République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mentouri Constantine
Faculté des Sciences de l'Ingénieur
Département d'Informatique

N° ordre : 18/TS/2012
N° Série : O5/INF/2012

THESE

Pour obtenir le diplôme de

Docteur en Sciences

En

Informatique

Présentée par

Louardi BRADJI

Adaptation des techniques de l'Extraction des Connaissances à partir des Données (ECD) pour prendre en charge la qualité des données

Sous la direction de M.

Pr : Mahmoud BOUFAIDA

Soutenue le : 01/03/2012

Devant le jury composé de

Mr Zaidi SAHNOUN	Président	Professeur, Université Mentouri de Constantine
Mr Okba KAZAR	Examinateur	Professeur, Université de Biskra
Mme Fouzia BENCHIKHA	Examinateur	M.C.A, Université de Skikda
Mr Ramdane MAAMRI	Examinateur	M.C.A, Université Mentouri de Constantine
Mr Mahmoud BOUFAIDA	Rapporteur	Professeur, Université Mentouri de Constantine

A mon fils, Yahia

A tous les proches qui me sont chers et m'ont soutenu

A la mémoire de ma mère

Remerciements

Je ne pense pas que quelques mots de remerciements puissent suffire pour exprimer le sentiment de profonde gratitude et de reconnaissances que j'éprouve à mon directeur de thèse Monsieur Mahmoud BOUFAIDA, Professeur à l'Université de Mentouri de Constantine (Algérie), pour m'avoir encadré avec diligence, disponibilité totale et une clairvoyance remarquable pour ces travaux. Qu'ils trouvent ici mes remerciements les plus sincères.

Je remercie Monsieur Zaidi SAHNOUN, Professeur à l'Université Mentouri de Constantine, pour l'honneur qu'il me fait d'avoir accepté d'être le président du jury.

Je remercie également Monsieur Okba KAZAR, Professeur à l'Université de Biskra (Algérie), Monsieur Ramdane MAAMRI, Maître de Conférences à l'Université de Mentouri de Constantine, et, Mme Fouzia BENCHIKHA, Maître de Conférences à l'Université de Skikda (Algérie) d'avoir accepté d'être les examinateurs de ma thèse. Je les remercie pour l'attention avec laquelle ils ont lu et évalué ce travail.

Je remercie également ma seconde famille dans laquelle je pense avoir réussi à trouver ma place ... Il s'agit du l'équipe SIBC du laboratoire LIRE. Permanents ou associés, chercheurs-enseignants ou secrétaires, encore là ou partis... toutes ces personnes que j'ai côtoyées et qui, d'un point de vue scientifique ou relationnel, de prés ou de loin, m'ont apporté leur aide ou leur soutien. Je fais le choix de ne pas les citer de peur d'en oublier. Mais je tiens tout de même à remercier nominativement Madame Zizette BOUFAIDA et Monsieur Nacereddine ZAAROUR, Professeurs à l'Université de Constantine.

Merci à mon père, mon fils Yahia et ma femme. Je leur éternellement reconnaissant de m'avoir assuré de leur confiance, de leurs encouragements et de leur indéfectible soutien sans borne au cours des années de ma thèse. J'ai également une pensée affectueuse pour mes frères et mes sœurs.

Table de matières

Introduction générale

Introduction générale…… 1

Chapitre 1 : Qualité des données et des connaissances

1. Introduction…… 9

2. Donnée, information et connaissance…….. 9

3. Qualité et concepts associées…….. 10

4. Qualité des données…… 11

4.1. Principaux problèmes de la non qualité des données…….. 11

4.2. Etude des principaux travaux sur la qualité des données…….. 12

4.2.1. Principales problématiques de recherche…… 12

4.2.2. Classification des travaux de recherche …… 14

4.2.3. Principales méthodologies de la gestion de la qualité des données…….. 14

4.2.4. Modélisation de la qualité des données…….. 15

4.2.5. Évaluation et amélioration de la qualité des données…….. 17

5. Qualité des connaissances…….. 18

5.1. Typologie de connaissances…… 19

5.2. Ingénierie des connaissances …… 19

5.2.1. Acquisition des connaissances …… 20

5.2.2. Formalisation des connaissances…… 21

5.2.3. Système à base de connaissances…… 22

5.2.4. Système à base de règles…… 23

5.3. Gestion de la qualité et l'ingénierie de connaissances…….. 26

5.3.1. Classification des travaux de recherche…….. 26

5.3.2. Gestion de la qualité pendant l'acquisition des connaissances…….. 27

5.3.3. Gestion de la qualité pendant apprentissage automatique…… 27

5.4. Limites des travaux de recherche sur la gestion de la connaissance…… 29

6. Nettoyage des données…… 29

6.1. Présentation du processus de nettoyage des données…… 30

6.2. Etude de quelques travaux de recherche sur le nettoyage des données…… 31

6.2.1. Nettoyage déclaratif…… 31

6.2.2. Nettoyage à base de règles…… 31

6.3. Étude comparative de quelques outils du nettoyage des données…… 32

6.4. Incorporation des utilisateurs dans le nettoyage des données…… 33

6.5. Limitations des outils du nettoyage à base de règles…… 33

7. Conclusion …… 35

Chapitre 2: Extraction des connaissances à partir des données

1. Introduction…… 37

2. Processus d'extraction des connaissances à partir des données…… 37

2.1. ECD et fouille des données …… 38

2.2. Présentation du processus d'ECD …… 39

2.2.1. Préparation des données…… 39

2.2.2. Découverte des connaissances…… 40

2.2.3. Évaluation et réitération…… 42

2.2.4. Gestion de la qualité des connaissances dans l'ECD…… 43

3. Présentation du processus d'entreposage des données…… 44

3.1. Entrepôt des données vs entreposage des données…… 44

3.2. Présentation du processus d'extraction, transformation et chargement des données…… 45

3.2.1. Extraction des données…… 46

3.2.2. Transformation des données…… 46

3.2.3. Chargement des données…… 47

3.2.4. Rafraîchissement des entrepôts des données..... 47

3.2.5. Gestion de la qualité des données par l'ETC..... 48

3.2.6. Traçabilité des données..... 48

3.2.7. Etude comparative de quelques outils d'ETC..... 48

4. Présentation de quelques travaux de recherche sur la qualité dans l'ECD..... 49

4.1. Gestion de la qualité des données dans l'ECD..... 50

4.2. Gestion de la qualité des connaissances dans l'ECD..... 50

4.2.1. Etude de quelques travaux de recherche..... 50

4.2.2. Inconvénients des travaux de recherche sur la qualité des connaissances dans l'ECD..... 51

5. Présentation de quelques travaux de recherche sur la qualité dans l'ED..... 52

5.1. Principales méthodologies de gestion de la qualité de l'ED..... 52

5.1.1. Présentation de l'approche GQM..... 53

5.1.2. Présentation de la méthodologie DWQ..... 53

5.2. Etude de quelques travaux de recherche sur la qualité des EDs..... 53

6. Entreposage des règles et des connaissances..... 55

7. Entrepôt des données à base de règles..... 55

8. Intégration des processus ED et ECD..... 56

9. Conclusion..... 56

Chapitre 3: Un système d'entreposage des règles et des connaissances orienté qualité

1. Introduction..... 58

2. Rappel de quelques concepts et propriétés de la logique..... 59

3. Formalisme unifié de représentation des règles..... 60

3.1. Motivation 60

3.2. Conception du métamodèle générique du système de nettoyage des données à base de règles..... 61

3.2.1. Détermination des principales caractéristiques des règles…….. 61

3.2.2. Cycle de vie d'une règle…….. 64

3.3. Description du formalisme unifié de représentation des règles…….. 64

3.3.1. Définition d'une règle…… 65

3.3.2. Forme de la règle…… 65

3.3.3. Composantes de l'environnement d'une formule…….. 69

3.3.4. Métamodèle des règles du formalisme proposé…….. 71

3.3.5. Détermination des propriétés des règles…… 71

4. Système de gestion de l'entrepôt des règles…… 73

4.1. Identification des structures des données du SGER…….. 73

4.2. Description des composantes du système de gestion de l'entrepôt des règles…….. 75

4.2.1. Installation du système…… 75

4.2.2. Construction de l'environnement des règles…….. 76

4.2.3. Interfaces utilisateurs…… 77

4.2.4. Entreposage des règles…… 77

4.2.4.1. Extraction des connaissances et des règles…….. 77

4.2.4.2. Transformation des connaissances et des règles…….. 80

4.2.4.3. Chargement des règles…….. 81

4.2.5. Système de gestion des règles…… 84

4.2.6. Système de gestion de la qualité…… 84

5. Conclusion …… 84

Chapitre 4: Adaptation des processus d'ECD et d'entreposage des données

1. Introduction…….. 86

2. Objectifs et apports de l'adaptation du processus d'entreposage des données…….. 87

2.1. Objectifs du processus adapté…….. 87

2.2. Apports du processus adapté…….. 87

3. Présentation du processus adapté d'entreposage des données……. 88

3.1. Composantes du processus adapté d'entreposage des données …… 88

3.2. Description du processus ETCTC…….. 90

3.2.1. Définitions de quelques concepts de base…… 90

3.2.2. Règle d'évaluation de la qualité…… 91

3.2.3. Description des différentes phases du processus ETCTC…….. 91

3.2.3.1. Phase d'extraction des données …… 92

3.2.3.2. Phase de monotransformation des données…… 92

3.2.3.3. Phase de monochargement des données…….. 92

3.2.3.4. Phase de transformation des données……. 92

3.3. Processus de propagation des données corrigées…… 93

3.3.1. Motivation et apports de la propagation des données corrigées…….. 93

3.3.2. Description des différentes phases du processus de propagation des données corrigées…….. 94

3.3.2.1. Envoi de l'ensemble corrigé…….. 95

3.3.2.2. Evaluation et amélioration de la qualité des données corrigées…….. 95

3.3.2.3. Propagation des données validées…… 97

3.3.3. Capture de changement des données corrigées…… 98

3.4. Processus de la traçabilité des données corrigées…….. 98

3.4.1. Algorithme de la traçabilité des données corrigées…… 98

3.4.2. Création de la structure des données de TDC …… 99

3.4.3. Initialisation de la structure…… 100

3.4.4. Traçabilité des données corrigées…….. 100

3.4.5. Traçabilité des données dérivées …….. 101

4. Présentation du processus ECD adapté…… 101

4.1. Entreposage des données …… 102

4.2. Transformation des données…… 102

4.3. Évaluation des données…… 102

5. Conclusion…… 103

Chapitre 5: Validation et performances

1. Introduction …… 104

2. Expérimentation: étude de cas et implémentation…… 104

2.1. Présentation de l'étude de cas…… 104

2.2. Démarche de l'expérimentation …… 105

2.3. Sources des données opérationnelles …… 106

2.4. Elicitation des connaissances…… 110

2.5. Fonctionnalités de l'outil ETCTC_ED…… 112

2.5.1. Création des connexions …… 112

2.5.2. Création des bases des données…… 113

2.5.3. Extraction, transformation et chargement des données…… 115

2.6. Algorithmes développés lors de l'expérimentation…… 115

2.6.1. Algorithmes de gestion de la qualité des règles…… 115

2.6.1.1. Démarche de collection des dimensions et des métriques de qualité…… 115

2.6.1.2. Algorithme d'initialisation des valeurs des métriques de qualité des règles 116

2.6.1.3. Algorithme de calcul des métriques par échantillonnage…… 118

2.6.1.4. Algorithme de statut de règle…… 118

2.7. Evaluation de l'expérimentation …… 119

3. Validation théorique…… 121

4. Performances du système proposé…… 123

5. Conclusion …… 123

Conclusion générale et perspectives

Conclusion générale et perspectives…… 124

Bibliographie

Références bibliographiques……. 128

Annexe

Annexe: Code JAVA de l'implémentation du processus ETCTC…… 142

Table des Figures

Figure 1 : Métamodèle de travail proposé…… 3

Figure 2 : Relations entre Donnée, information et connaissance, et outils associées…….. 10

Figure 3 : Classification des problèmes de la qualité des données…….. 12

Figure 4 : Principales problématiques de recherche…….. 13

Figure 5 : Cycle de vie de Total Data Quality Management (TDQM)…….. 15

Figure 6 : Ingénierie des connaissances (cycle de vie)…….. 20

Figure 7 : Classification des formalismes de représentation des connaissances…… 22

Figure 8 : Classification des règles…… 23

Figure 9 : Métamodèle UML des règles…….. 24

Figure 10 : Architecture générique d'un Système à base de règles…….. 26

Figure 11: Processus du nettoyage des données…….. 30

Figure 12 : Processus d'extraction des connaissances à partir des données (ECD)…… 38

Figure 13 : Quelques méthodes de fouille de données…… 41

Figure 14 : processus d'ECD et qualité…… 43

Figure 15 : Processus d'Entreposage des Données…….. 45

Figure 16 : Métamodèle générique du système de nettoyage de données à base de règles 62

Figure 17: Cycle de vie d'une règle…… 64

Figure 18 : Exemple d'unification de deux règles…… 69

Figure 19 : Métamodèle UML d'une règle…… 71

Figure 20 : Système de gestion de l'entrepôt de règles…….. 74

Figure 21 : Forme générale du formulaire-questionnaire…….. 79

Figure 22 : Algorithme de transformation des dimensions de la qualité…… 83

Figure 23 : processus adapté d'entreposage des données…… 89

Figure 24 : Processus de propagation des données corrigées…… 94

Figure 25 : Algorithme d'évaluation et amélioration de la qualité des données corrigées….. 96

Figure 26 : Algorithme d'évaluation de la qualité d'une règle…… 97

Figure 27 : Algorithme de la traçabilité des données corrigées…… 99

Figure 28 : Structure des données corrigées…… 100

Figure 29 : Processus adapté d'ECD à base de règles…… 101

Figure 30 : Modèle physique de la base de données de la pharmacie…… 107

Figure 31 : Modèle physique de la base des données du bureau des entrées…… 107

Figure 32 : Modèle physique de la base des données du laboratoire…… 108

Figure 33 : Diagramme de classe UML de métabase…… 109

Figure 34 : Implémentation de la métabase sous MySQL…… 109

Figure 35 : Formulaire-Questionnaire de la collection des informations…… 111

Figure 36: Interface d'ajout des connexions…… 112

Figure 37: Interface DriverGateWay…… 113

Figure 38: Interface de création des bases des données…… 113

Figure 39: Interface de création des tables…… 114

Figure 40: Interface de création des champs. …… 114

Figure 41 : Gestion de la dimension de qualité : Exactitude…… 116

Figure 42 : Algorithme d'initialisation des valeurs d'une Règle…… 116

Figure 43 : Algorithme de calcul des métriques par échantillonnage…… 118

Figure 44 : Algorithme de calcul du statut d'une règle…… 118

Figure 45 : Histogramme de la qualité des entrepôts des données et SDO…… 119

Figure 46 : Histogramme du temps estimé de transformations des données…… 120

Table des tableaux

Tableau 1: Description des différentes problématiques de recherche de qualité des données…….. 13

Tableau 2 : Quelques approches proposées de modélisation de la qualité des données……. 16

Tableau 3 : Dimensions récurrents de la qualité des données…….. 17

Tableau 4 : Formes des règles…….. 25

Tableau 5 : Définition des critères de qualité des connaissances…….. 28

Tableau 6 : Critères de qualité des connaissances et des règles…… 29

Tableau 7 : Critères de comparaison des outils de nettoyage des données…… 32

Tableau 8 : Comparaison des outils de nettoyage de données…….. 34

Tableau 9 : Quelques outils ETC (Propriétaires et Source-Ouverte) …… 48

Tableau 10 : représentation et unification des différents types de règles…… 60

Tableau 11: Caractéristiques des règles…….. 63

Tableau 12 : Interprétation des statuts d'une règle…….. 65

Tableau 13 : Exemples de quelques ensembles complets d'opérateurs…… 66

Tableau 14 : Quelques propriétés de base des règles…….. 72

Tableau 15 : Caractéristique informatique des sources des données opérationnelles de l'établissement sanitaire…….. 106

Table des algorithmes

Algorithme de transformation des dimensions de la qualité……. 83

Algorithme d'évaluation et amélioration de la qualité des données corrigées……. 96

Algorithme d'évaluation de la qualité d'une règle……. 97

Algorithme de la traçabilité des données corrigées……. 99

Figure 42 : Algorithme d'initialisation des valeurs d'une Règle…… 116

Figure 43 : Algorithme de calcul des métriques par échantillonnage……. 118

Figure 44 : Algorithme de calcul du statut d'une règle…….. 118

Introduction générale

Introduction générale

Les systèmes d'information deviennent de plus en plus complexes et diversifiés en raison notamment de l'émergence de nouvelles technologies. L'accroissement continu de la volumétrie des données numériques ainsi que la multiplicité des sources de données de plus en plus hétérogènes, conjugués aux besoins pressants des entreprises à exploiter ces données dans un processus d'aide à la prise de décisions ont fait émergé de nouvelles problématiques que les technologies émergentes d'extraction des connaissances à partir des données et d'entreposage de données continuent à étudier et à leur chercher des solutions. Ceci nécessite la définition de nouvelles approches pour les architectures, l'intégration, la modélisation, l'interrogation, l'optimisation

L'Extraction des Connaissances à partir des Données (ECD) [1] [1, 2, 3] apparue dans la communauté de l'intelligence artificielle, a pour but l'identification de structures inconnues, valides, et potentiellement exploitables dans les bases de données. L'ECD propose un cadre général dans lequel sont regroupées les méthodes qui permettent de faire face aux problèmes d'organisation des données et de leur exploitation, plus particulièrement : l'entreposage des données et la fouille des données [4]. L'entreposage des données [2] a pour objet d'organiser des très grands volumes de données, de les structurer et de les préparer à l'analyse. Il est centré sur le processus d'Extraction, Transformation et Chargement de données (ETC) [3][5]. Les données sont généralement stockées dans des bases des données spécialisées dites : Entrepôts des Données (ED)[4] [6]. La Fouille des Données (FD)[5] a pour but d'extraire des connaissances à partir des données par des méthodes de structuration (apprentissage non supervisé) ou par des méthodes explicatives (apprentissage supervisé), une fois les données acquises et préparées. Comme la FD est étroitement liée au processus d'ECD, la plupart des travaux de recherche utilisent ces deux termes de manière interchangeable.

Le niveau de qualité des données a un impact direct sur les applications basées sur le e-business : une non-qualité peut entraîner une augmentation des coûts et avoir des répercussions sur le niveau de service proposé au client. Si par exemple on ne sait pas précisément quels sont les résultats de l'entreprise (Chiffre d'affaire, bénéfices etc.), les produits qui se vendent bien ou mal, on peut prendre des décisions qui vont à l'encontre des intérêts stratégiques de l'entreprise. De même, des données erronées sur le client nuisent à la bonne connaissance du client et peuvent avoir des répercussions graves sur la perception qu'il a de l'entreprise, qui se traduiront par des annulations de commande, etc. Enfin, la mauvaise qualité des données a également un impact sur la satisfaction des employés. En effet, ils ne peuvent mener à bien leurs tâches

[1] En anglais : Knowledge Discovery from Data (KDD)
[2] En anglais: Data Warehousing
[3] En anglais: Extract, Transform and Load (ETL)
[4] En anglais: Data Warehouse (DW)
[5] En anglais: Data Mining (DM)

quotidiennes du fait de la non-qualité des données et doivent souvent recourir à des corrections et à des nettoyages des données. Ces problèmes relèvent de la responsabilité du gestionnaire et de lui seul et non pas individuellement d'un employé isolé. Le projet de gestion de la qualité des données requiert l'attention des gestionnaires.

Par conséquent, la qualité des données[6] et, au sens large la qualité des informations n'a cessé de prendre une place de premier plan au sein des communautés de recherche en ECD et ED. En effet, l'extraction des connaissances et la prise des décisions peuvent être réalisées sur des données de qualité médiocre (i.e. des données inexactes, incomplètes, ambiguës, incohérentes et contenant des doublons). On peut alors s'interroger sur le sens à donner aux résultats de ces analyses et remettre en cause la qualité des connaissances ainsi "élaborées" et le bien-fondé des décisions prises.

Les solutions ont maintenant atteint une certaine maturité et la place de l'ECD et des EDs est désormais au cœur des préoccupations d'un nombre important d'entreprises et organisations. Il n'en reste pas moins que de nombreux défis restent ouverts pour les chercheurs et les industriels travaillant dans ces domaines [7]. Les approches traditionnelles de ces deux domaines ne sont plus adaptées à un contexte dans lequel il faut appréhender non seulement de gros volumes de données mais s'intéresser à la qualité des informations [8]. La qualité des données suscite un vif intérêt, et ce thème émerge désormais comme un champ de recherche à part entière. La mauvaise qualité des données est l'un des problèmes principaux rencontrés par les entreprises lorsqu'elles mènent des projets décisionnels.

La qualité des données est un facteur clé pour le succès de projets décisionnels. De ce fait, Les deux processus d'ECD et d'ED utilisent des métriques sur la qualité des données pour pouvoir les exploiter à des fins d'évaluation et d'amélioration de la qualité des données. Le nettoyage des données[7] fait partie des stratégies d'amélioration automatique de la qualité des données [9]. Cependant, la mise en œuvre des systèmes de nettoyage des données avec les architectures actuels d'ECD et d'ED n'est ni incrémentale ni interactive. Ainsi les solutions actuelles de nettoyage des données se limitent à la simple implémentation d'une solution logicielle, le plus souvent des programmes ad hoc pour la recherche des doublons, corrections des noms et adresses. Notre observation sur les travaux existants sur la qualité des données, nous a conduit à la nécessite de proposer des nouvelles architectures d'ECD pour la mise en œuvre d'un système complet de la gestion de la qualité des données et des connaissances.

La littérature atteste que l'implication de l'utilisateur dans la gestion de la qualité des données et des connaissances est très importante tant en ECD qu'en ED [10, 11]. Le défi de cette implication consiste à exploiter automatiquement les informations des utilisateurs en connaissances nouvelles sous le contrôle des experts. Cette solution est lourde à mettre en œuvre et à maintenir, en particulier au niveau de l'évolution des besoins d'analyse. Ainsi, nous proposons de définir des nouvelles approches et systèmes de gestion de la qualité des données.

