus métier existant en entreprise pour chaque domaine. Ceci permet de définir la cartographie de processus métier que nous pouvons regrouper en type de workflow (le chapitre I présente les différents types du workflow). Chaque processus est défini par ses sous-processus et activités, ses entrées/sorties et les ressources requises pour atteindre ses objectifs. Une structure d'organisation est nécessaire pour établir la liste des responsables des différents processus du système.

En se basant sur le diagramme de cas d'utilisation de la Figure 3.1, nous pouvons identifier trois parties importantes qui traitent des aspects différents du système d'entreprise, qui sont l'aspect fonction (cartographier les processus et les décomposer), l'aspect ressource (identifier les ressources requises pour les tâches) et l'aspect organisation (établir la structure d'organisation et attribuer des responsables à ses processus). Ces parties sont appelées module (paquetages dans UML). L'objectif de la

section suivante est de définir les éléments nécessaires pour chaque paquetage.

3.3 Proposition d'un méta-modèle (structure d'un workflow)

Un processus métier reçoit des objets en entrée et leur ajoute de la valeur, moyennant des ressources, tout en fournissant des objets de sortie (produits/services) remplissant les besoins et les exigences d'un client (atteindre les objectifs) internes ou externes à l'entreprise. Chaque processus est en communication avec d'autres et peut être décomposé en sous-processus. Une activité transforme des entrées en sorties en utilisant les ressources requises et disponibles pendant une durée bien définie. Un processus workflow est constitué par un ensemble d'activités, chaque activité est un ensemble de tâches qui sont partiellement ordonnées, et chaque tâche est accomplie par un rôle approprié, qui peut être joué par un ou plusieurs acteurs. Un acteur qu'il soit humain ou outil peut jouer plusieurs rôles dans un processus.

3.3.1 Les travaux associés

Devant la multiplication du nombre de méta-modèles concernant les concepts de processus workflow, et de par nos préoccupations, nous avons retenu les méta-modèle définis dans [45] [38] [20] [22]: en tant que référence de base car elle se focalise sur les processus métier et les définit d'une façon proche de nos besoins. En effet, ces méta-modèles ont élaboré des concepts qui permettent de diminuer la complexité de l'entreprise ou d'un système pour faciliter la modélisation, que nous jugeons intéressants pour regrouper et gérer les processus métier.

En se basant sur ces travaux, nous proposons un méta-modèle qui permet à l'entreprise de développer ses applications par l'instanciation du méta-modèle ou en se basant sur d'autres applications déjà élaborées; ce qui engendre la capitalisation et la réutilisation du savoir-faire de la modélisation métier sous forme de modèles.

3.3.2 La description du méta modèle proposé

Notre méta-modèle contient trois modules, chaque module contient les concepts qui lui correspond et qui sont présentés par des classes d'objet au niveau d'implémentation, ces

modules sont :

❖ Le module « Fonction » : il permet de représenter les processus d'une organisation, elle s'intéresse à la description d'un processus en terme de ses constituants, indépendamment des moyens et des ressources mis en œuvre pour les réaliser. Il offre plusieurs niveaux de visualisation en permettant de décomposer un processus en activités et l'activité en tâche.

❖ Le module « Ressource » : consiste à définir les aptitudes des ressources et les compétences des ressources humaines pour identifier celles requises pour la réalisation d'une tâche. A cet effet nous avons introduit les concepts suivants :

a. Ressource : information sur un support physique.

b. Document : une ressource sur support informatique : la structure d'un document est un ensemble de champs informationnels. Un document est une collection organisée en « sections » de champs de type simple dont les valeurs peuvent contenir des liens vers d'autres documents ou fichiers attachés.

c. Etat : le statut d'un document relativement à l'exécution d'une tâche. L'état d'un document sert à dénoter sa situation relativement au flux (quelle est la tâche faite ou à faire ?). Aux états d'un document peuvent être associés des modes d'accès aux champs du document (pas d'accès, lecture ou édition). Lorsqu'un document est en entrée d'une tâche, les acteurs du rôle pouvant réaliser cette tâche ont accès à ce document.

d. Structure de Document : un ensemble de champs et de sections décrivant le contenu informationnel du document.

e. Outil : moyen technique permettant la réalisation d'une tâche.

❖ Le module « Organisation » : Les modèles de ce niveau concernent les moyens humains, techniques et organisationnels utilisés pour accomplir les objectifs détaillés sous forme d'activités. A chaque activité une ou plusieurs tâches correspondent. Une Tâche (tactique, stratégique, opérationnelle) est réalisée de manière collaborative par un ensemble d'acteurs auxquels sont assignés des rôles. Un acteur exerce un ou plusieurs rôles; un rôle est attribué à plusieurs acteurs. Le module Organisation contient les éléments suivants :

- a. Acteur : une personne physique pouvant jouer un rôle et occupant un poste, ou un dispositif technique (outil).
- b. Equipe : ensemble homogène de rôles.
- c. Rôle : ensemble de responsabilités confiées à un ou plusieurs acteurs dans le déroulement d'une tâche.

Les éléments de ces modules définissent notre méta-modèle qui est présenté par le diagramme de classe suivant, chaque module est représenté par un paquetage, et les éléments sont des classes, les propriétés de chaque classe seront définies et discutées dans la chapitre suivant.

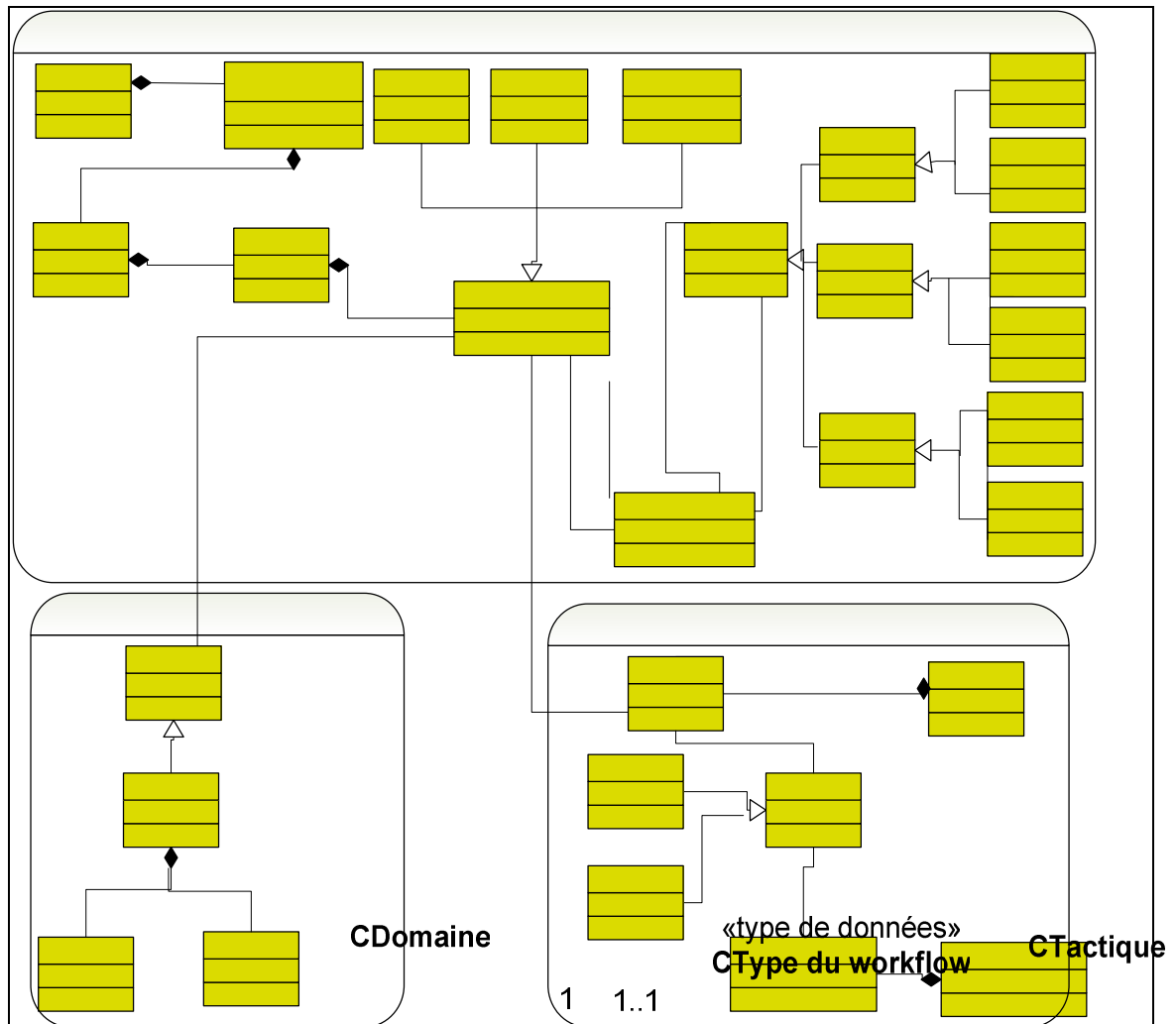


Figure 3.2 : Le méta-modèle de workflow proposé

Nous avons défini dans ce méta-modèle une classification des tâches, cette classification sera utilisée dans l'étape de sélection comme nous le présente dans le

CProcessus

CActivité

1

*

1

1..*

Fc

CStra

CTa

chapitre 5. On distingue trois grands types de tâches qui sont manipulées dans n'importe quelle entreprise :

- ❖ Une tâche stratégique : « Grande » décision, prise par l'autorité supérieure, qui oriente la vie future de l'entreprise.
- ❖ Une tâche tactique (administrative) : une tâche intermédiaire qui détermine les actions et les moyens nécessaires pour atteindre une tâche stratégique.
- ❖ Une tâche opérationnelle : mesure concrète destinée à exécuter au mieux une tâche stratégique et tactique.

Exemple : Le rôle des autorités organisatrices du transport ne se limite pas aux paiements de subsides. Elles sont également responsables de la régulation et de l'organisation du secteur. On distingue trois types de responsabilité:

La responsabilité stratégique concerne le choix d'une politique de transport : les objectifs à atteindre en termes de parts de marché et de rentabilité, la sélection et la régulation des opérateurs, la définition des moyens alloués pour le financement.

La responsabilité tactique consiste à déterminer les caractéristiques du service nécessaires en vue d'atteindre les objectifs stratégiques: choix du système d'horaire, des tarifs...

Responsabilité opérationnelle concerne la production et la vente de services de transport.

3.3.3 Les propriétés du méta-modèle proposé

▪ Notre méta modèle qui décrit le workflow dédié à la plateforme BPR\CBR, ne doit pas se contenter de fournir des concepts dédiés à la description des processus d'entreprise seulement, mais elle doit également permettre de définir des critères de mesure, qui assistent la sélection du cas plus proche dans l'étape de sélection.

▪ Nous avons défini dans ce méta modèle une classification des tâches (tactique, stratégique, opérationnelle), cette classification sera utilisée dans l'étape de sélection de la conception de la plateforme.

▪ Le méta modèle que nous proposons ici se limite à la phase de définition des processus et des critères de suivi. Nous ne considérons pas encore les aspects dynamiques des processus métier (les problèmes qui peuvent être produits au cours de l'exécution du processus), nous avons choisi de restreindre le cadre de notre travail à la modélisation du workflow.

3.4 La modélisation du workflow

Un processus d'entreprise peut être perçue comme un système complexe ce qui implique qu'il est nécessaire de le modéliser [Wil Van & All, 02] afin d'une part, de le rendre intelligible et, d'autre part, d'autoriser le raisonnement d'un acteur ayant un projet bien défini en son sein.

3.4.1 Les travaux associés

De nos jours, il existe diverses techniques et standards pour décrire un Workflow. D'un côté, il existe les approches textuelles pour lesquelles la description de la gestion de flux est définie par un langage basé sur un alphabet textuel, d'un autre côté, il existe les approches graphiques où la modélisation fait appel à un alphabet graphique.

Parmi les formalismes orientés texte, on trouve les Grammaires de Structures de Procédures Généralisées (GPSG) [19]. Ce formalisme est basé sur la définition d'une grammaire textuelle qui se compose d'un ensemble de règles, qui définissent les politiques de coordination des procédures Workflow. Et on distingue le langage XPDL (Xml Process Definition Language) [56] pour la définition du Workflow, développé par le consortium international de gestion de Workflow (WfMC).

Le point faible des formalismes et standards basés sur des grammaires textuelles est l'absence de représentation graphique, qui est primordiale pour la description du flux entre les activités du workflow, car les relations causales et temporelles existant entre les activités peuvent être détectées et exprimées par des outils visuels.

Dans les approches graphiques, on trouve les méthodes SADT, OMT, OOD, OOSE et UML, présentées dans le chapitre I, le problème de ces modèles réside dans leur ambiguïté qui rend difficile leur interprétation (l'exécution par le moteur de workflow).

Voilà notre motivation pour développer un environnement qui garde les caractéristiques des grammaires textuelles et qui pourrait offrir la capacité de représentation graphique.

Dans ce travail, nous présentons un Environnement de Description des workflow basé sur une représentation graphique, qui décrit la coordination des tâches de Workflow. L'environnement est basé sur la définition de graphes de flux des tâches sur la définition d'un ensemble de règles de réécriture de graphes, à l'aide de langage XML. Ces règles permettent de produire automatiquement des processus Workflow exécutable par un moteur, et en plus réutilisable pour la re-modélisation des workflow.

3.4.2 La modélisation du workflow pour le BPR

Le workflow ainsi décrit (le méta-modèle proposé), il reste à lui associer un modèle afin de capitaliser et de manipuler les solutions qui lui sont associées. Les processus doivent être représentés d'une manière intelligible pour les différents intervenants. Il existe une grande variété de représentations des processus, cependant cette représentation doit fournir des capacités d'analyse pour obtenir des modèles fiables et précis. Pour cela, nous considérons qu'un workflow peut être vue comme un plan des tâches, c'est à dire que chaque élément (tâche) est associé à une action à exécuter. Nous définissons un processus comme «un enchaînement partiellement ordonné d'exécution des tâches qui, à l'aide de moyens techniques et humains, transforme des flots d'entrée en flots de sortie en vue de réaliser un objectif ». Son comportement est défini par un graphe d'état, qui précise les différents états du processus (état initial, états intermédiaires et état final) et les transitions entre ces états.

3.4.3 Description de la gestion du flux des tâches

Avant de définir les fondations de notre modèle proposé, on doit spécifier quels sont les différents types de structures du flux qu'on va considérer dans notre modèle. La gestion du flux des tâches qu'on a définie est composée de trois types de structures de contrôle nommées séquence, branchement et synchronisation. Ces trois structures de contrôle sont similaires à celles qui sont utilisées par [21], [30] et sont en conformité avec l'Interface 1 du standard de Système de Gestion de WorkFlow [56] (défini dans le chapitre 1).

3.4.3.1 Structure de séquence des tâches

On parle de séquence lorsqu'au cours d'un processus, les tâches sont exécutées les une à la suite des autres, et que c'est le seul itinéraire possible. La séquence des tâches est désignée par un enchaînement entre ces tâches. Une tâche prédécesseur A doit se terminer avant le démarrage de la tâche successeur B. Autrement dit, la fin d'une tâche déclenche le commencement de la tâche successeur.

3.4.3.2 Structure de branchement des tâches

Les branchements (split) des tâches dessinent l'exécution en parallèle de plusieurs tâches. Il y a deux types de branchement : le branchement multiple et l'aiguillage. Le premier type (multiple) (AND-split) lorsqu'un itinéraire unique se sépare en deux ou plusieurs itinéraires différents dans le but de réaliser deux ou plusieurs tâches en parallèle. On parle d'aiguillage (OR-split) lorsqu'un itinéraire s'ouvre sur plusieurs itinéraires possibles, et que le cas d'exécution suit que l'un de tous les itinéraires possibles, selon les conditions de transition.

3.4.3.3 Structure de synchronisation des tâches

Les synchronisations (join) des tâches permettent de définir la convergence de plusieurs tâches vers une tâche unique. Il y a deux types de synchronisations : on a le type rendez-vous (AND-join) lorsque deux ou plusieurs activités parallèles convergent vers un itinéraire unique et que l'on assure la synchronisation des itinéraires, c'est-à-dire qu'on ne passera à la tâche suivante que lorsque toutes les tâches parallèles seront achevées. Et il y a le type jonction (OR-join) lorsque deux ou plusieurs itinéraires convergent vers une même tâche. Il ne s'agit pas de la synchronisation de plusieurs itinéraires, mais plutôt de la jonction de plusieurs itinéraires alternatifs. La figure suivante présente les différents types de dépendance inter-tâches.

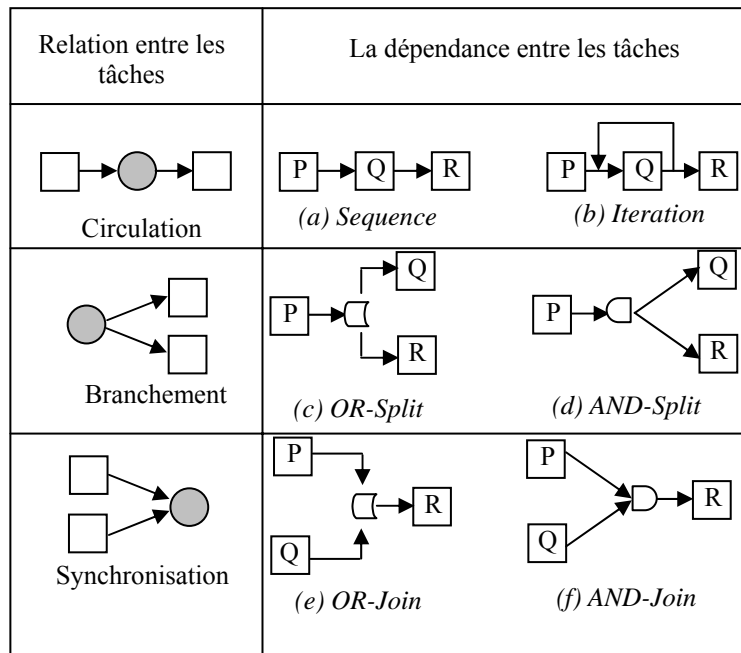


Figure 3.3: Les types de dépendance entre les tâches

3.4.4 Graphe de workflow

Les flux des tâches sont décrits avec un langage graphique qui représente l'implantation informatique du modèle workflow. Des méthodes comme Merise, et plus récemment la notation UML, ont utilisé ce type de représentation graphique basée sur des colonnes. Les diagrammes d'activité d'UML sont proches de notre modèle graphique de spécification des flux. Cette étude adopte une notation qui est inspirée du diagramme d'activité d'UML et les méthodes de modélisation présentées dans le chapitre 1.

Nous décrivons la coordination dans les processus Workflow par un graphe orienté, il est défini comme une couple (V, E) , où l'ensemble des nœuds V représente les tâches du Workflow, et E l'ensemble des arcs représente les transitions entre les tâches.

Pour distinguer les différents nœuds du modèle (nœuds connecteur, nœuds tâche, etc.), les nœuds sont constitués de deux champs : un type et une étiquette. Ces champs sont initialisés à la création du nœud. Seule l'étiquette pourra par la suite être modifiée. Dans la suite de ce mémoire, on représentera un nœud de façon graphique selon les conventions suivantes:



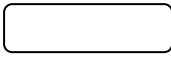
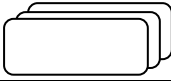

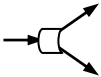
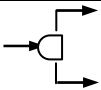

- L'étiquette est indiquée par une chaîne de caractères placée au centre du nœud,

- Le type est indiqué par la forme géométrique du nœud (deux nœuds de même forme ont le même type).

Les nœuds employés pour définir la coordination dans les processus Workflow sont:

- Nœud frontal : ce type de nœuds représente soit la tâche de départ START, soit la tâche de terminaison du processus END,
- Nœud branchement (split) : ces nœuds expriment l'exécution en parallèle de plusieurs tâches. Ces nœuds sont soit des branchements multiples soit des aiguillages.
- Nœud synchronisation (join) : ces nœuds expriment la jonction de plusieurs tâches sur une tâche unique. Ces nœuds représentent soit des rendez-vous soit des jonctions.
- Nœud tâche: ce sont les nœuds de base permettant la représentation des processus. Un nœud tâche représente l'abstraction d'une étape d'un processus lors de laquelle une action élémentaire est exécutée.

Le Tableau suivant montre la notation utilisée pour représenter les différents types de nœuds définis. Les types de base sont les nœuds qui représentent les tâches et ceux qui représentent le branchement et la synchronisation des tâches.

Classe	Type	Nom
Nœud frontal		Point de départ (START)
		Point d'arrêt (END)
Nœud tâche		Tâche Élémentaire
		Tâche Composée
Enchaînement		Transition
Nœud de branchement (split)		OR-Split
		AND-Split
Nœud synchronisation		OR-Join



(join)		AND-Join
Un rôle		Couloir d'activité d'un rôle

Tableau 3.1: La notation proposée

3.4.5 Propriétés du modèle proposé.

Notre modèle doit respecter les propriétés suivantes:

Propriété 1 : Un nœud tâche peut avoir plus d'un nœud prédécesseur et d'un nœud successeur. Car notre définition des nœuds de branchement et de synchronisation prend en charge le contrôle de flux.

Propriété 2 : notre modèle est toujours connexe. Le début et la fin d'un processus sont délimités par les nœuds START et END.

3.5 Etude de cas

Un processus métier se décompose en activité, les activités en tâches élémentaires et affecter chaque tâche à un rôle, selon les règles de gestion en vigueur. L'ensemble des tâches doit être exécuté selon un ordre défini, par les bons rôles au bon moment, afin de réaliser correctement le processus. On peut considérer le workflow comme un mode de division du travail entre rôles, qui facilite la coordination des actions entre elles.

Nous illustrons notre approche de modélisation et de représentation de cas par un exemple, qui est « Demande de remboursement des frais », le type de ce workflow est administratif. Cette description du processus est réalisée à l'aide de notre modèleur de processus graphique (les figures suivantes ont été réalisées à l'aide de cet outil).

Trois rôles sont nécessaires à la réalisation de ce workflow : demandeur, contrôleur et approuvateur. L'activité de chaque rôle est constituée de l'ensemble des tâches qui se trouvent placées dessous. Le contrôleur a pour responsabilité dans ce processus de contrôler la note récapitulante les frais engagés par un demandeur. Ce demandeur a rempli cette note à partir des accusés de réception, tickets et autres factures liés à une action

particulière.

Si la note de frais est incorrecte, le contrôleur peut la retourner en y joignant une explication au demandeur. Si elle est correcte, un approbateur doit la valider pour que le remboursement soit réalisé.

La Figure 3.4 décrit le déroulement du processus « Demande de remboursement des frais » engagés par des collaborateurs dans le cadre de leur travail.

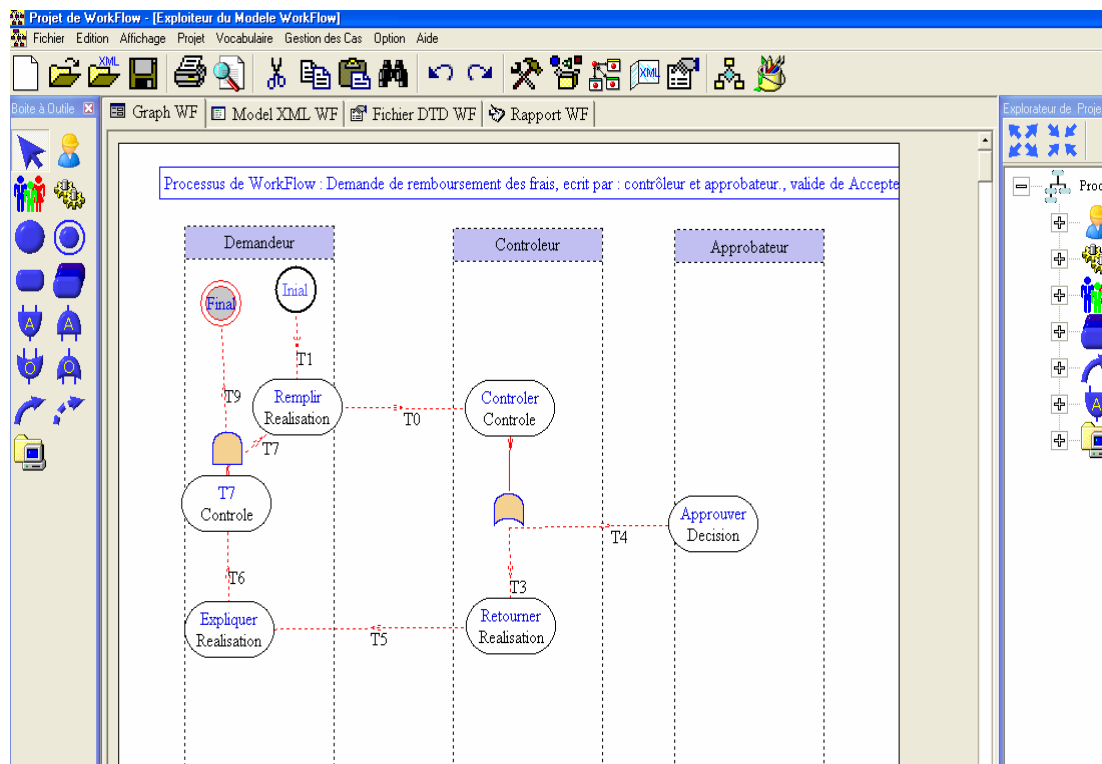


Figure 3.4: Modèle de workflow pour «Demande de remboursement des frais».