[6] En anglais, Data Quality (DQ)
[7] En anglais, Data Cleaning (DC)

Leur conception doit être basée sur les méthodes à base de règles qui permettent la représentation des connaissances de l'utilisateur et par la suite, facilitent leur intégration dans l'ECD en assurant leur évolution et leur mise à jour [12]. L'un des défis actuels de ces solutions est de fournir des moyens pour extraire et gérer des connaissances utiles (avant leur transformation sous forme de règles) et de leur qualité. Cependant, la littérature atteste que peu de travaux génériques ont été effectués sur la problématique de l'amélioration de la qualité des données basée sur l'extraction des connaissances et, plus particulièrement, du nettoyage des données à base de règles. Ces travaux s'intéressent généralement aux règles de production "Si A alors B". De bons résultats sont obtenus avec les systèmes à base des règles pour certains domaines d'applications tels que la médecine, la géologie, la finance, etc. Cependant, il apparaît vite que la formalisation sous forme de règles de la connaissance expertise est une tâche difficile voire impossible dans certains domaines [13]. De ce fait, nous proposons de trouver et proposer un nouveau formalisme de représentation des connaissances qui supporte l'évolutivité et la flexibilité des connaissances (non seulement pour la prise en compte de nouvelles informations mais aussi pour la gestion de la qualité de ces informations).

La figure 1 que nous avons établi comme un métamodèle de ce travail illustre les différents points essentiels à l'amélioration de la qualité dans les systèmes décisionnels basés sur l'extraction des connaissances. Elle montre que l'ECD et l'entreposage des données sont des supports nécessaires à la réalisation d'une architecture qualifiée de décisionnelle. Elle montre aussi que l'ED est un composant essentiel de l'ECD. D'où nous voyons la nécessite de couplage des processus d'ECD et ED pour extraire des connaissances utiles pour améliorer leur performance en termes de qualité. Dans la figure 1, nous avons essayé de définir les principaux acteurs participants à la gestion de la qualité des données: données, connaissances, utilisateurs et processus (transformation(s)), ainsi que leurs interactions.

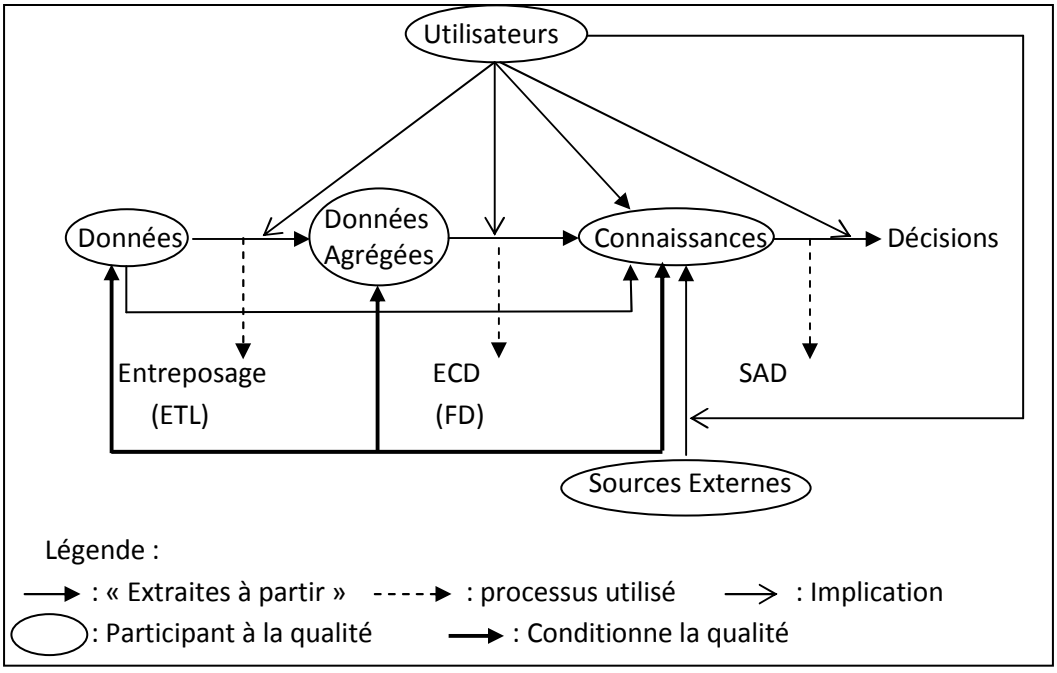

Figure 1 : Métamodèle de travail proposé.

Dans ce contexte, cette thèse se concentre sur l'amélioration de la qualité des données et des connaissances qu'est devenu un des sujets d'intérêt tout à la fois émergent dans le domaine de la recherche et critique dans les entreprises tant en ECD qu'en ED. Notre proposition consiste à la réalisation d'une nouvelle architecture d'ECD et d'ED qui implique l'utilisateur dans la gestion de la qualité des données et des connaissances en lui permettant d'exprimer ces connaissances. Comme il est difficile d'être exhaustif dans la collection des connaissances expertises avant leur transformation en règles au moment de la conception d'ECD parce que des nouvelles connaissances peuvent émerger et d'autres connaissances existantes peuvent être changées à tout moment, cette thèse tient compte de la nécessité d'une personnalisation d'ECD et ED, qui placerait l'utilisateur au cœur de la gestion de la qualité des données. La personnalisation n'est pas un principe nouveau, elle fait l'objet de nombreux travaux dans des domaines tels que la recherche d'informations, les bases de données. Cependant, elle constitue un axe de recherche récent dans les domaines d'ECD et d'ED, alors même que les caractéristiques de ces derniers lui sont favorables. En effet, la volumétrie des données et des connaissances connue pour être importante dans les ECD et le rôle central que joue l'utilisateur dans le processus décisionnel, au niveau de l'analyse en ligne, sont deux éléments qui justifient pleinement le recours à la personnalisation [14]. Dans notre travail, la personnalisation consiste à la création d'un entrepôt de règles pour assurer une meilleure gestion des données et de leur qualité. Ces règles seront exploitées par le système (processus) d'amélioration de la qualité et seront aussi validées par les experts et automatiquement.

Les principales motivations de notre travail qui nous ont poussé à choisir l'adaptation du processus d'ECD, se résument aux points suivants:

1. L'ECD et ED ont servi principalement à la prise de décision comme le montre la figure 1. Un objectif majeur est le maintien de la complémentarité de ces deux processus qui ne doivent plus être considérées comme des processus isolés mais davantage comme des étapes du processus d'aide à la décision. Ces processus se complètent et l'un des enjeux de ce travail est d'organiser et d'exploiter leur synergie et leur collaboration pour la conception d'un système permettant une gestion total et continue de la qualité des données et des connaissances dans les systèmes d'aide à la décision.
2. L'utilisation des techniques de l'ECD pour extraire des connaissances utiles des données elles mêmes pour l'optimisation des performances des bases des données est avancée depuis quelques années [15]. Cependant, peu de travaux ont été entrepris dans cette optique notamment pour l'amélioration de la qualité des données et des connaissances. C'est pourquoi nous nous intéressons à l'adaptation du processus de l'ECD pour une meilleure prise en charge de la qualité des données pendant l'entreposage des données et l'extraction des connaissances.
3. Très fréquemment, la qualité des données est un processus non systématique traité cas par cas, de façon ponctuelle. Non systématique veut dire qu'il n'existe pas de processus établi qui puisse tendre vers une stratégie globale de gestion de la qualité des données [9].

4. La complexité et le volume toujours croissant des données, auxquels s'ajoutent de nouveaux formats et l'arrivée des outils de nouvelle génération qui doivent intégrer les données les plus récentes de l'entreprise. Un exemple est l'intégration de données orientées « transaction » (c'est-à-dire des données opérationnelles dites aussi de production) dans des programmes analytiques utilisés pour la prise de décision.
5. Les outils de nettoyage des données et de la gestion de la qualité des données sont de plus en plus embarqués dans les outils d'ETC [16]. Ces outils n'intègrent pas toute la problématique de la gestion de la qualité des données [17]. Par conséquence l'adaptation de l'ECD implique automatiquement l'adaptation de processus ETL.
6. Le problème de la qualité des données et de connaissances doit être abordé sous un angle pluridisciplinaire : FD, Aide à la décision, gestion des connaissances, traitement automatique des langues, linguistiques et logique [18]. Cependant, on n'a pas constaté des travaux qui tiennent compte de tous les domaines qui participent à la chaine de production des connaissances : données, ECD et gestion de connaissances.
7. Bien que les architectures décisionnelles (ED et ECD) soient considérées centrées utilisateurs, leur (utilisateurs) prise en compte dans ces architectures a finalement été peu étudiée.
8. Les travaux actuels sur l'extraction des connaissances se focalisent sur la qualité des données sur lesquelles va porter le processus d'ECD comme un facteur décisif pour la performance de la qualité des connaissances extraites. Cependant, ce travail se focalise aussi sur la qualité des connaissances avec lesquelles nous évaluons et améliorons la qualité des données. Cela veut dire que la relation " ... conditionne la qualité de … " entre données et connaissances est symétrique (bidirectionnelle). L'exploitation de cette symétrie est un facteur crucial pour la réussite de tel système de gestion de la qualité des données.
9. Du fait d'énormes quantités de connaissances qui peuvent être extraites, la validation des connaissances est l'une des étapes les plus problématiques d'un processus d'ECD. Afin d'appréhender ce volume de connaissances et de trouver des connaissances utiles pour la prise de décisions (amélioration de la qualité dans notre cas), l'utilisateur (l'expert des données) a besoin de véritablement définir des critères de qualité pour mesurer la rentabilité des ces connaissances.
10. Les connaissances sont souvent d'origine expérimentale ou heuristique. Cependant, très peu de travaux génériques dans la littérature existent sur les problématiques d'incorporation de connaissances expertes dans le processus d'ECD. Le problème majeur lié à l'incorporation est le formalisme et l'acquisition des connaissances. Les formalismes utilisés pour la modélisation et la représentation des connaissances ne permettent pas un bon niveau d'abstraction et restent liés à l'implémentation. Un autre aspect du problème est que les connaissances d'un expert sont subjectives.
11. Une connaissance est généralement transformée sous forme de règles. Ainsi l'utilisation des systèmes à base de règles, fondements des systèmes experts, permet de rendre le nettoyage des données plus évolutive et interactive. Les travaux ayant porté sur

l'augmentation de la flexibilité dans l'ED font généralement recours à des langages à base de règles. Certains travaux ont apporté une réponse au traitement des exceptions dans le processus d'agrégation, rendant ce dernier plus souple. D'autres travaux ont proposé des modèles d'ED à base de règles pour résoudre des différents problèmes liés à la cohérence des données ainsi que à la performance, l'optimisation et l'évolution des modèles. Ainsi, les langages à base de règles ont permis de rendre plus flexible l'ED. Cependant, dans le contexte de la gestion de la qualité des données peu de travaux font recours aux systèmes à bases de règles.

12. L'entreposage des données se porte sur des copies des données opérationnelles. Ainsi le problème de la cohérence mutuelle des données est un facteur de qualité qui doit être prise en compte lors de la conception des projets de gestion de la qualité des données. Nous avons identifiées beaucoup de travaux dans la littérature sur la problématique de la propagation des mises à jour des données opérationnelles vers l'ED (rafraîchissement de l'ED). Cependant nous n'avons pas constaté des travaux sur la propagation des données améliorées vers les sources opérationnelles. Cela nous permet d'éviter de commettre les mêmes traitements d'amélioration de la qualité à chaque opération d'entreposage et de fouille de données. Donc une gestion totale de la qualité des données doit assurer la qualité des données originales et de leurs données copies et dérivées. La propagation permet aussi aux utilisateurs de détecter les erreurs introduites lors des opérations de nettoyage des données.
13. La gestion de la qualité basée sur les connaissances requiert l'incorporation d'un système à base de règle dans le processus d'ECD. Les systèmes à base de règles traditionnels sont conçus comme des systèmes fermés, autonomes, c'est-à-dire que leur fonctionnement est conçu indépendamment de l'environnement dans lequel ils évoluent. Ils ne permettent pas aussi la gestion de la qualité des règles. Ces limitations en font des systèmes coûteux à concevoir, peu robustes, difficiles à maintenir et mal adaptés à un environnement d'extraction des connaissances et d'entreposage des données. De ce fait, il est nécessaire de concevoir un système à base de règles propre à l'ECD et les projets de gestion de la qualité à base de connaissances.

Dans ce travail, nous nous intéressons plus particulièrement aux problèmes de la qualité des données exposées ci-dessus qui sont devenus des verrous à la nécessité d'obtenir des données et des connaissances de meilleure qualité, il convient donc nécessaire de considérer la qualité des données dans l'ECD comme un problème d'aide à la décision centrée sur l'utilisateur. De ce fait l'élément central de notre contribution dans cette thèse est la proposition d'un nouveau processus d'ECD adapté à la qualité des données et des connaissances. Afin de réaliser cette architecture, nos principales contributions peuvent être ventilées en quatre volets : l'entreposage des règles pour la gestion de la qualité des données et de connaissances, extension du processus ETL pour le maintien de la réplication des données, traçabilité des données nettoyées et capture des changements des données et l'adaptation du processus d'ECD. Ainsi, nous définissons dans cette thèse:

- **Un système d'entreposage des règles pour la gestion de la qualité des données et de connaissances:** Cette contribution est fondée sur l'observation suivante : la qualité des données est fortement liée à la qualité des connaissances qui sont généralement transformées sous forme de règles. Ainsi, nous avons proposé d'incorporer ces règles dans le processus ECD afin d'assurer leur gestion et d'améliorer leur qualité. Les systèmes actuels de gestion de règles gèrent et supportent spécifiquement les règles métier des entreprises et par conséquent ne permettent pas la gestion des connaissances et des règles de nettoyage des données. De ce fait, nous proposons un système évolutif, interactif et itératif d'entreposage des règles pour assurer la gestion des règles et de leur qualité pour optimiser les performances en termes de qualité du nettoyage des données et par la suite la qualité des données offertes pour l'aide à la décision. A la différence des systèmes actuels de gestion des règles qui permettent uniquement la représentation des règles de production, le système que nous proposons dans ce travail supporte les différents types de règles. L'utilisation du concept de structure de la théorie de la logique des prédicats est le fondement du formalisme que nous représentons dans ce travail. L'adaptation des certains concepts de la logique des prédicats, nous a permis de regrouper toutes les caractéristiques des règles et de leur qualité (dimensions de qualité) dans une seule entité qui tient compte de l'évolution et du changement des règles. L'utilisation de la logique des prédicats s'appuie sur le fait que la validité et l'exactitude d'une règle dépend uniquement de l'environnement dans laquelle est appliquée. Dans ce système, nous proposons aussi une méthode d'acquisition des connaissances par élicitation afin de permettre l'incorporation de l'expertise dans la gestion de la qualité des données.
- **Un mécanisme d'extension du processus ETC.** Cette proposition est basée sur les deux observations suivantes : (1) une donnée nettoyée (c.à.d. sa valeur a changé due aux opérations de nettoyage des données) doit être propagée vers les sources des données à partir des quelles cette donnée est formée, et (2) les opérations de nettoyage des données peuvent introduire de nouvelles erreurs sur les données. De ce fait, nous avons proposé d'ajouter des nouvelles étapes dans l'ETC pour permettre la propagation et la validation des données corrigées. Cette extension a trois bénéfices. Premièrement, elle permet la propagation des données afin d'assurer la cohérence mutuelle des données. Deuxièmement, elle permet d'éviter de refaire le même travail d'amélioration de la qualité à chaque entreposage des données et/ou FD. Finalement, elle permet aux utilisateurs de valider les données nettoyées et les règles.
- **Un modèle de traçabilité des données nettoyées et capture des changements des données.** La propagation des données et leur validation nécessitent de conserver la traçabilité de ces données. De ce fait, nous avons proposé des algorithmes pour la traçabilité des données nettoyées, et pour la capture des changements des données nettoyées. Nous avons réalisé des algorithmes spécifiques aux données nettoyées afin d'accélérer le processus d'ECD et de permettre à l'utilisateur ou système à tout moment d'intervenir pour éviter la non qualité des données nettoyées.
- **Une adaptation du processus d'extraction des connaissances des données.** Cette contribution est une conséquence de trois propositions précédentes.

Ce mémoire est organisé de la manière suivante:

Dans le chapitre 1, nous nous attachons à présenter les différents aspects fondamentaux liés à la gestion et la gouvernance de la qualité des données et des connaissances. Nous expliquons la notion de la qualité des données et des connaissances, et leurs différents constituants. Nous fournissons un panorama des différentes approches de gestion de la qualité des données et connaissances ainsi que l'extraction et l'impact des connaissances sur les données. Nous nous intéressons particulièrement dans ce chapitre aux méthodes existantes de nettoyage des données à base de règles et aux travaux relatifs à l'acquisition des connaissances.

Le chapitre 2 est un état de l'art sur le processus d'ECD. Nous présentons l'architecture des processus d'ECD et d'entreposage des données. Nous décrivons en détail le processus d'ETC et, plus particulièrement, la préparation des données qui est la clef de voûte de ces deux processus. Nous terminerons ce chapitre par une étude comparative de quelques travaux de recherche portant sur la gestion de la qualité des données et plus particulièrement sur le nettoyage des données dans les processus d'ECD et d'ED.

Les chapitres 3, 4 et 5 sont consacrés à la présentation de nos contributions d'adaptation du processus d'ECD pour la prise en charge de la qualité des données.

Dans le chapitre 3, nous commençons par la présentation du fondement théorique de notre travail. Puis nous détaillons notre formalise de représentation des connaissances (règles) qu'est la base du processus d'entreposage des règles qui nous proposons dans ce travail. Finalement nous décrivons les différents composants du système d'entreposage des règles proposé et leur interaction.

Le chapitre 4 décrit le processus proposé d'entreposage des données ainsi que celui d'ECD. Nous détaillons plus particulièrement notre processus ETCTC proposé pour permettre l'extraction, transformation et chargement et propagation des données. Dans ce chapitre nous décrivons aussi nos algorithmes de capture des changements des données et de traçabilité des données nettoyées qui sont essentiels pour l'optimisation des performances de notre système.

Comme notre contribution est plus conceptuelle que pratique, le chapitre 5 présente une validation théorique du processus ETCTC ainsi que les résultats de l'expérimentation que nous avons réalisée dans un établissement sanitaire.

Le chapitre 6 conclut la thèse, discute des résultats obtenus du travail que nous avons mené dans cette thèse et ouvre sur de nouvelles perspectives de recherche.

Chapitre 1:
Qualité des Données et des Connaissances

« Une connaissance, c'est une information validée par l'expérience »
A. Einstein

1. Introduction

Longtemps associée au domaine industriel, la qualité est un concept récent dans le monde académique, la qualité en recherche ayant longtemps opposé créativité de la recherche et rigueur du management.

La qualité des données est un thème qui revient dans toutes les phases de la gestion des données de base. Le présent chapitre met en exergue l'importance de la qualité des données et des connaissances. Il s'agit là d'aspects fondamentaux liés à la gestion de la qualité tant pour les données que pour les connaissances. Nous nous devons d'évoquer certaines généralités sur la gestion de la qualité afin d'éclaircir nos propres futurs. Il s'agit en effet de revenir sur ce qu'est la gestion de la qualité, comment est elle conçue, quelles sont leurs caractéristiques, etc.

Ainsi, ce chapitre s'attache tout d'abord à définir ce que recouvre le terme de qualité tant dans les bases des données que dans les bases des règles afin de comprendre les difficultés que soulève cette opération. Puis nous représentons les stratégies de conception des systèmes de gestion de la qualité, les principaux éléments relatifs à leur modélisation, avant d'évoquer brièvement les récents travaux sur ce point et de revenir sur la notion de nettoyage des données à base de règles qui est cruciale pour nos travaux. Nous détaillons aussi la problématique d'acquisition des connaissances qui est aussi fondamentale pour nos travaux. Enfin, nous concluons ce chapitre en introduisant la gestion de la qualité des connaissances.

L'ensemble des problèmes entravant l'intégration de la gestion de la qualité des données dans les systèmes de gestion des bases données ainsi que les éléments qui permettraient d'aider à leur résolution sont également mis en exergue au sein de ce même chapitre.

2. Donnée, information et connaissance

Plusieurs définitions peuvent être trouvées dans la littérature pour définir les concepts de : donnée, information et connaissance. Ces concepts sont récurrents, c'est pourquoi il est important de préciser et de comprendre le sens que nous leur attribuons dans cette thèse. La figure 2 donne les définitions de ces concepts et leurs interdépendances ainsi que les outils associés à chaque concept.

Pour établir la distinction entre donnée, information et connaissances, nous allons reprendre les définitions introduites dans le contexte de la gestion de connaissances. Une donnée est un élément brut qui n'a pas encore été interprété. Et c'est là toute la différence entre information et donnée. En effet, une information est une donnée interprétée (ou réinterprétée) [19, 20, 21]. En d'autres termes, la mise en contexte d'une donnée crée de la valeur ajoutée pour constituer une information. La connaissance est considérée comme une information comprise, c'est-à-dire assimilée et utilisée, qui permet d'aboutir à une décision. De ce fait, nous constatons que l'information est étroitement liée à la donnée et la connaissance ainsi que la frontière qui

sépare information de donnée nous semble beaucoup trop mince. C'est pourquoi dans le cadre de cette thèse nous estimons que les données et les informations sont similaires et interchangeables.

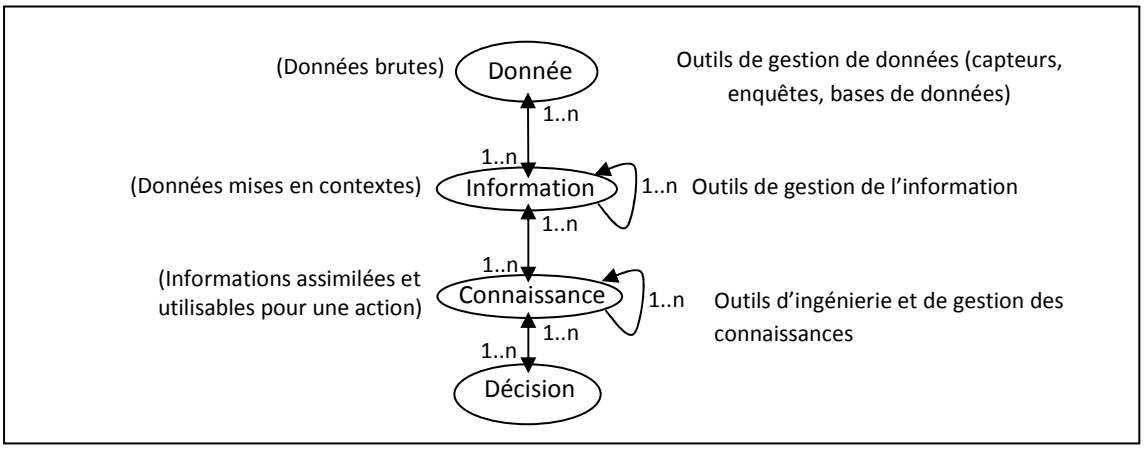

Figure 2 : Relations entre donnée, information et connaissance, et outils associées [21,22].

Une connaissance peut affecter l'utilisation des données, et inversement, une donnée peut réfuter ou affaiblir une connaissance [23]. Le constat de cette synergie données – connaissances appelle à réaliser un couplage de ces deux concepts au sein des systèmes d'information afin de fournir aux utilisateurs des connaissances pertinentes adaptées à la prise de décision.

3. Qualité et concepts associés

La qualité est une préoccupation que l'on trouve dans beaucoup de domaine [24]. De ce fait la première difficulté réside dans l'absence de consensus sur la notion de qualité [9]. Comme la communauté aujourd'hui préconise également l'application dés le début des normes et standards internationaux, nous nous intéressons ici aux définitions données par l'organisation internationale de standardisation (ISO : International Standard Organization) et par Organisation de Coopération et de Développement Economiques (OECD : Organisation for Economic Cooperation and Development).

ISO : Les normes telles que l'ISO 10006 pour la gestion de projet, l'ISO 20000 pour la qualité de service, l'ISO 25000 pour le logiciel et l'ISO 27001 pour la sécurité informatique introduisent la notion de « qualité » qui apparaît très présente dans ces normes de management. La norme ISO 9000 définit la qualité comme «L'ensemble des propriétés et caractéristiques d'un produit ou d'un service qui lui confère l'aptitude à satisfaire des besoins exprimés ou implicites» [25, 26]. Pratiquement, la qualité d'un produit signifie qu'il est adapté au besoin qu'il est censé satisfaire. La notion de qualité s'applique aussi bien à des produits qu'à des services [27].

OECD : La qualité est vue comme un concept à facettes multiples. Les caractéristiques de qualité dépendent des perspectives, des besoins et des priorités d'utilisateur, qui changent à travers des groupes d'utilisateurs [28]. Ainsi cette définition est complémentaire à la définition

ISO 9000 en y ajoutant le contexte d'utilisation et le domaine de l'application c.à.d. que les besoins sont définis par l'utilisateur dans le cadre d'une application donnée.

La prise en compte du point des vues des utilisateurs dans la qualité, ont permis de séparer la qualité en deux parties : qualité interne et qualité externe. La qualité interne est l'ensemble des propriétés et caractéristiques d'un produit ou service qui lui confère l'aptitude à satisfaire aux spécifications de contenu de ce produit ou de ce service. La qualité externe est définie comme étant l'adéquation des spécifications aux besoins de l'utilisateur. Elle est liée aux besoins des utilisateurs et varie donc d'un utilisateur à un autre ou également pour un même utilisateur et d'une application à une autre [29,30].

Les définitions s'accordent sur le fait que la qualité d'un objet peut se décomposer en une multitude de dimensions, catégories, critères, facteurs, paramètres ou attributs. La qualité est de ce fait une notion relative fondée sur les besoins, multidimensionnelle et mesurable par le biais des indices dites : mesures de qualité (ou d'intérêt). Chaque dimension peut être mesurée de manière subjective (orientées utilisateurs) en recueillant préalablement la perception (buts, connaissances et croyances) des utilisateurs, soit de manière objective (orientées objets) au travers de suivi automatique des indicateurs spécifiques [31,32].

4. Qualité des données.

Si nous ne mettons pas en place aucune gestion de la qualité des données, le système pourra rapidement être saturé de données manquantes, superflues voir incorrectes. Le problème de la qualité des données se pose avec d'autant plus d'acuité que les volumes à traiter augmentent et que les applications tendent à se diversifier. Outre cela, les pressions réglementaires et les exigences de contrôle interne obligent les entreprises à s'intéresser de plus en plus à la qualité de leurs données.

4.1. Principaux problèmes de la non qualité des données

Les problèmes de qualité des données se répandent de façon endémique à tous les types de données (structurées ou non) et dans tous les domaines d'applications. Les conséquences des données de mauvaise qualité sur les prises de décision et les coûts financiers qu'elle engendre sont considérables.

Nous avons effectué une recherche bibliographique d'une manière intensive afin de mieux cerner les principales causes de la mauvaise qualité des données [33,34, 35, 36]. Nous avons constaté que la majorité des travaux de recherche répartit les problèmes de la mauvaise qualité des données en problèmes mon-sources et multi sources, aux niveaux de schéma ou des instances. Dans ce travail, nous nous intéressons aux problèmes au niveau des instances. Dans la figure 3, nous avons résumé les différents problèmes de la non qualité des données.

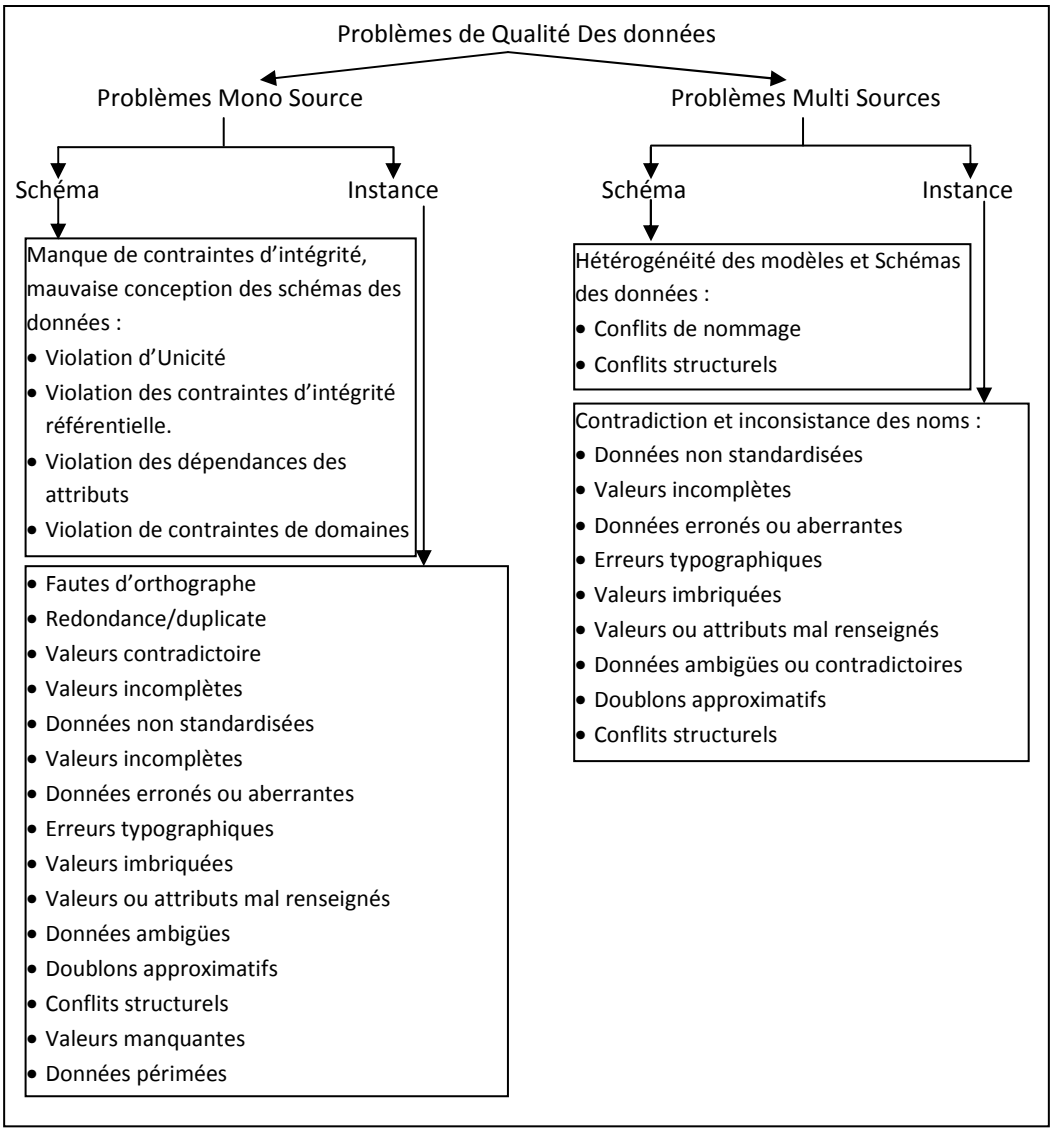

Figure 3 : Classification des problèmes de la qualité des données.

4.2. Etude des principaux travaux de recherche sur la qualité des données

Lorsque nous appliquons la définition de la qualité à la donnée, nous constatons que la qualité des données ne dépend pas uniquement des caractéristiques intrinsèque des données mais aussi des processus et utilisateurs qu'utilisent ces données [37].

4.2.1. Principales problématiques de recherche

Le domaine de la qualité des données comme le montre la figure 4 fait partie intégrante de différents sujets de recherche ainsi que divers domaines d'application. En pratique des applications pouvant traiter un point ou un autre de la gestion de la qualité des données sont utilisées par de nombreuses entreprises. Les corrections manuelles sont néanmoins encore fréquentes. Peu d'entreprises ont mis en place une gestion prenant en charge le cycle complet de gestion de la qualité des données. Les enjeux liés à ce domaine sont encore peu appréhendés par

le management des entreprises ce qui freine le développement d'initiatives sur la gestion globale de la qualité des données.

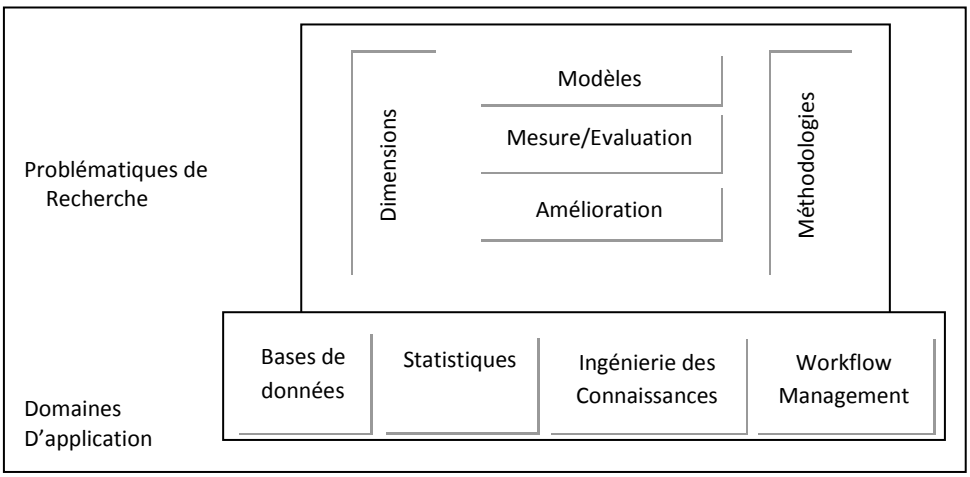

Figure 4 : Principales problématiques de recherche [38].

L'intérêt à une gestion totale et continue de la qualité des données a conduit à la définition de différentes méthodologies de gestion de la qualité des données. Le tableau 1 montre que ces méthodologies concerne principalement les dimensions qualité, les modèles, les techniques, les outils, ainsi que les méthodologies adaptées aux nouveaux types de données et systèmes d'informations [8, 39].

Problématique	Description
Dimensions	Sont appliquées dans les modèles qualité, les techniques, les outils et les structures.
Modèles	Sont utilisées dans les bases des données et représentent les dimensions et autres aspects liés à la qualité des données.
Techniques	Sont un ensemble d'algorithmes, d'heuristiques et de processus pour répondre à un problème spécifique sur la qualité des données.
Méthodologies	Fournissent des directives pour choisir à partir des techniques et des outils existants, la mesure de la qualité des données la plus efficace pour améliorer un système d'information spécifique
Outils	Sont nécessaires aux techniques et aux méthodologies, représentent des processus automatisées avec une interface qui permettent à l'utilisateur l'exécution manuelle de certaines techniques.

Tableau 1: Description des différentes problématiques de recherche de qualité des données [40].

De ce fait, ces dernières années, plusieurs travaux se sont focalisés sur la formalisation des dimensions, des indicateurs, des techniques et des méthodes pour une gestion efficace de la qualité des données [16, 41].

Dans la suite de cette section, nous détaillons les travaux liés à la gestion de la qualité des données.

4.2.2. Classification des travaux de recherche

Les différents travaux de recherche sur la qualité des données sont regroupés selon leur but qui est [42] :

1. Analyse et définition des dimensions qui impactent la qualité des données et les mesures associées.
2. Création d'un standard universel présentant l'ensemble des dimensions opérationnelles de la qualité des données.
3. Proposition d'une assistance à l'utilisateur pour qu'il définisse et évalue lui-même la qualité des données qu'il acquiert et manipule.

Parmi ces trois courants de recherche sur la qualité des données, différents travaux se sont plus particulièrement intéressés à l'une des trois activités suivantes [43, 44, 45]:

1. Modélisation de la qualité des données.
2. Mesure et évaluation de la qualité des données.
3. Amélioration de la qualité des données.

4.2.3. Principales méthodologies de la gestion de la qualité des données

La recherche dans les travaux des recherche existants sur la qualité des données, nous a permis de constater que les praticiens et les académiques se sont penchés sur le problème de la qualité des données dans les bases des données relationnelles et dans les systèmes d'information. Par conséquence, ils ont développés des méthodologies adéquates liés à la mise en œuvre des projets d'évaluation et d'amélioration continue de la de qualité des données. D'une manière générale, chaque méthodologie est adaptée à son domaine et type d'application [38, 46, 47, 48].

Basé sure cette recherche, nous avons identifié différentes méthodologies de gestion de la qualité des données [38, 47, 49, 50, 51]. Les principales méthodologies sont :

- Total Data Quality Management (TDQM)
- Total Information Quality Management (TIQM)
- A Methodology for Information Quality Assessment (AIMQ)
- Data Quality Assessment (DQA)
- Information Quality Measurement (IQM)
- Canadian Institute for Health Information Data Quality (CIHIDQ)
- An Object-Oriented framework for DQ Management (OODQM)
- QUality Evaluation Data Integration Systems (QUADRIS)
- Data Quality Broker (DQB)

Dans ce travail, nous nous intéressons à la méthodologie TDQM qu'est présenté comme une source profolique de travaux au sujet de la qualité des données. La Figure 5 montre que TDQM fonctionne selon un processus (cycle continue) en itérations consistant en : Définition, Mesure, Analyse et Amélioration de la qualité des données. La phase de définition consiste à

décrire les dimensions qualité ainsi que les exigences de la qualité des données associées. Pendant la phase de mesure, le TDQM produit les métriques pour chaque dimension ainsi que les méthodes de calcul associées. La phase d'analyse identifie les causes de la non qualité des données selon les résultats de la phase d'analyse, et il évalue aussi leurs impacts. Finalement, la phase de l'amélioration fournit les méthodes et les techniques de correction et d'amélioration de la qualité des données [42, 48, 52, 53].

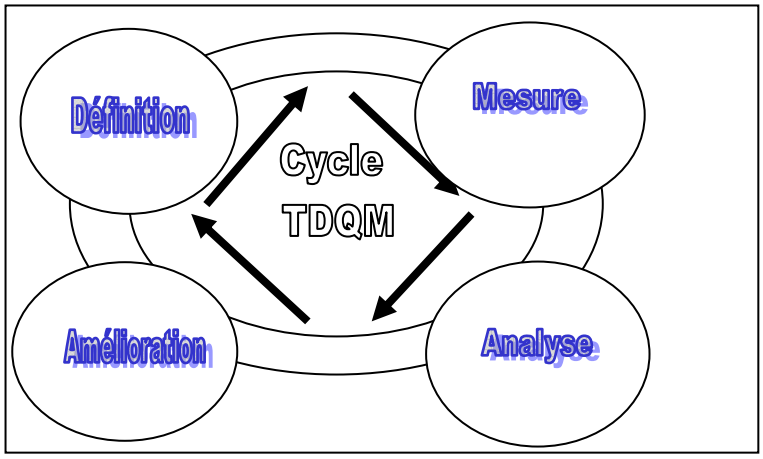

Figure 5 : Cycle de vie de Total Data Quality Management (TDQM) [54].

Cependant, quelle que soit la méthode utilisée, la mise en place d'un processus de gestion de la qualité des données nécessite la [45]:

- Définition des objectifs de qualité fixés par les usagers.
- Définition des processus de maintien des données: processus de saisie, de validation et de contrôle, des indicateurs de mesure permanente (phases de définition et mesure pour TDQM),
- Mise en place des outils techniques: formatage, validation, correction automatique, déduplication (phase d'amélioration pour TDQM), et
- Responsabilisation des collaborateurs concernés pour les amener à un réflexe permanent "Qualité des données" (phase d'analyse pour TDQM).

L'inconvénient majeur de ces méthodologies est qu'elles transposent les normes de qualité des produits industriels aux produits informationnels malgré les différences énormes entre ces deux objets (produit industriels et données). Face à cet inconvénient, d'autres méthodologies ont fait recours à la théorie de l'information, qui a le mérite de mettre en évidence le caractère relatif de la qualité des données. L'utilité principale de ces travaux est qu'elles tiennent de tous les acteurs engagés dans le flux d'information pour en évaluer la qualité, à savoir aussi bien le producteur que l'utilisateur [19].

4.2.4. Modélisation de la qualité des données.

Tout le monde s'accorde sur le fait que la qualité des données peut être décrite en termes de facteurs, critères, dimensions, catégories et attributs. L'analyse des travaux de recherche

menée dans [38, 55, 56] a recensé cinq approches concernant la modélisation de la qualité des données (voir tableau 2). Ces approches permettent de définir des diverses dimensions au sein d'un modèle de qualité des données. Les approches théoriques, empiriques et intuitives sont dites génériques parce qu'elles sont liées aux systèmes d'information en général. Par contre les approches spécifiques sont liées à des systèmes d'information dans des domaines bien précis.

Plus de deux cents dimensions ont été recensées dans les littératures. Cependant, dans tableau 3, nous présentons les dimensions identifiées récurrentes pour la qualité des données qui sont : Traçabilité, exactitude, ponctualité ou fraicheur, complétude, cohérence, compréhensibilité, accessibilité, utilisabilité et duplication [37, 55, 56,57].

Notons que ces approches abordent la qualité des données en considérant : la qualité de la représentation des données (modèle conceptuelle), la qualité de la gestion des données (processus de traitement), et la qualité des données (instances et valeurs).

Approches	**Catégorie**	**Dimensions**
théorique		Exactitude, Fiabilité, Ponctualité, Complétude, cohérence
empiriques	Intrinsèque	Crédibilité, Exactitude, Objectivité, Réputation
	Contextuelle	Valeur ajoutée, Relevance, Ponctualité, Quantité de données appropriée
	Représentative	Interopérabilité, Compréhension, Cohérence, Représentation concise
	Accessibilité	Accessibilité, Sécurité d'accès
intuitives	Contenu de la donnée	Exactitude, Complétude, Actualité, Cohérence
	Format de la donnée	Adéquation, Compréhension, Portabilité, Précision, flexibilité, Capacité, Cohérence
spécifiques		Actualité, Instabilité, Ponctualité

Tableau 2 : Quelques approches proposées de modélisation de la qualité des données [56].

Afin de modéliser les dimensions, diverses extensions des modèles ont été proposées en accord avec différents types de données existantes. Les modèles conceptuels (entité-objet et relationnel) et logiques ont également été étendus à la description de la qualité des données avec des valeurs qualité associées à chaque attribut donnant comme résultat un modèle des attributs qualité [59].

L'inconvénient major des approches proposées de modélisation de la qualité des données est qu'elles ne laissent que peu de choix à l'utilisateur, sans pour autant l'aider à construire des critères, dimensions et métriques de qualité ou bien l'assister dans leur spécification. D'une manière générale, nous résumons les limites de ces travaux dans les points suivants :

- Ces approches sont limitées dans leur applicabilité. Elles sont utiles seulement dans le domaine pour lequel elles ont été conçues ainsi la réutilisation de la définition de la qualité est limitée. En effet la majorité des modèles incorporent des critères de qualité les plus appropriés à leur domaine cible.
- La majorité des définitions proposées de la qualité des données ne distinguent pas le point de vue utilisateur et le point de vue système. Par exemple pour la fraîcheur des données, nous distinguons la fraîcheur comme un point de vue utilisateur et la fréquence.

Dimension	Définition
Traçabilité	Documentation détaillée et historique de la conception et de l'évolution du modèle conceptuel des données.
Exactitude	Quantité de valeurs correctes et sans erreur
Ponctualité (fraicheur)	Ensemble des facteurs qui capturent le caractère récent et le caractère d'actualité d'une donnée entre l'instant où elle a été extraite ou créée dans la base et l'instant où elle est présentée à l'utilisateur
Complétude	L'habilité d'une donnée à représenter le monde réelle.
Cohérence	Quantité de valeurs satisfaisant l'ensemble des contraintes ou règles de gestion définies
Compréhensibilité	La documentation et les métadonnées qui sont disponibles pour interpréter correctement la signification et les propriétés des sources des données.
Accessibilité	Ensemble des facteurs sur l'aptitude du système à rendre les données consultables et manipulables dans des temps adéquats
Utilisabilité	Mesure l'effectivité, l'efficacité, la salification avec lesquelles les utilisateurs spécifies perçoivent et utilisent les données.
Duplication	Mesure la redondance des données

Tableau 3 : Dimensions récurrents de la qualité des données [59, 60].