3.6 Conclusion

Dans ce chapitre nous avons présenté un embryon de méta-modèle dédié à la définition du suivi des processus. Les différents éléments qu'il contient ont été extraits à partir des travaux réalisés dans ce domaine. Les notions de processus, activités, tâches, modèle graphique du processus sont fréquemment utilisés par les experts de remodelisation. Les concepts ainsi définis peuvent ensuite être échangés et discutés entre experts afin d'arriver à un méta-modèle consensuel.

Nous avons proposé ainsi un modèle de représentation de workflow sous forme de graphe. Tout workflow est représenté sous forme d'un graphe orienté, ayant un point de départ et un point d'arrivée uniques. Les nœuds de ce graphe sont :

- Soit des tâches (composées ou élémentaires),
- Soit des connecteurs logiques : ils consistent à introduire des points de choix, de parallélisme et de synchronisation dans la procédure.

Le modèle proposé n'est pas exécutable, il faut fournir un mapping complet du modèle vers un langage d'exécution. Sa implique que notre système rendre le processus modélisé par les utilisateurs en un processus exécutable. Par la possibilité de générer automatiquement sa spécification correspond. Dans le chapitre suivant nous présentons la structure de la spécification de workflow, qui sera utilisée pour exécuter le workflow, et pour alimenter la base de cas afin de l'utiliser dans la re-modélisation.

CHAPITRE 4

REPRESENTATION DE CAS

4.1 Introduction

Le présent chapitre décrit la représentation d'un cas dans la plateforme BPR/CBR, dans cette étape les contributions les plus significatives peuvent être apportées. La littérature CBR actuelle propose des représentations de cas comportant des mots-clé, des termes complexes, des catégories et des paires attributs-valeurs. Toutefois, pour une application particulière, il n'existe pas une méthodologie pour déterminer le niveau de représentation adéquat ni de critères pour faire ce choix. Nous proposons dans ce chapitre notre approche pour représenter un cas (un schéma de workflow).

4.2 Les objectifs de la représentation de cas

La représentation du workflow sous forme de cas textuel peut servir à :

❖ L'exécution du workflow : La technologie workflow s'appuie beaucoup, sur la technique de modélisation puisqu'un workflow se doit implanter un modèle en répondant, en particulier, aux questions suivantes :

- Quelles sont les tâches à réaliser ?
- Quelles sont les compétences nécessaires pour réaliser ces activités ?
- Quand faut-il réaliser ces tâches?
- Quels sont les outils et les informations nécessaires à la réalisation de ces tâches? ...etc

Une fois le processus métier d'une entreprise modélisé, l'étape suivante consiste à transcrire ce modèle en une forme (informatique) exécutable qui permet de suivre et de contrôler l'exécution des processus. La description du cas générée à partir du modèle graphique sera utilisée pour exécuter le processus métier par un moteur de workflow. ٥

❖ Structuration d'un cas : la plateforme BPR/CBR utilise des spécifications de workflow comme des cas. Pour la représentation d'un cas, nous appuyons sur les langages existants pour garantir la portabilité et la compatibilité avec un maximum d'outils existants et à venir, ainsi que le résultat des recherches menées par la WFMC ; c'est pourquoi nous pensons à utiliser le standard XML, il est à l'heure actuelle le format idéal pour représenter le contenu des workflows [46]. Compte tenu des nombreuses normes qui sont définies autour d'XML ainsi que le nombre de plus en plus important d'outils utilisant XML.

4.3 Le langage de représentation d'un cas

Le plus souvent les processus sont considérés comme un ordre des activités, tâches, ou des fonctions qui représentent un objectif. Les activités peuvent être exécutées par des personnes ou dispositifs techniques (computer). Le format standard à l'heure actuelle pour la structuration des données est le langage XML [15]. Son importance croissante dans le domaine de l'informatique nous incite fortement à l'utiliser pour manipuler les données. L'importance de représenter les données par XML est discutée dans [46]. Nous résumons ces avantages en :

- a. Compatibilité et portabilité. XML est une recommandation du consortium W3C. Les outils compatibles avec cette recommandation peuvent donc lire tous les documents XML valides.
- b. Les méthodes d'accès aux fichiers XML sont normalisées grâce à la définition des interfaces (SAX et DOM) par le groupe W3C. En plus, des bibliothèques de fonctions implémentant ces interfaces sont disponibles sur Internet (la plupart gratuites), ce qui facilite le traitement de ces fichiers [1].
- c. XML est un langage flexible (on peut définir les balises qu'on veut),
- d. C'est un langage simple et lisible (c'est un langage auto descriptif, au format texte et présentant quelques règles de syntaxe) est un format texte structuré, hiérarchisé, clair et lisible,
- e. Il est adapté pour les échanges des données entre applications : dans la plateforme nous utilisons les schémas XML pour le transfert des données entre les sous plateformes, les couches, et les modules.

- f. XML sépare le contenu de la forme, ce qui autorise représentations multiples adaptée à l'utilisateur de même contenu.
- g. XML est un format extensible, il est possible d'ajouter des nouveaux éléments sans modifier les programmes existants traitant les schémas XML, d'où une évolution rapide des applications.
- h. XML est un langage pour définir des langages, proposant DTD et schéma pour spécifier et vérifier les formes des documents.

En générale, XML définit un ensemble de règles de formatage et de structure pour le document de façon à ce qu'il soit facile à générer et à interpréter par différents systèmes (et technologies), qu'il ne soit pas ambigu et enfin qu'il soit extensible si l'on souhaite lui ajouter de l'information. Un important avantage associé à XML pour la représentation des workflow est d'être utilisable sur le web. Il nous semble intéressant d'utiliser XML parce qu'il existe déjà des langages de workflow qui utilisent XML, comme les langages : XPDL (XML Process Definition Language), WFXML (WorkFlow XML) ; qui sont des langages standards pour représenter le workflow, et ils ont en conformité avec l'Interface 1 de standard de Système de Gestion de WorkFlow, sa nous permet d'utiliser des outils (moteur) existent.

En contrepartie, les inconvénients de XML qui peuvent être cités sont limités [Gardarin, 02], on peut être circonscrits :

- ❖ XML est un langage bavard augmente la taille des documents.
- ❖ XML est consommateur en ressource unité centrale et la mémoire, l'analyse de texte nécessite la puissance et la place mémoire.

4.4 Le passage du méta-modèle au cas workflow

Nous avons d'écrire dans le chapitre précédent un méta-modèle abstrait qui nous permet de représenter la structure d'un workflow. Il s'agit maintenant de lui faire correspondre une représentation concrète en XML. Dans ce but, nous présentons le modèle XML en document de définition de type (DTD), c'est la façon standard de définir des modèles XML. Le langage utilisé dans les DTD est très simple, et bien adapté à sa

prise en main par un utilisateur. Un DTD est une grammaire qui définit la structure d'un document XML [2]. Elle permet de valider que chaque composant d'un document XML est à la bonne place.

La façon de faire la correspondance entre les langage UML et XML est d'associer des entités XML aux classes UML, et les éléments (XML) aux attributs (UML), pour faire le passage de UML à XML nous avons basé sur la référence de [9]. Les règles de transformation d'un diagramme de classe en schéma XML sont résumées comme suit :

- Une classe UML est représentée comme un élément XML, contenant toutes les classes immédiatement dépendantes.
- Un attribut UML est représenté comme un élément XML englobant une valeur.

Si les avantages liés à XML sont nombreux et incontestables, mais il faut créer des documents XML valides et bien formés. Cela n'est pas nécessairement aisé. Pour y arriver, il faut utiliser un bon éditeur, capable d'éviter les erreurs, dans ce but nous avons utilisé l'éditeur oXygen v3.1, pour écrire un DTD valide.

4.5 Expression de notre représentation du cas en XML

La base de cas est constituée par des schémas représentant des modèles de workflow autant que cas. Ce dernier inclue l'ordre global des activités qui remplissent une condition générique de travail. Quand un schéma de workflow est modifié ou mis à jour, il est stocké comme un nouveau cas dans la base de cas. Les cas sont représentés avec des éléments syntactiques et sémantiques bien définis. Nous présentons ces éléments dans la section suivante.

4.5.1 La description d'un workflow

Les cas sont représentés par le langage XML avec des étiquettes appropriées pour les différents éléments structuraux. Un cas est manipulé aussi, comme une représentation XML d'un processus métier exécutable, qui peut être déployée sur le moteur de workflow. L'élément atomique de notre workflow processus est une « tâche », qui peut être l'envoi d'un message, la réception d'un message, l'appel d'une opération (attente d'une réponse),

ou une transformation de données. Les détails des éléments structuraux d'un cas, les étiquettes correspondantes et les représentations XML sont détaillés ci-dessus.

Les cas incluent un modèle graphique de processus, dénoté par l'étiquette de WSPGModel. Ce modèle de processus graphique sera défini ultérieurement.

Les éléments WSPreconds et WSPostconds pour le cas définissent les conditions préalables et les postes conditions qui s'assurent qu'un cas a été traité.

WSInputs et WSOupputs définissent les paramètres d'entrée et de sortie (exemple : budget) nécessaires pour ordonner le cas. Le cas définit également les ressources qui sont peut-être utilisées pour exécuter le cas (défini par ResList). La liste de tâche (TaskList) identifie les tâches qui définissent le schéma de workflow.

Les contraintes temporelles propres à un workflow portent sur sa date de début et de fin, on peut définir pour chaque date des contraintes de type 'au plus tôt le' et 'au plus tard le'. Chaque workflow disposera donc d'un couple (WSValidFrom, WSValidTo) où WSValidFrom et WSValidTo sont les intervalles de temps acceptés pour les instants START et finish.

Nous avons également raffiné notre méta modèle, par l'ajout des propriétés comme : WSAuthor, WSVersion, qui nous permet de sauvegarder la trace de l'évolution de processus, et qui va servir dans la re-conception du processus.

La première colonne du tableau suivant identifie l'élément et son étiquette dans XML, les autres colonnes décrivent la sémantique de l'élément et illustrent sa représentation informatique.

Élément, étiquette de XML	Description	Représentation
IDentifiant de Cas WSID	Chaque schéma est assigné à un Identifiant de cas unique	Un symbole (WSI2)
La version	La version du processus	Un numéro (02)

WSVersion		
Nom WSName	Le nom de cas	Texte
Description WSDesc	Une annotation textuelle.	Texte
L'auteur WSAuthor	Le responsable qui modélise le processus	Texte
Type WSType	Le type de workflow(Adhoc; Production; Collaboration; Administration)	Type énumération
Liste des Tâches TaskList	Liste des identifiants des tâches primitives et composée	Type List
Graphe du processus WSPGModel	La transformation du modèle graphique de workflow sous forme textuel	Un graphe dirigé attribué aux tâches, aux connecteurs, et aux transitions.
Inputs WSInputs	Liste d'arguments existents pour initialiser le workflow	Chaque argument est une donnée attribuée à une entité exemple : client
Output WSOutputs	Liste d'arguments produits par le workflow	Chaque argument est une donnée attribuée à une entité exemple : transaction terminée
Precondition WSPreconds	Conditions qui doivent exister dans l'état d'organisation courant pour initialiser le processus courant.	Liste de texte
Postcondition WSPostconds	Conditions qui doivent vérifier par l'exécution du workflow	Liste de texte
Types de ressource ResList	Liste des ressources utilisées par le workflow	Liste de texte
Valid From WSValidFrom	La date d'activation du processus workflow	Type date
Valid To WSValidTo	La date maximale de la validité du processus.	Type date

Tableau 4.1 : Les éléments d'un cas

4.5.2 La description d'une tâche

Une tâche est décrite par un nom, les ressources impliquées et les intervenants Humains (rôle). Dans le méta-modèle nous avons défini trois types des tâches : stratégique, tactique, opérationnelle, chaque type de tâche contient les informations suivantes :

Élément, étiquette de XML	Description	Représentation
ID Tâche, TID	Identifiant unique de tâche à travers toute descriptions de domaine.	Un symbole (TI-21)
Description TDesc	Description de la tâche	Texte
Paramètres de Tâche TRessource	Liste des ressources utilisées par la Tâche	Liste des paramètres comme (nom, adresse, document)
Pre-condition de la Tâche TPreconds	Conditions qui doivent exister dans l'état courant pour initialiser la Tâche.	Liste de texte
Post-condition de la Tâche TPostconds	Conditions qui doivent exister après l'état final de la Tâche.	Liste de texte
Type TaskType	Le type de la tâche: automatique ou manuelle.	Type énuméré
Rôle TRole	Le rôle responsable à la réalisation de cette tâche	Texte

Tableau 4.2 : Les éléments d'une tâche

Une transition est caractérisée par :

Élément, étiquette de XML	Description	Représentation
ID Transition, TRID	Chaque transition à un identifiant	Un symbole (TR-1)
Début de La transition TRFrom	La tâche qui déclenche la transition	Texte
Fin de La transition TRTo	La tâche déclenchée par la transition	Texte

Tableau 4.3: Les éléments d'une transition

4.5.3 La dépendance entre les tâches

Dans la modélisation du workflow il est important de montrer le flux des tâches. Comme nous avons défini dans le chapitre précédent, il y'a six types de dépendances entre les tâches sont possibles : Séquence, Itération, And-split, And-join, Or-split et Or-join.

Afin d'enregistrer la structure graphique du processus nous avons ajouté les propriétés OrSplit, AndSplit, OrJoin, et AndJoin. Ces dernières sont représentées par un ensemble de transitions, Les éléments structuraux de la structure graphique sont définis dans le tableau suivant :

Élément, étiquette de XML	Description	Représentation
AndJoin	Est un rendez-vous. Les prédécesseurs directs de tous les flux d'entrée doivent être terminés pour que le système passe à l'étape suivante.	List des transitions
AndSplit	Indique que le workflow peut continuer si tous les flux d'entrée sont traversés.	List des transitions
OrJoin	Est une disjonction parallèle : toutes les étapes de travail sont instanciées en parallèle.	List des transitions
OrSplit	Est une disjonction possible ; un prédicat est associé à chacune des branches possibles.	List des transitions

Tableau 4.4 : Les types de dépendance entre les tâches

4.5.4 La description du rôle et de l'acteur

L'exécution de chaque tâche est réalisée par un rôle. Dans cette partie, nous décrivons les moyens humains, techniques et organisationnels utilisés pour accomplir les objectifs attendus du processus, ils ont réalisés de manière collaborative par un ensemble

d'acteurs auxquels sont assignés des rôles. Un acteur exerce plusieurs rôles, et un rôle est attribué à plusieurs acteurs. Les éléments structuraux d'un acteur et un rôle sont énumérés dans les tableaux suivants.

Élément, étiquette de XML	Description	Représentation
ID rôle, RID	Chaque rôle à un identifiant.	Un symbole (R-4)
Chef rôle RChef	Le chef du rôle.	Texte
List des acteurs RLActor	List des acteurs qui réalise ce rôle	List des acteurs

Tableau 4.5 : Les éléments d'un rôle

Élément, étiquette de XML	Description	Représentation
ID acteur, AID	Chaque acteur à un identifiant	Un symbole (A-45)
Type AType	Le type de l'acteur : humain, système	Type énuméré

Tableau 4.6: Les éléments d'un acteur

4.5.5 La description du modèle graphique de workflow (WFGModel)

Le WSPGModel inclut les étiquettes des tâches séquentielles, et une liste des tâches simultanées (split, join). Le codage de XML montre une représentation partielle pour le workflow, pour compléter cette représenter nous avons ajouté les propriétés AndJoin, AndSplit, TOrJoin, et OrSplit dans la description d'une tâche. La figure suivante illustre un exemple d'utilisation de la structure proposée (WSPGModel).

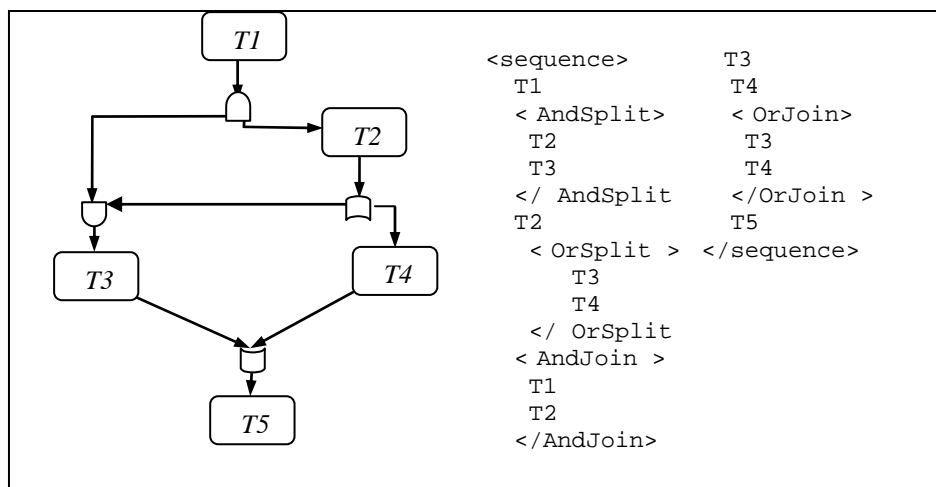


Figure 4.1: Exemple d'utilisation de la définition du modèle graphique de WF

Pour montrer la validité de notre structure graphique d'un modèle workflow, nous avons implémenté un prototype de notre plateforme, qui inclus une sous plateforme correspond à «Edition». Cette sous plateforme est très importante car elle permet de prendre en charge les processus des entreprises. Ces processus permettent de décrire les tâches, les organisations et les personnes impliquées, les informations nécessaires, et les applications requises pour les différentes étapes de ces processus.

Cette couche implémente notre approche de la modélisation d'un workflow, et vérifie la correspondance entre le fichier XML d'un workflow et le modèle correspond. En effet, Edition est une sous plateforme logicielle qui dispose d'un parseur XML, c'est à dire un interpréteur de code XML vérifiant la grammaire spécifiée par une DTD (Document Type Definition). Il est donc envisageable de l'adapter à l'édition de workflows, il faudra alors commencer par spécifier une grammaire de représentation d'un workflow par une DTD en XML, La figure suivante illustre notre définition d'une grammaire de description d'un workflow sous forme de DTD, elle est basée sur les éléments du méta modèle défini dans le chapitre précédent.


```

<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE WSCase [
<!ENTITY %String "#PCDATA"> <!--Définition d'un type-->
<!ELEMENT %TActivite"
    <!ELEMENT AcId(#PCDATA) > <!-- L'identifiant de l'activite-->
    <!ELEMENT AcNom(#PCDATA) > <!--Le nom de l'activite-->
">
<!ELEMENT %TDomaine"
    <!ELEMENT DID(#PCDATA) > <!-- L'identifiant du domaine-->
    <!ELEMENT DNom(#PCDATA) > <!--Le nom du domaine -->
">
<!ELEMENT %TFonction"
    <!ELEMENT FID(#PCDATA) > <!-- L'identifiant de la fonction-->
    <!ELEMENT FNom(#PCDATA) > <!-- Le nom de la fonction-->
">
<!ELEMENT %TTask"
    <!ELEMENT TTask(Technique|strategique|operationnelle) > <!--La typologie des tâches-->
">
<!ENTITY %Acteur"
    <!ELEMENT AId(#PCDATA) > <!--L'identifiant de l'acteur-->
    <!ELEMENT AType(systeme| humain) > <!-- commentaire-->
">
<!ELEMENT %Role"
    <!ELEMENT RId(#PCDATA) > <!-- L'identifiant du rôle-->
    <!ELEMENT RChef(#PCDATA) > <!--Le chef du rôle qui est un acteur-->
">
<!ENTITY %Condition"
    <!ELEMENT PreCond(#PCDATA) >
    <!ELEMENT PostCond(#PCDATA) >
"> <!--Le types d'une condition -->
<!ENTITY % WSCase "
<!ELEMENT WSCase (WSID, WSName, WSDomaine, WSFonction, WSType, WSDes, WSVersion,
WSAuthor, TaskList, WSInput, WSOOutput, ResList, WSValidFrom, WSValidTo, Cond,
WSRating, WSPGModel) > <!--Les elements d'un workflow-->
<!ATTLIST WSID PCDATA #REQUIRED>> <!--L'identifiant d'un cas -->
<!ELEMENT WSName(#PCDATA) > <!--Nom du workflow -->
<!ELEMENT WSDomaine %TActivite > <!--Le domaine du workflow-->
<!ELEMENT WSFonction %TFonction > <!--La fonction -->
<!ATTLIST WSType(Adhoc | Production | Collaboration| Administration) >
<!ELEMENT WSDes(#PCDATA) >
<!ELEMENT WSVersion(#PCDATA) > <!--numéro de version de ce workflow -->
<!ELEMENT WSAuthor(#PCDATA) > <!--le nom de l'auteur -->
<!ELEMENT TaskList(%Task+) > <!--La liste des tâches-->
<!ELEMENT WSInput(#PCDATA) > <!--les données utilisées-->
<!ELEMENT WSOOutput(#PCDATA) > <!-- les données sortantes -->
<!ELEMENT ResList (#PCDATA) > <!--Liste de type des ressources-->
<!ELEMENT Cond %Condition > <!--Les condition de réalisation-->
<!ELEMENT WSRating(#PCDATA) > <!--la priorité -->
<!ELEMENT WSPGModel(%WSPGModel) > <!--Le schema graphique du workflow-->
<!ELEMENT WSValidFrom (#PCDATA) > <!--La date de début de validation de workflow-->
<!ELEMENT WSValidTo(#PCDATA) > <!-- La date de fin de validation de workflow -->
<! ENTITY %Task"
<!ELEMENT Task(Task ID, Task Name ,TaskType, TaskDes , TParam ,Cond, TType,Agent ,
Procedure , TRole, TJoin, TFork, Imput, Output, DataItem) > <!--Les éléments d'une tâche-->
<!ATTLIST Task ID PCDATA #REQUIRED> <!--L'identifiant de la tâche -->
<!ELEMENT Task Name(#PCDATA) > <!--Le nom de la tâche -->
<!ELEMENT TaskDes(#PCDATA) >
<!ELEMENT TParam(#PCDATA+) >
<!ELEMENT Cond %Condition > <!-- Les condition de réalisation -->
<!ATTLIST TaskType (Manual|Automatic) > <!--Le type de la tâche -->
<!ELEMENT Imput(DataItem+) > <!-- les données utilisées -->
<!ELEMENT Output(DataItem+) > <!--Les résultats-->
<!ELEMENT DataItem (#PCDATA) >
<!ELEMENT TRole(%role) > <!--Le rôle responsable de la réalisation-->
<!ELEMENT TJoin(TranList, Type) > <!--Les transitions sortantes et la relation entre eux
-->
<!ELEMENT TranList (%Transition+) > <!-- liste des transitions -->
<!ELEMENT Type (OR|AND) > <!--La relation entre les transitions-->
<!ELEMENT TFork(TranList, Type) > <!-- Les transitions entrantes et la relation entre
eux -->

```

```

<!ENTITY %Transition "
    <!ELEMENT TRID(#PCDATA)> <!--L'identifiant -->
    <!ELEMENT TRFrom(#PCDATA+)> <!--La tâche de début-->
    <!ELEMENT TRTo(#PCDATA+)> <!-- La tâche de fin -->
"> <!--Fin de la définition d'une transition-->
" > <!--Fin de la définition d'une tâche -->
<!ENTITY %WSPGModel"
<!ELEMENT SEQUENCE (#PCDATA, Fork, Join)+> <!-- commentaire-->
<!ELEMENT Fork (#PCDATA +) >
<!ELEMENT Task (#PCDATA) >
<!ELEMENT Join (#PCDATA +) >
"> <!-- Fin de la définition de la représentation graphique du workflow -->
"> <!--La fin de définition d'un schéma de workflow-->
]> <!--La fin de la grammaire -->

```

Figure 4.2 : DTD de schéma de workflow

4.6 Etude de cas

Dans cette section nous présentons le déroulement d'une activité d'achat par la technologie du commerce électronique, afin d'aboutir à un modèle workflow, ce dernier contient les tâches :

Consulter : initialement le client consulte le catalogue des produits proposés par l'entreprise ainsi que les conditions et les contraintes de vente maintenues par le site.