4.2.5. Évaluation et amélioration de la qualité des données

Les travaux de recherche actuels dans le domaine de la qualité des données se concentrent sur la définition, la sélection, l'adéquation et l'application des méthodes d'évaluation et d'amélioration de la qualité des données [36, 43, 61]. De ce fait les différentes méthodologies existantes divergent selon les dimensions utilisées, les diverses phases ou étapes méthodologiques, ainsi que les stratégies et techniques utilisées. Généralement, ces méthodologies comprennent deux phases complémentaires : l'évaluation et l'amélioration de la qualité des données. L'évaluation et l'amélioration de la qualité des données reposent sur des techniques d'audit et de suivi de données (incluant, par exemple, le recensement des différents

types d'erreurs, l'élaboration de méthodes pour les détecter, l'estimation de leur fréquence d'occurrence dans la base, etc.) [55, 62, 63].

a) Audit des données

L'objectif d'un tel audit est d'obtenir une vision claire de la qualité des données présentes en bases. L'audit des données met en œuvre des programmes chargés de déterminer l'état actuel de la qualité des données. L'attention ne se focalise pas exclusivement sur les données mais aussi sur les exigences des utilisateurs ainsi que sur les processus de création des données. L'audit de la qualité des données est structuré de façon modulaire, chaque module étant centré sur aspect spécifique [16, 63, 64]. Cependant, cet outil s'attache à un nombre limité d'indicateurs à la fois. La mise en œuvre de l'outil d'audit complet est très onéreuse.

b) Évaluation de la qualité des données

L'évaluation de la qualité des données implique : (1) la sélection des facteurs de qualité à évaluer, (2) la sélection des métriques, (3) l'implémentation des algorithmes pour évaluer ces facteurs et (4) l'exécution des algorithmes pour mesurer la qualité des données produites pour le système [57, 59, 65].

c) Amélioration de la qualité des données

L'amélioration de la qualité des données passe par la mise en place d'une initiative permanant, continue et globale. Cette tâche consiste en la sélection de stratégies, processus et techniques pour cibler de nouvelles contraintes en termes de qualité. L'utilisation des métadonnées pour l'amélioration de la qualité est souvent mise en évidence, car les métadonnées permettent de stocker des informations complémentaires pour comprendre et évaluer les données [66]. Pour améliorer la qualité des données, des aspects comme l'identification des erreurs et leurs causes, des techniques de contrôle de qualité, etc., sont prises en compte [8, 67].

Toutes les méthodes d'amélioration de la qualité des données comprennent plusieurs étapes qui sont généralement regroupées en deux processus complémentaires : la détection des problèmes de la qualité des données (erreurs) et le nettoyage des données [68]. Cependant, dans les travaux récents sur la qualité des données, l'amélioration et le nettoyage sont utilisées d'une manière interchangeable.

5. Qualité des connaissances

Un des objectifs de notre travail est de proposer une approche méthodologique pour exploiter des connaissances du domaine et des connaissances expertes dans l'évaluation et l'amélioration de la qualité des données. Le principal goulot d'étranglement de la conception des approches à base de connaissance est la collecte et la formalisation de ces connaissances et de leur qualité.

Ainsi, dans cette section, nous présentons les différents aspects liés à la gestion et l'ingénierie des connaissances, et leur assistance dans les différentes applications (Systèmes ou méthodes à base de connaissances).

5.1. Typologie de connaissances

Plusieurs définitions peuvent être trouvées dans la littérature pour définir la connaissance [13, 69, 70]. Ces définitions s'accordent sur le fait que la connaissance est :

- Le produit de trois ressources : expérience, informations et savoirs acquis.
- Evolutive, corrective, adaptative et relative.
- Extraite (semi automatique) à partir des bases des données et/ou à partir d'autres ressources telle que les utilisateurs.
- Réutilisable

Les connaissances se manifestent sous différentes formes. Toutefois, la segmentation dichotomique la plus utilisée en sciences repose sur la distinction entre formes explicite et tacite. Les connaissances explicites sont des connaissances formalisées et codifiées. Elles sont de natures conceptuelles et abstraites, ce que leur permet ainsi d'avoir un large champ d'exploitation, avec toutefois une nécessaire adaptation au contexte. Les connaissances tacites ne sont pas formalisées et sont difficilement transmissibles. Ce sont les compétences, les expériences, etc. De ce fait la plupart des travaux de recherche s'intéresse aux connaissances explicites qui sont généralement transformables sous forme de règles [71-74].

5.2. Ingénierie des connaissances

Comme le montre la figure 6, l'ingénierie des connaissances concerne l'acquisition, la formalisation, la validation, le stockage, la diffusion, la manipulation et la maintenance des connaissances et des savoir-faire généralement détenus par les acteurs (souvent experts) d'une organisation dans un domaine donné. Les deux principales limites rencontrées par les systèmes issus d'une gestion de connaissances concernent leur déploiement dans les applications, souvent difficile, et leurs capacités à évoluer afin de maintenir des connaissances à jour, souvent faibles [15, 23, 76-78].

L'ingénierie de connaissance au début a été perçue comme un processus de transfert des connaissances qui consiste à transférer et transformer des expertises de résolution de problèmes en des programmes pour construire des Systèmes Experts (SE). Les Systèmes à Base de Connaissances (SBC) qui ont succédé les SE sont censés permettre l'extraction de la connaissance à partir des experts et de leur formalisation à l'aide de règles de production de la forme " si A alors B" et de leur modélisation selon différents points de vue [13, 71].

A la suite des développements des applications basées sur l'ingénierie des connaissances, la question de la modélisation et de l'acquisition des connaissances pour ces systèmes était apparue comme cruciale et problématique : la question de l'acquisition des connaissances justifia de nombreux travaux, que ce soit avec des problématiques très cognitives ou plus orientés vers des questions de niveau de représentation [73, 79, 80].

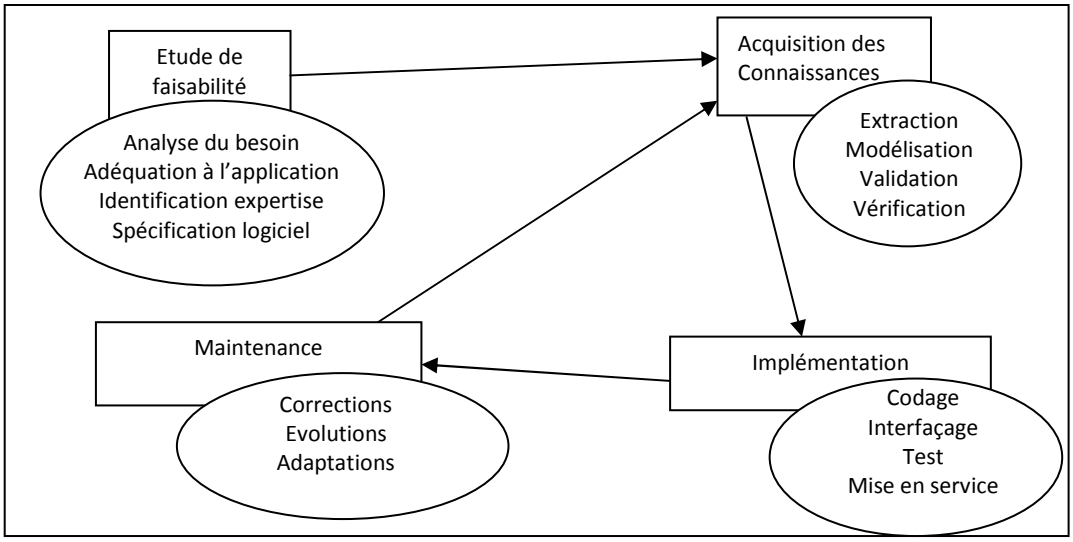

Figure 6 : Ingénierie des connaissances (cycle de vie) [23].

5.2.1. Acquisition des connaissances

L'acquisition des connaissances constitue la première phase de la conception d'un système d'information. Elle repose principalement sur une appropriation des méthodes et des informations manipulées par les experts d'un domaine. Le transfert de connaissances s'avère bien souvent empirique et il est difficile d'automatiser une telle tâche [43, 81].

a) Acteurs de l'acquisition des connaissances

Les acteurs du processus d'acquisition des connaissances sont [21]:

- Expert : spécialiste du domaine; Il détient les connaissances hétérogènes à modéliser ainsi que l'expérience et le savoir-faire,
- le cogniticien : spécialiste en représentation des connaissances et expert du domaine étudiée; il recueille, analyse et formalise les connaissances de l'expert du domaine.

b) Modes d'acquisition des connaissances

Les méthodes d'acquisition des connaissances peuvent être hiérarchiquement classées ainsi [13, 21, 79, 82]:

- La pratique du métier de l'expert : le cogniticien exerce lui-même le métier qu'il se propose de modéliser,
- L'observation de l'expert dans le cadre de sa fonction : le cogniticien se propose de suivre l'expert dans l'exercice de ses fonctions afin de recueillir les connaissances tacites de l'expert,
- Les interviews d'experts : le cogniticien pose les questions directement à l'expert,
- Les questionnaires : le cogniticien laisse une série de questions à l'expert qui répondra en temps voulu,

- Les tests ou les simulations : les connaissances tacites sont déduites à travers une série de tests de situation que le cogniticien propose à l'expert,
- La bibliographie : le cogniticien étudie tous les ouvrages et articles parus dans le domaine,
- Les données : le cogniticien déduit les connaissances tacites des données recueillies ou déduites par l'expert. Le cogniticien doit être un expert du domaine étudié.

Le processus d'acquisition des connaissances nécessite également de nombreuses discussions entre le cogniticien et un voire plusieurs experts du domaine lors des premières étapes d'identification et de conceptualisation. Ce processus constitue un élément essentiel de notre proposition.

Les travaux en acquisition des connaissances se sont clairement divisés en deux groupes ayant des objectifs relativement différents mais toutefois complémentaires : le transfert d'expertise à l'aide de techniques d'élicitation des connaissances[8] et l'acquisition de connaissances par apprentissage automatique[9] [83, 84].

5.2.2. Formalisation des connaissances

La représentation des connaissances revient à établir une correspondance entre le monde extérieur et un formalisme symbolique qui peut être traité par un ordinateur. Le domaine de la connaissance est trop vaste et varié pour être représenté et exploité par un formalisme unique. De ce fait plusieurs représentations sont utilisées. Leurs avantages réciproques sont surtout techniques [70].

La figure 7 présente les divers formalismes allant de procédural (plus structurée) au plus déclaratif (plus ouvert). Ces nombreux formalismes de représentation et de raisonnement ont été mis au point pour prendre en comptes les natures diverses et variées des connaissances. Les formalismes les plus couramment utilisés sont celles basées sur la logique : logiques propositionnelle (d'ordre 0) et prédicative (d'ordre 1) [85-87], le langage Prolog, les règles de production et les logiques de descriptions [88]. Il existe bien sûr d'autres formalismes utilisables dans différentes applications informatiques basées sur les objets, les frames, les scripts (décrivant une séquence d'événements), les réseaux sémantiques (réseaux de concepts liés par des associations) et les graphes conceptuels [70].

La représentation des connaissances par des règles est très répandue. Elle est proche de la formulation naturelle des raisonnements et est donc facile à utiliser. Les règles permettent de représenter des connaissances dynamiques. La syntaxe de représentation des règles est la suivante [72]: SI Prémisse(s) ALORS Conséquence(s).

[8] En anglais: Knowledge Elicitation
[9] En anglais: Machine Learning

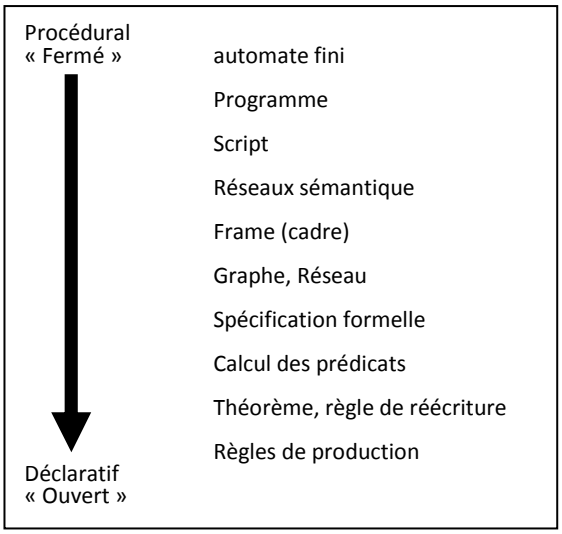

Figure 7 : Classification des formalismes de représentation des connaissances [70].

5.2.3. Système à base de connaissances

Un SBC est défini comme étant un système interactif qui utilise une représentation explicite des connaissances spécifique à un domaine donné et met en œuvre des procédures de raisonnement sur ces connaissances pour résoudre des problèmes du domaine. L'architecture générique d'un SBC se compose d'un SGBD connecté à un SBC [76, 82, 81, 89]:

- Le système gère des bases de données sous forme de relations générales, et met à la disposition des experts un ensemble d'outils de mise à jour des connaissances.
- Les bases de connaissances composées d'un ensemble de descriptions d'objets ou règles, contiennent l'expertise dans des domaines d'application. Chaque module peut être spécialisé dans un domaine plus précis.
- Les interfaces graphiques pour les utilisateurs.

L'avantage des SBC est que la base des connaissances est séparée de code qui exécute la connaissance sur les données [80 92]. Des différentes applications basées sur les SBCs ont été développées et ont démontré l'importance des SBCs dans des domaines aussi variés que la médecine, mécanique, l'enseignement, ... et pour des tâches différentes tels que tâches d'analyse (diagnostic, classification, assessement, supervision, prédiction, réparation de pannes et agencement, etc.) et les tâches de synthèse (conception/configuration, modélisation, planification, ordonnancement, réparation, etc.) [81, 89].

Cependant les travaux de recherche ont recensé un nombre de défis inhérents à la gestion des connaissances, et plus spécifiquement aux SBCs [82, 83] :

1. Les difficultés essentielles des SBCs sont liées aux caractéristiques des connaissances:
    - Elles sont implicites et nécessitent donc de définir des méthodes d'analyse de l'activité pour y accéder.
    - Elles sont évolutives et les modèles doivent pouvoir être mis à jour.

- Elles sont contextualisées, ce qui implique de prendre en compte les conditions dans lesquelles elles sont utilisées.

2. La difficulté de spécifier en avant la totalité des connaissances nécessaires à cause de la nature incrémentale du processus d'élicitation des connaissances [82].
3. L'acquisition des connaissances est coûteuse en termes de temps de construction [91].

5.2.4. Système à base de règles

Comme nous l'avons souligné, la représentation des connaissances par des règles est le formalisme le plus répandu. De ce fait dans cette section nous détaillons les Systèmes à Base de Règles (SBR) qui sont des SBC où la connaissance est décrite par une règle [76, 79].

On parle aussi aujourd'hui de systèmes à base de règles métier (BRMS = Business Rules Management System) ou encore de moteurs de règles « métier » [93].

a) Définition et types de règles

En informatique, une règle est une description de haut niveau permettant de contrôler et/ou de prendre une décision dans une entreprise ou une organisation. Ainsi, les règles décrivent ce que l'on doit faire, c'est-à-dire l'expertise « métier ». Ces règles peuvent être définies sous la forme de règles simples (du type SI << Conditions>> ALORS <<Actions>>), de tables de décision, ou encore d'arbres de décision [44, 94].

Les principaux types de règles que l'on rencontre sont les règles de dérivation, de contraintes d'intégrité, de production et de transformation. Les règles de dérivation, de contraintes, d'intégrité et de réaction ont plus de sens pour les experts métier alors que les règles de transformation et de production ont plus de sens pour des experts technique ou système [72, 74, 95, 102]. La figure 8 décrit selon la notation UML toute ces types de règles ainsi que les langages à base de règles les plus répandus pour chaque type (ILOG JRule, BlazeAdvisorRule, Jess, etc.).

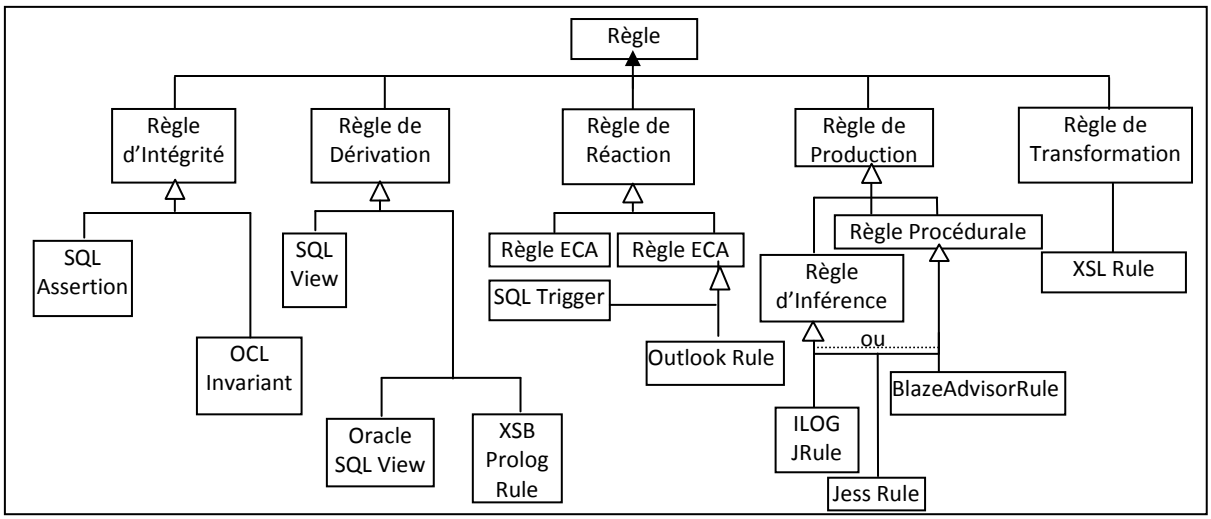

Figure 8 : Classification des règles [74].

La figure 9 présente le métamodèle des règles en utilisant une notation basée sur UML. Les entités Conclusion, Condition et post condition sont des formulations logiques et exactement des conjonctions des formulations logiques élémentaires [96]. En se basant sur ce métamodèle qu'est la base des différents travaux des recherches dans le domaine de l'ingénierie des connaissances et des bases des données, nous pouvons définir chaque type de règle comme suit [74, 102]:

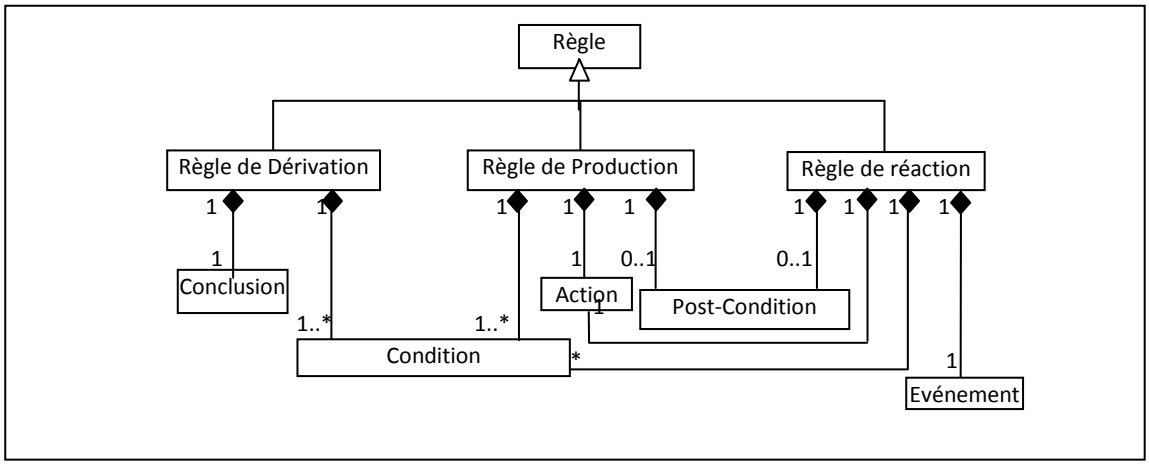

Figure 9 : Métamodèle UML des règles [72].

- Les règles ou contraintes d'intégrité : Ce sont les règles relatives aux données et connaissances incorporées au modèle afin d'assurer la cohérence de ces données. Elles portent généralement sur un attribut ou plusieurs attributs et peuvent concerner soit son format soit son domaine. Les langages les plus connus pour créer ces règles sont SQL et OCL. Ces règles peuvent être représentées par des règles actives ECA (Even-Condition-Action). Le principe de ces règles est la suivante : "lorsqu'un événement se produit, si une condition est remplie, alors une action est exécutée". La partie action décrit les traitements à réaliser suite à la production d'un événement et à l'évaluation à vrai de la condition.
- Les règles de dérivation sont construites d'une ou plusieurs conditions et d'une seule conclusion.
- Les règles de réaction : Ces règles sont constituées d'un événement déclencheur, d'une condition, d'une action et d'une éventuelle Post-Condition. Son principe est le suivant : lorsqu'un événement se produit avec la satisfaction de ses conditions alors une ou plusieurs actions seront déclenchées avec éventuelle satisfaction d'un Post-Condition. Les règles sans Post-Condition sont dites Event-Condition-Action (ECA) par contre celles avec Post-Condition sont dites : ECAP (voir figure 8).
- Les règles de production : ces règles comportent généralement une condition et une action. Ce sont les règles les plus utilisées avec les SBR.
- Les règles de transformation : Ce sont des règles de réécriture qui comportent généralement trois parties. La première partie pour les données à transformer, la

deuxième partie représente une condition et finalement la partie de transformation qu'est similaire à une action.

Vue l'importance de cette partie pour notre travail, nous synthétisons dans le tableau 4 la forme de chaque type de règles [92, 79, 88, 96, 102].

Type de Règle	Sous type de Règle	Forme
Réaction	ECA	**Quand** <Evénement> **Si** <Condition> **Alors** < Action>
	ECAP	**Quand** <Evénement> **Si** <Condition> **alors** < Action> Faire <Post-Condition>
Dérivation		**Si** <Condition> **Alors** <Conclusion>
Production		**Si** <condition> **Alors** <Action>
Intégrité		----------
Transformation		<Données>, **Si** <condition> **Alors** <Transformation>

Tableau 4 : Formes des règles.

b) Présentation d'un SBR

Comme le montre la figure 10, un SBR comporte trois modules et deux bases séparés et interdépendantes (cette séparation permet de respecter le caractère déclaratif de la représentation). La base de règles contient les connaissances du domaine mémorisées sous forme de règles (le plus souvent de production) interdépendantes (aucune règle ne fait appel à l'autre). La base de faits contient les données relatives au problème à résoudre. Le Moteur d'Inférence (MI) permet d'enchaîner et de contrôler les cycles d'applications des règles, en partant des données pour atteindre la solution du problème. Le cycle d'un MI est à trois temps : Sélection des règles applicables, Choix d'une règle et déclenchement d'une règle. Le choix d'une règle passe en deux temps : le filtrage qui permet de déterminer l'ensemble des règles en conflits puis la résolution qui permet au système de choisir les règles applicables. Le système explicatif qu'est une aide indispensable pour l'acquisition et la mise au point de la base de règles (resp. de connaissances) [90, 91]. D'une manière générale un SBR tel que CLIPS et JESS mis à la disposition des utilisateurs des outils pour programmer ou introduire des règles dans la base de règles. Cependant un SBR ne permet pas de générer des règles à partir des données et ne gère pas les règles incertaines [97]. D'autre part, la plupart des SBRs ne permettent pas de traiter tous les types de règles. Généralement chaque SBR gère un type donné de règles [80, 98].

c) Langages à base de règles

La plupart des langages à base de règles se repose sur des différentes versions de l'algorithme de Rete, pour l'amélioration de leurs performances. Cet algorithme évite d'appliquer les prémisses de toutes les règles sur les objets. Parmi ces langages nous citons : CLIPS, Jess, Drools, Ilog Rules, OpenRules, Gensym's, BlazeAdvisorRule. Selon le cas ces langages peuvent être : expert system shell, système de gestion de règles métiers et système de règles de production. Ces langages sont fondés sur la spécification JSR-94 qui décrit l'API entre Java et les systèmes de règle et sont orientés vers les règles de production. Cependant la plupart de ces langages ne sont que des interprètes des règles et des fois même des éditeurs de règles [99, 100]. Aussi bien ces langages ne permettent pas l'échange des règles entre eux ou avec des systèmes

hétérogènes (manque d'interopérabilité). Un autre inconvénient est qu'il n'existe jusqu'à présent aucun format de représentation des règles (même pour un type donné) qui soit une norme de facto [98].

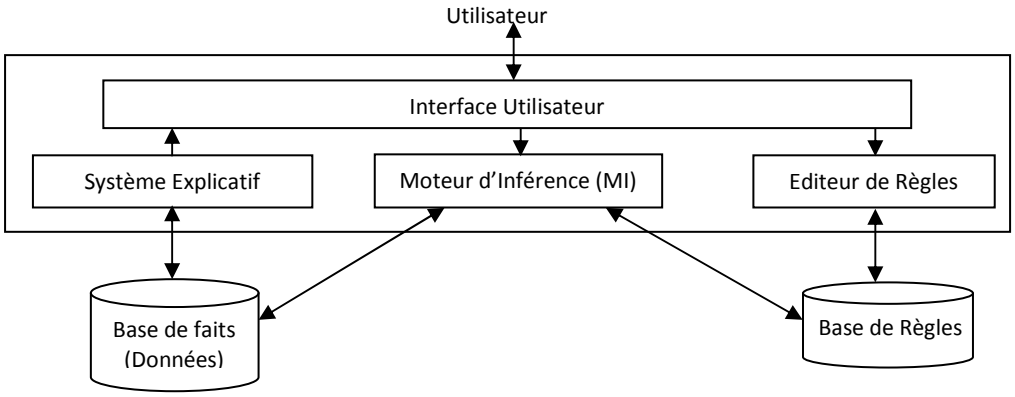

Figure 10 : Architecture générique d'un système à base de règles [100].

Cependant, certaines initiatives telles que Rule Markage language (en français, Langage de Marquage de règles) RuleML, Rule Interchange Format (RIF) et Semantics Business Vocabulary and Business Rules(SVBR) augurent d'une meilleur interopérabilité des règles dans le futur en proposant un langage standard de représentation de règles basée sur XML/RDF pour permettre leur échange dans le Web Sémantique et le Web Service [88, 102].

Le langage de marquage des règles RuleML offre un standard de règles et une plate-forme d'interopérabilité intégrant plusieurs langages de règle. Cependant, il ne crée pas qu'un seul standard mais crée une hiérarchie de sous langages dans laquelle les règles sont organises en cinq niveaux et plusieurs catégories. De ce fait, nous pouvons dire que le problème de développement d'un formalisme standard pour la représentation des règles qu'est l'un de nos objectifs dans ce travail n'est pas encore bien étudié.

5.3. Gestion de la qualité et l'ingénierie de connaissances

Bien que la qualité des connaissances conditionne la réussite des SBCs, peu de travaux ont été menés sur ce sujet. Ces travaux s'accordent sur le fait que le développement de bonnes connaissances implique une assimilation des évaluations critiques de la qualité de la connaissance et l'application des critères articulés les uns aux autres [9, 103].

5.3.1. Classification des travaux de recherche

La plupart des travaux de recherche utilisent des mesures subjectives liées aux intérêts des experts pour évaluation de ces connaissances parce que la connaissance est orientée humaine et aussi contextuelle. Une connaissance peut être valide dans un domaine et contradictoire dans un autre domaine [20, 104-107].

Les travaux de recherche sur la qualité des connaissances (ou control de la connaissance) se concentrent sur l'analyse et l'évaluation de la qualité des connaissances [103]. Cependant, nous n'avons pas constaté des travaux sur l'amélioration de la qualité des connaissances.

Comme une connaissance peut être acquise à l'aide des techniques d'élicitation des connaissances ou par apprentissage automatique, la gestion de la qualité varie d'une technique à une autre.

5.3.2. Gestion de la qualité pendant l'acquisition des connaissances

La plupart des outils d'acquisition des connaissances tels que : Acquire, CommonKADS (Common Knowledge Acquisition and Documentation structuring), Epistimics, Expect (Integrated Environment for Knowledge Acquisition) et Protégé2000 se reposent sur des méthodes pas à pas pour l'ingénierie des connaissances qui impliquent les experts du domaine dans la détermination de la qualité de ces connaissances [77].

Quelque soient l'outil et le domaine d'application, les critères de qualité des connaissances les plus fréquemment mentionnées dans la littérature, sont : Exactitude, Complétude, Fraicheur, Cohérence, Utilité individuelle, Simplicité, Nouveauté, Conformité, Formalité et Utilité Collective dont les définitions générales sont données dans le tableau 5 [10, 103, 107]. Certaines critères sont subjectives et d'autres sont intersubjectives. Pendant l'évaluation, Les experts sélectionnent un échantillon de faits ou des données pour mesurer les critères de qualité. En pratique, les méthodes d'évaluation se focalisent sur la complétude et l'exactitude [108-110]. Cependant, ces travaux ne tiennent pas compte de la multiplicité et la diversité des avis des experts et utilisateurs dans les méthodes de calcul des mesures. Notons bien que ces critères sont valables pour les connaissances implicites ou explicites [111, 112].

5.3.3. Gestion de la qualité pendant l'apprentissage automatique

Les travaux de recherche sur la qualité des connaissances au cours de l'apprentissage automatique s'intéressent généralement aux connaissances explicites et plus spécifiquement les règles. Ces travaux se concentrent aussi sur l'analyse et l'évaluation de la qualité des connaissances en définissant des différents critères de qualité. Ainsi, Ils proposent :

- Des critères d'intérêt et leurs méthodes de mesure.
- Des algorithmes et des méthodes pour la détection de l'inconsistance, redondance et non terminaison des connaissances dans les bases de connaissances.

a) Critères et mesures de qualité des connaissances

La plupart des travaux de recherche se focalisent sur la détermination des critères d'intérêt des règles. Dans le tableau 6, nous citons les critères le plus fréquemment utilisés dans les travaux de recherche [18, 64, 103, 104]. Ces mesures évaluent des connaissances selon différents point de vue et rejettent celles qui sont trop mauvaises en utilisant un seuil de qualité minimal [106, 114].

Le support et la confiance constituent les deux mesures les plus communément utilisées pour évaluer des règles. Tout d'abord parce qu'ils sont grandement intelligibles. Ensuite parce qu'ils sont à la base des algorithmes d'extraction des connaissances ou règles [114]. Nous allons donner les définitions le plus fréquemment utilisées dans ces travaux au support et à la confiance d'une règle.

Critères de qualité	Définition	Nature
Exactitude	C'est le degré de vérité d'une connaissance.	subjective
Complétude	Toutes les informations nécessaires pour la connaissance sont renseignées	subjective
Fraîcheur	C'est le degré d'actualité d'une connaissance	subjective
Cohérence	Une connaissance ne doit pas être contradictoire (avec elle-même ou avec autre connaissance)	subjective
Utilité Individuelle	C'est le degré par lequel une connaissance augmente la capacité de l'utilisateur de cette connaissance à atteindre ses buts ou à résoudre ses problèmes	subjective
Simplicité	Une connaissance doit être facile à apprendre	subjective
Nouveauté	Les connaissances nouvelles et inattendues tendent à attirer les attentions et donc stimule l'énergie cognitive ce qui facilite leur assimilation.	subjective
Formalité	Une connaissance doit être interprétée de la même manière partout. La définition d'une connaissance doit se reposer sur un langage complètement débarrassée de toute ambiguïté.	intersubjective
Conformité	C'est un critère pour qu'un groupe coopère à partir du moment où la connaissance acquise est vue comme étant collectivement utile.	intersubjective
Utilité Collective	Certaines formes de connaissance sont un avantage pour le collectif mais sont inutiles pour l'individu.	intersubjective

Tableau 5 : Définition des critères de qualité des connaissances.

Le support évalue la généralité d'une règle. Il s'agit de la proportion d'individus qui vérifient la règle (n_{ab}) dans le jeu de données (n) [18, 110, 114] :

$$\text{Support (règle } (a \rightarrow b) = {}^{n_{ab}}/_{n}$$

La confiance évalue la validité d'une règle. Il s'agit de la proportion d'individus qui vérifient la conclusion (n_{ab}) parmi ceux qui vérifient la prémisse (n_a) :

$$\text{Confiance (règle } (a \rightarrow b) = {}^{n_{ab}}/_{n_a}$$

En se basant sur les limites de ces deux indices qui ne permettent d'évaluer que certains critères de la qualité des connaissances, des différents travaux de recherche ont préposés de nombreux indices de qualité pour compléter le support et la confiance [35, 44, 113, 115-117].

b) Détection des anomalies

Des différentes solutions (algorithmes et méthodes) sont proposées pour la détection de certaines anomalies de bases des connaissances. Ces solutions peuvent être appliquées sur la base des connaissances ou bien sur les règles pendant et/ou avant leurs insertions dans la base de

connaissances. Les anomalies le plus fréquemment menés par ces travaux de recherche sont : Duplication (Redondance) des règles, Inconsistances, Circularité (non terminaison) et Utilisabilité [18, 35, 103, 112].

Critères d'intérêt	Définition	Mesures d'évaluation	Nature
Généralité	Elle exprime la généralité d'une règle	Support	Objective
Validité	Elle exprime la justesse de la règle.	Confiance	Objective
lisibilité	Une connaissance doit être compréhensible partout et pour tout		Subjective
Nouveauté	Il exprime le degré de la valeur ajoutée par la connaissance		Subjective
Surprise	Toute règle inconnue ou contradictoire		Subjective
Utilité	Il exprime le gain attendu d'une règle.		Subjective

Tableau 6 : Critères de qualité des connaissances et des règles [103].

5.4. Limites des travaux de recherche sur la gestion de la connaissance

Nous pouvons résumer les limites des travaux courants dans le domaine de l'ingénierie des connaissances dans les points suivants:

- Bien que l'élicitation des connaissances et l'apprentissage automatique soient complémentaires, nous n'avons pas constaté des travaux qui tiennent compte de synergie élicitation-apprentissage automatique.
- Manque d'une méthode de gestion totale te continue de la qualité des connaissances. Généralement les travaux s'intéressent au post-traitement de la qualité de la connaissance.
- Transformation des indices de mesure de qualité des connaissances en parallèle avec les transformations des connaissances en règles. Cela permis d'exploiter l'historique de la qualité des connaissances.
- Manque d'un formalisme uniforme et unifie pour la représentation des règles et de leur qualité.
- Les travaux de recherche sur la qualité ne tiennent pas compte des propriétés qui distinguent les connaissances des données au profit de l'amélioration de leur qualité.
- Manque d'interactivité : Malgré que la qualité est centrée humaine, les travaux ne actuels permettent l'incorporation de l'utilisateur dans les différentes étapes du processus de gestion de la qualité des données.
- Exploitation des connaissances elles-mêmes pour l'amélioration de la qualité.

6. Nettoyage des données

Dans cette section, nous présentons le processus de nettoyage des données ainsi que les travaux réalisés.

6.1. Présentation du processus de nettoyage des données

Le nettoyage des données fait partie des stratégies d'amélioration automatique de la qualité des données [33, 34, 52]. Le problème de nettoyage des données qui consiste à détecter et éventuellement corriger des incohérences et des erreurs trouvées dans des jeux des données originaux, est bien connu dans le domaine de l'aide à la décision et des bases des données [118, 119]. La figure 11 montre que le nettoyage des données est un processus itératif et interactif qui comporte trois phases.

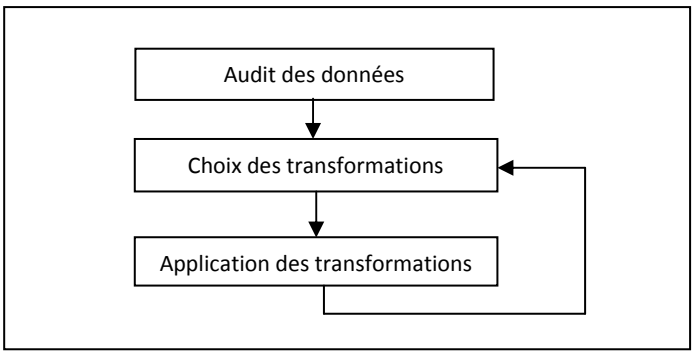

Figure 11: Processus du nettoyage des données [39].

La phase de l'audit des données consiste à vérifier l'état des données afin de détecter les anomalies (problèmes de la qualité des données). L'utilisateur ou le système sélectionne l'ensemble des transformations à exécuter sur les données de mauvaise qualité pendant la phase de choix des transformations. Pendant la dernière phase, le système applique les transformations sur les données. Si les résultats du nettoyage sont inacceptables, le processus remet en cause le choix des transformations. L'implication de l'utilisateur peut être soit explicite, soit implicite. Dans le cadre d'une implication explicite, l'utilisateur doit effectuer des interactions directes avec le système tandis que lors d'une implication implicite, le système s'adapte automatiquement à l'utilisateur [16, 118].

Le nettoyage des données traite deux catégories de problèmes. La première catégorie concerne les problèmes liés aux données erronés tels que l'inconsistance des données, les erreurs de frappes, les valeurs manquantes, les fautes d'orthographe, données obsolètes, outliers (hors-norme), les valeurs incorrectes, les valeurs aberrantes, les valeurs nulles, etc. La deuxième catégorie concerne le problème de détection et élimination des doublons exacts et des données approximativement redondantes (déduplication) qu'est considéré dans la plupart des travaux de recherche comme un axe de recherche à part entière [36, 61]. Notons bien que certains travaux considèrent l'étape de standardisation et normalisation des données comme une opération du nettoyage des données appliquées au début de la phase de l'application des transformations [65].

La plupart des travaux de recherche se focalisent sur les problèmes de déduplication des données et le les problèmes de nettoyage des conflits des noms et des adresses [33, 35].

6.2. Etude de quelques travaux de recherche sur le nettoyage des données

La recherche bibliographique que nous avons réalisée sur les travaux de recherche dans le contexte du nettoyage des données [16, 35, 56, 59, 65, 68, 120-123], nous a permis de constater que les outils de nettoyage des données sont classifiés selon leur fonctionnalité comme suit : le nettoyage déclaratif et le nettoyage à base de règles.

6.2.1. Nettoyage déclaratif

Les outils du nettoyage déclaratif des données permettent aux utilisateurs de modéliser leur propres processus de nettoyage des données. Ces outils se basent sur l'utilisation d'un modèle d'exécution, d'un langage de spécification déclaratif et des algorithmes qui permettent aux utilisateurs d'exprimer les spécifications de nettoyage des données de façon déclarative [16, 56, 65, 120]. Par conséquent, ces outils sont indépendants du domaine d'application. Parmi les outils académiques de nettoyage des données déclaratif, citons AJAX et Potter's Wheel à partir de les quelles sont inspirés la plupart des outils. AJAX par exemple est une extension du langage déclaratif SQL permettant de spécifier chaque transformation des données (à l'aide des opérateurs de matching, merging, mapping et clustering) nécessaire au processus de nettoyage des données [118]. Potter's WHEEL est un outil interactif permettant à l'utilisateur de visualiser les diverses règles de nettoyage applicable sur les données et ainsi l'utilisateur peut modifier ces règles afin que le nettoyage devienne plus précis [125, 126].

6.2.2. Nettoyage à base de règles

Partant de l'idée que les connaissances de domaine sont modélisables sous forme de règles et peuvent être utilisées pour résoudre des problèmes du domaine et effectuer du raisonnement, certains travaux de recherche ont établi des outils de nettoyage des données à base de règles. IntelliClean est le premier système de nettoyage à base de règles proposé dans la littérature de nettoyage des données [124]. Dans ce système, les règles sont soit générées automatiquement, soit spécifiées par l'utilisateur afin de personnaliser le nettoyage des données. IntelliClean concentre sur la détection des données redondantes. Il formalise les connaissances sous forme de règles de production de la forme: "SI <conditions> ALORS <action>" et sont écrites en langage JESS [101]. IntelliClean comporte trois phases. La phase de prétraitement qui consiste à la standardisation, conversion et normalisation des données. La phase de traitement qui permet la détection des enregistrements redondants. Pendant la phase de validation et vérification, l'utilisateur intervient pour vérifier les enregistrements redondants non détectées ainsi que pour valider les résultats de la deuxième phase. Si des enregistrements redondants sont validés faux par l'utilisateur alors le système remet en cause les règles [36, 65,125]. Cependant, l'analyse de ces travaux de recherche, nous a permis à identifier certaines limites qui sont:

- La majorité des outils à base de règles ne disposent pas d'un module d'acquisition des connaissances.
- La validation et la vérification des résultats de nettoyage par les utilisateurs sont couteuses en termes de temps.

- La validation et la vérification des résultats de nettoyage par les utilisateurs peuvent introduire des erreurs.
- Ces outils ne disposent pas d'un système de gestion des règles du nettoyage et de leur qualité malgré que la qualité des résultats de nettoyage est fortement conditionnée par la qualité des règles.

6.3. Étude comparative de quelques outils du nettoyage des données

Pour affiner la lecture des outils et systèmes portant sur le nettoyage des données le plus fréquemment utilisés, nous proposons une grille d'analyse basée sur des critères de comparaison que sont jugés pertinents par les travaux de recherche menés sur le nettoyage des données [127]. Dans le tableau 7, nous décrivons brièvement la définition de chaque critère de comparaison.

Critère	Définition
Interactivité	Il exprime la capacité du système d'impliquer l'utilisateur dans le nettoyage des données.
Interopérabilité	Elle désigne la capacité du système d'échanger les données et les opérations de nettoyage des données avec d'autres systèmes.
Optimisation	Il exprime la capacité du système d'utiliser et d'employer les techniques du traitement parallèle et du partitionnement des données afin d'optimiser le temps du nettoyage.
Base de règle	Elle indique si le système utilise une base de règle.
Connaissances du Domaine	Elle indique si le système intègre un système de gestion des connaissances (acquisition).
Déclarative	Elle indique si le système suit l'approche du nettoyage déclaratif.
Standardisation et Normalisation	Elle indique si le système réalise en premier les opérations de base de transformations des données : normalisation, standardisation et conversion.
Nettoyage des anomalies	Il indique si le système détecte et probablement corrige les données erronées (sauf la redondance).
Détection des Duplicates	Elle indique si le système détecte les données redondantes.
Enrichissements des Données	Elle indique si le système peut exporter des données externes pour enrichir les métadonnées des données.
Audit des Données	Elle indique si le système évalue l'état actuel de la qualité des données.
Analyse des Données	Elle indique si le système est doté des opérations d'analyse ces données.

Tableau 7 : Critères de comparaison des outils de nettoyage des données.

Basé sur l'étude comparative et l'exploration de la littérature de nettoyage des données que nous avons effectuées [127, 128], nous avons caractérisé dans tableau 8 chacun des outils de

nettoyage de données en les positionnant par rapport aux critères définis dans le tableau 7. L'analyse des résultats présentés dans le tableau 8, nous a permis de constater que :

- Peu des outils et systèmes qui sont interactifs (33 % des travaux). Cette interactivité est fonctionnelle (faible) car l'utilisateur doit uniquement choisir ou visualiser des opérations préétablies par les concepteurs.
- Rareté des systèmes et outils qui permettent l'interopérabilité (8% des travaux).
- La plupart des travaux s'inspirent de l'approche déclarative (83% des travaux). Par contre les outils à base de règles ou connaissances représentent 17% des travaux.
- Les systèmes qui exploitent le parallélisme des processus et la répartition des données ne représentent que 18 % des travaux.
- La plupart des travaux traite le problème des données redondantes (détection et correction des duplicates).
- Mettre en œuvre les processus de nettoyage des données complet est un véritable défi. De fait que la plupart des outils dépendent du domaine de l'application (médical, finance, etc.) et sont généralement appropriés à des problèmes spécifiques de qualité.

L'approche de nettoyage déclaratif des données -qu'est le plus souvent ad hoc, spécialisé et fragmenté- offre de bonnes performances dans certains nombreux domaines d'applications (nettoyages des conflits des noms, des adresses, email et numéros de téléphone) mais qui ne sont pas systématiquement utilisables pour d'autres domaines. En outre, les travaux de recherche ont démontré que les outils inspirés de cette approche ne sont pas structurés. Cependant, les outils inspirés de l'approche à base de règle qui sont systématiques et structurés offrent de façon exhaustive toutes les solutions possibles [9]. Cela correspond aux propriétés d'efficacité, exactitude, actualité, cohérence et de complétude des SBRs [90].

6.4. Incorporation des utilisateurs dans le nettoyage des données

Comme nous l'avons souligné auparavant, l'interactivité des systèmes et outils existants est fonctionnelle ou autrement dit faible. Cela veut dire que l'utilisateur ne peut pas écrire ces propres opérations de détection ou correction des problèmes de qualité des données. De ce fait, certains travaux de recherche se sont penchés sur le problème de l'interactivité dans les systèmes du nettoyage des données [45, 119, 129-132]. Partant de l'idée que les outils de nettoyage des données peuvent introduire des erreurs ainsi que certains problèmes de qualité doivent être corrigé manuellement, les auteurs de ces travaux proposent un support pour l'implication de l'utilisateur dans le nettoyage des données. Leur idée consiste à l'implication de l'utilisateur dans la vérification des résultats intermédiaires durant le nettoyage afin de détecter et possiblement réparer les résultats intermédiaires erronés. Néanmoins, cette solution est difficile car généralement les utilisateurs finaux ne peuvent pas évaluer les résultats intermédiaires.