L'inscription : Afin de pouvoir participer au cycle d'achat, le client doit être abonné dans ce site autant que client. Il doit introduire ses coordonnées (nom, adresse, numéro de compte bancaire ...) auprès d'un formulaire qui lui sera fourni par le site. Tout client doit avoir un numéro qui le différencie des autres.

La demande d'inscription sera vérifiée et examinée par une application appropriée, pour cela la demande d'inscription sera soit acceptée ou bien rejetée.

La commande : une fois le client est inscrit, il peut effectuer une opération d'achat en remplissant une commande indiquant la liste des produits avec les éventuelles quantités. Comme le cas d'une demande d'inscription, la commande doit être vérifiée, afin de pouvoir décider son acceptation ou son rejet.

Traitement de la commande : La commande du client sera prise en considération par un agent commercial qui chargé d'effectuer le traitement de la commande

Le paiement : après avoir accepté la commande, une facturation est enchaînée par la suite, elle consiste à remplir une facture, indiquant le montant globale des produits. La facture sera envoyée au concernée pour finaliser l'opération et effectuer la livraison des achats.

La livraison : le service des ventes et facturation est chargé de la préparation du bon de livraison en deux exemplaires, un sera envoyé au client et l'autre sera envoyé au responsable du service livraison. Le client et le responsable de service de livraison peuvent maintenant procéder à la livraison.

L'activité d'achat électronique, est constitué de cinq tâches élémentaires : consulter, inscrire, commander, trait_cde (traitement de la commande) et livrer; et une tâche composée, facturer. Pour chaque tâche élémentaire on associe le rôle chargé de sa réalisation et l'ensemble de données manipulées par cette tâche. La tâche complexe facturer doit être décomposée en tâches élémentaires.

Le modèle suivant est une représentation graphique du workflow 'achat électronique', réalisé par notre système.

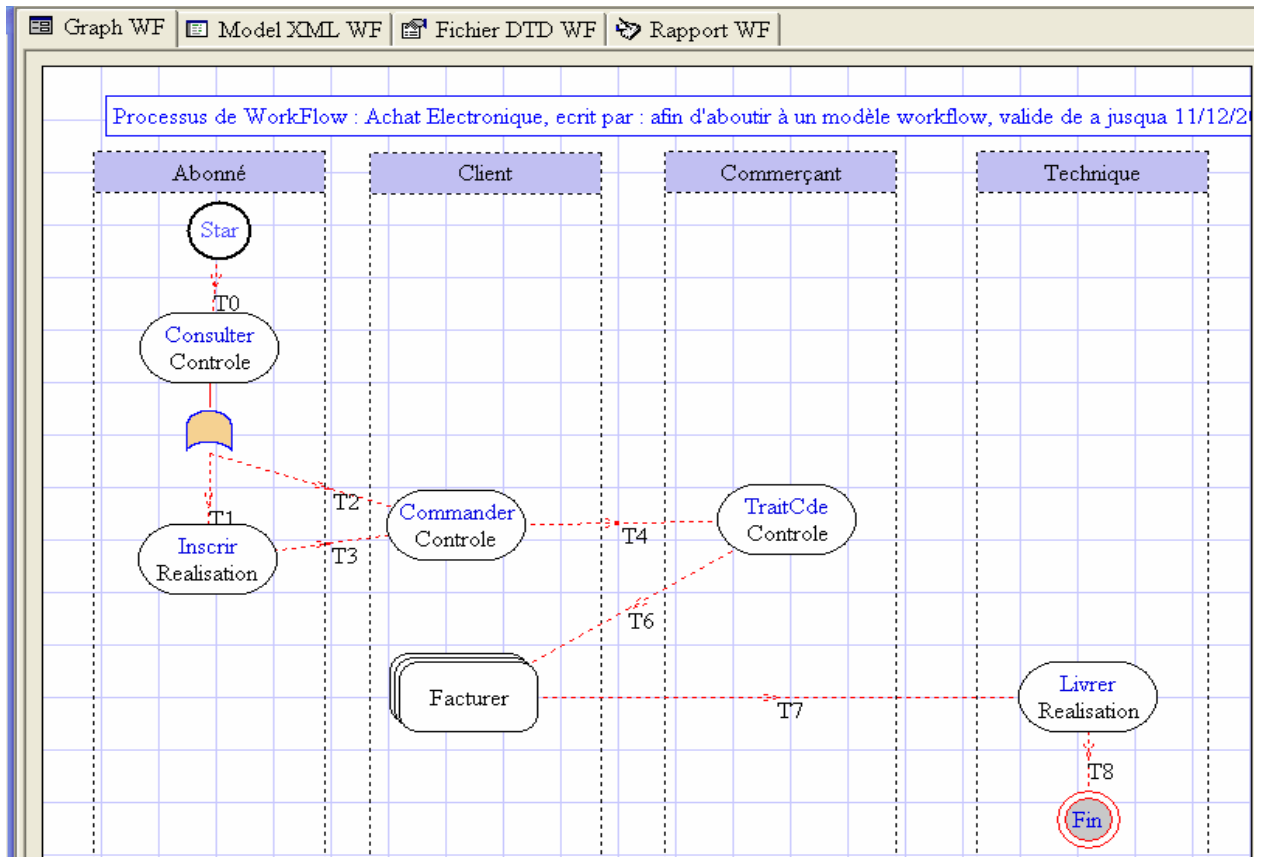


Figure4.3 : Modèle de processus achat électronique

Ce schéma de workflow est stocké en tant que cas dans la base de cas. La figure suivante montre une partie de la spécification du workflow de 'L'achat électronique' :

<pre>?xml version="1.0" encoding="UTF-8" ?> _ <workflow> _ <ProprieteProcessus id="11"> NomPorc>Achat Electronique</NomPorc> <VersionProc>1</VersionProc> <AuteurProc>afin d'aboutir à un modèle workflow</AuteurProc> <DescriptionProc>Ce processus le déroulement d'une activité d'achat par la technologie du commerce électronique</DescriptionProc> <TypeProc>Adhoc</TypeProc> <RangProc>2</RangProc> <DonneEntreeProc>auteur1</DonneEntreeProc> <DonneeSortieProc>a</DonneeSortieProc> <PreconditionProc>a</PreconditionProc> <PosteconditionProc>a</PosteconditionProc> <DateDebutProc>a</DateDebutProc> <DateFinProc>11/12/2003</DateFinProc> </ProprieteProcessus> _ <ListeActeurs nbacteurs="4"> _ <Acteur id="0"> <nomacteur>Djaidida</nomacteur> <typeacteur>Humain</typeacteur> </Acteur> _ <Acteur id="1"> <nomacteur>Mohamed</nomacteur> <typeacteur>Humain</typeacteur> </Acteur> _ <Acteur id="2"> <nomacteur>Aya</nomacteur> <typeacteur>Humain</typeacteur> </Acteur> _ <Acteur id="3"> <nomacteur>Omar</nomacteur> <typeacteur>Humain</typeacteur> </Acteur> </ListeActeurs> _ <ListeActivites nbactivites="1"> _ <Activite id="0"> <nomactivite>Activite0</nomactivite> </Activite> </ListeActivites> _ <ListeRoles nbroles="4"> _ <Role id="0" X="107" Y="74" W="150" H="30"> <nomrole>Abonné</nomrole> <chefrole>Djaidida</chefrole> _ <listeacteur nbacteur="2"> <acteur>0</acteur> <acteur>2</acteur> </listeacteur> </Role> _ <Role id="1" X="285" Y="75" W="150" H="30"> <nomrole>Client</nomrole> <chefrole>Aya</chefrole> _ <listeacteur nbacteur="1"> <acteur>2</acteur> </listeacteur> </Role> _ <Role id="2" X="475" Y="74" W="150" H="30"></pre>	<pre>H="30"> <nomrole>Technique</nomrole> <chefrole>Mohamed</chefrole> _ <listeacteur nbacteur="1"> <acteur>1</acteur> </listeacteur> </Role> </ListeRoles> _ <ListeTaches nbtaches="8"> _ <Tache id="0" X="113" Y="115" W="40" H="40"> <nomtache>Star</nomtache> <typetache>Initial</typetache> <description /> <precondition /> <postecondition /> <duree>0</duree> <mode>Controle</mode> <donnee /> <typedonnee /> <role>Abonné</role> <activite>Activite0</activite> </Tache> _ <Tache id="1" X="107" Y="198" W="90" H="50"> <nomtache>Consulter</nomtache> <typetache>Elementaire</typetache> <description /> <precondition /> <postecondition /> <duree>0</duree> <mode>Controle</mode> <donnee /> <typedonnee /> <role>Abonné</role> <activite>Activite0</activite> </Tache> _ <Tache id="2" X="106" Y="347" W="90" H="50"> <nomtache>Inscrire</nomtache> <typetache>Elementaire</typetache> <description /> <precondition /> <postecondition /> <duree>0</duree> <mode>Realisation</mode> <donnee /> <typedonnee /> <role>Abonné</role> <activite>Activite0</activite> </Tache> _ <Tache id="3" X="266" Y="323" W="90" H="50"> <nomtache>Commander</nomtache> <typetache>Elementaire</typetache> <description /> <precondition /> <postecondition /> <duree>0</duree> <mode>Controle</mode></pre>
---	--

Figure 4.4: La spécification du workflow 'Achat électronique'

4.7 Conclusion

La meilleure exploitation des données ne sera garantie qu'en prévoyant une représentation facile à gérer, mais surtout portable et pérenne de ces données. C'est la raison pour laquelle, on a pensé tirer profit de la technologie XML; en effet l'intérêt de l'utilisation du XML comme format de stockage des données réside dans ces caractéristiques.

La représentation de cas (codée dans XML) a été conçue avec la considération de soutiens de recherche des modèles de processus similaires au problème, en effet, la recherche d'un cas codé en XML implique l'exploitation de la structure des cas pour rechercher les modèles similaires. Le prochain chapitre montre l'implémentation et l'utilisation de la structure de cas pour la conception de la plateforme BPR/CBR.

CHAPITRE 5

IMPLEMENTATION DU SYSTEME

5.1 Introduction

Nous avons constaté dans le premier chapitre le besoin de développement d'un système de re-modélisation des modèles de workflows aux utilisateurs. Notre idée est de leur fournir un framework afin de faciliter et d'accélérer le processus de re-modélisation des modèles.

Dans le troisième chapitre nous avons traduit la structure du workflow en différentes rubriques (informations sur les tâches, les transitions, la structure graphique...) à l'aide de balises dans un document XML. Ce qui va nous permettre d'implémenter l'étape de re-mémoration du CBR.

Ayant opté pour la technique raisonnement à base de cas (CBR) pour la gestion des connaissances dans la re-modélisation de processus métier. C'est un processus cyclique comportant quatre étapes :

- ◆ Recherche du cas le plus semblable,
- ◆ Réutilisation des solutions proposées pour résoudre le problème,
- ◆ Mettre à jour la solution proposée si nécessaire, et
- ◆ Retenue de la nouvelle solution comme partie du nouveau cas [1]

Nous présentons dans ce chapitre une méthode d'évaluation de similarité mise en œuvre dans un système d'indexation sémantique de documents XML (documents textuels semi-structurés). L'avantage de ces documents est qu'ils possèdent une structure qui facilite leur présentation, ainsi que leur interprétation et leur exploitation dans plusieurs contextes. Cependant, très souvent, la majeure partie de l'information reste

contenue dans la structure graphique du workflow, l'utilisation exclusive de la recherche textuelle n'est donc pas suffisante. Nous proposons un système de re-mémoration permettant d'exploiter à la fois la structure graphique et le contenu textuel des documents.

Le présent chapitre décrit le framework proposé ; il présente le processus de développement utilisé et décrit les outils réalisés avec ce framework. Nous présentons au début les différentes méthodes de re-modélisation afin que notre plateforme puisse les réaliser. Nous donnons d'abord une vision globale du framework et son architecture en insistant sur l'utilisation de raisonnement à base de cas. Puis nous détaillons l'architecture, en respectant tous les modules définis.

5.2 Les méthodes de re-modélisation d'un workflow

Il y a trois possibilités pour re-modéliser un processus métier, qui sont :

1. Adopter une approche « feuille propre », c.-à-d. le processus est conçu à partir de zéro.
2. La re-modélisation par « la dépendance de domaine » : un processus existe déjà utilisé comme un point de départ.
3. Utilisation d'un modèle de référence comme un patron pour le nouveau processus : le CBR est procédé dans cette méthode.

Le choix entre les différentes possibilités de re-modélisation est discuté dans la littérature [34]. Les adversaires de l'approche 'feuille propre' identifient quatre inconvénients principaux, qui sont comme suit [34]:

- Il y a le danger de concevoir un autre système inefficace.
- L'approche 'feuille propre' ne construit pas sur la connaissance et n'éprouve pas qui a été accumulée temps fini et risque des erreurs du passé.
- Les ouvriers peuvent ne pouvoir pas se relier au nouveau processus pendant qu'il soutient peu de ressemblance au travail qui est effectué.

La deuxième possibilité (approche) qui utilise un processus déjà existe est souvent préconisé. Cependant, nous avons pris en considération les trois méthodes de re-

modélisation pour développer notre plateforme ; afin que l'utilisateur puisse re-modéliser son processus par la méthode qu'il lui convient.

Les étapes de re-modélisation considérées par notre plateforme en utilisant l'approche raisonnement à base de cas, peuvent être définies comme suit:

- ❖ **Elaboration du problème cible :** La première étape consiste à formuler le cas problème à résoudre, cette formulation est appelée «Initialisation du modèle Entré ». Dans cette étape l'expert cherche à trouver les tâches pertinentes demandées qui servent à formuler le problème. Le système vérifie la validité du modèle initial de l'expert pour l'aider à corriger son modèle en cas d'erreur.

- ❖ **Remémoration du problème source :** Dans cette étape la plateforme compare le cas problème à résoudre à un ensemble de problèmes déjà résolus, en utilisant des règles de calcul de similarité. Ces problèmes sont résolus et enregistrés dans une base de cas. Le système analyse le problème de l'expert et lui propose les meilleurs choix en le justifiant par leurs degrés de similarité. L'expert choisit un cas solution qui lui semble le plus proche possible du cas problème courant.

- ❖ **Adaptation de la solution du problème source au problème cible :** Après avoir choisi un cas solution qui soit le plus proche du cas problème, la solution du cas problème (modèle de workflow) est déterminée en copiant et en adaptant la solution du cas remémoré (cas solution) pour tenir compte des différences entre les spécifications des problèmes. La solution obtenue est proposée à l'utilisateur ; qui doit adapter le modèle de workflow associé au cas problème pour construire un modèle pour le problème à résoudre.

- ❖ **Mémorisation du problème cible :** la dernière étape de résolution consiste à ajouter la solution du cas problème à la base de cas, en attachant le nouveau cas problème à un cas problème prototype, c'est à dire, en l'insérant dans un type de workflow. Ce dernier regroupe tous les problèmes qui sont attachés au même type. Il arrive que la solution du cas problème n'appartienne à aucune fonction, domaine de fonction, ou domaine de business déjà existants; dans ce cas, l'expert peut créer un nouveau dont le cas problème est lui-même le cas problème prototype.

❖ Révision : la solution du nouveau problème obtenue par l'adaptation est proposée. Les retours sur le succès ou l'échec résultant de la proposition permettent de réaliser des corrections. La révision permet d'identifier d'éventuelles causes d'échecs, et de proposer des adaptations supplémentaires en fonction d'une analyse des échecs. Cette étape n'est pas prise en compte dans notre travail, nous avons laissé cette étape comme perspective.

Après la définition des étapes de re-modélisation par CBR, nous montrons dans les sections suivantes notre méthode de réalisation de ces différentes étapes.

5.3 L'architecture de la Plateforme 'BPR/CBR'

Nous distinguons trois parties principales pour concevoir la plateforme BPR-CBR, qui sont : la première implémente la mémoire (dépôt), qui correspond à la base de cas et la base de vocabulaire, l'autre implémente la première méthode de re-modalisation « feuille propre », et alimente la base de cas et la base de vocabulaire. La dernière partie pour implémenter la deuxième et la troisième méthodes de re-modélisation (La dépendance de domaine et CBR).

Pour que la plateforme puisse prendre en considération toute les approches de re-modélisation d'un workflow décrites précédemment, l'architecture de la 'BPR/CBR' se compose de :

- ❖ La sous plateforme «Edition » : elle permet de représenter un workflow sous forme graphique ou textuelle, pour réaliser cette fonction, nous avons défini deux couche dans cette partie qui sont : la couche 'Modeleur' sert à définir le workflow graphiquement, et la couche « Parseur XML » qui génère la spécification du workflow sous forme textuel (document XML).
- ❖ La sous plateforme «Gestion des connaissances » : elle contient les couches 'Gestion de la base de cas', 'CBR' et 'Gestion de la base de vocabulaire' afin de réaliser les activités de KM, à savoir la création, le partage, et la réutilisation des workflow.

❖ Le dépôt : représente la mémoire de la plateforme BPR/CBR, il se compose de deux mémoire : 'la base de cas' et 'la base de vocabulaire, afin d'assurer la disponibilité des données utilisées par les sous plateforme «Edition» et «Gestion des connaissances».

Chaque sous plateforme comprend des couches qui sont constituées par des modules, et chaque module inclut des sous modules. L'architecture globale de notre plateforme est montrée par Figure suivante :

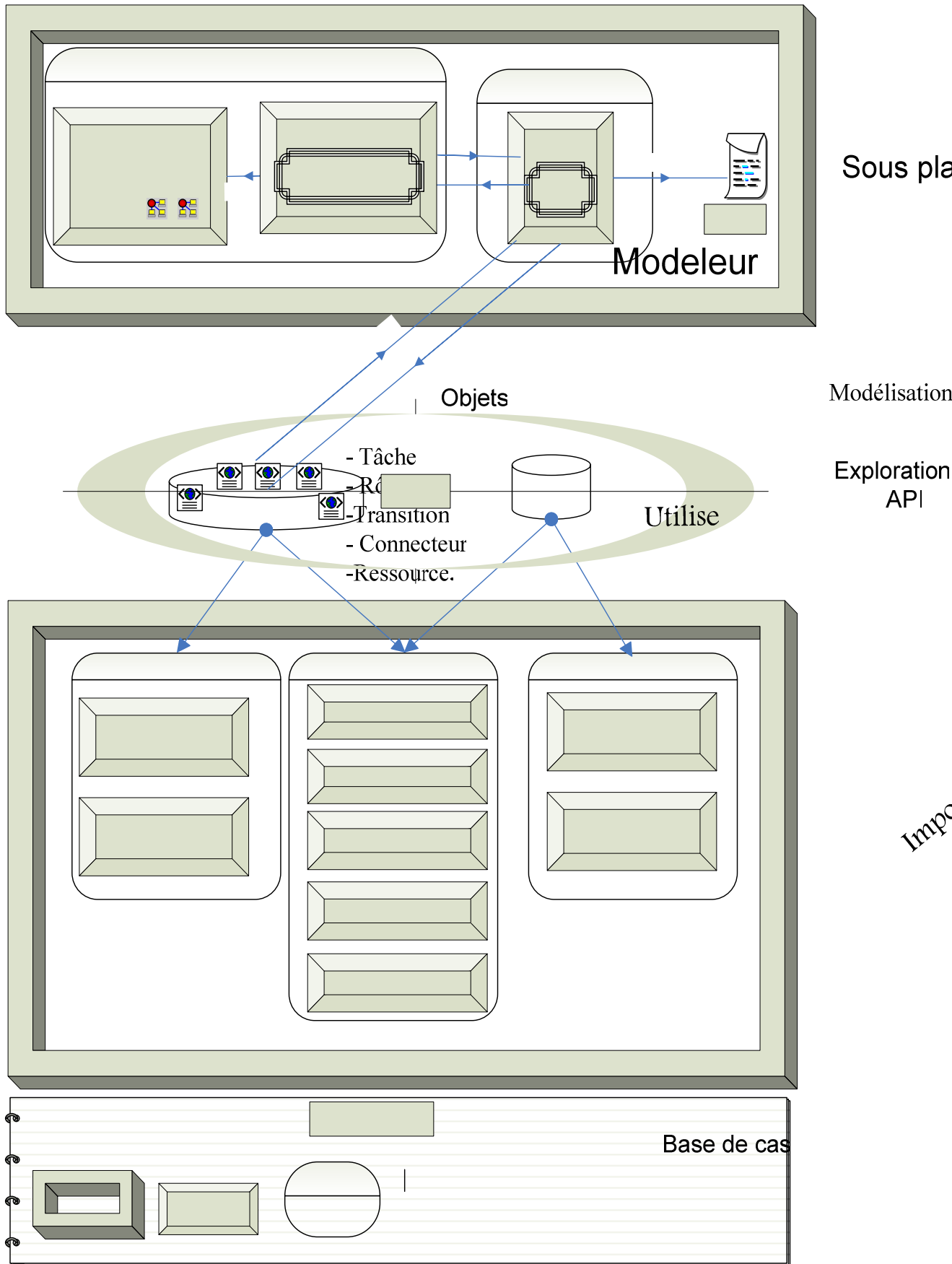


Figure 5.1: L'architecture de la plateforme BPR/CBR

Sous plateforme 'G...

Notre système est constitué de deux sous plateformes (Edition, Gestion des connaissances), et d'une mémoire (dépôt) qui inclut la base de cas et la base de vocabulaire, comme le montre la Figure 5.1.

La sous plateforme 'Edition' : la modélisation consiste à réaliser une description fidèle et exhaustive du processus. Pour cela nous avons défini un outil de modélisation de processus 'Modeleur'. Cette sous plateforme aide le concepteur à manipuler les couches 'Gestion de la base de cas', 'Gestion de la base de vocabulaire' à travers des fonctions interactives. Il reçoit les besoins de l'utilisateur pour modéliser le workflow, interroge l'utilisateur pour des informations spécifiques, traite les réponses, construit la description du modèle du workflow (modèle textuel), et montre le modèle graphique de workflow conçu.

Le sous framework 'Gestion des connaissances' est constitué de trois couches (Gestion de la base de cas, CBR et Gestion de la base de vocabulaire) ; La mise en œuvre de ces couches est proposée afin de développer des outils et des méthodes pour l'acquisition, la représentation, le partage et la réutilisation des modèles workflow, pour assurer les tâches de KM comme l'acquisition, la réutilisation et le partage de la connaissance.

La couche 'Gestion de la base de cas' se compose de deux modules : 'Consultation des cas' et 'Edition des cas'. Le module 'Consultation des cas' aide à parcourir la hiérarchie de la base de cas pour chercher un cas. Le module 'Edition des cas' édite le cas spécifique de workflow sélectionné via le module 'Consultation des cas', et fournit des fonctions pour supprimer et, ajouter un nouveau cas dans la base de cas.

Le système 'CBR' est constitué de cinq modules : 'Initialiser le cas problème', 'Rechercher', 'Sélectionner', 'Adapter' et 'stocker'. Le premier module scrute les cas pour trouver les workflows proches au cas problème. Le module 'Sélectionner' classe les cas trouvés et présente ce classement à l'utilisateur pour choisir le meilleur cas, ce dernier sera modifié pour refléter les besoins de l'utilisateur par le quatrième module (adapter), puis le module 'Stocker' enregistre le nouveau modèle dans la base de cas.

La couche ‘Gestion de la base de vocabulaire’ comporte deux module : ‘Consultation de vocabulaire’ et ‘Edition de vocabulaire’. Ce dernier assiste la suppression, l’ajout, la modification ou l’addition dans la base de vocabulaire. Le module ‘consultation de vocabulaire’ aide à parcourir les éléments de la base de vocabulaire tel que :

- Les noms des tâches (Exemple ‘recevoir commande’ ‘accepter chèque’),
- Les noms qui constituent les noms des tâches (Exemple ‘commande’, ‘chèque’),
- Les verbes qui constituent ces noms des tâches (Exemple ‘recevoir’, ‘accepter’),
- Les noms qui peuvent être liés à un verbe. (Exemple, le verbe ‘accepter’ est lié à ‘commande’ ou ‘chèque’, ...etc.),
- Et les verbes qui peuvent être liés à un nom. (Exemple, le nom ‘commande’ est lié à ‘recevoir’, ‘refuser’, ‘accepter’, ...etc.).

Le dépôt compose de ‘La base de cas’ et ‘ La base de vocabulaire’. Le succès d’un système CBR dépend de la maintenance effective de cette base de cas. Cette dernière enregistre un ensemble de cas d’après une hiérarchie de « est un ‘is a’ » [Aamodt, 94]. Les couches de cet arbre hiérarchique sont arrangées par ordre de niveau d’abstraction de domaine de métier. Nous avons introduit la notion de ‘la base de vocabulaire’ pour nous permettre une meilleure recherche de cas en utilisant la recherche sémantique. La base de vocabulaire maintient les noms de tâche et ces noms/verbes. Les noms et les verbes sont manipulés séparément à partir les noms de tâche.

5.4 La sous plateforme ‘Modeleur’

Une plate-forme BPR doit inclure un environnement de modélisation des processus [3] où les processus sont construits et modifiés de manière graphique et où sont définis les rôles (et les employés qui peuvent remplir ces rôles), les règles (qui définissent comment doivent se dérouler les processus) et les liens entre les différents processus.

Dans cette partie nous présentons les systèmes qui constituent la sous plateforme ‘Edition’ et les modules qu’ils contiennent, en déterminant les interfaces entre ces modules comme le montre Figure 5.1. L’interface ‘Exploration API’ du module « Modélisation », lui permet d’envoyer le modèle workflow créé à la couche « Parseur XML » pour générer

la spécification qui lui est associée, et d'importer la spécification de workflow à partir de la couche « Parseur XML » pour concevoir sa représentation graphique.

L'architecture de cette sous plateforme est composée de deux couches : « Modeleur » et « Parseur XML ». La première couche contient trois modules :

- ◆ Le module «Modélisation» utilise le module 'Objet' pour réaliser la modélisation d'un processus, le résultat de ce module est une représentation graphique d'un modèle workflow. L'extension du fichier contenant le graphe est (.PWF).
- ◆ Le module « Objet » contient des objets graphiques qui facilitent la modélisation d'un workflow pour l'utilisateur, nous définissons dans la partie suivante ces objets.
- ◆ Le transfert des données entre les modules «Modélisation» et «Scriptor» est assuré par l'interface «API DOM». Après la modélisation graphique d'un processus workflow au niveau du module «Modélisation», les objets du processus seront envoyés à la couche «Parseur XML» à travers l'interface «API DOM», pour générer la spécification du workflow (fichier XML) qui correspond à un cas, cette génération est assurée par le module «Scriptor».

La deuxième couche « Parseur XML » contient le module «Scriptor». Ce dernier est responsable de la génération des cas (document XML), le résultat de ce module est un fichier textuel (d'extension .XML) qui correspond à un cas (modèle).

5.4.1 Le module Scriptor

Au niveau de ce module se fait la représentation textuelle d'un processus workflow qui correspond à un cas. Le modèle créé par la couche 'Modeleur' est envoyé vers 'Scriptor' pour générer un cas sous forme textuelle (XML). Le fichier généré doit être conforme au méta modèle, pour cela deux étapes de traitement sont prévues :

- a).** Le passage : dans cette étape, il faut transformer la représentation du méta-modèle qui représente un workflow (présenté dans le chapitre 3) d'UML vers le langage XML, afin de générer la spécification de workflow (cas) ; il s'agit de présenter la grammaire (DTD) qui

inclut la structure du workflow conforme au méta modèle. Nous avons défini ce passage dans le chapitre précédent.

b). La validation : il s'agit de vérifier si la spécification (document XML) généré par le système est conforme au DTD (définie dans l'étape précédente), et par conséquent conforme au méta-modèle de workflow. Pour réaliser cette étape XML offre des mécanismes pour vérifier la validité du document produit et pour extraire les données de ce document, cette opération est possible à l'aide d'un outil appelé analyseur ou parseur. On distingue deux types de parseurs XML [12] :

- ❖ Les parseurs validants (validating) permettant de vérifier qu'un document XML est conforme à sa DTD.
- ❖ Les parseurs non validants (non-validating) se contentant de vérifier que le document XML est bien formé (c'est-à-dire, respectant la syntaxe XML de base).

Pour analyser un document, les parseurs utilisent actuellement deux types d'approches [15] :

- Les API basés sur un traitement hiérarchique : Les analyseurs construisent une structure hiérarchique contenant des objets représentant les éléments du document, et dont les méthodes permettent d'accéder aux propriétés. La principale API utilisant cette approche est DOM (Document Object Model).
- Les API basés sur un mode événementiel permettent de réagir à des événements (comme le début d'un élément, la fin d'un élément) et de renvoyer le résultat à l'application utilisant cette API. SAX (Simple API for XML) est la principale interface utilisant l'aspect événementiel.

L'interface SAX impose un parcours séquentiel du document XML traité, ce qui rend à priori son utilisation plus lente. A l'inverse de l'interface SAX, DOM permet un accès direct aux nœuds de l'arbre représentant le document XML traité mais, nécessite par contre la construction d'un arbre en mémoire contenant l'intégralité des éléments du document. Ainsi, pour de gros documents, l'utilisation de DOM devient difficile.

L'objectif du module 'Scriptor' est de concevoir un document XML qui contient le plus efficacement les informations d'un workflow. Ceci nous a amené à opter pour l'interface DOM.

5.4.2 Le module Objets

La modélisation graphique d'un processus workflow est réalisée par l'utilisation d'une notation graphique, cette dernière est présentée dans le chapitre 3. Les objets graphiques que nous avons définis pour la modélisation sont :

- Tâche initiale : Cet objet représente la tâche initiale d'un modèle de processus, n'a pas de prédécesseur et peut avoir un ou plusieurs successeurs ; elle est représentée par un cercle.
- Tâche finale : cet objet représente la fin d'un processus workflow. Il existe une et une seule fois dans un modèle de processus. Cet objet possède au moins un prédécesseur et n'a pas de successeur.
- La tâche élémentaire: pour représenter une tâche élémentaire, nous avons utilisé un rectangle étiquetée par le type de la tâche ; la tâche est de type stratégique, de type tactique ou de type opérationnel.
- La tâche composée : une tâche composée est composée de plusieurs tâches élémentaires. Pour représenter la tâche composée nous avons utilisé la notation graphique définie dans le chapitre 3
- Transition : La transition d'une tâche à une autre est représentée par une flèche reliant les tâches suivant une règle d'enchaînement.
- Rôle : Un rôle est un ensemble d'acteur, il est responsable de réaliser des tâches, pour représenter un rôle dans le graphe nous avons utilisé un couloir d'activité.
- Connecteur : cet objet peut être :
 - La règle Split-And : Le connecteur ne peut être franchi tant que les branches incidentes ne sont pas parcourues.
 - La règle Split-Or : Il faut que la branche incidente soit parcourue pour qu'une branche sortante choisie soit exécutée.
 - La règle Fork-And : Les branches sortantes sont exécutées en parallèle.
 - La règle Fork-Or : Il suffit qu'une des branches incidentes soit parcourue pour que la branche sortante soit exécutée.

5.4.3 Le module modélisation

Un processus se décrit sous forme d'un graphe ; les nœuds représentent les tâches élémentaires et composées, les arcs représentent les transitions d'une tâche vers une autre. Ce module utilise le module 'Objet' pour représenter les éléments métier d'un processus. Nous avons représenté dans le chapitre précédent notre approche de modélisation de workflow, et la notation utilisée.

Dans cette sous plateforme nous avons défini un éditeur graphique prend en considération toutes les notations graphique définies précédemment, cet éditeur permet aux experts de modéliser un processus métier. La figure suivante représente l'interface qui montre cet éditeur:

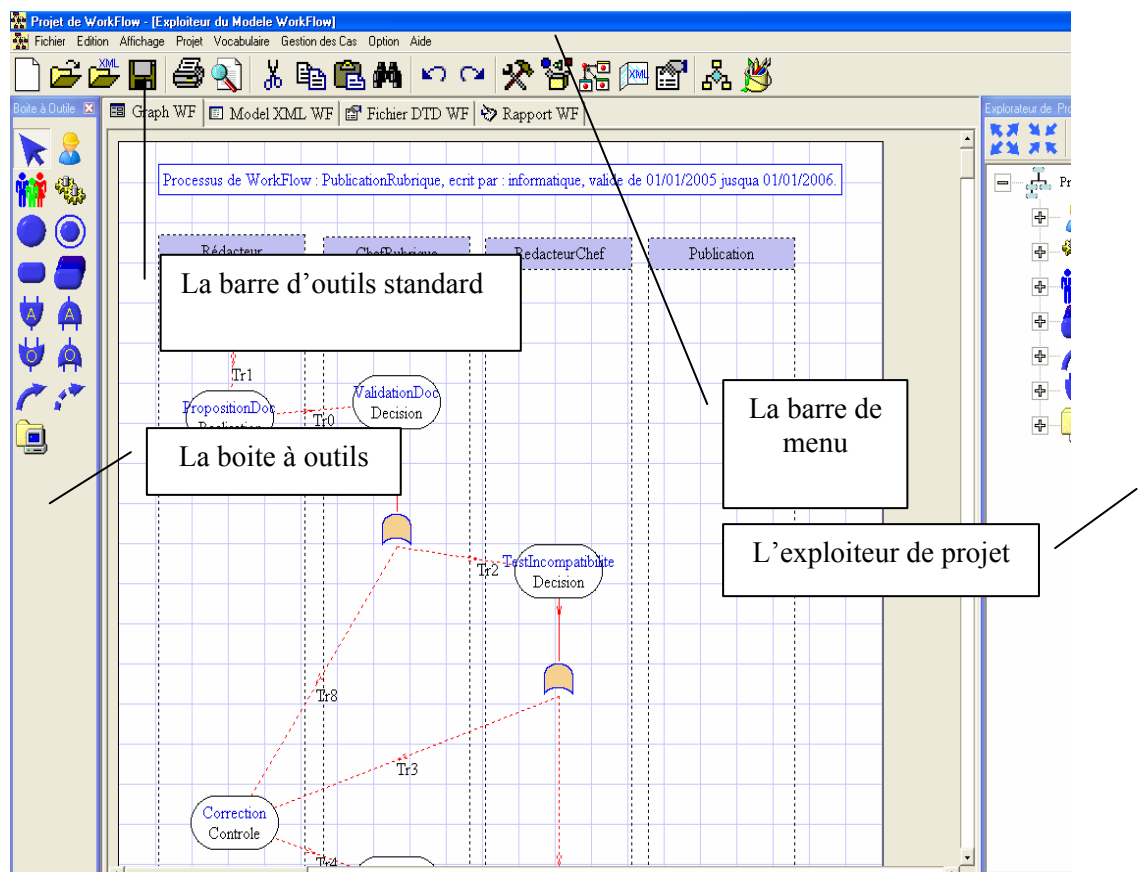


Figure 5.2 : L'interface de sous plateforme 'Edition'

Le diagramme de composant suivant représente les différents composants qui interviennent à la communication entre les deux modules : 'Scriptor' et 'Modélisation', en utilisant les interfaces DOM.

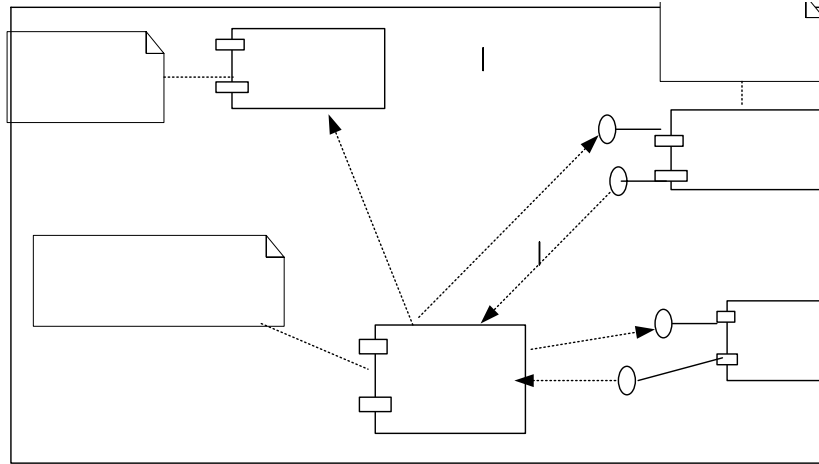


Figure 5.3 : Le diagramme de composant système

5.5 Le sous framework 'Dépôt'

Transmettre l'expérience en vue de la réutiliser nécessite une structuration de la mémoire collective. De manière à refléter ce mode de travail, Nous proposons un modèle de mémoire qui permet de réaliser toutes les fonctions de tout les module de la plateforme, tout en mettant à jour les connaissances et les informations dont les concepteurs ont besoin dans leurs activités. Notre modèle se décompose donc en deux parties : la base de cas et la base de vocabulaire. Ces deux bases se différencient par leur contenu, par la façon de les construire, ainsi que par la manière dont leurs versions sont gérées. Cette partie propose l'architecture et mode d'utilisation de ces mémoires.

5.5.1 La base de cas

Dans cette partie, nous abordons le problème de la structuration des cas afin d'améliorer les performances de la phase de remémoration du système CBR. Dans notre système de Raisonnement à Partir de Cas, les cas sont décrits sous forme textuelle avec une certaine structure ; il est primordial de pouvoir indexer ces cas selon leur contenu et raisonner dessus par similarité.

Confiance
contient
le DTD

Fichier
WF.DTD

Les interfaces DOM
fournit pour la
generation textuelle

Nous avons étudié dans le deuxième chapitre, les notions liées à la mémoire dans un système de raisonnement à partir de cas en général. Nous avons vu qu'il existe deux grandes approches pour l'organisation de la mémoire : l'approche de la mémoire plate et l'approche basée sur l'organisation hiérarchique de la mémoire.

Dans une mémoire plate, les cas sont stockés d'une manière séquentielle dans un fichier simple, la méthode de remémoration consiste à appliquer une fonction d'appariement sur chacun des cas, l'avantage de cette approche est qu'elle garantit une bonne précision pendant la remémoration des cas à cause de l'examen de toute la base de cas. En plus, la mémorisation d'un nouveau cas est facile et se traduit par l'ajout de ce cas à la fin de la mémoire. En revanche, le temps de remémoration augmente linéairement avec la taille de la base de cas.

L'organisation hiérarchique de la mémoire consiste à organiser les cas dans une mémoire avec plusieurs niveaux de hiérarchie. Cette méthode rend l'opération de remémoration plus efficace en terme de temps de calcul, mais elle complique la mémorisation des nouveaux cas à cause de la structure compliquée de la mémoire. Nous décrivons dans la suite notre choix de l'organisation des cas en détaille.

Mémoriser les workflows pour les réutiliser est une tâche complexe. Ce stockage des schémas nécessite à structurer dans une manière systématique. Notre base de cas est organisée sous forme d'un arbre hiérarchique de cas à partir du sommé nœud (domaine de métier) au nœud feuille (les tâches). Nous avons structuré la base de cas par la hiérarchie « est un », comme le montre la Figure 5.4. S'il y'a un nouveau workflow crée, il stocke dans la location approprié au chemin hiérarchique depuis le noeuds business au nœud workflow.

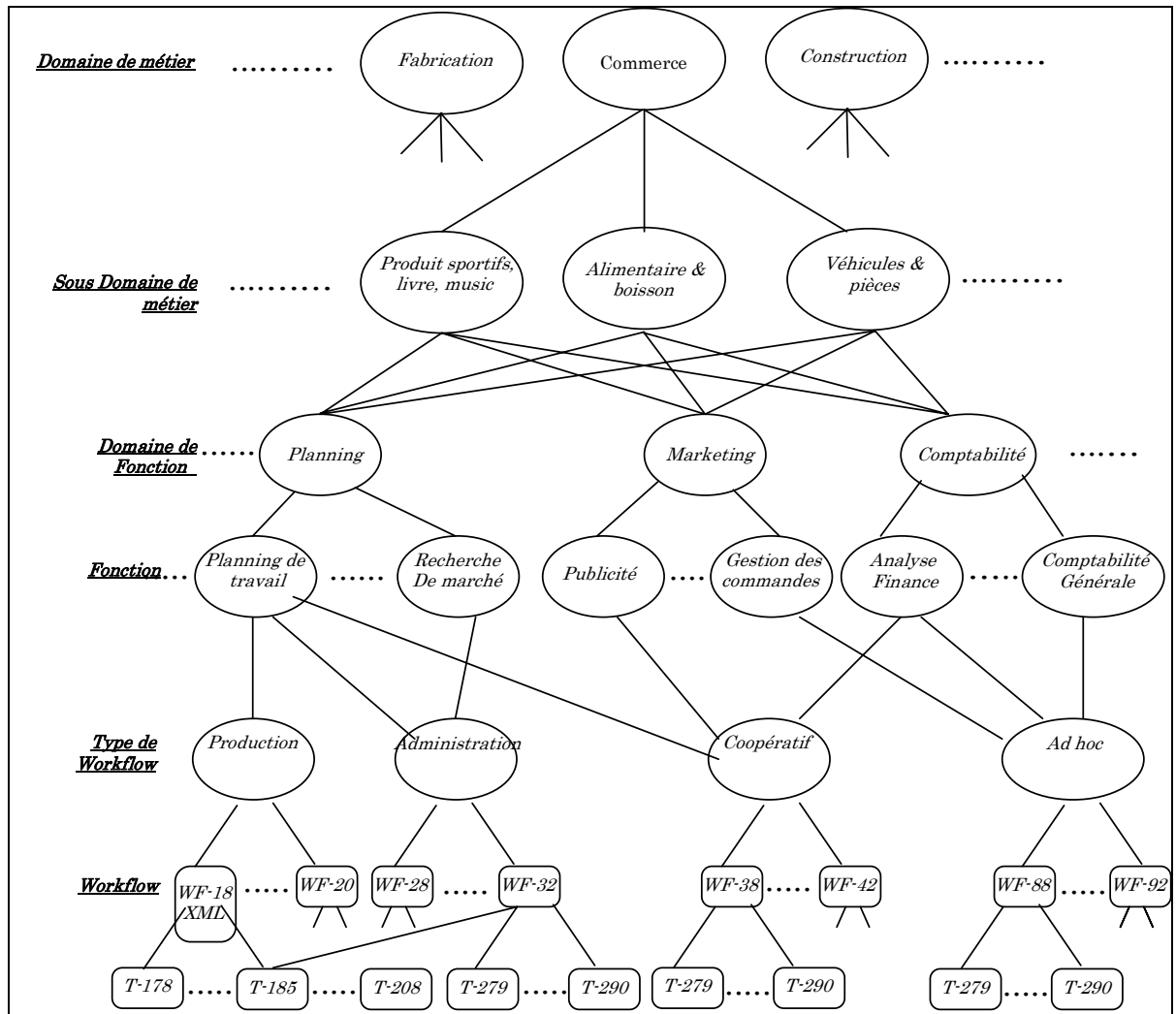


Figure 5.4 : La structure de la base de cas

Les quatre nœuds supérieurs (domaine de métier, sous-domaine de métier, domaine de fonction et fonction) représentent plus des caractéristiques abstraites des cas, alors que les trois derniers nœuds (type de workflow, workflow et tâche) représentent plus des caractéristiques spécifiques. Plusieurs tâches constituent un workflow, ce dernier appartient à un type, et plusieurs workflows constituent une fonction, et différentes fonctions constituent un domaine de fonction. Il existe des tâches qui peuvent être utilisées par plusieurs workflows.

Notre base de cas est représentée par le diagramme de classe montré par la

Figure 5.5. La classe Workflow possède une relation avec la classe Tâche; qui est en relation avec la classe Connecteur, ce qui implique un workflow est constitué de plusieurs tâches, les quelles sont liées par des connecteurs. La classe Tâche possède aussi

une relation avec les classes Ressource et Rôle, par ce que chaque tâche est réalisée par un rôle et produit une ressource en sortie à partir d'une ressource en entrée. La partie présentée (les classe Tâche, Ressource, Rôle et Connecteur) est indexée pour remémorer un cas, en utilisant un ensemble représente l'ensemble des synonymes du nom de la classe, ce qui implique, chaque classe Tâche, Ressource ou Rôle peut être représenté par un Ensemble des Terme Similaire (ensemble des synonymes) au nom de la classe, avec un degré de similarité. Ces ensembles de termes similaires sont les indexes principaux pour trouver les cas similaires à un cas problème à partir de la base de cas. Nous présentons la forme et l'utilisation de l'ensemble de termes similaires dans la section 5.6.1.2.2

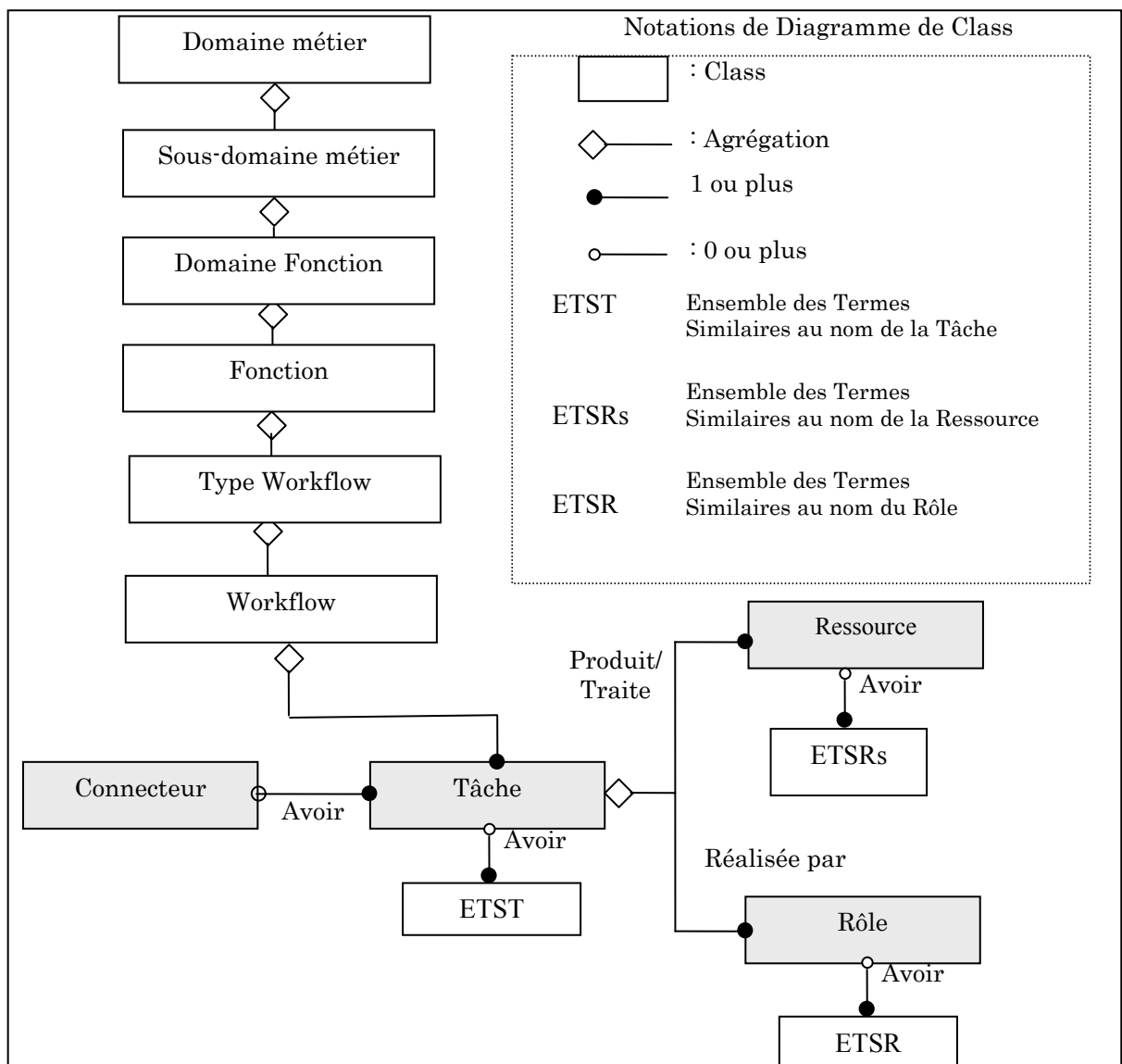


Figure 5.5 : Le schéma de la base de cas

5.5.2 La base de vocabulaire

Pour montrer l'utilité de la base de vocabulaire et comment elle peut être utilisée, son schéma de données est illustrée par la Figure 5.6. Dans ce schéma la relation «synonyme» dénote le degré de similarité entre deux vocabulaires sémantiquement. L'entité nom-tâche représente le service fourni par la tâche (le but). Il est composé par des paires des verbes et des noms qui constituent les noms des tâches. Par exemple, cette entité peut contenir la description de la tâche 'prendre commande' ou 'préparer livraison_demandée'.

Chaque entité nom-tâche possède des termes synonymes, par exemple la tâche 'recevoir une commande' a les tâches 'prendre une commande', 'obtenir une commande' comme des termes synonymes. Ce synonyme de la tâche est utilisé pour calculer le degré de similarité entre deux tâches en terme de sémantique, c à d si les noms de deux tâches sont similaires sémantiquement, alors elle sont regardées comme des tâches similaires.