6.5. Limitations des outils du nettoyage à base de règles

Même si les outils de nettoyage de données basés sur les règles permettent de traiter en pratique un nombre restreint d'applications, elles montrent néanmoins quelques limites :

	Interactivité	Interopérabilité	Optimisation	à base de règle	Connaissances du Domaine	déclarative	Standardisation et Normalisation	Nettoyage des anomalies	Détection des duplicate	enrichissement des données	Audit des Données	Analyse des Données
AJAX [118]	N	N	N	N	N	O	O	O	O	O	N	N
Potter's Wheel [125,126]	O	N	O	N	N	O	O	N	N	N	N	O
IntelliClean[124]	N	N	N	O	N	N	O	N	O	N	N	N
FraQL [56]	N	N	I	N	N	O	O	O	O	I	N	N
ARKTOS [59]	O	N	O	N	N	O	O	O	N	N	N	N
Bellman [56]	N	N	N	N	N	O	N	N	O	N	O	O
Febrl [133]	O	N	I	N	N	N	O	N	O	N	O	O
Talend Open Studio [56]	N	N	N	N	N	N	O	O	O	N	N	N
D-Dupe [56]	N	N	N	N	N	O	N	N	O	N	N	N
ODCF [63]	I	N	N	O	O	N	N	O	O	N	O	N
IQ driven DC [41]	O	N	N	O	N	O	N	N	O	N	N	N
Ontology based DC [120]	N	O	N	N	O	O	N	O	O	N	N	N
Grammar Based DC [134]	N	N	N	O	N	O	O	N	O	N	N	N
N : non, O : oui, I : Inconnu, DC : Data Cleaning, IQ: Information Quality												

Tableau 8 : Comparaison des outils de nettoyage de données.

- Ces systèmes ne disposent pas d'un système de gestion de connaissances : Une règle est une connaissance transformée. Cette connaissance est souvent d'origine expérimentale et heuristiques. Donc, l'incorporation d'un système de gestion des connaissances et des règles dans le processus de nettoyage des données est indispensable afin de rendre le nettoyage des données plus évolutif, interactif et extensible. Cela nécessite la mise en place des approches, des méthodes et des techniques d'identification, de représentation et d'opérationnalisation de connaissances.
- Ces systèmes ne permettent pas la gestion de qualité des règles : La qualité des résultats de nettoyage des données est conditionnée par la qualité des règles et des connaissances. Les systèmes existants tel que IntelliClean -qu'est la base des travaux à base de connaissance- consacre la dernière phase pour la validation et la vérification des résultats de nettoyage. Cela veut dire que l'utilisateur répète les mêmes tâches faites détection et correction des erreurs (redoublants dans le cas d'IntelliClean) à chaque nettoyage. Il balaye la même base des données pour rechercher des redoublants non détectées par le système. De ce fait cette tâche est couteuse en termes de temps surtout dans le cas des bases des données volumineuses. Cela est dû principalement à l'absence d'une gestion de qualité des règles dans ces systèmes. De ce fait vient la nécessité d'incorporation de la gestion de la qualité dans les systèmes de gestion de connaissances. L'objectif attendu de cette incorporation est d'éviter la vérification et la validation des résultats de nettoyage des données.
- Les systèmes actuels à base de règles permettent uniquement la transformation des connaissances sous forme des règles de production. Cependant les connaissances doivent être transformables sous forme des règles des différents types. Cela nous ramène à dire que ces système sont incomplets. D'où la nécessité d'adapter ces systèmes à tout type de règles. Par exemple IntelliClean consacre la première phase à la transformation des données (la standardisation et la normalisation) par des programmes ad hoc [118]. Par conséquent, si nous voulons ajouter ou mettre à jour une opération de transformation, nous devons mettre en cause ces programmes de transformation des données. Pourtant, la représentation des transformations par des règles permettent de mettre à jour uniquement ou ajouter la règle en question.
- Très fréquemment, le nettoyage des données à base de règles est un processus non systématique traité cas par cas, de façon ponctuelle. Non systématique veut dire qu'il n'existe pas de processus établi qui puisse tendre vers un nettoyage des données complet.

7. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur la qualité des données et des connaissances dont l'objectif essentiel est d'incorporer explicitement un système de gestion des connaissances et leur qualité dans les systèmes de nettoyage des données à base de règles.

Après l'identification des limitations des différents travaux et approches dans les différents domaines nécessaires à l'amélioration de la qualité des données, nous avons pu

démontrer que la réalisation d'un système de gestion totale et continue de la qualité des données nécessité l'adaptation des processus d'acquisition des connaissances et de nettoyage des données.

Nous avons aussi démontré la nécessité d'un formalisme unifie et uniforme pour la représentation de tout type de règles. Cela est important car les SBRs actuels permettent uniquement la représentation des règles de production.

L'état de l'art que nous avons présenté dans ce chapitre sur la gestion des données et des connaissances et de leur qualité est très utile dans la première phase de notre travail, à savoir les outils d'acquisition des connaissances et la gestion de leur qualité ainsi que le nettoyage des données à base de règles. Nous avons montré que la mise en œuvre d'un SBRs dans les systèmes d'amélioration de la qualité est fondamental pour une gestion totale te continue de la qualité. Cependant, il est nécessaire d'adapter les processus d'acquisition des connaissances les SBRs afin d'être incorporable dans les systèmes de gestion de la qualité des données.

Comme l'acquisition des connaissances peut être réalisée par élicitation ou par apprentissage automatique, le second chapitre est consacré à l'étude de l'extraction automatique des connaissances.

Chapitre 2: Extraction des Connaissances à partir des Données(ECD)

« La connaissance sans la sagesse, est de l'intelligence artificielle.»
Juliana M. Pavelka

1. Introduction

L'Extraction des connaissances à partir des données constitue un champ de recherche important dans lequel de nombreux problèmes restent à résoudre. La mesure de la qualité des connaissances extraites est une étape clef qui a donné lieu à de nombreux travaux de recherche. Ces travaux ont démontré que la qualité de connaissances est fortement liée à la qualité des données à partir des quelles sont extraites ces connaissances. L'ECD importe des données à partir des différentes sources de production des données et les stockent généralement dans des EDs où les données sont prêtes pour la fouille par les algorithmes de fouille de données.

Généralement, le processus d'ECD, sous la supervision d'un spécialiste, se déroule en quatre phases : l'acquisition des données, le prétraitement et mise en forme des données, la fouille de données, et finalement la validation et la mise en forme des connaissances produites. Cependant l'entreposage des données a pour objet d'organiser des très grands volumes de données, de les structurer et de les préparer à l'analyse en les stockant dans les EDs qui sont des composants fondamentaux des processus d'ECD.

L'ECD et l'ED sont vus comme des supports complémentaires nécessaires à la réalisation d'une architecture qualifiée de décisionnelle. Dans cette architecture le processus d'entreposage des données est en amont du processus d'ECD. Certes, l'entreposage des données peut se concevoir comme un processus isolé mais elle est également une étape essentielle du processus d'ECD. Et c'est suite à ces deux constats que beaucoup de travaux de recherche se focalisent sur l'organisation de la synergie et la complémentarité des ces deux processus qui ont un défi commun qu'est l'amélioration de la qualité des données. Cela nécessité une étude détaillée des deux processus et des aspects et technologies associées.

De ce fait dans ce chapitre, nous présentons en détails ces deux processus et nous situons le problème de l'amélioration de la qualité des données et des connaissances par rapport à ces deux processus. Nous évoquerons aussi rapidement les technologies et les aspects associés à ces deux processus et que nous avons jugés utiles à la gestion de la qualité des données et des connaissances : Processus d'Extraction, Transformation et Chargement des données (ETC), Réplication des données dans les EDs, Traçabilité des Données, Capteur des changements des données et Entreposage des règles et des connaissances.

2. Processus d'extraction des connaissances à partir des données

L'ECD est une nécessité imposée par le besoin des entreprises de valoriser les données qu'elles accumulent dans leurs bases. En effet, le développement des capacités de stockage et les vitesses de transmission des réseaux ont conduit les utilisateurs à accumuler de plus en plus de données. Certains experts estiment que le volume des données double tous les ans [135, 136]. Il

existe dés lors un très grand intérêt à développer des techniques permettent d'utiliser au mieux tous ces stocks de données afin d'extraire un maximum de connaissances utiles.

2.1. ECD et fouille des données

La Fouille de données est une activité consistant à extraire des informations nouvelles, implicites, non triviales, inconnues auparavant et potentiellement utiles dans des données volumineuses, souvent sous la forme de régularités : des motifs cachés, des tendances et relations inattendues, etc. La FD est souvent confondu par des non-spécialistes avec l'ECD. Il existe deux visions différentes pour le processus d'ECD : vision académique et vision industrielle. Pour chaque vision, nous pouvons trouver des multiples architectures de ce processus [1-3, 78, 137-139]. Nous avons synthétisé ces architectures dans la figure 12 qui décrit une architecture générique pour l'ECD.

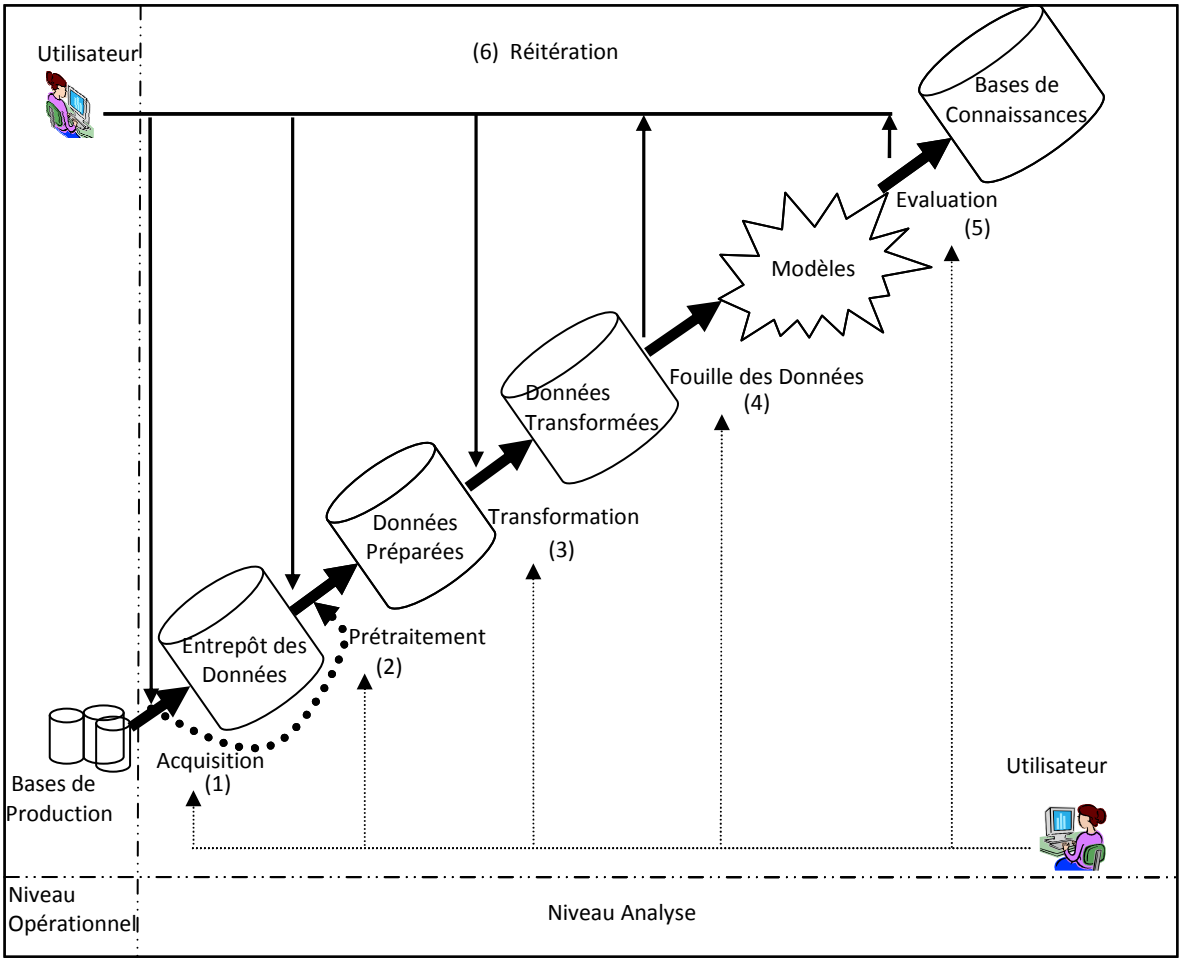

Figure 12 : Processus d'extraction des connaissances à partir des données (ECD).

Les différents travaux de recherche que nous avons constatés s'accordent sur le fait que [29, 137, 139, 140]:

a) L'ECD est un processus semi-automatique, itératif et interactif mettant en œuvre des algorithmes de recherche de régularités dans les bases de données dans le but ultime de

découvrir des connaissances intéressantes, significatives et réutilisables, et dont la FD n'est qu'une étape (voir figure 12).

b) L'utilisateur fait partie intégrante du processus: L'interactivité est liée aux différents choix que l'utilisateur est amené à effectuer. L'interactivité est liée au fait que l'utilisateur peut décider de revenir en arrière à tout moment si les résultats ne lui conviennent pas car l'extraction des connaissances est dirigée par l'objectif de l'utilisateur.

2.2. Présentation du processus d'ECD

L'ECD se réfère à une démarche complète -qui comporte cinq étapes- d'exploitation des données intégrant leur prétraitement pour permettre l'application des algorithmes de FD suivie de la validation des modèles obtenus parvenant ainsi au stade de connaissances [5, 138, 141, 142]. En pratique, ce n'est pas toujours le cas. En effet, dans de nombreux travaux, certaines de ces étapes sont fusionnées. De ce fait, nous voyons l'ECD en trois étapes principales : Préparation des données, Découvertes des connaissances et Evaluation-Réitération [143].

L'utilisation industrielle ou opérationnelle de l'ECD permet de résoudre des problèmes très divers, allant de la gestion de la relation client à la maintenance préventive, en passant par la détection de fraudes ou encore l'optimisation de sites web.

2.2.1. Préparation des données

Les connaissances produites à partir de données de mauvaise qualité ont des conséquences graves sur les décisions prises par les utilisateurs, quelque soit le domaine d'application. Pour cela, la qualité des données et des connaissances est un sujet d'intérêt dans le processus d'ECD. Toutes les applications dédiées à l'ECD requièrent différentes formes de préparation des données avec de nombreuses techniques de traitement, afin que les données passées en entrée aux algorithmes de FD ne contenant pas d'incohérences, de doublons, de valeurs manquantes ou incorrectes.

a) Vue générale sur le processus de préparation des données

La figure 12 montre que pendant la préparation des données, le processus d'ECD s'intéresse à (1) l'acquisition des données où il procède à la sélection, l'intégration et le nettoyage des données cibles, (2) le prétraitement des données où l'ECD exécute des opérations complémentaires du nettoyage et sélection des données, et (3) la sélection et transformation des données où l'ECD réalise les opérations standards de normalisation, de standardisation et de conversion des données [78, 136,139]. La préparation est déterminante pour l'ECD car les données sont attachées souvent d'erreurs (des valeurs manquantes, des valeurs inconsistantes, des valeurs incomplètes, des valeurs nulles, des valeurs aberrantes, des duplicates, etc.).

La préparation de données occupe environ 60 à 80% du temps impliqué dans l'extraction de données. La difficulté de cette préparation peut être comprise à partir de deux perspectives : les problèmes relatifs aux données et les problèmes relatifs au processus [2, 144].

b) Acquisition des données

La sélection vise à cibler de façon grossière, l'espace des données qui va être exploré. La sélection met en œuvre des requêtes ad hoc ou des moteurs de requêtes des bases des données comme le langage SQL pour rapatrier les données potentiellement utiles et pertinentes selon le point du vue de l'utilisateur pour la tâche à accomplir [137, 142].

Le nettoyage des données met en œuvre des techniques d'amélioration de la qualité des données (remplit les valeurs manquantes, élimine les données redondantes et résout les inconsistances). De ce fait, le nettoyage des données consiste à retravailler des données de mauvaise qualité, soit en les supprimant, soit en les modifiant de manière à en tirer le meilleur profit [78, 144].

L'intégration des données résout le problème des données hétérogènes. En effet, les données proviennent souvent de plusieurs sources de données différentes et hétérogènes (bases de données relationnelles, fichiers XML, etc.) où une même donnée peut être référencée de deux manières différentes dans deux sources. L'intégration des données permet d'uniformiser les formats, afin de pouvoir les stocker de manière uniformisée dans un ED, où les données sont prêtes pour la fouille par les algorithmes de FD [139, 144].

c) Prétraitement des données

Il peut parfois que les EDs ou les bases des données contiennent encore des données de mauvaise qualité. Ces données doivent être nettoyées si cela n'est pas été fait précédemment [13, 144].

d) Transformation des données

Une fois les données préparées, il est souvent nécessaire de les transformer de façon à ce que les algorithmes de FD soient plus pertinents. Les données sont transformées ou consolidées dans un format approprié et adéquat à la tâche de FD (Agrégation de certaines valeurs, Normalisation s'autre données, etc.). Enfin, la réduction des données consiste à réduire le volume total des données de façon à ce que les algorithmes disposent d'assez de ressources (mémoire et calculatoire) pour les traiter. La problématique de cette étape est de choisir les données à supprimer telles que la sortie des algorithmes n'en soit pas significativement modifiée, c'est-à-dire supprimer des données particulières en respectant le comportement général des données. La réduction consiste souvent à supprimer des attributs considérés comme non pertinents, agréger certaines valeurs, etc. [3, 139]. Les transformations dépendent du type de FD à entreprendre [1, 2, 78].

2.2.2. Découverte des connaissances.

La deuxième étape du processus d'ECD consiste à l'application des algorithmes de FD sur les données. Il existe différentes tâches de la FD, chacune répondant à un problème différent [18]. Il n'existe pas de technique de FD supérieure à toutes les autres pour tous les problèmes. Il faut choisir une technique en fonction des besoins des utilisateurs ainsi que des avantages et inconvénients de chacune d'elles. Il est possible de combiner plusieurs méthodes et techniques

pour obtenir une solution optimale globale [144]. La question est ici de savoir ce que l'on cherche dans les données. La fouille peut être faite sur l'exhaustivité de la base de données ou sur un échantillon. Ce choix dépend des outils utilisés, de la puissance machine, du budget, etc. [18].

a) Typologie des méthodes de fouille de données

Les différentes méthodes de FD sont classifiées par rapport aux tâches qu'ils permettent d'effectuer et se distinguent les unes des autres suivant le type de données considérées (voir figure 13) [64, 144-146]:

Apprentissage / Tâches	Supervisées	Non supervisées
Segmentation		• K moyennes (K-means) • K plus proches voisins (PPV-raisonnement à partir de cas) • Réseaux de neurones avec cartes de Kohonen • ….
Classification Prédiction	• Arbre de décision • Réseaux de neurones avec perception • Réseaux bayésiens • Machines à Vecteur Supports • Programmation logique Inductive • ….	• K plus proches voisins (PPV-raisonnement à partir de cas) • Règles temporelles • Recherche des séquences • Reconnaissance de formes • …..
Prédiction	• Arbres de décision • Réseaux de neurones	• K plus proches voisins (PPV-raisonnement à partir de cas) • ..
Association		• Règles d'associations

Figure 13 : Quelques méthodes de fouille de données [146].

- La *classification* consiste à examiner les caractéristiques d'un objet et lui attribuer une classe. Ces classes sont préalablement définies. La structure de classification la plus connue est l'arbre de décision. Le problème revient alors à induire un arbre de décision à partir des données. Une classification peut également être une liste de règles de classification ou encore un réseau bayésien.
- La *prédiction* est une tâche qui découle souvent de la classification car une fois la structure permettant la classification générée à partir des données, la prédiction revient à déterminer à quelle classe appartient tout nouvel élément ou à prédire la valeur d'un attribut en fonction d'autres attributs.
- La *catégorisation* ou *Segmentation* sert à générer des groupes significatifs à partir des données de telle façon que la similarité entre les données d'un même groupe et la dissimilarité entre différents groupes soient les plus grandes possibles. La notion de similarité trouve son complément dans la notion de distance qui mesure l'écart dans l'espace [147].
- L'*Association* et la *corrélation* est également une tâche couramment répandue dans la FD. Elle consiste à extraire des données des dépendances (détermination des attributs qui sont

corrélés), par exemple des règles d'association du type : "si $A_1(x)$ et $A_2(x)$…… et $A_n(x)$ alors B(y) où $A_i(x)$ signifie l'objet A_i a la caractéristique x)". Les corrélations peuvent également être formulées sous forme de réseau bayésien.

b) Typologies des modèles

Les structures extraites par les algorithmes de FD se divisent en deux grandes catégories : les motifs et les modèles. Les modèles sont des structures représentant le comportement global des données, à un haut niveau d'abstraction. Les modèles peuvent être prédictifs, comme les arbres de décision, ou descriptifs, comme les clusters de données. Les motifs sont des affirmations au niveau local, c'est-à-dire à propos d'une partie des données ou des variables. Par exemple, une expression régulière, qui peut être utilisée pour décrire un motif séquentiel dans un langage, n'affirme rien sur les données au niveau global, mais uniquement que telle ou telle donnée vérifie ponctuellement la contrainte représentée par l'expression régulière [136,139]. D'une manière générale, les structures peuvent être : Arbres de décision, Règles, Réseaux de neurones, équations mathématiques, regroupements, graphes, histogrammes, etc.

2.2.3. Évaluation et réitération

Pendant cette phase, l'ECD évalue les modèles afin d'extraire des connaissances utiles et intéressantes à l'utilisateur et relance le processus si les résultats sont jugés mauvais.

a) Evaluation des modèles

Cette opération identifie les modèles intéressants représentant les connaissances, en se basant non seulement sur des mesures d'intérêt mais aussi sur l'avis de l'expert. La présentation des résultats à l'utilisateur se fait grâce à des différentes techniques de visualisation. Ce n'est qu'à partir de la présentation qu'on peut employer le terme de connaissance. Il y a principalement deux techniques de validation qui sont la validation statistique et la validation par expertise.

In fine, l'évaluation des modèles issus du processus d'ECD, est généralement effectuée par un spécialiste (expert, analyste, ...). Cette tâche de post-traitement est souvent très lourde et un moyen de la faciliter consiste à aider le spécialiste en lui fournissant des critères de décision sous la forme de mesures de qualité ou d'intérêt des modèles. Ces mesures doivent être conçues afin de combiner deux dimensions : l'une objective liée à la qualité des modèles (statistique), l'autre subjective liée aux intérêts d'experts. Bien que les techniques utilisées en FD et en gestion des connaissances soient très différentes, elles partagent l'objectif de produire des modèles de connaissances pertinents pour les décideurs, avec une préoccupation commune d'évaluation de la qualité des modèles produits [18].