L'entité 'Vocabulaire' représente les termes (des verbes et des noms) stockés dans la base de vocabulaire. Chaque élément (verbe ou nom) possèdent des synonymes. La relation 'Apparier' implique que chaque verbe (nom) lié à plusieurs noms (verbes), et chaque combinaison d'un verbe (nom) et un nom (verbe) forme un nom de tâche, par exemple, le verbe 'recevoir' est lié à plusieurs noms comme 'Commande' 'Accusé' et 'chèque', et le nom 'Commande' est aussi lié à plusieurs verbes comme 'Recevoir', 'Refuser' et 'Transférer'. En plus, quand l'utilisateur estime le verbe (nom) qui constitue le nom d'une tâche particulière, notre système aide l'utilisateur à trouver facilement les noms (verbes) liés à ce verbe (nom).

La relation 'SynonymeV' représente les similarité entre les noms des tâches ou les items vocabulaires (verbes et noms), c'est une relation binaire, récursive. La propriété Poids mesure combien deux items (tâches) sont similaires sémantiquement.

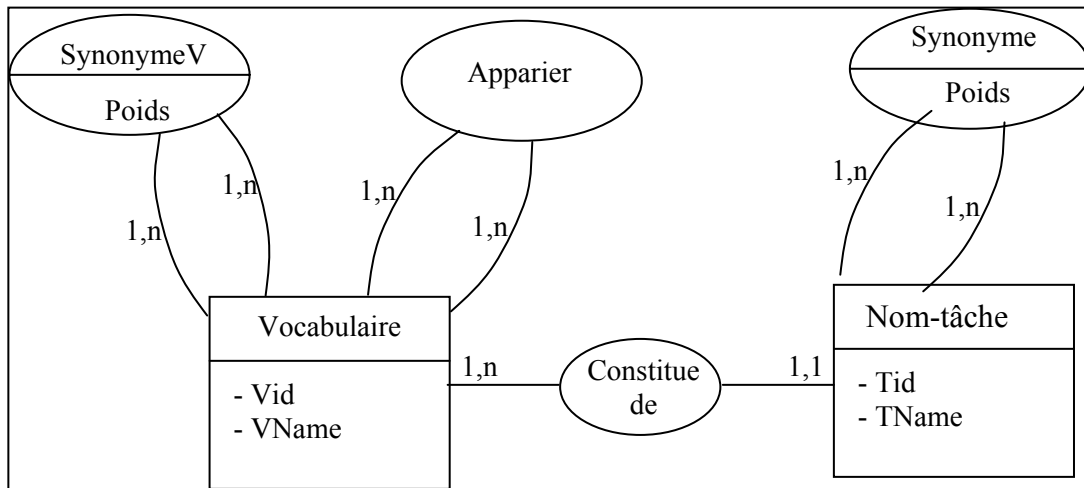


Figure 5.6: Le schéma de la base de vocabulaire

5.6 La sous plateforme 'Gestion des connaissances'

Dans cette partie, nous décrivons le processus de re-modélisation par les deux approches : « Dépendance de domaine » et « CBR ».

La re-modélisation de processus métier (en utilisant la plateforme BPR/CBR) peut se poser à l'utilisateur par deux approches (la Figure 5.7) :

a) Re-modélisation des workflow par l'approche 'Dépendance de domaine' : consiste à re-modéliser un processus trouvé par l'utilisateur, et en utilisant la hiérarchie 'Dépendance de domaine' de la base de cas.

b) Re-modélisation des processus par la méthode 'CBR' : correspond à la re-modélisation d'un processus similaire au processus problème, la différence entre cette approche et l'approche 'Dépendance de domaine' réside dans la recherche de processus, la recherche du processus par l'approche 'CBR' s'effectue sans intervention de l'utilisateur, par contre, la recherche par 'Dépendance de domaine' est réalisée par l'utilisateur, mais avec l'aide de système.

La re-modélisation par l'approche 'Dépendance de Domaine' peut être la première possibilité si l'utilisateur connaît la location du processus problème, car elle est rapide et facile (dans cette recherche on a pas les étapes : 'rechercher les cas similaires' et la 'sélection', de l'approche CBR). Cependant, si cette recherche est insatisfaisante, la méthode CBR peut être adopter. Les deux méthodes de re-modélisation de processus sont détaillées ultérieurement.

v

Figure 5.7 : Le processus de re-modélisation par la plateforme BPR/CBR

Dans la première possibilité de re-conception, l'utilisateur re-modélise son processus par l'approche 'Dépendance de domaine', suivant les étapes suivantes (montrée dans la figure ci-dessus):

[a] Il commence la recherche des cas à l'aide de la hiérarchie 'Dépendance de domaine' de la base de cas, où il parcourt la base de cas couche par couche. Pour soutenir la recherche de l'utilisation nous avons défini le module «Parcourir la base de cas»

[b] Si l'utilisateur trouve le processus métier [c, d], ce dernier est rarement répond exactement aux besoins d'utilisateur, il nécessite une adaptation [e], et ce nouveau cas est intégré dans la base de cas pour future utilisation [f]. Nous illustrons cette méthode de re-modélisation par le diagramme de séquence suivant :

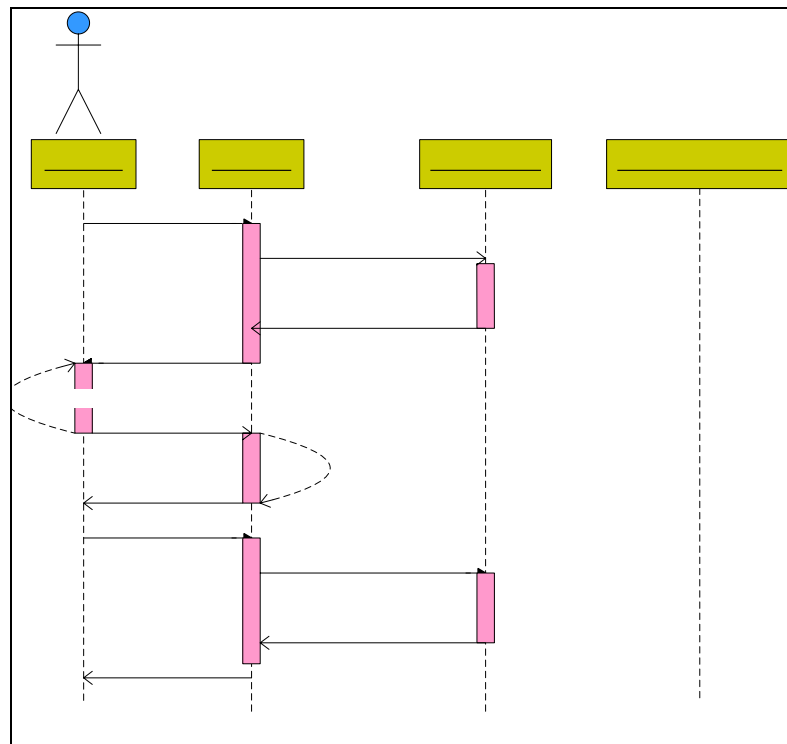


Figure 5.8 : Séquence de la re-modélisation par 'dépendance de domaine'

Après l'échec de la première approche de re-modélisation, puisque l'utilisateur n'a pas des informations complètes sur le processus re-modélisé, il doit adopter la deuxième possibilité, c'est l'approche CBR [1] (montrée dans la figure Figure 5.7). Pour chercher le

processus re-modélisé par l'approche CBR, il nécessite un modèle initial, ce dernier est initialisé par le système avec la contribution de l'utilisateur, ce dernier connait les tâches primordiales du problème, et ensuite il lance une recherche pour les noms de ces tâches dans la base de vocabulaire [2]. En appliquant nos règles de similarité, notre plateforme BPR/CBR peut trouver des workflow similaires au modèle initial ; dans l'étape de sélection l'utilisateur intervient pour décider le processus le plus similaire à son problème [3, 4, 5], ce cas sélectionné est adapté pour les besoins d'utilisateur [6]. Puis stocké dans la base de cas. Le diagramme de séquence suivant montre les étapes de cette méthode de re-modélisation.

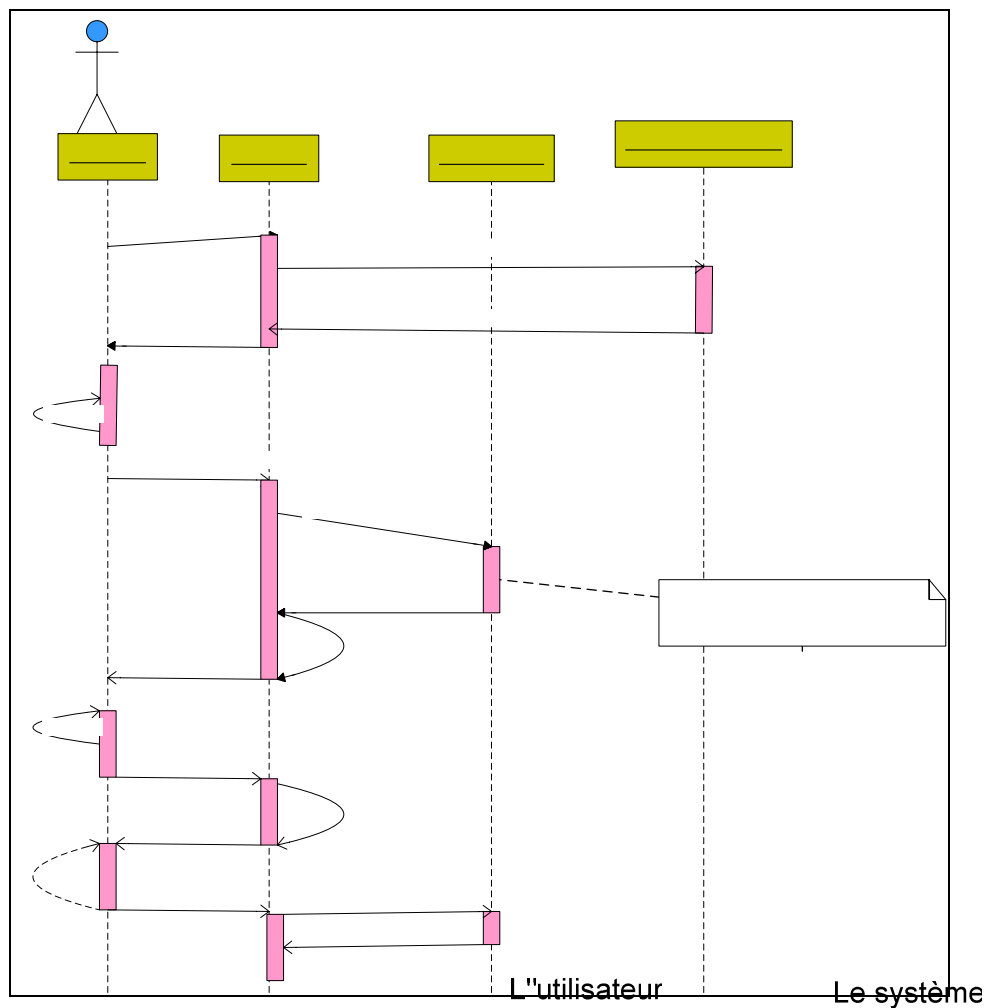


Figure 5.9: Séquence de la re-modélisation par 'CBR'

5.6.1.1 La Re-modélisation par l'approche 'Dépendance de domaine' Tâche primordiale

Nous avons identifié deux étapes principales pour réaliser la méthode « Dépendance de domaine », la première étape celle de processus de recherche, et la

Noms Similaires

Selection

deuxième étape c'est l'adaptation et le stockage du nouveau cas. Cette partie présente notre approche de réalisation de ces étapes.

5.6.1.1.1 Processus de recherche

La recherche d'un cas en utilisant l'approche Dépendance de domaine est une sorte de recherche, qui examine tout les nœuds (cas) dans la base de cas commençant par la racine (le nœud domaine de métier) et suivant la hiérarchie Dépendance de domaine, jusqu'au nœud workflow, comme nous montre la Figure suivante.

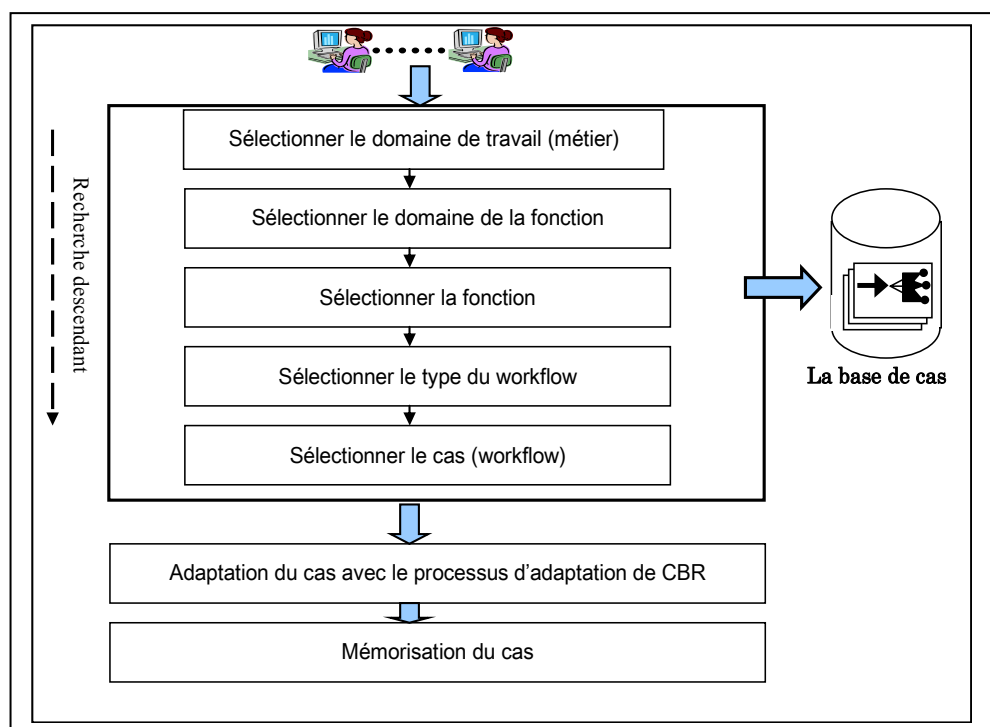


Figure 5.10 : Le processus de recherche des cas par dépendance de domaine

Si l'utilisateur trouve le cas demandé par cette approche, il peut l'utiliser sans changement ou adaptation, selon ses besoins. Mais dans le cas général, le cas trouvé ne répond pas exactement aux besoins de l'utilisateur, une adaptation du cas est nécessaire pour répondre aux besoins d'utilisateur (exemple : combler les lacunes et supprimer les parties inutiles). Dans cette méthode de re-modélisation, le cas trouvé sera adapté en utilisant le processus d'adaptation de l'approche CBR, durant cette étape la partie qui ne répond pas aux besoins d'utilisateur est substituée par une autre partie réutilisable d'un autre cas. Finalement, si une solution est créée par cette approche, ce nouveau cas doit être sauvegardé dans la base de cas pour future utilisation.

Pour illustrer l'utilisation de l'approche 'Dépendance de domaine' nous utilisons l'exemple suivant. Si l'utilisateur cherche les cas de workflow liés à la fonction 'processus commande', il nécessite dans la première recherche les nœuds de la couche domaine de métier et sélectionne 'Commerce Détaillé', parmi ses sous-domaines de métier, il choisit aussi le nœud 'vente produit, achat produit,..'; ensuite, il sélectionne le nœud 'Distribution' dans la couche 'domaine de fonction', puis il choisit la fonction 'processus commande', finalement il peut examiner les workflow par leurs type, ce dernier figure dans une liste appelée 'workflow', et finalement, l'utilisateur choisit les workflow comme 'paiement, acceptation...'

La Figure suivante illustre l'interface graphique du prototype de notre plateforme, quand l'utilisateur parcourt la base de cas (les modèles de workflow). Cette image d'écran montre les boutons et les menus qui permettent l'accès à la couche « Gestion de la base de cas ». Un cas est affiché sous forme graphique et textuelle (XML).

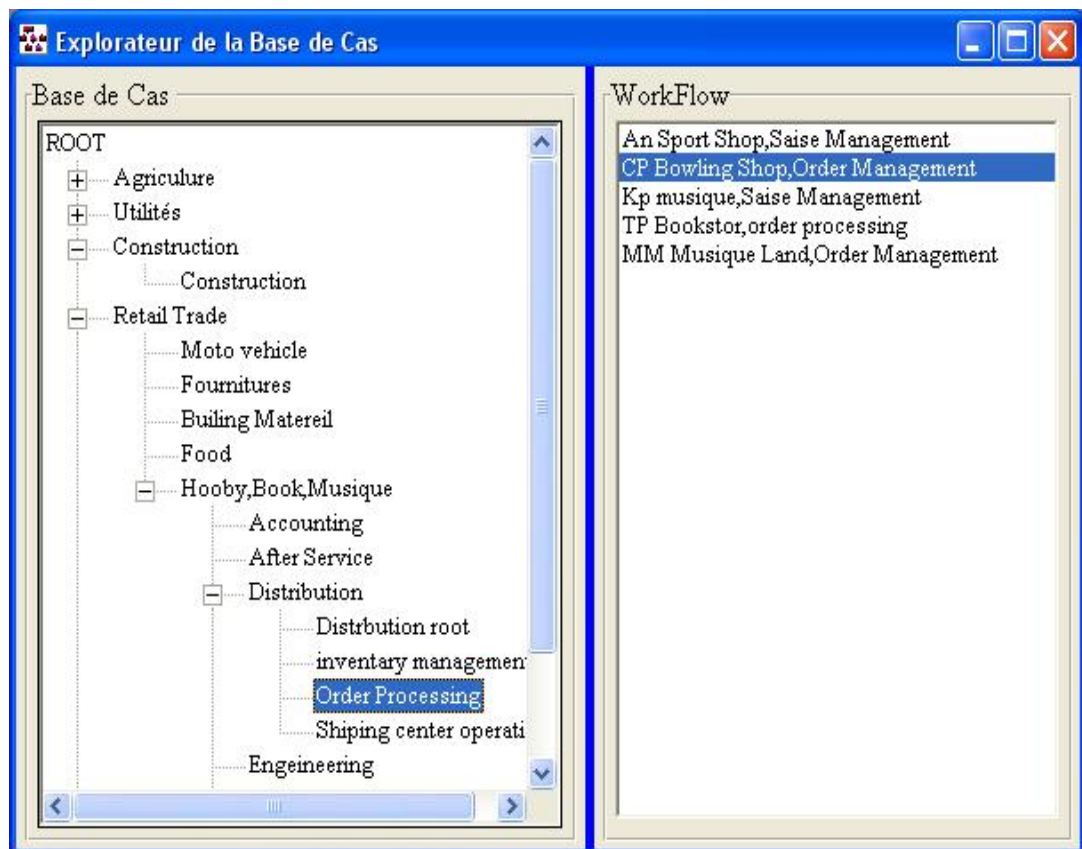


Figure 5.11 : L'interface d'utilisateur de 'Gestion de la base de cas'

5.6.1.1.2 Adaptation et mémorisation d'un cas

L'adaptation d'un cas est définie comme réalisation des changements pour étendre un cas qui peut être utilisé dans la situation courante. Reconnaître quelles sont les nouvelles exigences a modifié et comment effectuer cette modification, sont les considérations principales. L'adaptation de l'ancien modèle est, en effet, la conception d'une nouvelle solution. Ce cas adapté doit être stocké après l'étiquetage comme un nouveau modèle de workflow dans sa propre location dans la base de cas pour la réutilisation. Les deux étapes d'adaptation et mémorisation sont les mêmes étapes de la deuxième approche de la re-modélisation : 'CBR', nous les expliquerons dans la section ci-dessous.

5.6.1.2 Re-modélisation en utilisant l'approche Raisonnement à base de cas

Quand l'utilisateur ne trouve pas des cas similaires au moyen de la recherche de cas par Dépendance de domaine, il doit employer l'approche raisonnement à base de cas. Cette stratégie est considérée comme une méthode descendante. Nous montrons dans cette section notre approche de réalisation de cette méthode de re-modélisation.

5.6.1.2.1 Le module 'Initialiser le problème' (Input model)

Pour initialiser le modèle entré (initial), l'utilisateur doit connaître les tâches primordiales de la solution. Mais il n'est pas facile pour l'utilisateur de deviner les tâches principales exactement par son imagination, il nécessite un mécanisme pour lui aider dans son travail. Pour cette raison, la base de vocabulaire contient les noms des tâches, et elle est structurée et gérée.

Pour initialiser un workflow, premièrement, l'utilisateur aperçoit les tâches principales du processus workflow, il trouve chaque nom de ces tâches à l'aide de la couche 'Gestion de la base de vocabulaire' de la plateforme. Si l'utilisateur entrait un verbe (nom), le système suggère la contrepartie noms (verbes) qui est connectée à ce verbe (nom). Par exemple, si le verbe entré est 'recevoir', le système montre les noms des tâches

liés à ce verbe, comme ‘chèque’ à partir de la tâche ‘recevoir chèque’ et ‘rapport’ à partir de la tâche ‘recevoir rapport’.

L'utilisateur répète ce processus pour obtenir les différentes tâches nécessaires pour formuler le modèle entré. Quand il termine toutes les tâches, il commence à initialiser le modèle entré par l'affectation des dépendances entre les tâches, en terme comment ces tâches doivent interconnecter afin que le travail sera effectué correctement. Nous illustrons les étapes d'initialisation du modèle entré par l'organigramme suivant :

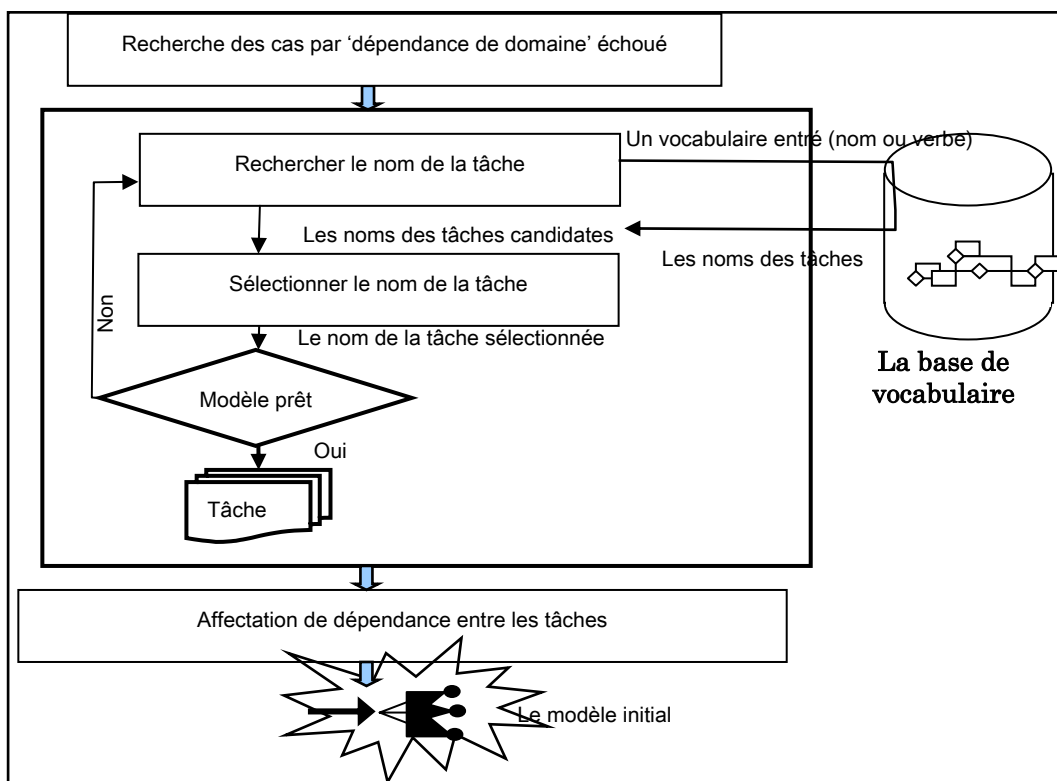


Figure 5.12 : Le processus d'initialisation du modèle entré

Le prototype de la plateforme proposée prend en considération cette étape. La Figure 5.13 montre l'interface utilisateur pour initialiser un modèle d'entrée, si l'utilisateur rentre un (des) mot clé (exemple : le nom ‘Commande’) dans le contrôle ‘Recherche un cas’ et click sur le bouton ‘Chercher’, le système lui affiche les noms des tâches qui inclus

le mot 'commande' dans la liste 'Tâches'. Parmi ces tâches trouvées, l'utilisateur sélectionne la tâche qu'il veut par un double click. Au même temps, la description de la tâche sélectionnée est montrée dans le contrôle 'Définition de la Tâche'. S'il click sur le bouton 'Confirmation' pour finir la recherche de la tâche, cette dernière est montrée dans l'espace de modélisation nommé 'Cas de workflow'. Pour montrer les flux entre les tâches principales définies par l'utilisateur, il sélectionne le connecteur qu'il veut parmi celles défini dans le Chapitre 3 qui sont : 'or-split', 'and-split', 'or-join', 'and-join', et séquence. Dans la partie gauche de l'écran l'utilisateur peut voir le modèle graphique entré qui inclut dans notre exemple les tâches 'recevoir commande', 'recevoir chèque' et la tâche 'préparer commande'.

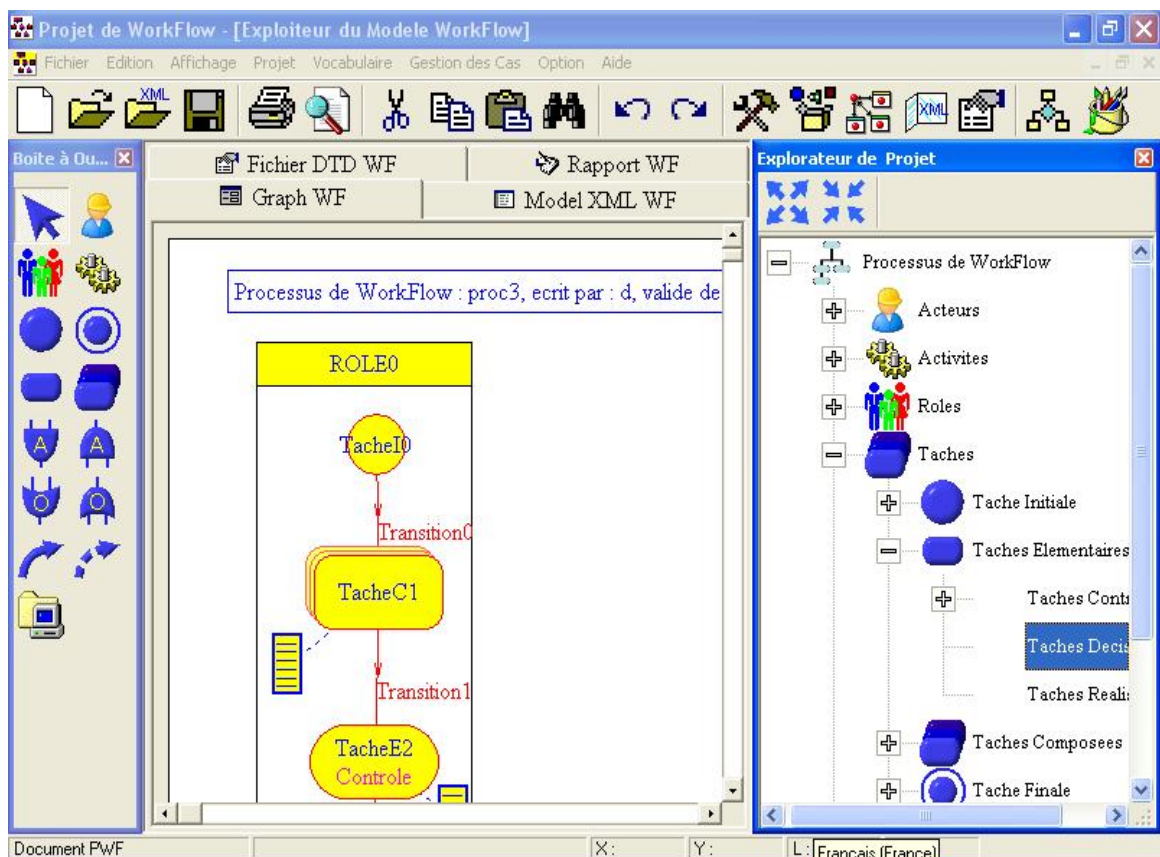


Figure 5.13 : L'interface d'utilisateur pour initialiser le problème (modèle entré)

5.6.1.2.2 Le module Rechercher

Le système nécessite de définir des techniques de remémoration pour retrouver les cas les plus pertinents, et le plus rapidement possible. Le système doit également posséder une (ou des) mesure(s) de similarité pour effectuer une sélection fine des cas. Nous cherchons une mesure qui permet de, maximiser la similarité entre le problème posé dans

le cas problème et la solution posée dans le cas solution, et à minimiser l'effort d'adaptation entre les deux cas.

Si le modèle entré est prêt, il est possible maintenant de déterminer comment il est similaire aux cas dans la base de cas. Cette similarité est déterminée en fonction des tâches, des ressources, des rôles et de la dépendance entre les tâches.

Comme nous montre la Figure 5.14 en introduisant le modèle initial, le système calcule le degré de similarité entre le problème et les cas existents dans la base de cas, en utilisant les règles de récupération qui sont sous forme de formules, en suite le système classe les cas trouvés par ordre décroissant de leurs degrés de similarité. Parmi la liste des cas proposée, l'utilisateur sélectionne le cas qui a le plus grande similarité comme un cas proposé, ce dernier est envoyé à l'étape suivante 'Adaptation' afin qu'il réponde aux besoins utilisateur.

Nous présentons ci-après notre méthode de remémoration des cas en fonction du nombre d'étapes qu'ils nécessitent pour réaliser cette phase.

1) Choix des critères de sélection

Notre étude pour la détermination des critères de similarité s'appuie sur des ouvrages dédiés à la technologie workflow [3] [27][6] [34]. Le but de cette étude est de déterminer des critères permettant de caractériser un problème de workflow en fonction des actions (tâches) à effectuer et des données à traiter. Les critères que nous proposons sont issus d'une classification des termes retenus pour décrire un workflow.

Nous avons distingué deux grandes catégories de critères : ceux qui sont liés à la description de la tâche et ceux qui sont liés à son contexte. La figure suivante illustre les critères utilisés (étapes) pour rechercher un cas :

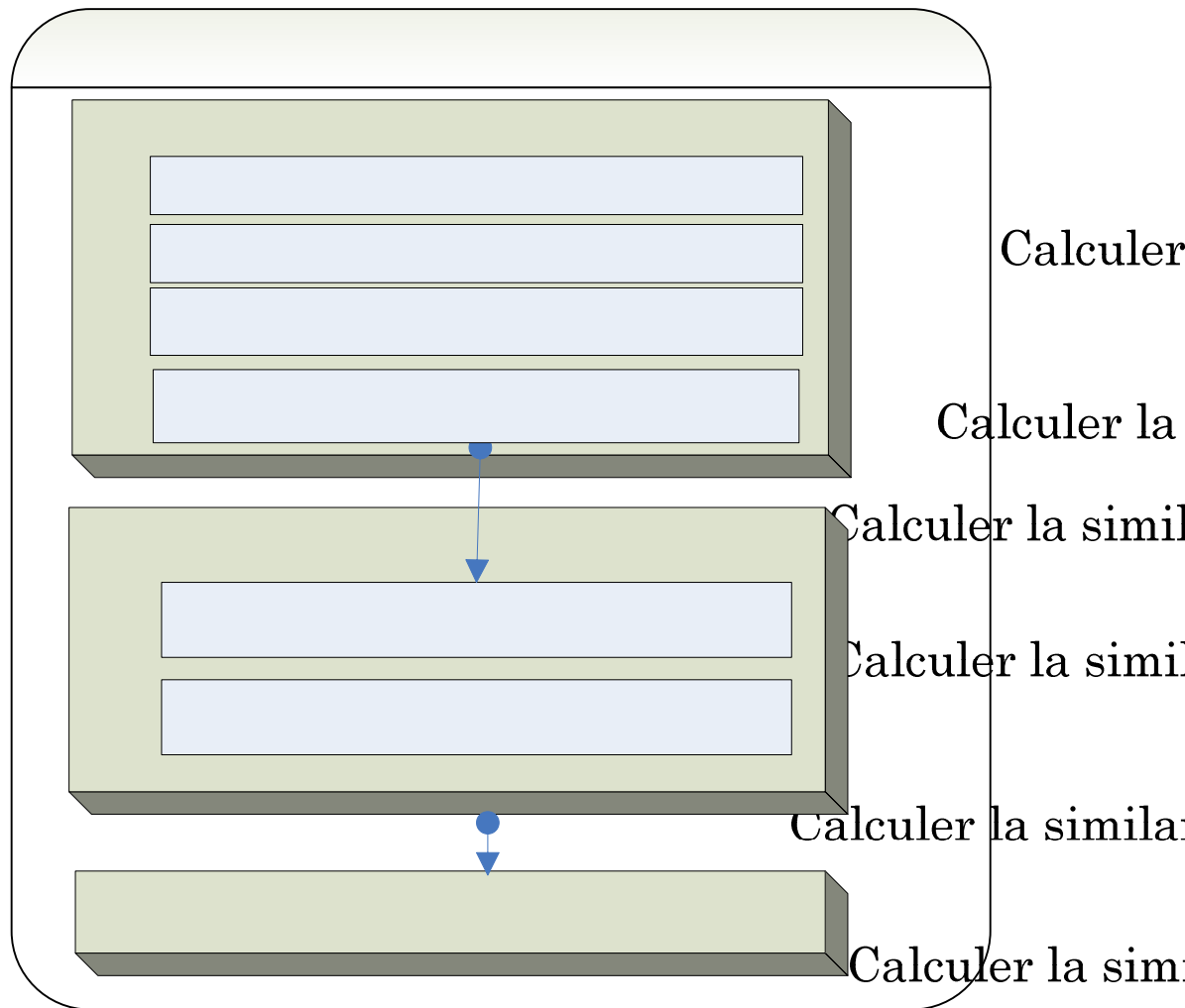


Figure 5.14 : Les étapes de calcul la similarité entre les cas

2) Calculer la similarité entre les tâches

La Figure 5.14 montre que l'opération de calcul de similarité entre les tâches est constituée de quatre étapes : calcule la similarité entre les noms des tâches, entre les types des tâches, entre les noms des ressources utilisées, et calcule la similarité entre les noms des rôles. Le calcul de Degré de Similarité entre les Nom des Tâches (DSNT) est effectué par la comparaison du nom de la tâche i (du modèle initial) avec les noms d'autre tâches (inclut dans l'ensemble des termes similaires) (la formule I et II). Si le nom de la tâche spécifique i du modèle initial est inclus dans cette l'ensemble des termes similaires, il peut avoir un degré de similarité. Par exemple, le nom de la tâche 'recevoir commande' possède l'ensemble des termes similaires :

$E = \{ \text{recevoir commande}, 1, \text{prendre commande}, 1, \text{demande commande}, 0.8, \dots \}$
 tel que le terme similaire 'recevoir commande' a un degré de similarité 1 et 'demande commande' a un degré de similarité 0.8, etc.

Pour représenter l'ensemble des termes similaires entre deux modèles nous avons défini la structure suivante :

<p>Similar-Term (Tn) = { Tn₁, w₁, Tn₂, w₂, Tn₃, w₃, ..., Tn_k, w_k }</p> <p>Tn: le nom de la tâche</p> <p>Tn_k: le k^{ème} élément de l'ensemble des termes similaires</p> <p>w_k: le poids de k^{ème} élément de l'ensemble des termes similaires.</p>

Équation I : La représentation de l'ensemble des termes similaires

Pour illustrer comment calculer la similarité entre deux workflows, nous utilisons l'exemple de processus métier 'achat électronique' présenté dans la figure suivante.

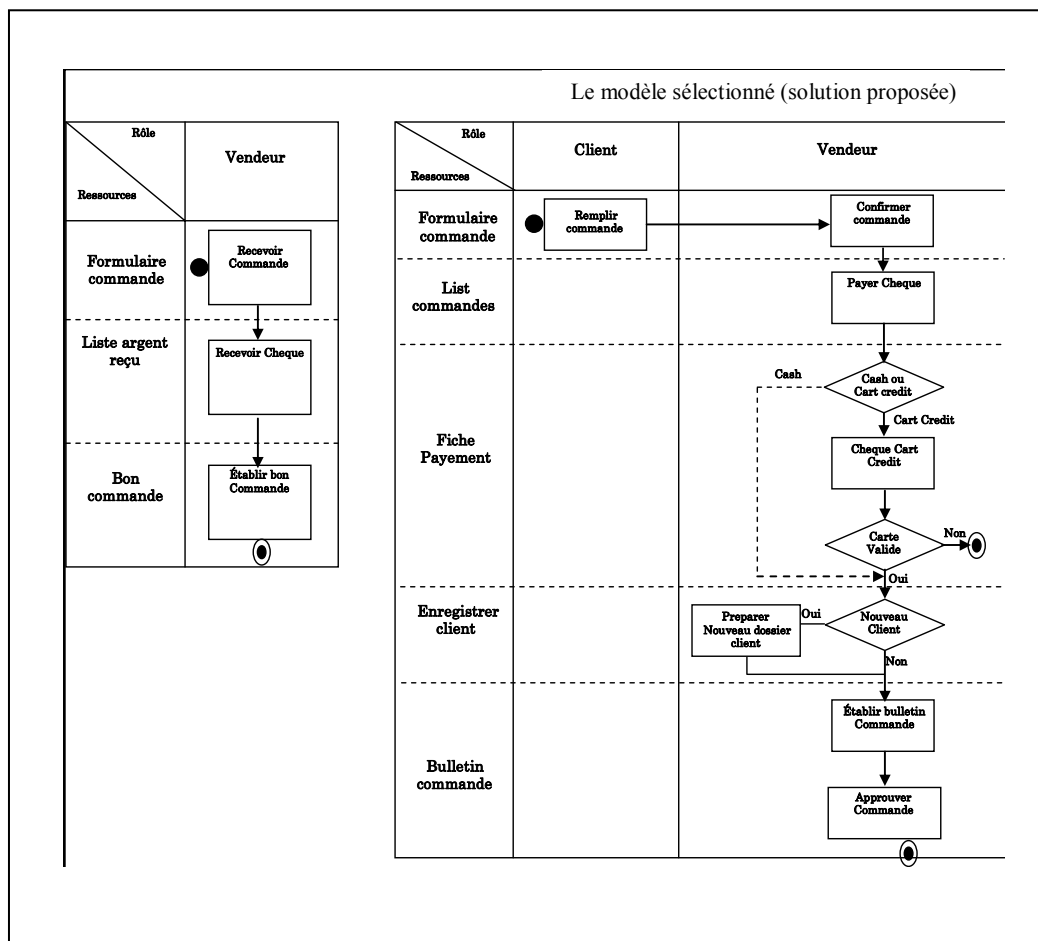


Figure 5.15 : Exemple d'un modèle entré et un modèle trouvé

2.a) Le calcul de degré de similarité entre les noms des tâches (DSNT)

Pour calculer le DSNT nous définissons la règle suivante :

$$T_n = \sum_{i=1}^{i=n} Tn_i / n$$

T_n : Le degré de similarité total des noms des tâches du problème.

Tn_i : Le degré de similarité du nom de la tâche i .

n : le nombre des tâches du problème.

Équation II : La formule de calcul de similarité des noms des tâches

En appliquant la règle de DSNT sur l'exemple de 'l'achat électronique', nous remarquons que les tâches du modèle entré ('recevoir commande', 'Recevoir chèque', 'établir bon commande') correspondent aux tâches ('confirmer la commande', 'payer chèque', 'établir bulletin commande') respectivement. Chaque tâche possède un degré de similarité calculé comme suit :

'Recevoir commande'=0.8, selon l'ensemble des termes similaires suivante :
 Sim-Term(confirmer commande) = {confirmer commande, 1.0, affirmer commande, 1.0, tenir commande, 0.8, recevoir commande, 0.8}

'Recevoir chèque'=1.0, d'après l'ensemble des termes similaires suivante :
 Sim-Term(payer chèque) = {payer chèque, 1.0, recevoir chèque, 1.0, examiner chèque, 0.7, contrôle payer, 0.7 }

'Etablir bon commande'= 1.0, suivant l'ensemble des termes similaires suivante :
 Sim-Term (établir bulletin commande) = {établir bulletin commande, 1.0, établir bon commande, 1.0, préparer bulletin commande, 1.0}

En appliquant la règle de DSNT précédente, nous trouvons le résultat
 DSNT=(0.8+1.0+1.0) /3=0.93.

2.b) Le calcul de Degré de Similarité entre les Types des tâches (DSTT)

Dans le chapitre 3, nous avons défini 3 types de tâches effectuées au sein d'une entreprise qui sont : la tâche 'Tactique', la tâche 'Stratégique' et la tâche 'Opérationnelle'. Dans cette partie nous montrons l'utilité de cette décomposition.

Pour avoir une similarité plus précise entre deux tâches, nous comparons les types des deux tâches. Nous avons défini des valeurs de degré de similarité entre deux types des tâches dans le tableau suivant :

	stratégique	tactique	opérationnelle
stratégique	1	0.3	0
tactique	0.6	1	0.3
opérationnelle	0	0.6	1

Tableau 5.1 : Définition du degré de similarité du type des tâches

Pour le même exemple de l'achat électronique, les degrés de similarité entre les types des tâches (calculés par la règle de DSTT définie) donnés par le tableau suivant :

Tâche	Type	Correspond	Tâche	Type	Le degré de similarité
recevoir commande	opérationnelle	→	confirmer commande	tactique	0.6
Recevoir chèque	opérationnelle	→	payer chèque	opérationnelle	1
Etablir bon commande	opérationnelle	→	Établir bon commande.	opérationnelle	1

Tableau 5.1: Calculer la similarité

Pour calculer la similarité totale entre les types des tâches nous avons défini la formule III. En appliquant cette formule pour l'exemple précédent nous avons trouvé le résultat suivant : $DSTT = (0.6+1+1)/3=0.86$.

$$T_t = \sum_{i=1}^{i=n} Tt_i / n$$

T_t : Le degré de similarité total des types des tâches du problème

Tt_i : Le degré de similarité du Type de la tâche i

n : le nombre des tâches

Équation III : La formul de calcule de similarité des types des tâches

2.c)Le calcul de degré de similarité de la ressource (DSRs) et du rôle (DSRI)

Le calcul de degré de similarité entre les ressources et entre les rôles est montré dans les formule IV et V, il est basé sur le même concept de processus de calcule de DSNT.

$$R_{ST} = \left(\sum_{e=1}^l R_{Se} \right) / l$$

R_{ST} : Le degré de similarité total entre les ressources du problème.

R_{Se} : Le degré de similarité d'une ressource.

e : index des ressources.

l : le nombre des ressources.

Équation IV : La formul de calcule la similarité des ressources

En appliquant cette règle sur l'exemple, nous trouvons que : les ressources du modèle entré ('formulaire commande', 'liste argent reçu', 'bon commande') correspondent à celles de la base de cas ('formulaire commande', 'liste commande', 'bulletin commande'). Chaque ressource a son degré de similarité, calculé comme suit :

'Formulaire commande'= 1.0, suivant l'ensemble des termes similaires suivant : $\text{Sim-Term}(\text{formulaire commande}) = \{ \text{formulaire commande} , 1.0, \text{bulletin commande}, 0.5, \text{liste commande}, 0.5 \}$.

‘Liste argent reçu’= 0.0, selon l’ensemble des termes similaires suivant :
 $\text{Sim-Term}(\text{liste commande}) = \{\text{liste commande}, 1.0, \text{formulaire commande}, 0.5, \text{bulletin commande}, 0.5\}$

‘bon commande’= 1.0., d’après l’ensemble des termes similaires suivant :
 $\text{Sim-Term}(\text{bulletin commande}) = \{\text{bulletin commande}, 1.0, \text{bon commande}, 1.0, \text{formulaire commande}, 0.5, \text{liste commande}, 0.5\}$

D’après la règle ci-dessus $\text{DSRs} = ((1.0+0.0+1.0)/3=0.67)$,

Pour calculer le degré de similarité entre les rôles nous proposons la formule suivante :

$$RLT = \left(\sum_{e=1}^l RL_e \right) / l$$

RLT : Le degré de similarité total entre les rôles

RL_e : Le degré de similarité du nom du rôle

e : index du rôle

l : le nombre des rôles

Équation V : La formul de calcule la similarité des rôles

Le modèle entré (de l’exemple de ‘l’achat électronique’) contient un seul rôle, ‘vendeur’. Son degré de similarité est 1.0, d’après l’ensemble des termes similaires suivant : $\text{Sim-Term}(\text{vendeur}) = \{\text{homme affaire}, 0.8, \text{vendeur}, 1.0, \text{assistant vente}, 0.5\}$.

3) Calculer la similarité de dépendance entre les tâches (DSDT)

Le degré de similarité de dépendance entre les tâches est calculé par la comparaison de la distance (transition) et du type de dépendance (connecteur) entre deux tâches de modèle entré (input) avec celle de dépendance entre les deux tâches correspondantes d’un cas de modèle de workflow, comme le montre l’équation suivante. par conséquent, pour avoir le degré de similarité de dépendance entre les tâches, il faut faire la correspondance de deux tâches du workflow candidat et le workflow problème.

$$Pd_T = \sum_{i=1}^{i=m} Pd_i / m$$

Pd_T : Le degré de similarité totale des distances des dépendances.

Pd_i : Le degré de similarité de la distance de dépendance i

i : index de la dépendance

m : le nombre de distance de dépendance du problème

Équation VI : Formule de calcul de similarité de distance de dépendance

3.a) Calculer le degré de similarité de distance des dépendances (DSDD)

Une distance de dépendance correspond à une transition (entre deux tâches); cette dernière est définie dans le chapitre 3 ; dans cette section, nous montrons l'utilité de cet élément. Le degré de similarité de distance d'une dépendance spécifique est décidé par : combien la similarité de distance d'une dépendance spécifique du modèle initial est correspond avec une seule dépendance dans le modèle candidat. Il est calculé par le nombre des dépendances (transitions) entre les deux tâches, et s'il y'a plusieurs chemins entre deux tâches, le plus court chemin est choisi comme la dépendance correspondante.

Par exemple, dans la Figure 5.16, il y'a deux chemins de A à B [$A \Rightarrow C \Rightarrow D \Rightarrow B$] et [$A \Rightarrow C \Rightarrow B$], mais ce dernier chemin est le chemin choisi comme dépendance correspond au problème, par ce que c'est le plus petit chemin. Ainsi, la distance de dépendance entre la tâche A et la tâche B de la solution égale 2, et son degré de similarité est 0.8, d'après le Tableau 5.2, ce dernier contient les distances possibles et le degré de similarité correspond.

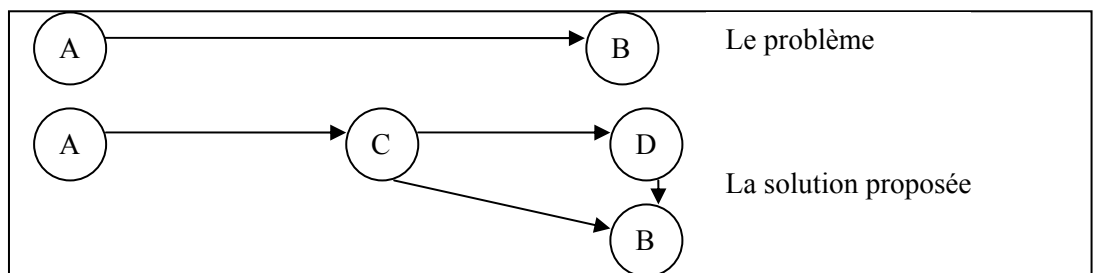


Figure 5.16 : Exemple de distance de la dépendance

La Distance	1	2	3	4	5	6 et plus
Le Degré	1	0.8	0.6	0.4	0.2	0

Tableau 5.2 : Définition du degré de similarité de distance de la dépendance

Pour illustrer la règle de similarité de distance, nous utilisons l'exemple précédent. Le cas de workflow inclus deux chemins à partir de la tâche 'collecter commande' à la tâche 'établir bon commande' ; ['recevoir commande'=>'recevoir chèque'=>'enregistrer client'=>'établir bon commande'] et ['collecter commande'=>'recevoir chèque'=>'établir spécification commande']. La dépendance choisie sera le deuxième chemin, par ce que c'est le chemin le plus court, et son degré de similarité est 0.8 d'après le tableau précédent.