Les connaissances produites s'expriment généralement sous forme d'un concept général qui enrichit le champ sémantique de l'usager par rapport à une question qui le préoccupe. Elles peuvent s'exprimer comme un modèle mathématique ou logique pour la prise de décision. Les modèles explicites, quelle que soit leur forme, peuvent alimenter un SBC [136].

b) Réitération

Par nature le processus d'ECD est itératif guidé par l'utilisateur. À chaque itération, cet utilisateur juge la qualité des résultats extraites par rapport à ses attentes et modifie les algorithmes de FD et les données s'il n'est pas tout à fait satisfait. Il relance ensuite les algorithmes pour tenter d'extraire des informations plus intéressantes. Cette rétroaction de l'utilisateur sur les étapes du processus dans le cycle d'ECD est représentée sur la figure 12 par des flèches en pointillées (étape 6 de la figure 12). Les attentes du décideur sont les intérêts de qualité à travers lesquels les connaissances seront évaluées. Les attentes les plus fréquemment utilisées sont : Lisibilité, Nouveauté (surprise), Validité et utilité [18]. Cependant, il existe d'autres intérêts caractérisant chaque type de connaissances. Ainsi, l'extraction de connaissances peut être vue comme une boucle vertueuse qui s'enrichit à chaque itération de manière explicite ou implicite par les connaissances de l'utilisateur [139].

2.2.4. Gestion de la qualité des connaissances dans l'ECD

Certes, l'offre en matière d'outils d'ECD, et plus spécifiquement de FD est en net accroissement. Toutefois la qualité des connaissances extraites reste un problème difficile et ouvert. De ce fait, l'une des tâches de la FD consiste à élaborer des mesures permettant d''evaluer la qualité des connaissances extraites. Cependant, le problème de la qualité dans l'ECD ne sera pas résolu pendant une seule phase de nettoyage des données. Dans la figure 14, nous avons essayé de positionner la qualité par rapport au processus d'ECD où nous avons constaté qu'elle revienne dans toutes les phases.

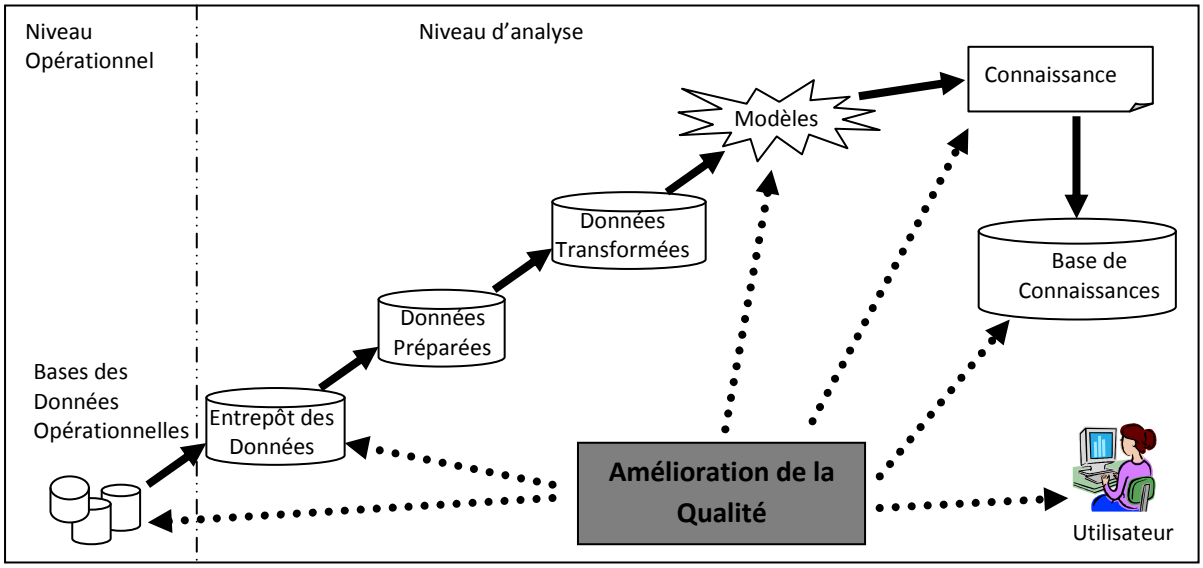

Figure 14 : processus d'ECD et qualité

La figure 14 montre que la qualité dans l'ECD requiert un système de gestion totale de la qualité des données et des connaissances. La recherche bibliographique que nous avons effectuée, nous a permis de constater que les travaux de recherche académiques et industriels actuels ne proposent pas des solutions pour une gestion complète de la qualité dans les ECDs [127, 128, 148]. Sur ce constat, nous déduisons que la gestion de la qualité doit être abordée sous

un angle pluridisciplinaire : bases des données, EDs, les bases de connaissances, FD et a fortiori l'expertise humaine. Cela, nous conduisons à dire que qualité des résultats d'un processus d'ECD est conditionnée par la qualité des éléments suivants : (1) données opérationnelles, (2) ED, (3) modèles extraites par FD, (4) connaissances du domaine, (5) base de connaissances (évolution de la qualité de connaissance), et (6) algorithmes utilisés.

Par conséquent, dans ce travail, nous proposons l'adaptation du processus d'ECD pour une meilleure prise en charge de la qualité des aspects décrits ci-dessus. Les éléments (1) et (2) impliquent l'adaptation du processus d'entreposage des données à l'ECD et à la gestion de la qualité des données. Les éléments (3), (4), (5) et (6) nécessitent l'adaptation de l'ECD pour permettre l'extraction de la connaissance du domaine à partir des différentes sources (humain, revue, etc.) et l'intégration d'un système de gestion des connaissances et de leur qualité.

Notons bien que dans ce travail, nous considérons que l'ED et l'ECD sont des processus complémentaire.

3. Présentation du processus d'entreposage des données

L'Entrepôt des données (ED) est un élément essentiel du processus d'ECD qui conditionne les résultats en aval surtout la qualité des connaissances extraites. Il est aussi en amant du processus d'ECD dans les systèmes d'aide à la décision. Sur ce constat, dans notre travail, nous considérons que l'ECD est étroitement liés à l'ED. De ce fait, cette section décrit les principales étapes et approches de conception d'un ED ainsi que la gestion de leur qualité.

3.1. Entrepôt des données vs entreposage des données.

Le concept d'ED a pris forme au début des années 90, il est devenu depuis la clé de voûte de ce que l'on appelle l'informatique décisionnelle. La définition la plus appropriée de l'ED est : l'ED est une collection de données intégrées, orientées sujet, non volatiles et historisées, organisées pour le support d'un processus d'aide `a la décision. C'est une base de données d'aide à la décision qui est entretenue de manière séparée des sources à partir des quelles est formée [4, 149, 151, 152]. Par contre l'entreposage des données est le processus responsable de la conception, administration, construction et le rafraichissement de l'ED [153, 154]. Le Magasin de Données[10] (MD) est un extrait de l'ED. Ces données extraites sont adaptées à une classe de décideurs ou à un usage particulier [3, 149, 155]. Les MDs peuvent être dépendants ou indépendants de l'ED. Les MDs dépendants sont construits à partir de l'ED grâce aux outils ETC. Cependant les MDs indépendants sont construits directement à partir des sources des données opérationnelles. Dans la littérature des EDs, nous avons constaté des différentes architectures d'entreposage des données [156-159]. Dans la figure 15, nous décrivons une schématisation de l'architecture générique d'un ED qui intègre tous ces composants de base.

L'ED est alimenté grâce à des outils d'Extraction, Transformation et Chargement des données (ETC). Ces outils ont pour vocation d'extraire et de structurer des données en

[10] En anglais Data Mart

provenance des Sources de Données Opérationnelles (SDO) [152, 155]. Cette opération est difficile car les données sont hétérogènes, réparties et complexes. Le processus d'ETC réalise également un nettoyage des données suivi généralement d'une phase d'agrégation au sein des EDs [159].

La plupart des processus ETC stockent les données avant leur chargement dans l'ED dans une base temporaire dite Zone de Préparation des Données[11] (ZPD). La ZPD est une structure intermédiaire (stockage tampon) qui stocke les données issues des SDOs avant leur transformation et leur intégration dans l'ED. Bien souvent le modèle de données de la ZPD est un modèle relationnel classique assez proche des modèles des SDOs [151, 149].

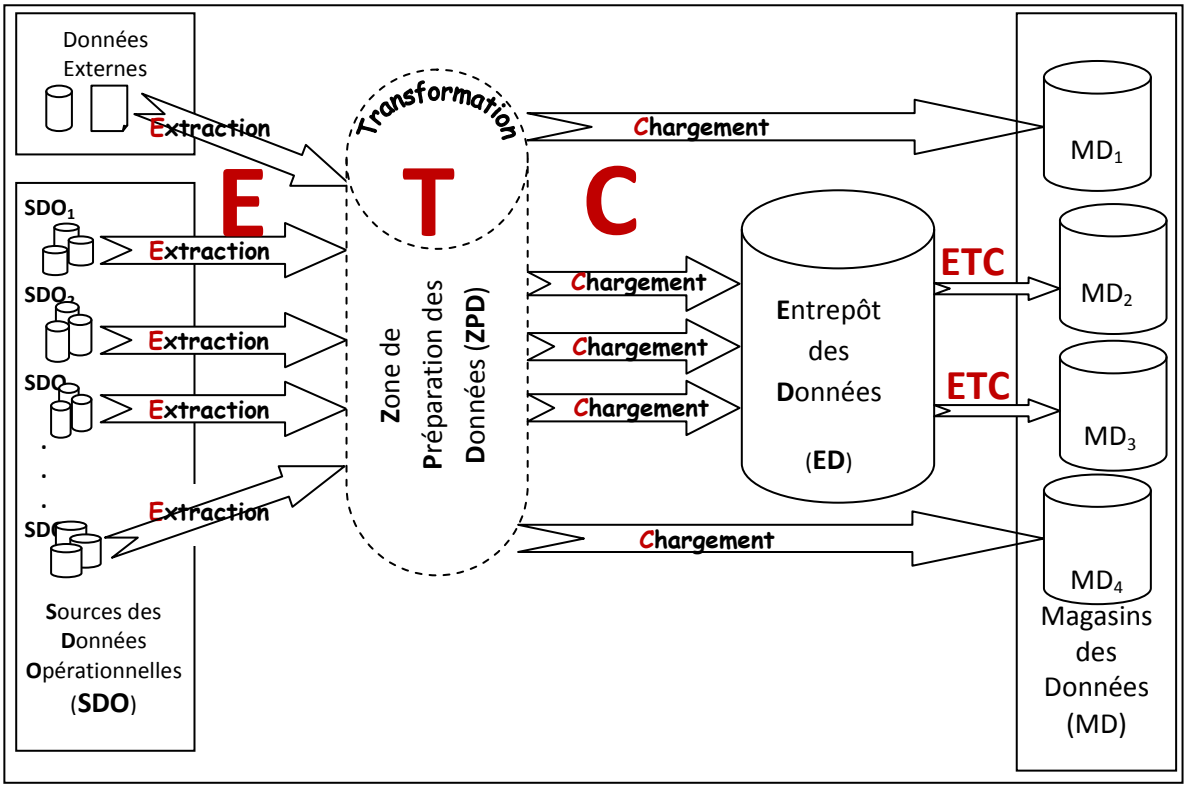

Figure 15 : Processus d'entreposage des données [152].

3.2. Présentation du processus d'extraction, transformation et chargement des données

L'ETC est un outil fondamental du processus d'entreposage des données. Il propose de nombreuses fonctionnalités avancées tels que la gestion de multiples sources de données en entrée et en sortie, et de différents types de sources (bases de données relationnelles, fichiers plats, …), transformations complexes (requêtes d'agrégation, …), parallélisme offrant de meilleures performances... Les interfaces graphiques sont plus accessibles. Les ETC basés sur la génération de code produisent non plus du C ou du COBOL mais du SQL qui peut être exécuté sur de multiples plateformes [5, 160-162].

[11] En anglais, Operational Data Store (ODS)

3.2.1. Extraction des données

L'Extraction assure la connexion à la majorité des SDO_i afin de pouvoir identifier, sélectionner et récupérer les données nécessaires à la formation de l'ED. Elle assure aussi un mécanisme de synchronisation pour la réactualisation. Elle peut s'effectuer selon deux méthodes à l'aide de [17, 143]:

- Déclencheurs [12](méthode push) : Ceux-ci sont déclenchés lors de modifications sur les SDO_i et poussent ces données modifiées vers l'ETC. Cette méthode nécessite d'intervenir au niveau des SDO_i pour mettre en place ces déclencheurs ce qui n'en fait pas la méthode la plus répandue ;
- Requêtes (méthode pull) : l'ETC interroge les SDO_i pour extraire les données. Cette méthode est la plus répandue car elle ne nécessite pas la modification des SDO_i pour mettre en place les processus ETC.

L'extraction peut considérablement charger les systèmes de gestion des SDOi et perturber les applications au niveau de ces SDO_i. Les données extraites sont stockées dans la ZPD. L'extraction doit permettre de se connecter aux SDOi via divers connecteurs : ODBC, JDBC, SQL natif, Fichiers plats ou encore avec des connecteurs spéciaux. Les SDOi et les sources externes peuvent être des SGDB (MySQL, Oracle, SQL Server, etc.), de Fichiers plats (Txt, Excel, XML, etc.) ou d'autres données spécifiques [162].

Pendant l'extraction, les données seront analysées afin de connaitre les propriétés de celles-ci : savoir par exemple si une donnée est de type entier ou chaîne de caractères. Cette opération est simple lorsque les données proviennent des SGBD mais s'avère difficile lorsqu'elle provient des fichiers plats. L'extraction s'occupe aussi de certaines tâches de l'amélioration de la qualité des données surtout les fautes de frappe [160].

3.2.2. Transformation des données

Les transformations (Standardisation, Normalisation, Conversion, Enrichissement des données, Discrétisation, réduction, fusion, filtrage, etc.) sont le cœur de l'ETC. Elles permettent d'effectuer différentes opérations sur les données afin de les concilier et d'obtenir un format qui respecte celui de la ZPD. L'utilisateur définit des correspondances entre les schémas des ODS_i et de l'ED. L'ETC s'appuie sur ces correspondances pour appliquer les transformations nécessaires sur les données et résoudre ainsi l'hétérogénéité sémantique. La tendance actuelle est de proposer une interface graphique permettant de définir visuellement les correspondances. L'amélioration de la qualité des données est effectuée conjointement avec cette phase grâce à un processus du nettoyage des données. Les différentes transformations s'effectuent dans la ZPD. La phase transformation pour les ED doit s'occuper à la fin généralement de différentes agrégations (par exemple effectuer les commandes SQL tels que SUM (somme), COUNT (comptage) ou AVG (moyenne) [17].

[12] En anglais: Triggers

La transformation génère aussi des clés de substitution (Surrogate key). Ces clés sont utilisées pour remplacer les clés primaires qui proviennent des SDO_i. Ces clés accroissent les performances des outils ETC : recherche des données avec une meilleure indexation, historisation des changements des données, et l'indépendance des SDO_i [162].

3.2.3. Chargement des données

Il permet de charger les données obtenues à l'issue de la phase de transformation vers l'ED. Une fois, une donnée est chargée dans l'ED, elle sera automatiquement effacée de la ZPD. Les données chargées doivent être marquées et datées. A tout moment le système peut identifier la provenance de données et connaître leur date de chargement. Le chargement assure aussi la qualité des données (un entier ne doit pas être inséré dans un champ fait pour les caractères) [152, 159].

3.2.4. Rafraîchissement des entrepôts des données

Les actualisations (mises à jour appliquées aussi bien au contenu des données qu'à leur structure) effectuées dans les SDO_i doivent être reportées sur l'ED [163, 164]. De ce fait le processus d'entreposage des données doit permettre, outre l'initiale construction de l'ED, le rafraîchissement de l'ED [121]. Le rafraîchissement de l'ED est un procédé un peu différent de la construction initiale de l'ED. Les données anciennes (déjà présentes dans l'ED) ne seront jamais remplacées par les nouvelles données, mais archivées. L'ajout des nouvelles données ne doit pas modifier les données existantes. Une technique pour garantir la cohérence des données consiste à générer de manière automatique pour les nouvelles données les valeurs des clés des tables de l'ED au moment de leur chargement [15, 166].

Nous avons constaté que le rafraîchissement d'un ED peut être effectué de manière complète (toutes les données des SDO_i sont chargées) ou incrémentale (seules les nouvelles données des SDO_i seront chargées) [154].

Dans le cas d'un rafraîchissement incrémentale, l'ETC doit être capable d'identifier les « nouvelles » données. De ce fait, nous avons remarqué que les systèmes actuels d'ECD intègre un outil qui permet de détecter et récupérer les changements effectuées sur les SDO_i dites *Capture des Changements des Données* [13] [17]. Cet outil exploite les informations suivantes avec des différentes façons [167]:

- Si les données sources sont datées, l'ETC peut se reposer sur ces informations en effectuant des comparaisons de données entre les SDO_i et l'ED,
- Des déclencheurs peuvent être mis en place au niveau des sources de données. Ceux-ci se déclenchent à la mise à jour des données et stockent ainsi les changements effectués dans un espace réservé, et
- Les logs de transactions peuvent être analysés afin de tracer les changements.

[13] En anglais : Change Data Capture (CDC)

3.2.5. Gestion de la qualité des données par l'ETC

Selon les besoins, une phase d'évaluation et d'amélioration des données, plus spécifiquement de nettoyage de données, est réalisée. Celle-ci est effectuée conjointement à la phase de transformation. Elle vise à améliorer la qualité des données à transférer vers l'ED. Les données « à nettoyer » sont : les doublons, les données erronées, les valeurs manquantes, etc. Les techniques employées sont : le rejet des données, le déduplication, l'introduction de valeurs fixes, moyennes, etc.… [161, 168, 169].

3.2.6. Traçabilité des données

La Traçabilité Linéaire des Données[14] (DLT) est une opération très importante pour la traçabilité des données de l'ED. Elle permet de déterminer les données des SDO_i à partir des quelles est formée une donnée de l'ED. La recherche bibliographique que nous avons effectuée sur le sujet de la traçabilité des données dans les EDs, nous a permis de constater que la DLT est modélisé sous la forme d'un graphe des transformations des données. C'est un facteur important pour la vérification, analyse et l'authentification des données des SDO_I [170, 171]. Cependant, le problème majeur que nous avons noté dans les EDs et qui échoue la mise en œuvre des DLTs dans les EDs est que généralement, les transformations sont irréversibles. Par conséquent, nous avons consacré une partie de ce travail à la réalisation d'un modèle de traçabilité des données spécifique aux données nettoyées.

3.2.7. Etude comparative de quelques outils d'ETC

Après avoir présenté les différents concepts liés à l'ED où nous avons concentré sur le processus ETC qui est la colonne vertébrale de l'entreposage des données, nous présentons ici les résultats d'une étude comparative que nous avons effectuée sur quelques outils existants d'ETC [5, 17, 116, 161, 162]. Cette étude nous a permis de distinguer trois groupes de ces outils: ETC maison, ETC propriétaire et ETC source ouverte. Dans le tableau 9, nous décrivons les principaux outils d'ETC source ouverte et propriétaires. Les ETC maisons sont crées par les entreprises pour leurs propres applications. L'avantage de ces outils est qu'ils s'adaptent facilement aux spécificités métiers et aux différents types de données utilisés. Dans la plupart de ces outils, les transformations sont des programmes ad hoc.

ETC source ouverte	ETC propriétaire
Talend Open Studio, Pentaho Data Integration , Clover ETL, Enhydra Octopus, KETL, Scriptella, BenETL, OpenESB, Jitterbit, Apatar	IBM Information Server, InfoSphere DataStage, SAS Data Integration Studio, Oracle Warehouse Builder, (OWB), Sap BusinessObjects Data Integration

Tableau 9 : Quelques outils ETC (Propriétaires et Source-Ouverte).

Ces outils traitent les problèmes de la qualité des données par un ensemble des transformations. Ils utilisent des audits des données pour détecter les données de mauvaise

[14] En Anglais :Data Lineage Tracing (DLT)

qualité et puis les corrigent si possible par des solutions basées sur les statistiques (utilisation de la moyenne, par défaut, la suppression). Ces solutions donnent des données valide mais au même temps peuvent introduire des erreurs (Par exemple la moyenne permet de calculer une valeur valide mais pas exacte) [9]. Cependant, cette étude comparative, nous a permis d'identifier plusieurs problèmes qui rendent difficile la conception et la construction des ETC robustes. Nous résumons ci-dessous les principaux problèmes :

- Il n'existe pas une méthodologie générique (surtout académique) de conception et modélisation du processus ETC.
- Les outils existants d'ETC ne permettent pas une gestion complète de la qualité des données (car ce sont des programmes ad hoc). Ils ne permettent pas par exemple d'introduire des métriques de mesure de la qualité des données.
- Les outils existant n'exploitent pas le parallélisme au cours de la phase de transformation des données, ce que rend l'entreposage des données couteux en termes du temps.
- Les outils existants ne permettent pas l'incorporation des connaissances expertes ou du domaine.
- Les outils existants n'exploitent pas les techniques de FD pour extraire des connaissances à partir des données pour l'auto-amélioration de la qualité des données.
- Les outils existants ne permettent pas la propagation des données corrigées vers leurs sources (SDO_i dans notre cas). Cela est important car il évite de refaire les mêmes opérations d'améliorations de la qualité des données pendant le rafraichissement de l'ED et la construction d'un nouvel ED.
- Les outils existants ne permettent pas la validation des corrections faites sur les données par les utilisateurs.

De ce fait, pour se situer par rapport aux travaux existants de gestion de la qualité des données et des connaissances, nous proposons une nouvelle architecture pour le processus ETC pour prévenir aux problèmes décrits ci-dessus. Cette proposition part de l'idée que la plupart des transformations faites pendant l'ETC sont formalisable sous forme de règle où chaque transformation se déclenche suite à une événement ou condition. Ainsi, nous jugeons utile la réalisation d'un ETC à base de règles. Les avantages et les bénéfices attendues de cette proposition sont multiples: une interactivité forte, l'évolution et l'extensibilité des ETC.

4. Présentation de quelques travaux de recherche sur la qualité dans l'ECD

La figure 14 que nous avons établie pour situer la qualité par rapport au processus d'ECD, montre que la qualité concerne les données et les connaissances et qu'elle doit être gérée tout au long du processus. Cependant, la plupart des travaux de recherche se focalisent sur l'amélioration de la qualité des données pendant la transformation des données et validation des connaissances extraites [1, 75, 144].