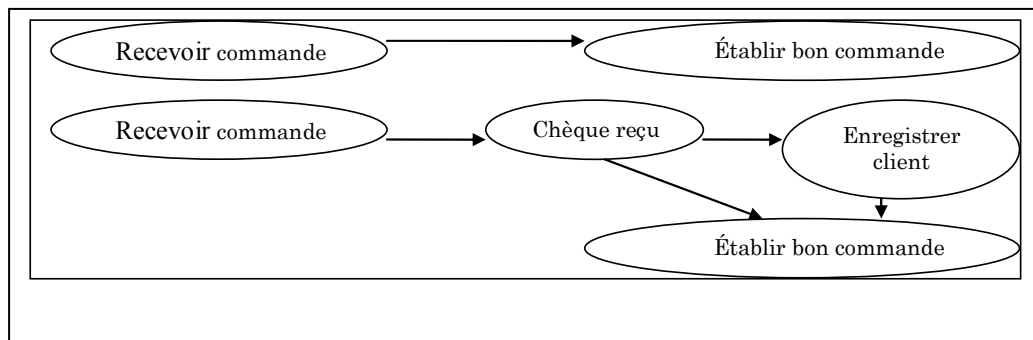


Figure 5.17 : Un exemple de distance de dépendance entre les tâches

3.b) Calculer le degré de similarité de type des dépendances (DSTD)

Le type de dépendance correspond aux structures de contrôle définies dans le chapitre 3 (circulation, branchement et synchronisation). Cette section montre l'utilisation de ces éléments pour calculer la similarité entre deux workflows. La formule de calculer la similarité totale de type de dépendance est la suivante :

$$Ptr = \sum_{i=1}^{i=m} Pt_i / m$$

Pt_T : Le degré de similarité totale des types des dépendances du problème

Pt_i : Le degré de similarité du type de dépendance i

i : index de la dépendance

m : le nombre de type de la dépendance du problème

Équation VII : Formule de calcul de similarité de type de dépendance

Le degré de similarité de type de la dépendance inter-tâches est calculé par l'application des résultats du Tableau 5.3, dans ce dernier, nous avons défini le degré de similarité de chaque type de la dépendance existe entre deux tâches, comme nous avons les

défini dans le chapitre 3, si deux dépendances appartient au même catégorie le degré de similarité sera égale à 1 si non 0.

		Synchronisation		Branchement		Circulation	
		Or-join	And-join	Or-Split	And-Split	séquence	Itératif
Synchronisation	Or-Join	1	0.5	0	0	0	0
	And-join	0.5	1	0	0	0	0
Branchement	Or-Split	0	0	1	0.5	0	0
	And-Split	0	0	0.5	1	0	0
Circulation	Séquence	0	0	0	0	1	0.5
	Itératif	0	0	0	0	0.5	1

Tableau 5.3 : Degré de similarité de type de dépendance.

- Dans notre modèle entré le type de dépendance entre la tâche : ‘recevoir commande’ et la tâche ‘recevoir chèque’ est ‘séquence’ ; et le type de dépendance entre les tâches du modèle proposé: ‘confirmer commande’ et ‘payer chèque’ est aussi ‘séquence’, d'où le degré de type de dépendance est 1.0, et la distance de dépendance est 1.0.
- Le type de la dépendance entre les tâches : ‘recevoir chèque’ et ‘établir bon commande’ est différent avec le type de la dépendance correspondre du cas sélectionné, donc la similarité du type et de la distance est 0.0.

En appliquant les règles définis , nous avons trouvé les résultats suivants:

$$DSDD = ((1.0+0.0)/2=0.5)$$

$$DDTD = ((1.0+0.0)/2=0.5)$$

3.c) Calculer la similarité totale

L'équation suivante nous montre, la règle que nous avons défini pour calculer le degré de la similarité totale. Après le calcul de DSNT, DSTT, DSRs, DSRI, DSDD et DSTD, nous pouvons maintenant juger que deux modèles de workflow sont similaires ou non, par ce que la similarité entre deux workflows ne dépend pas seulement à la similarité

des tâches, des rôles et des ressources ; en plus ; la similarité entre les dépendances n'est pas un facteur de décision, c'est juste un facteur additionnel qui augmente le degré de la similarité totale ; cela nous oblige d'intégrer tout les facteurs, pour calculer la similarité.

$$ST = \frac{(cf1*DSNT+cf2*DSTT+cf3*DSRs+cf4*DSRI+cf5*DSDD+cf6*DSTD)}{cf1+cf2+cf3+cf4+cf5+cf6}$$

ST= la Similarité Totale

DSNT : Le Degré de Similarité total des Noms des Tâches du problème

DSTT : Le Degré de Similarité total des Types des Tâches du problème

DSRs: Le Degré de Similarité total des Ressources du problème

DSRI : Le Degré de Similarité total des Rôles du problème

DSDD : Le Degré de Similarité totale des Distances des Dépendances du problème

DSTD : Le Degré de Similarité totale des Types des Dépendances du problème

Cf(1..6) : Coefficient défini par l'utilisateur

Équation VIII : La formule de calcul la Similarité Totale

Pour notre modèle entré, nous avons pris $cf1=cf2=cf3=cf4=cf5=cf6=1$, donc la similarité totale est $ST=(0.93+0.63+0.67+1.0+0.5+0.5)/6=0.70$.

5.6.1.2.3 Le module 'Sélectionner'

Le but de ce module est de sélectionner un cas pour proposer une solution à l'utilisateur. Pour le réaliser, nous proposons les modèles trouvés pendant la phase de recherche par ordre décroissant de leurs degrés de similarité total, et l'utilisateur sélectionne le modèle solution, ou le système lui propose un modèle. Si la similarité totale entre deux modèles est inférieure au seuil (défini par l'utilisateur), c'est-à-dire que le problème est un nouveau workflow il faut le modéliser par la méthode 'feuille propre'.

La figure suivante présente la similarité totale mesurée entre notre modèle entré et quatre cas de modèle dans la base de cas. Le résultat est ordonné par ordre décroissant de degré de similarité totale.

Num de Cas	Nom de Cas	Total	Tach	Document	Role	Distance	Type
	TP_Telemarketing	0.72	0.93	0.67	1.00	0.50	0.50
WF : 2	KP_Discount	0.66	0.82	0.56	0.83	0.50	0.58
WF : 3	SS_Life_insurance	0.58	0.70	0.49	0.81	0.44	0.48
WF : 4	LG_Cridet_Card	0.52	0.66	0.58	0.69	0.30	0.38
WF : 5							
WF : 6							

Figure 5.18 : L'interface utilisateur pour les cas candidats

5.6.1.2.4 Le module 'Adapter'

Si le cas sélectionné ne s'adapte pas exactement aux besoins de l'utilisateur, il faut le comparer avec le modèle initial pour l'améliorer. Nous proposons les étapes suivantes pour la comparaison.

Premièrement, les composants de modélisation qui sont déterminés à l'origine par l'utilisateur ; le modèle initial est mieux adapté pour répondre au besoins utilisateur que le cas sélectionné. Par exemple, dans la Figure 5.19 l'utilisateur sélectionne le nom de la tâche du modèle entré, 'recevoir une commande' plutôt que celle du modèle trouvé 'confirme la commande'.

Deuxièmement, les composants qui sont déterminés par la politique de travail (c'est la règle appliquée pour obtenir des résultats satisfaisants) sont adoptés à la base du modèle sélectionné (exemple : la dépendance entre les tâches). Comme le montre Figure 5.19, l'utilisateur sélectionne les dépendances entre les tâches et les tâches liées anticipé et adapte les au politique de travail.

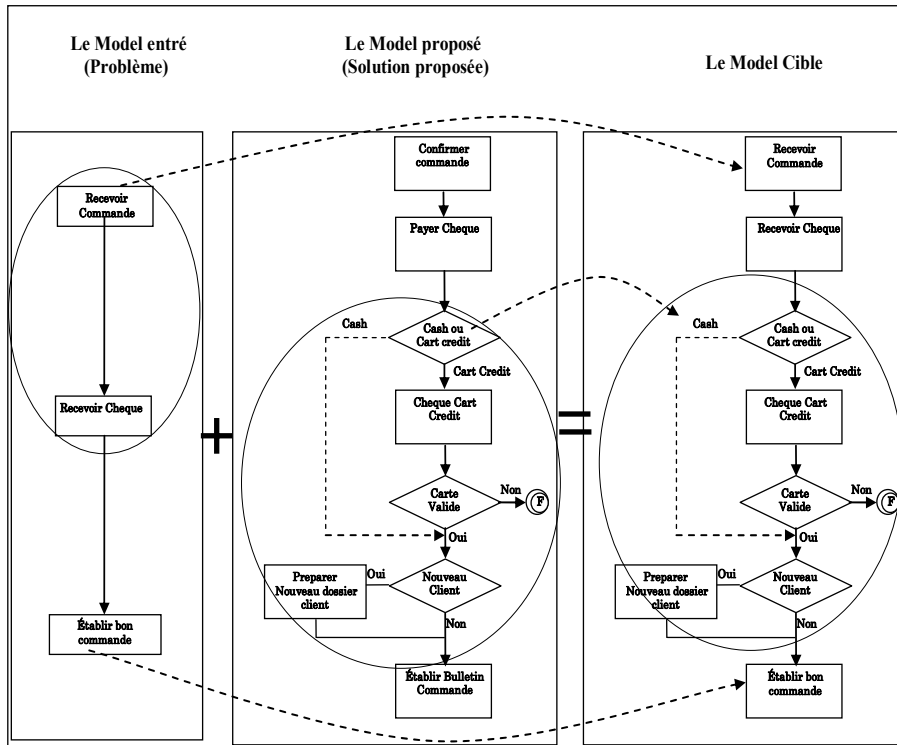


Figure 5.19 : Un exemple d'adaptation d'un cas

Troisièmement, pour les composants qui demandent une réflexion comme l'ensemble des termes similaires, les deux modèles (entré et sélectionné) sont combinés. Par exemple, dans le cas de l'ensemble de terme similaire de la Figure 5.20, le nom de la tâche du modèle entré ('recevoir une commande') et celle du modèle sélectionné ('confirmer la commande') sont combinés pour devenir un ensemble des termes similaires de cette tâche.

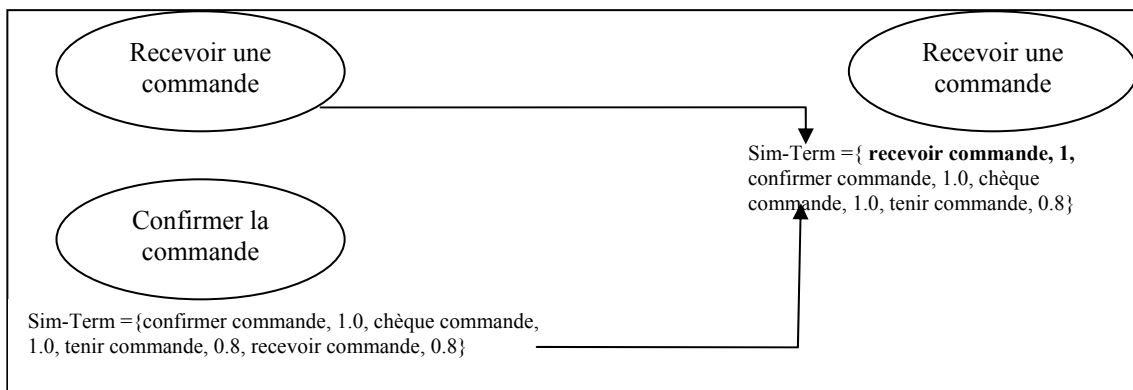


Figure 5.20 : Exemple d'adaptation d'un ensemble des termes similaires

5.6.1.2.5 Le module Mémoriser

Une fois l'étape de l'adaptation terminée, on peut définir les nouveaux cas à intégrer dans la base de cas. Par conséquent, pour manager tous les modèles de workflow (pour une utilisation future) dans la base de cas, les modèles workflows sont organisés par la hiérarchie Dépendance de domaine, et catégorisés par l'utilisation des mécanismes : spécialisation et héritage. Un workflow est généré par l'héritage des workflows existants. Et pour ajouter un nouveau cas, la spécialisation aide à trouver l'alternative du workflow qui représente la nouvelle solution. Afin de sauvegarder un workflow dans la base de cas, nous utilisons la hiérarchie Dépendance de domaine présentée dans la Figure 5.5.

Après la représentation de notre approche de conception de la plateforme BPR/CBR, nous exposons dans la figure suivante l'architecture détaillée du module CBR.

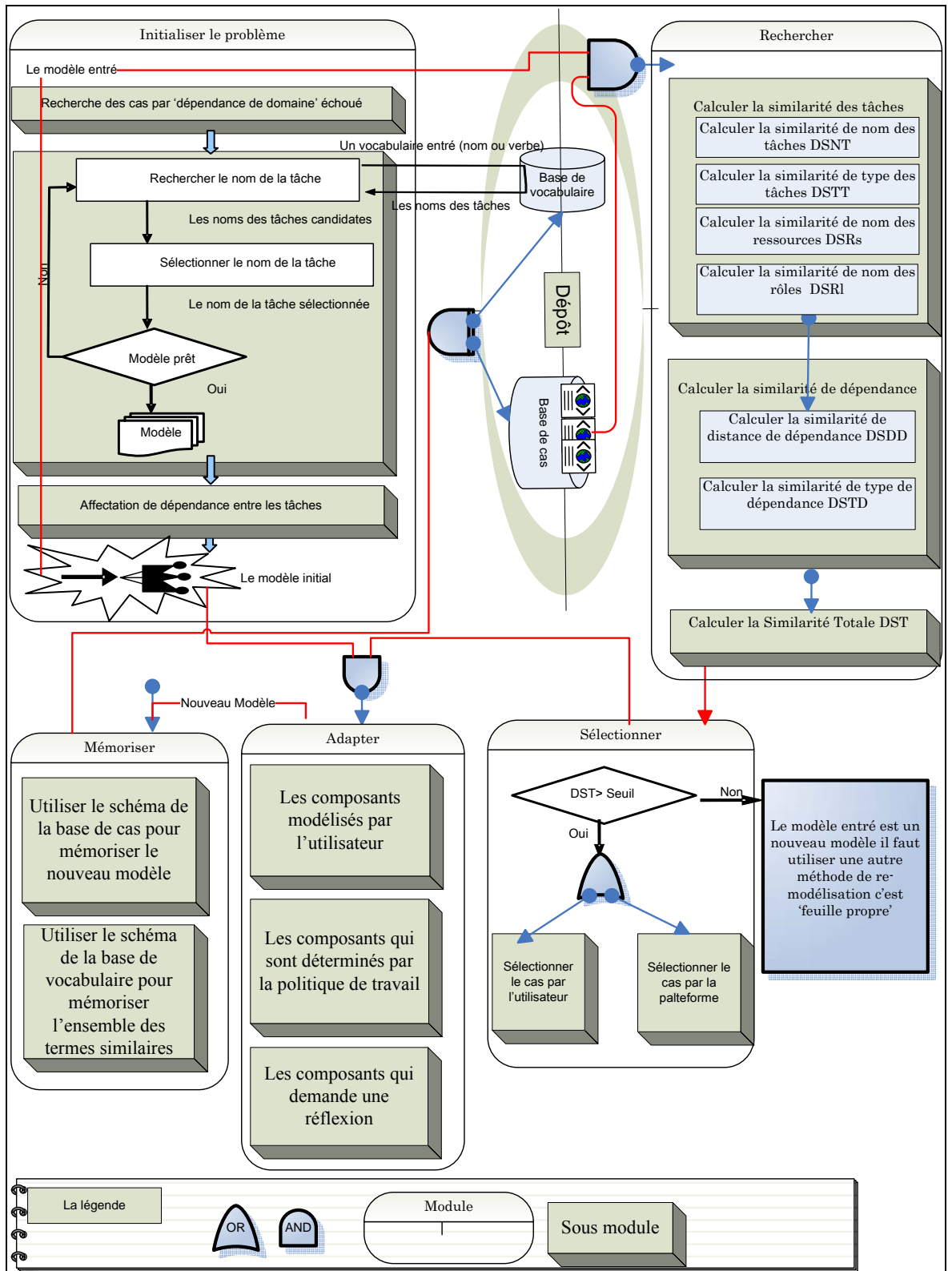


Figure 5.21 : L'architecture du module 'case based reasoning'

5.7 Le prototype BPR/CBR

Les études présentées dans ce chapitre et les chapitres précédents, ont été menées suffisamment en détail pour qu'un prototype soit réalisé. Le prototype BPR/CBR a permis de tester et de valider nos propositions, et aussi de mettre en évidence leurs avantages et leurs points faibles. La démarche mise au point par notre prototype se décompose en quatre étapes :

- ❖ Tout d'abord le processus métier est re-modélisé par un expert métier dans le formalisme proposé (chapitre 3). Cette description du processus est réalisée à l'aide d'un modéleur de processus graphique, par une des méthodes de re-modélisation présentées avant.
- ❖ Ensuite, une fois le modèle stabilisé, la spécification qui lui correspond est générée grâce au module Scriptor (un outil de génération développé par BPR/CBR). Cette génération est entièrement automatisée et donc immédiate et totalement transparente pour l'utilisateur.
- ❖ Puis le workflow généré stocké dans la base de cas pour d'autres utilisations.
- ❖ Enfin, le workflow peut être exécuté à l'aide d'un moteur de workflow. Cette étape n'est pas prise en compte dans notre projet.

Afin de réaliser ces fonctions nous avons implémenté un prototype de système pour réaliser le framework proposé. Nous définissons par suite, les fonctions principales accomplies par notre prototype.

5.7.1 Menu

Les fonctions du système sont catégorisées en huit main-menus. Les deux menus majeurs (menu 'cas' et menu 'vocabulaire') correspondent aux fonctions de l'architecture des couches: Gestion de la base de cas, Gestion de vocabulaire. Pour les fonctions de support, cinq menus sont activés dans le prototype proposé (Fichier, Edition, Format, Affichage, Windows, Aide et le menu A propos). Un résumé de bar de menu est montré dans le tableau suivant.

Menu Principal	Sous-menu	Description
Fichier	Nouveau (Ctrl+N)	Crée un nouveau fichier
	Ouvrir (Ctrl+O)	Ouvre un fichier existant
	Fermer	Ferme le fichier en cours
	Enregistrer (Ctrl+S)	Enregistre le fichier en cours
	Enregistrer sous	Enregistre le fichier en cours sous un autre nom
	Mise en page	Mettre le canevas en portrait ou en paysage
	Aperçu avant impression	Aperçu du fichier en cours
	Imprimer (Ctrl+P)	Imprime le fichier en cours
	Quitter	Sort de l'application en cours
Edition	Annuler (Ctrl+Z)	Annule la tâche précédente
	Couper (Ctrl+X)	Déplace l'objet(s) sélectionné(s) dans le presse-papier.
	Copier (Ctrl+C)	Copie l'objet(s) sélectionné(s) dans le presse-papier.
	Coller (Ctrl+V)	Colle l'objet(s) coupé(s) dans la zone sélectionnée.
	Sélectionner tout (Ctrl+A)	Sélectionne tout les objets dans le fichier en cours
	Effacer (Del)	Efface l'objet sélectionné
Format	Police	Spécifie la police utilisée dans l'objet sélectionné
	Alignement	Aligne l'objet sélectionné
Gestion des cas	Rechercher des cas	Parcourt les cas dans la base des cas
	Récupérer des cas	Récupère des cas similaires à partir de la base des cas
	Effacer un cas	Efface le cas en cours dans l'écran
	Ajouter un rôle	Ajoute l'élément rôle dans le canevas
	Ajouter une ressource	Ajoute l'élément ressource dans le canevas
	Ajouter une tâche	Ajoute l'élément tâche dans le canevas
	Ajouter un agent	Ajoute l'élément Agent dans le canevas

	Ajouter une dépendance entre les tâches	Ajoute l'élément dépendance entre les tâches dans le canevas
Gestion du vocabulaire	Synonymes	Gère les synonymes
	Noms des tâches	Gère les noms des tâches
	Noms du Workflow	Gère les noms du Workflow
Fenêtres	/	Arrange la fenêtre active dans des styles variés : cascade, horizontale, verticale.
Aide	/	Fournit la fonction d'aide
A propos	/	Montre les informations a propos de la version, du copyright et autres.

Table 5.2: Résumé de bar de menu

5.7.2 Gestion de la base de cas

Les fonctions de gestion de la base de cas consistent à parcourir, ajouter, supprimer, changer et à sauvegarder les cas. L'utilisateur sélectionne le menu 'Gestion de Cas' dans la barre de menu, la forme 'Gestion des Cas' est apparue comme le montre la Figure 5.22. Dans le contrôle l'arbre d'affichage (TreeView) de la partie gauche, l'utilisateur peut parcourir tout les cas de la base, par un double click sur le nœud. Quand il atteint le nœud désiré, la liste des workflows qui appartient à cette fonction montrera dans le contrôle ListControl 'Workflows' dans la partie droite. S'il click deux fois sur un workflow, la forme 'Cas de Workflow' est apparue pour montrer le modèle graphique de workflow qui lui correspond (Figure 5.23). On utilisant la barre d'outil pour créer des éléments de modèle, l'utilisateur peut adapter ce modèle de workflow par l'addition, suppression et modification des éléments de modèle selon les besoins. Si l'utilisateur veut supprimer un élément de modèle, il sélectionne simplement un ou plusieurs éléments de modèle et après appuie sur le bouton 'Supprimer'. Quand l'utilisateur termine l'adaptation, il peut sauvegarder le modèle de workflow adapté comme un nouveau cas en appuyant sur le bouton 'Enregistrer'.

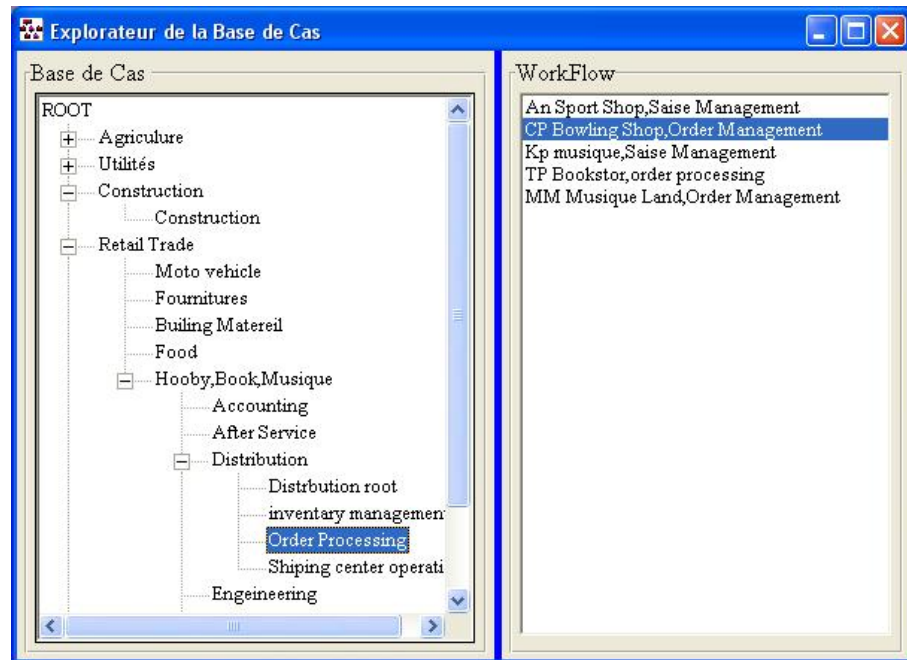


Figure 5.22 : L'interface utilisateur pour la gestion des cas

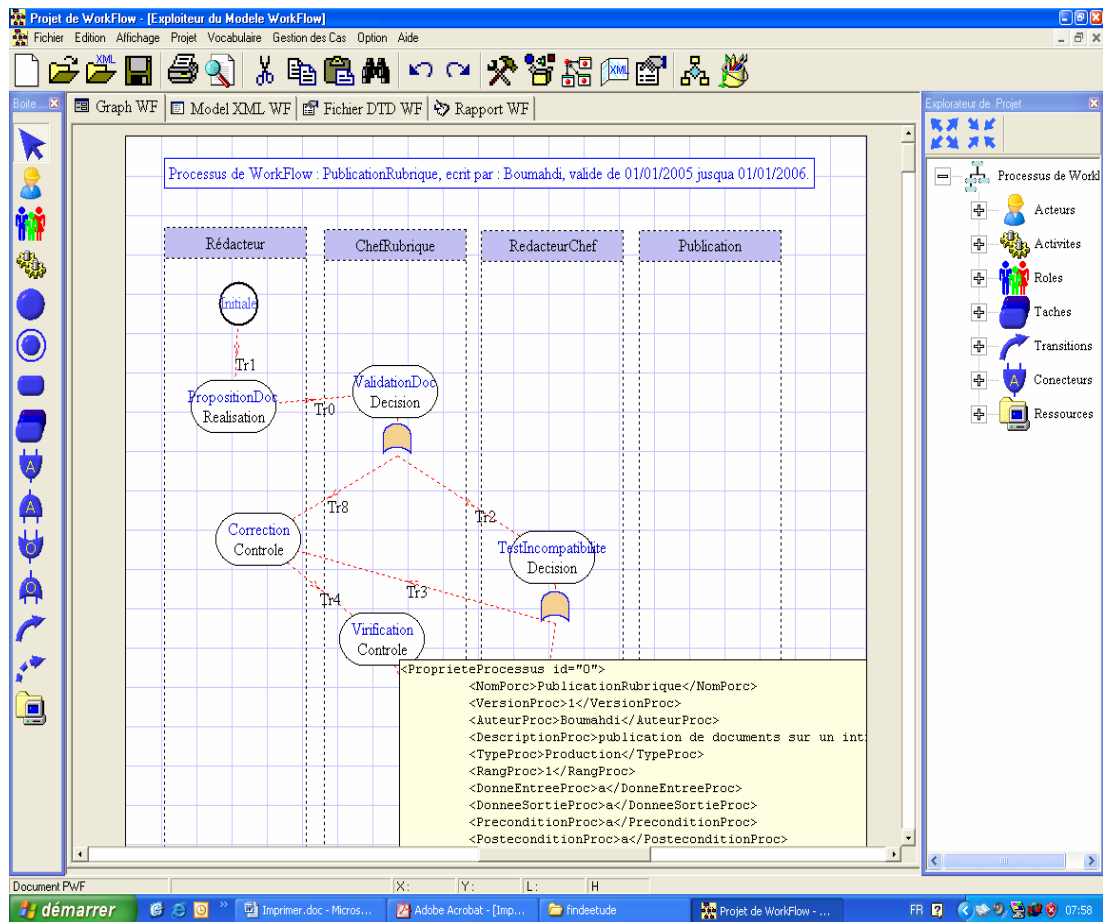


Figure 5.23 : L'interface utilisateur pour un cas de workflow

5.7.3 Gestion de la base de vocabulaire

Les fonctions de gestion de la base de vocabulaire sont : parcourir, création, suppression, changement et enregistrement de vocabulaire des noms des tâches et noms des workflow.

Ces fonctions supportent le processus de CBR par la préparation des correspondances verbe/nom pour le modèle entré de processus initial et la gestion de l'ensemble de termes similaires pour le cas de processus sélectionné.

5.8 Conclusion

Nous avons présenté dans ce chapitre notre approche pour résoudre le problème de re-modélisation du workflow. Cette approche repose sur la technique de raisonnement à base de cas textuel, permettant d'exploiter une base de cas antécédents et de proposer des solutions à des nouveaux problèmes.

Dans ce travail, nous avons décrit une plateforme à la base de Règles à Partir de Cas (RàPC) pour aider la réutilisation des connaissances en BPR. À l'aide de celui-ci, l'expert en BPR peut rechercher un modèle existant qui résout un problème proche du sien et l'adapter à la nouvelle situation. Il peut ainsi réutiliser ses propres connaissances ou celles qu'un autre expert en BPR aura modélisées au préalable, réalisant alors un « partage des connaissances ». BPR/CBR est un outil d'aide à la re-modélisation d'un processus entreprise. Ce prototype nous a permis de valider notre approche basée sur l'utilisation du Règles à Partir de Cas. Nous l'avons testé avec succès sur des applications de petites tailles. Les caractéristiques du prototype BPR/CBR proposé consistent aux éléments suivants :

- La capacité d'établir des définitions de processus (nom, description, date version, composants, ...etc.) ;
- La capacité de modéliser le cheminement séquentiel, parallèle, branchement, et itératif au moyen des composants graphiques tels que AndSplit, OrSplit, AndJoin, OrJoin ;
- Support de gestion de la version (il peut y avoir plusieurs versions de même processus) ;

- La définition des attributs utilisés dans le processus ;
- Spécifications des tâches ; et
- la vérification de l'exactitude (syntactique) d'une définition de processus (cas) et le traçage de toutes omissions ou contradictions.

CONCLUSION

L'enjeu de cette thèse consiste à justifier, proposer et expérimenter l'utilisation de CBR dans le cadre de la gestion des connaissances workflows.

Les travaux de recherche présentés dans ce mémoire ont porté sur la re-modélisation des workflows à partir d'une démarche de Gestion des connaissances. Nous avons traité cette problématique de manière conceptuelle et cherché à poser les premiers fondements d'une architecture de la plateforme BPR/CBR dans l'entreprise. Notre plateforme a été testée sur plusieurs applications de petite taille. Les résultats obtenus montrent que le système BPR/CBR a des propriétés intéressantes et de nombreux avantages.

Deux objectifs principaux sous-tendent notre démarche : une aide à la résolution des problèmes et un enrichissement incrémental de la base de cas. En parallèle à ces deux objectifs, deux hypothèses nous ont guidés : il existe une base de cas et une base de vocabulaire du domaine incomplètes, et l'utilisateur peut exprimer une partie du workflow à re-modéliser à l'aide des tâches existantes.

Pour aider l'utilisateur dans la phase de re-modélisation, une notation de workflow sous forme de décomposition en tâches est proposée. Ainsi, la re-modélisation d'un workflow peut être effectuée par trois approches différentes : par « feuille propre », « la dépendance de domaine » et par « CBR ». Ceci permet aussi de faciliter l'acquisition des connaissances, puisque l'utilisateur a à sa disposition des cas similaires et qu'il peut les combiner et les compléter pour résoudre des nouveaux problèmes.

L'objectif global de ce travail était de faciliter la re-modélisation des processus métiers par des techniques plus efficaces et mieux adaptées aux perceptions des utilisateurs. Nous résumons comment cet objectif a été atteint par les propositions

concrètes de cette thèse et plus généralement par les apports de l'approche à base de CBR. Enfin, nous donnons quelques perspectives à notre travail.

Dans le contexte de BPR, le CBR peut être appliqué pour aider le processus de re-modélisation. D'un autre côté, la technique CBR peut servir à réutiliser efficacement les méthodes de re-conception afin d'essayer d'améliorer l'exécution du workflow. La plateforme proposée et l'outil de CBR sont des stratégies de support de transfert de la connaissance qui aident la re-modélisation de processus métier.

S'accumulant les cas dans l'outil de CBR aide à organiser, restructurer et apprendre la connaissance acquise après la re-conception d'un processus. En effet, Ce travail actuel fournit à toute organisation qui veut re-modéliser ses processus un outil qui soutient la création, le partage et le transfert de la connaissance.

Notre travail de recherche a été initialisé par l'étude de la relation qui existe entre les trois technologies : BPR, KM et le CBR. Les approches traditionnelles de résoudre des problèmes de re-conception de workflow sont peu adaptées [Selma & all, 03]; la dernière approche proposée est le KM, elle consiste à réutiliser l'expérience et les bonnes pratiques connus pour l'élaboration d'un nouveau processus métier. Aussi avons-nous centré notre mémoire sur une adaptation des principes d'une nouvelle approche de l'ingénierie des connaissances : l'approche à base de CBR. Afin de mettre en relief les principales contributions de nos travaux, nous rappelons les différentes étapes de notre démonstration.

Nous avons montré dans la première partie pourquoi et comment il est possible d'intégrer toutes les technologies KM et CBR dans un système complet et cohérent, pour la re-modélisation des workflows.

Le premier chapitre nous a permis de revenir aux origines de la re-modélisation des workflows en tant que problématique. Cela nous permettra de trouver un certain nombre de points communs et de convergences avec la Gestion des Connaissances. Ceci fait, nous nous intéresserons aux liens entre le workflow et la gestion des connaissances.

Dans un second chapitre, nous avons identifié les étapes de réalisation d'une architecture générique d'une plateforme de Gestion des Connaissances workflow à base de

CBR. Dans un premier temps, nous avons extraire la synergie et la relation qui existe entre le KM et le CBR. Puis, nous avons réalisé un état de l'art de la méthode CBR. A partir de l'analyse de cette étude, nous avons mis en relief les étapes essentielles pour concevoir une plateforme à base de CBR.

Nous avons, pour finir, illustré dans la deuxième partie, la conception de la plateforme BPR/CBR proposée aux chapitres 3, 4 et 5, en appuyant sur les correspondances établies au chapitre II. Pour cela, nous avons détaillé notre méthode de réalisation de toutes les étapes du CBR : depuis « Acquisition de connaissance de domaine » jusqu'à « Implémentation du système ».

Les chapitres 3 et 4 nous ont permis de réaliser la première et la deuxième étape de conception de la plateforme BPR/CBR. Les résultats obtenus dans ces chapitres seront utilisés dans le chapitre V pour décrire les composants de BPR/CBR. Nous avons également proposé, pour chaque problème du système CBR, une solution nouvelle et originale.

Nous avons présenté notre approche pour résoudre le problème de gestion des connaissances workflow. Cette approche repose sur la technique de raisonnement à base de cas textuel permettant d'exploiter une base des cas antécédents et de proposer des réponses. Les contributions originales que nous avons apporté se situent principalement au niveau de l'acquisition des connaissances et l'architecture de la plateforme, ces contributions sont résumées en :

- Le méta-modèle proposé est basé sur les travaux de workflow déjà réalisés, nous avons également ajouté des critères de workflow pour nous aider dans la mémorisation des cas, et notre principale motivation est la description graphique d'un workflow que nous avons pris en considération dans la représentation des cas.
- Le Modèle de workflow proposé est une description abstraite et de haut niveau de la coordination, puisqu'il est graphique. Notre description de haut niveau permet, même à des non-spécialistes, de comprendre facilement la coordination modélisée, de la modifier et de la faire évoluer. D'autre part, la modélisation sous forme de diagramme d'activité permet de représenter toutes les étapes de traitement d'un processus métier.

La présentation des différentes fonctionnalités à l'aide d'une interface graphique conviviale améliore la coopération entre le système et l'utilisateur. Ce dernier peut créer ses workflows d'une façon facile.

- La base de cas : dans ce travail, nous avons proposé la hiérarchie dépendance de domaine de la base de cas pour aider l'entreprise à développer des modèles de workflow. Ces modèles sont partageables entre les acteurs métier de l'entreprise.

- La base de vocabulaire : la remémoration d'un cas se base sur une similarité, qui peut être vue comme un ensemble de règles. Cela nous a conduit à rechercher les cas similaires d'un problème sémantiquement, notre idée dans ce contexte est de construire une base de vocabulaire.

- Le processus de re-mémorisation de cas est important pour la performance du système et la qualité des solutions obtenues. Nous avons proposé des règles de recherche des cas similaires, les règles proposées basées sur les critères « tâche - rôle - ressource ».

- Pour ce qui est de l'adaptation, les approches actuelles de CBR textuel ne proposent pas des techniques dans ce sens [41]. Par contre, il est crucial pour notre application de pouvoir réutiliser et modifier des modèles antérieurs. D'où notre motivation de défricher la voie de l'adaptation des cas textuels de workflow, pour l'identification des portions de workflow à adapter.

- Au niveau de la plateforme BPR/CBR nous avons réalisé un prototype interactif destiné à aider un expert en BPR à re-modéliser ses modèles workflow, dans un premier temps pour les capitaliser, et ensuite pour les réutiliser. Grâce à l'interactivité du système, l'acquisition des connaissances s'effectue directement auprès de l'expert, et ainsi l'intégration progressive des modèles est réalisée au fur et à mesure de leur apparition.

Le prototype BPR/CBR dispose :

- d'un outil de représentation graphique de modèles de workflow qui permet l'expression des règles ou des contraintes à appliquer sur le modèle. La simplicité et la convivialité de cet outil autorisent une utilisation par un large public,

- Des solutions de transformation de modèles s'appuyant sur le standard XML: formalisation des règles de transformation, l'exécution de ces règles via le module

‘scriptor’ permettant de transformer un modèle graphique (issu de la couche MODELER) en un modèle textuel (dans la base de cas) ou inversement.

Les propositions et les expérimentations exposées dans ce mémoire marquent des étapes dans notre projet de recherche ; que nous souhaitons poursuivre à court terme et à long terme sur les aspects suivants :

- Pour restreindre l’étendue du problème, les tests ont été réalisés sur des petites applications. Il sera intéressant de diversifier le contenu de nos bases (base de vocabulaire et base de cas), en intégrant des applications réalisant des traitements plus diversifiés (allant de la restauration à l’interprétation...etc.), et s’appliquant à des processus de différents domaines. Ceci nous permettra également d’enrichir le vocabulaire utilisé pour la description des cas, de compléter notre ensemble de critères et de proposer à l’utilisateur des listes de termes plus exhaustives.
- La validation : consiste à apporter la preuve que ce système réalise effectivement ce qu’il prétend réaliser. Dans l’application réelle, il est nécessaire d’affiner la base de cas et la base de vocabulaire actuelles.
- Le passage du prototype BPR/CBR à un système totalement utilisable dans le service où il est appliqué. Pour cela, plusieurs axes doivent être développés :
- Architecture réseau : Pour améliorer l’efficacité de notre architecture, il serait intéressant d’étendre notre plateforme BPR/CBR afin de la rendre rapide, robuste, efficace et multi-utilisateurs et enfin, adopter une architecture client/serveur ou intranet.
- La base de vocabulaire : d’autres perspectives à plus longue échéance s’inscrivent dans la continuité de ce travail. Poursuivant l’objectif de faciliter l’acquisition des connaissances, il est intéressant d’initialiser la base de vocabulaire. Pour améliorer l’efficacité de notre architecture, il serait intéressant de construire la base de vocabulaire par une ontologie.

- L'ajout d'un module d'explication : BPR/CBR effectue son raisonnement en choisissant le(s) cas le plus similaire, et en transférant la solution au problème actuel. Il serait utile d'ajouter un module d'explication qui explique à l'utilisateur pourquoi une telle solution a été proposée par le système.
- Le développement d'une interface graphique plus conviviale pour le système BPR/CBR : par rapport aux travaux actuellement opérationnels, plusieurs travaux techniques sont indispensables. D'une part, du point de vue ergonomique, il est indispensable de proposer une interface graphique permettant à l'utilisateur de : couper/coller les tâches des cas proposés, créer de nouvelles tâches et les insérer dans la base existante ... etc. Cette interface a été construite mais elle demande plus des fonctions interactives.
- Nous devons également envisager la réalisation d'un moteur de workflow se charge d'exécuter les modèles workflow. À partir des règles définies dans le modèle de processus, l'exécution du processus est contrôlée de façon automatisée par le moteur de workflow. Ce dernier se charge de distribuer les tâches aux personnes chargées de leur réalisation et d'effectuer lui-même quelques actions automatiques (contrôle l'ordre d'exécution,...etc.).

De plus, les perspectives d'ouverture de ce travail et en particulier dans le cadre de raisonnement à partir de cas sont :

- L'étape de révision : nous n'avons pas réalisé cette étape dans ce travail, pour réaliser la révision il faudra faire une étude sur les bonnes pratiques de re-modélisation d'un workflow.
- L'étude de changement des lois de recherche : nous pouvons envisager l'étude de l'effet de changer les lois de recherche des cas sur les performances du système (la capacité d'apprentissage). Les mesures de similarité sont fondées sur les critères : tâche, rôle et ressource. Nous pouvons donc étudier expérimentalement l'effet de changer les mesures de similarité en choisissant par exemple d'autres types de mesure (changer ou ajouter des critères de sélection).

- Le mécanisme de mémorisation, c'est une bonne technique qui soutient le partage de la connaissance et d'adaptation des meilleures pratiques, dans la re-modélisation des processus métiers, notre plateforme prend en considération cette étape, mais il faut étudier et voir leurs conséquences pour le valider.

Comme on vient de le voir par les perspectives de cette recherche, ce travail reste ouvert. Mais dans l'état actuel, il propose déjà un cadre permettant de sensibiliser les chercheurs développant des applications à l'intérêt de gérer les connaissances pour la diffusion du savoir faire.

APPENDICE A
LISTE DES SYMBOLES ET DES ABRIVIATIONS

BRP	: Business Process Redesign
CBR	: Case Based Reasoning
DOM	: Document Object Model
DTD	: Document Type Definition
GC	: Gestion des Connaissances
KM	: Knowledge Management
RàPC	: Raisonnement à Partir de Cas
SAX	: Simple Api for XML
UML	: Unified Modeling Language
XML	: Extensible Markup Language

REFERENCES

1. Aamodt, A and Plaza, E 'Case-Based Reasoning: Foundational Issues, Methodological Variations and Systems Approaches' AI Communications, Vol 7 No 1 (1994) pp 39-59.
2. Aha DW, Becerra-Fernandez I, Maurer F, Muñoz-Avila H (eds) Exploring Synergies of Knowledge Management and Case-Based Reasoning. AAAIPress, 1999.
3. Alec Sharp, Patrick McDermott, "Workflow Modeling—Tools for Process Improvement and Application Development", Teamfly, 2001.
4. Amrit Tiwana, " Knowledge Management Toolkit ", Prentice Hall, 1999.
5. Baizet Y. - La Gestion des Connaissances en Conception Application à la simulation numérique chez Renault - DIEC, Thèse de doctorat de l'université Joseph Fourier, Grenoble1, Soutenue le 12 mars 2004.
6. BEAUBOUCHER NATHALIE, ANAIS : RAISONNEMENT à PARTIR DE CAS EN RÉOLUTION DE PROBLÈMES, Thèse de doctorat université PARIS VI 1994.
7. Grady Booch, matthew fuchs, magnus cristerson UML for XML schema mapping specification, 1999.
8. BOUMAHDJ Fatima, FAREH Massaouda, 'analyse et conception d'un outil de définition de workflow', mémoire PFE, université de bilda, 2003.
9. Erwan Breton, Jean Bézivin, Un méta-modèle de gestion par les activités, 2001, sciences.univ-nantes.fr

10. Bryan Bergeron, “ Essentials Of Knowledge Management ”, John Wiley & Sons, 2003.
11. Cauchard Valérie Ficet, Réalisation d’un système d’aide à la conception d’applications de Traitement d’Images : une approche basée sur le Raisonnement à Partir de Cas. Thèse de doctorat université Caen 1999.
12. Dieng, Rose, Corby, Olivier, Giboin, Alain, Golebiowska, Joanna, Matta, Nada, Ribière, Myriam, « Méthodes et outils pour la gestion des connaissances », Paris: Dunod(2000).
13. Edward Waltz, Knowledge Management in the Intelligence Enterprise, Artech House, 2003 (Technique KM)
14. El Sawy Omar A., Redesigning Enterprise Process for e-business, édition McGraw-Hill, 2003.
15. Erik T. Ray, Learning XML, edition oreilly, 2001.
16. Fuchs B., Lieber J., Mille A., Napoli A., « Vers une théorie unifiée de l'adaptation en raisonnement à partir de cas. », Actes de RàPC99, Palaiseau.
17. Fuchs Béatrice, Représentation des connaissances pour le raisonnement à partir de cas, Thèse de doctorat université Jean Monnet de Saint-Etienne 1997.
18. Georges Gardarin, XML - Des bases de données aux services Web, edition Dunod , 2002.
19. Glance Natalie S., Pahgani Daniele S. et Pareschi Remo, “ Generalized Process Structure Grammars (GPSG) for Flexible Representations of Work,” Computer Supported Cooperative Work 96, ACM 0-89791-765-0/96/11, Cambridge MA USA, 1996.
20. Hamdah Mohamed, « modélisation de workflow dans un environnement orienté objet », mémoire de magistère USTHB, 2002/2003.

- 21.Kappel G., Rausch-Schott S., et Retschitzegger W., “ Coordination in Workflow Management Systems - A Rule-Based Approach,” Coordination Technology for Collaborative Applications, LNCS 1364, 1997.
- 22.Karl Wiig, “ People-Focused Knowledge Management ”, Elsevier, 2004.
- 23.Luc Lamontagne, RAISONNEMENT À BASE DE CAS TEXTUEL POUR LA RÉPONSE AUTOMATIQUE AU COURRIER ÉLECTRONIQUE, Rapport, Université de Montréal, 2001.
- 24.Layna Fischer, “ Workflow Handbook ”, Future Strategies, 2002.
- 25.Serge K. Levan, « le projet workflow », Eyrolles, 1999
- 26.Malhotra Yogesh. "Business Process Redesign: An Overview," IEEE Engineering Management Review, vol. 26, no. 3, Fall 1998. (URL: <http://www.kmbook.com/bpr.htm>)
- 27.Margaret May, “Business Process Management ”, Prentice Hall, 2003.
- 28.MAY MARGARET, Business Process Management, édition Prentice Hall 2003.
- 29.Michael Papazoglou, Joachim W. Schmidt, and John Mylopoulos, “ Workflow Management”, The Mit Press Cambridge, 2002.
- 30.Nurcan Selmin, “Main Concepts for cooperative Work Place Analysis”, in proceedings of the Telecooperation Conference of the 15th IFIP World Computer Congress 1998, 31 August – 4 September 1998, Vienna.
- 31.O'Leary DE , Using AI in Knowledge Management: Knowledge Bases and Ontologies. IEEE Intelligent Systems 13: 334-39, 1998.
- 32.Papazoglou Michael, Joachim W. Schmidt, and John Mylopoulos, “ Workflow Management ”, The Mit Press Cambridge, 2002.

- 33.Rao Madanmohan, “ Knowledge Management Tools And Techniques ” , Elsevier, 2005.
- 34.Reijers, Hajo A.“Design and Control of Workflow Processes”, thèse de doctorat, Eindhoven University Press Facilities, 2002
- 35.Rosina Weber, Randy Kaplan, « Knowledge-based knowledge management», University, Philadelphia, 2003
- 36.Mélissa Saadoun « Le Workflow Pour automatiser les procédures et tâches répétitives », INEDIT, 2002
- 37.Sankar K. Pal, Simon C. K. Shiu. “ Foundation Of Soft Case-Based Reasoning ”, John Wiley & Sons, 2004.
- 38.Selma Limam Mansar and Farhi Marir, “Case-Based Reasoning as a Technique for Knowledge Management in Business Process Redesign”, Eindhoven University of Technology, The Netherlands, 2003
- 39.Sorenson C, Snis U , Codify or Collaborate. From Expert Systems to Systems of Experts for Knowledge Creation in Manufacturing. In: Proc. of IRIS 23. (2000). Online: <http://iris23.htu.se/proceedings/PDF/109final.PDF>
- 40.Staab S, Studer R, Schnurr HP, Sure Y (2001) Knowledge Processes and Ontologies. IEEE Intelligent Systems 16, 1: 26-34
- 41.Sugumaran Vijayan, “ Intelligent Support Systems Technology : Knowledge Management ”, Irm Press , 2002.
- 42.Watson Ian, Applying Case-Based Reasoning: Techniques for Enterprise Systems. Morgan Kaufmann, 2000.
- 43.Watson Ian, Knowledge Management and Case-Based Reasoning: a perfect match?, AAAI Press, (2002)
- 44.Watson Ian, Applying Knowledge Management Techniques for Building Corporate Memories, Morgan Kaufmann Publishers, 2003.

45. Workflow Management Coalition Interface 1: Process Definition Interchange Process Model Document WfMC-TC-1016-P, novembre 1998, http://www.wfmc.org/standards/docs/TC-1016-P_v11_IF1_Process_definition_Interchange.pdf.

46. Berthold Daum, Udo Merten, "System Architecture with XML", édition AMAZON, 2003.

47. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, F. Lorensen, « OMT Modélisation et conception orientée objet » Prentice-Hall 1991

48. Ross, D. T. and Brackett, J. W., An approach to structured analysis. Computer Decisions 8, 9, 1976, pp. 40-44.

49. Wil van der Aalst and Kees van Hee, Workflow Management Models, Methods, and Systems, The MIT Press Cambridge Massachusetts London England, 2002

50. Kolodner, J.; (1993) "Case-Based Reasoning", Morgan Kaufmann.

51. BOOCH.G, Object oriented analysis Yourdon Press, 2ème édition 1991.

52. I. Jacobson, « OOSE Object Oriented Software Engineering », Addison Wesley 1992.

53. Dieng, Rose, Corby, Olivier, Giboin, Alain, Golebiowska, Joanna, Matta, Nada, Ribière, Myriam, « Méthodes et outils pour la gestion des connaissances », Paris: Dunod(2000).

54. S.khochafiane, M. Buckiewicz, « Groupware et Workflow », Masson, 1998

55. Marc Blau, « Gestion électronique documentaire systèmes de gestion de workflow et possibilités d'intégration de ces systèmes », diplôme postgrade en informatique et organisation, université de Lausanne, soutenue 2001-2002.

Disponible à l'adresse : <http://inforge.unil.ch>.

56. Work group 1, "Interface 1 : Process Definition Interchange Process Model, " Workflow Management Coalition, 12 Novembre 1998.

57.FREY, « La gestion de documents à travers une application Workflow », diplôme postgrade en informatique et organisation, université de Lausanne, soutenue 1999-2000.

Disponible à l'adresse : <http://inforge.unil.ch>.

58.kettani n, mignit d.paré p.,rosenthal-sabroux c, « de merise à uml », eyrolles, paris,2001