4.1. Gestion de la qualité des données dans l'ECD

Malgré que la qualité des données conditionne la qualité des connaissances extraites dans le processus d'ECD et vice versa, nous avons constaté que les travaux de recherche dans le domaine de la qualité dans les ECD (respectivement l'ED) se limitent uniquement à l'application de quelques transformations (nettoyage de données) basées sur les statistiques pour détecter et corriger quelques erreurs : valeurs manquantes, données incomplètes, doublons, données erronées, valeurs aberrantes, données incohérentes. De plus, les EDs dans l'ECD ne se forment pas via du processus ETC mais généralement par le biais des requêtes ad hoc ou des moteurs de requêtes (query engine) des bases des données comme le langage SQL [10, 136]. Ce constat confirme notre proposition d'adaptation de l'ETC au processus ECD.

L'exploration des travaux de recherche que nous avons effectuée, nous a permis d'identifier très peu de travaux génériques qui traitent la problématique de la qualité des données dans l'ECD. Ces travaux s'appuient sur l'intégration des métadonnées de qualité des données au processus d'ECD par fusion des indicateurs renseignant la qualité des connaissances extraites [18]. Cependant, ces travaux :

- gèrent la qualité des données pendant leur l'acquisition à partir des SDO et la qualité des connaissances après leur transformation sous forme de règle d'association. De ce fait, nous pouvons constater que ces travaux sont spécifiques aux règles d'association. En outre, ils ne permettent pas une gestion de la qualité tout au long du processus ECD.
- négligent l'amélioration de la qualité des données et des connaissances.
- évaluent la qualité des données par le biais des indicateurs de la qualité des connaissances qui sont déjà formées à partir de ces données mêmes. Cette méthode peut aggraver la situation de la qualité si la qualité des données originales est mauvaise. Pour pallier ce déficit, il s'avère donc important d'évaluer et d'assurer préalablement la qualité des données originales.

4.2. Gestion de la qualité des connaissances dans l'ECD

Nous discuterons brièvement dans cette section les travaux de recherche sur la gestion de la qualité des connaissances ainsi que les problèmes associés.

4.2.1. Etude de quelques travaux de recherche

Comme nous l'avons évoqué dans le premier chapitre, les travaux de recherche s'intéressent généralement à la détermination des intérêts et des métriques de qualité pour l'évaluation de la qualité des connaissances et plus spécifiquement les règles d'association. Ces métriques sont généralement ad hoc selon les besoins d'une application et/ou d'un usage [56]. Nous avons constaté que l'objectif principal de ces travaux n'est pas la qualité au sens large mais c'est pour diminuer le nombre de connaissances extraites qui est généralement élevé surtout dans le cas des règles d'association [94, 162]. L'exploration de quelques travaux de recherche, nous a permis de constater que ces travaux s'intéressent généralement à quatre dimensions de qualité pour l'évaluation des connaissances qui sont: Lisibilité, Nouveauté, Validité et utilité. Ces mesures sont calculables à partir de trois critères objectifs de qualité (Généralité, Validité et

Fiabilité) et quatre critères subjectifs (sens commun, actionabilité, nouveauté et surprise) [56, 68]. Ces travaux partent du fait que le choix d'une bonne mesure d'évaluation de la qualité des connaissances extraites est la clé du succès d'un projet d'ECD. A cause de la concentration de la plupart des travaux de recherche sur les règles d'association, nous nous présentons brièvement ces travaux.

Du fait des grandes quantités de règles que produisent les algorithmes de FD, le post-traitement est une étape nécessaire mais difficile dans un processus de recherche de règles d'association. Il consiste en une seconde opération de fouille, mais alors que la FD est réalisée automatiquement, la fouille de règles est généralement laissée à la charge de l'utilisateur. En pratique, il est très laborieux pour ce dernier de rechercher des règles intéressantes dans les listes de règles obtenues à la sortie des algorithmes. Différentes solutions ont été proposées pour aider l'utilisateur dans sa tâche. On distingue trois formes de solutions [106, 108, 116, 168]:

- De nombreux indices de qualité ont été développés afin d'évaluer les règles selon différents points de vue. Ils permettent à l'utilisateur d'identifier et de rejeter les règles d'association de faible qualité, mais aussi d'ordonner les règles acceptables des meilleures aux plus mauvaises.
- Une autre solution consiste à organiser une exploration interactive des règles pour l'utilisateur. Plusieurs logiciels et langages de requêtes ont été conçus dans cette optique.
- La tâche de l'utilisateur peut également être facilitée en lui soumettant des représentations visuelles des règles. Elles facilitent la compréhension et accélèrent l'appropriation des règles par l'utilisateur.

Malgré ces différents travaux, plusieurs problèmes demeurent. Tout d'abord, les indices de qualité sont nombreux et souvent redondants entre eux. La signification de ces indices n'est pas non plus très claire fréquemment pour l'utilisateur. D'une manière générale, il est difficile de choisir quels indices à appliquer. Ensuite, l'interactivité dans le post-traitement de règles d'association est souvent faible: les interactions ne sont pas pleinement adaptées à la tâche de l'utilisateur, et en particulier elles ne tiennent pas compte de la spécificité des données, c'est-à-dire le fait qu'il s'agisse de règles. De ce fait pour mieux prendre en considération l'utilisateur, nous proposons que le processus d'ECD doit être considéré non pas sous l'angle de la FD mais sous l'angle de l'utilisateur, comme un système d'aide à la décision centré sur l'utilisateur.

4.2.2. Inconvénients des travaux de recherche sur la qualité des connaissances dans l'ECD

Les travaux de recherche sur la qualité des connaissances dans l'ECD présentent certains inconvénients que nous résumons ci-dessous:

- On constate la domination de l'approche Généralité-Validité (Support-Confiance dans le cas des règles d'association) qui recherchent de façon exhaustive les connaissances dont les indices de qualité dépassent des seuils fixés préalablement par l'utilisateur. Cela peut conduire à rejeter des connaissances de bonne qualité
- L'étude bibliographique que nous avons effectuée, nous a permis de constater que l'évaluation de la qualité des connaissances extraites dépend étroitement de la technique

de FD utilisée (règles d'associations, arbre de décision, etc.) [10]. Basé sur ce constat, l'un des objectifs visé ans cette thèse est la mise en œuvre d'un système générique d'évaluation de la qualité des connaissances indépendamment des algorithmes de FD.

- Ces travaux ne permettent pas l'amélioration de la qualité des connaissances (ils acceptent ou bien rejettent la connaissance (élagage des connaissances)) en utilisant les deux mesures confiance et support ou des mesures calculables généralement à partir de ces mesures. Cela peut conduire à l'utilisation des règles de mauvaise qualité et aussi à élaguer des règles de bonne qualité. Donc il est important de gérer la qualité des règles d'une manière continue et interactive.
- Malgré que les donnée et les connaissances sont deux concepts différentes, les travaux de recherche sur la gestion de la qualité existants traite la connaissance comme une donné, et indépendamment des données à partir des quelles cette connaissance est formée. C'est-à-dire qu'ils ne détiennent pas compte des propriétés qui caractérisent les connaissances des données.
- Les travaux de recherche se sont penchés sur les règles d'associations malgré que la connaissance puisse être décrite sous différentes formes (règles d'association, équations, etc.). Cela est dû principalement à la facilité de l'automatisation (algorithmes) de l'évaluation de la qualité (la plupart des mesures sont basées sur le support et la confiance). L'évaluation des autres types de connaissances est effectuée par des spécialistes. C'est une tâche très couteuse en termes de temps et moyen humains.
- Manque des méthodologies de la gestion totale et continue de la qualité des connaissances et des données dans l'ECD: les travaux proposés ne permettent pas la gestion de la qualité tout au long du processus ECD.

5. Présentation de quelques travaux de recherche sur la qualité dans l'ED

Comme l'objectif de ce travail est la gestion de la qualité des données dans le processus d'ECD en tenant compte de la complémentarité de l'ECD et de l'ED, nous nous intéressons dans cette section aux principales méthodologies et aux quelques travaux de recherche sur la qualité des données dans les EDs et de savoir si elles sont adaptables à l'ECD.

5.1. Principales méthodologies de gestion de la qualité de l'ED

En général, les travaux existants académiques ou industriels sur la qualité des données dans les EDs suivent deux méthodologies : Qualité de l'Entrepôt des Données[15] (QED) et Gestion totale de la qualité des données[16] (TDQM). Cette dernière, nous l'avons présenté en détail dans le premier chapitre. Ces deux méthodologies s'inspirent de l'approche But-Question-Métrique[17] (GQM) qu'est utilisé pour capturer les corrélations entre les différentes métriques de qualité identifiées pour atteindre des objectifs de qualité particuliers [173].

[15] En anglais: Data Warehouse Quality (DWQ)
[16] En anglais: Total Data Quality Management (TDQM)
[17] En anglais: Goal-Question-Metric (GQM)

5.1.1. Présentation de l'approche GQM

C'est une approche d'identification des métriques qui suit une démarche en trois phases qui sont : (1) Enumération des objectifs principaux du projet, (2) Dérivation des questions pour chaque objectif, et (3) Sélection de ce qui doit être mesuré pour répondre aux questions [174].

5.1.2. Présentation de la méthodologie DWQ

Le DWQ est un projet centré sur la modélisation formelle de la qualité de données pour optimiser la conception d'un ED intégrant la gestion des métadonnées de qualité. Le DWQ s'appuie sur des modèles formels pour la qualité. Les résultats comportent des méta-modèles de données formels destinés à la description de l'architecture statique d'un ED. Les outils associés comportent des facilités de modélisation incluant des caractéristiques spécifiques aux EDs comme la résolution de sources multiples, la gestion de données multidimensionnelles (éventuellement agrégées) et des techniques pour l'optimisation de requêtes et la propagation incrémentale des mises à jour [158].

Nous avons identifié deux stratégies qui peuvent être suivies pour la mise en œuvre de la méthodologie DWQ : Gestion de la qualité des données proactive[18] (ProDQM) et Gestion de la qualité des données réactive[19] (ReaDQM). La stratégie ReaDQM (orienté symptômes) se concentre sur les données et vise à l'amélioration de leur qualité par détection et correction des erreurs pendant l'acquisition des données. Cependant, la stratégie ProDQM (Orientée causes) analyse les causes de la déficience de la qualité des données et ensuite implémente un système d'amélioration continue de la qualité des données afin de prévenir à la non qualité des données [51, 175, 179, 181].

5.2. Etude de quelques travaux de recherche sur la qualité des EDs

L'avantage majeur de la méthodologie DWQ est qu'elle ne considère pas uniquement la qualité de l'ED mais elle considère aussi la qualité de son architecture. DWQ étend l'architecture de l'ED par la construction d'un modèle des métadonnées de la qualité des données et l'intègre dans les différents modèles de l'ED (Conceptuel, Logique et Physique). Les données sur la qualité sont acquises, stockées et manipulées dans le modèle des métadonnées. De ce fait les métadonnées de l'ED devraient gérer continuellement les composantes architecturelles de l'ED et les facteurs de la qualité des données [51 ,175-182]. Les principaux travaux de recherche reposants sur les méthodologies DWQ et/ou TDQM sont :

1. Qualité de l'ED basée sur les agents technologiques [51] : ce projet exploite la théorie des agents informatiques pour l'implémentation de la méthodologie DWQ dans les ED.
2. Qualité de l'ED orientée conception, utilisation et évolution [180]: Ce travail de recherche propose une architecture d'ED avec une base des métadonnées pour décrire toutes les

[18] En anglais: Proactive Data Quality Management (ProDQM)
[19] En anglais : Reactive Data Quality Management. ReaDQM (ReaDQM)

composantes architecturales de l'ED par un ensemble des métamodèles à la quelle est ajoutée un métamodèle de la qualité pour définir les dimensions et les facteurs de la qualité.

3. Qualité des données basée sur les métadonnées [179]: Ce projet se base sur les fondements de la méthodologie TDQM selon la stratégie ProDQM.
4. Qbox-fondations [181]: Ce projet issu du projet QUADRIS présenté dans le premier chapitre est une plateforme de métadonnées consacrée à la mesure et l'évaluation de la qualité des données. Son avantage majeur est qu'il permet l'interopérabilité de plusieurs outils de qualité.
5. AQUAWARE [182]: Ce projet utilise l'approche web services afin de favoriser l'interopérabilité entre les outils de qualité exploitant de l'ED (outils OLAP, Outils FD, etc.).
6. Gestion de la qualité des données des entrepôts orientés-objet [26] : Ce système présente un modèle orienté-objet pour la gestion de la qualité des données dans les ED. Il repose sur l'approche orienté-objectif. Une fois l'objectif de la qualité de données est défini, le système gère la qualité des données par l'interaction entre ses composantes qui sont : besoins de la qualité, acteurs, objets de test de la qualité des données, et les problèmes de la qualité des données.
7. Cycle de vie de développement d'un ED [55]: Ce travail basé sur les expériences personnelles, la littérature des ED et la qualité des données propose un framework qui comporte six phases : Planification, Analyse, exigences, Développement, Implémentation et Mesure. Il s'inspire de la méthodologie TDQM et la stratégie ProDQM
8. TDQM dans l'ED basé sur l'approche IP (Information-Produit) [46]: Ce travail adapte la méthodologie TDQM appliqué dans le domaine de la fabrication des produits aux entrepôts des données. Il s'inspire de la méthodologie TDQM et la stratégie ProDQM.
9. Nettoyage des données avec gestion intelligible de la qualité [175]: Ce projet de recherche propose un système de gestion de la qualité des données dans l'ED qui répond aux exigences de la norme ISO 9001 :2000 sur la qualité. Ce projet considère le système d'intégration des données comme étant un système de fabrication des produits industriels. De ce fait, il propose quatre activités de base pour une gestion totale de la qualité des données : Politique, Planification, Control et Assurance de la qualité. Cependant, ce projet ne tient pas compte de la différence entre produit et donnée.

L'étude comparative que nous avons réalisée sur ces travaux de recherche, nous a permis de remarque que la plupart des travaux combinent les deux méthodologies et suivent la stratégie ProDQM et se focalisent sur la détermination des dimensions et des facteurs de la qualité des données et leur gestion comme étant des métadonnées. Cependant, ces travaux présentant des limites que nous résumons comme suit:

- Manque d'une fondation formelle sur la qualité des données.
- L'amélioration de la qualité des données est marginalement adressée dans ces travaux.

- La qualité des données est étudiée indépendamment du processus d'entreposage des données.
- Malgré que l'ETC soit la clé de voute de l'entreposage des données, ces travaux ne tiennent pas compte de ce processus dans la gestion de la qualité. Ainsi, il est difficile de rendre ces travaux applicables. Cela a conduit les entreprises à investir beaucoup plus dans le nettoyage des données [183, 184].
- Ces travaux ne permettent pas l'amélioration des SDO à partir des quelles sont formées les données de mauvaise qualité.
- Ces travaux ne tiennent pas compte de la complémentarité et la synergie des processus d'ED et d'ECD.

6. Entreposage des règles et des connaissances

Comme l'un de nos objectifs est la mise en œuvre d'un système de gestion des règles et des connaissances, nous avons effectué une recherche bibliographique sur l'entreposage des règles et des connaissances.

Notre recherche bibliographique, nous a permis de constater que le concept d'Entrepôt des Règles (ER) est introduit et développé pour supporter le partage des connaissances et la collaboration entre les entreprises [185,186]. Les auteurs de [63] ont proposé la création d'un ER sémantique pour un audit continu des données. Le Concept d'entrepôt de connaissances est introduit dans la littérature comme synonyme au dépôt ou base de connaissance. Le premier objectif de l'entrepôt de connaissances est de fournir aux décideurs une plateforme d'analyse intelligente qui améliore les différentes phases du processus de gestion de connaissance. Les entrepôts des connaissances existants s'intéressent uniquement aux connaissances explicites [185, 186]. Cependant, le mot entrepôt dans ces concepts n'implique pas la présence d'un processus d'entreposage des règles ou des connaissances similaire à celui des données. C'est une base des règles ou des connaissances avec quelques opérations de vérification de la redondance, inconsistance, circularité des règles et des connaissances. Donc la gestion de la qualité des règles et des connaissances est totalement marginalisée dans ces travaux.

7. Entrepôt des données à base de règles

Nous présentons dans cette section, les travaux de recherche portant sur la prise en compte des points de vue des utilisateurs pendant la construction et le rafraîchissement des EDs. Cela permet de faire émerger des conclusions utiles à notre travail sur la personnalisation des EDs et des ECDs.

La prise en compte des préférences, des usages et des interactions du décideur, appelée personnalisation, constitue un champ de recherche qui reste à explorer dans le domaine des systèmes décisionnels intégrant des EDs multidimensionnels [187]. La personnalisation se repose sur l'utilisation d'un langage à base de règle [12]. Les principaux travaux que nous avons identifiés dans ce contexte sont :

- Evolution de l'ED dirigée par les connaissances[20] [7, 188].
- Modèle d'ED à base de règles[21] [12].

Ces travaux nous semblent prometteuse dans la voie de renforcer l'interaction entre l'utilisateur et le système d'aide à la décision en permettant à celui-ci d'intégrer ses propres connaissances. Ils reposent sur l'utilisation des règles dites règles d'agrégation ou règles d'analyse sous forme de <Si condition Alors conclusion> ou les règles de type ECA [7, 12, 189]. Cependant ces travaux concernent uniquement la phase de l'analyse (c'est-à-dire après la construction de l'ED) de l'ED.

L'atout majeur de ces travaux est qu'il peut faire intervenir l'utilisateur en lui laissant la possibilité d'introduire ses connaissances dans le système décisionnel.

8. Intégration des processus ED et ECD

L'utilisation des techniques de l'ECD pour extraire des connaissances utiles des données elles mêmes pour l'optimisation des performances des bases des données est avancée depuis quelques années. Nous avons constaté un travail prometteur pour l'auto-administration des EDs. Ce travail a développé un outil qui recommande une configuration d'index et de vues matérialisées permettant d'optimiser le temps d'accès aux données. Cet outil applique les techniques de FD sur les requêtes des utilisateurs pour construire une configuration d'index et des vues afin de partager efficacement l'espace de disque alloué pour stocker ces structures [15]. Cependant, nous avons constaté que peu de travaux ont été entrepris dans cette optique notamment pour l'amélioration de qualité des données et des connaissances.

La plupart de ces travaux ont pour but d'extraire des règles permettant la prédiction de valeurs manquantes dans les cas où l'attribut sur lequel la prédiction est faite est soit continu soit discret [188-190].

9. Conclusion

Au cours de ce chapitre, nous avons présenté le processus d'ECD et le processus d'entreposage des données. Nous avons montré que ces deux processus sont complémentaires dans les systèmes d'aide à la décision et que l'ED est un élément essentiel du processus d'ECD. De ce fait, nous avons détaillé ces deux concepts et les concepts associées et nous avons montré l'utilité de leur adaptation afin de permettre une meilleure prise en charge de la qualité des données et des connaissances.

La recherche bibliographique que nous avons effectuée, nous a permis de constater que tous les travaux de recherche s'accordent sur le fait que la qualité des données et des connaissances est le défi commun de ces deux processus. Les travaux réalisés conservent généralement des opérations pendant la phase de préparation des données pour l'évaluation et

[20] En anglais : data Warehouse Evolution Driven by Knowledge : WEDrik
[21] En anglais : Rules based Data Warehouse : R-DW

(éventuellement) l'amélioration de la qualité des données en se basant le plus souvent sur les statistiques.

Ces méthodes sont insuffisantes pour permettre une gestion complète et continue de la qualité des données et des connaissances. Ainsi, nous proposons l'adaptation le processus d'ECD pour une meilleure prise en charge de la qualité des données et des connaissances. Dans cette proposition, nous présentons un nouveau processus d'ECD afin de permettre l'acquisition des connaissances automatique et par élicitation ainsi que l'assurance de leur qualité.

La présentation des connaissances sous forme de règles est très prometteuse car les règles peuvent être exploitées pour l'évaluation et l'amélioration de la qualité des données et des connaissances.

Cela requiert la mise en œuvre d'un système complet d'entreposage des règles et des connaissances qui permet la gestion des connaissances et de leur qualité afin d'être utilisables au profit de la gestion de la qualité des données. Le chapitre suivant détaille cette proposition.

Chapitre 3:
Un système d'entreposage des règles et des connaissances orienté qualité

« Le prix s'oublie, la qualité reste.»
Proverbe français

1. Introduction

Nous avons montré dans les chapitres précédents que le problème de la qualité dans les systèmes d'ECD n'est pas encore bien considéré dans les travaux de recherche. Nous avons aussi précisé que telle solution de gestion de la qualité doit être totale et continue. La gestion totale de la qualité se repose sur l'incorporation des différents acteurs de l'ECD et plus spécifiquement: donnée, connaissance, utilisateurs et processus. Une solution continue doit permettre de gérer la qualité tout au long du processus ECD. Nous avons montré que les travaux existants sur la gestion des données et des connaissances ne sont pas adaptables aux ECD.

Par conséquent, nous proposons dans ce travail l'adaptation du processus de l'ECD pour une meilleure prise en charge de la qualité des données. Pour atteindre cet objectif, nous avons proposé un panel de contributions permettant de répondre partiellement au problème de la qualité dans l'ECD :

- Un système d'entreposage des règles et de connaissance orienté qualité.
- Une adaptation du processus d'entreposage des données.
- Une adaptation du processus ECD.

Notons bien que dans ce travail, nous considérons la complémentarité des processus d'entreposage des données et d'ECD ainsi que la synergie des connaissances et donnée. Les deux dernières contributions seront détaillées dans le chapitre suivant.

Dans ce chapitre, nous présentons notre contribution d'entreposage des règles et des connaissances orienté qualité. Nous avons conçu un système qui répond aux limites des travaux de recherche dans le domaine de l'ingénierie des connaissances et, plus spécifiquement, de l'acquisition des connaissances par élicitation ou par apprentissage automatique et de gestion de leur qualité. L'objectif majeur est de pouvoir définir un système de gestion de la qualité des connaissances et des données à base de règles afin de bénéficier des connaissances extraites des données et de domaine au profit de l'évaluation et l'amélioration de la qualité. Notre principale contribution, dans ce chapitre, se repose sur la proposition de:

- Un formalisme unifié (valable pour tous les types de règles) pour la représentation et la gestion des règles et de leur qualité. Ce formalisme est fondé sur la théorie de la logique d'ordre 1, plus spécifiquement, la théorie de structure qui a été introduite dans la logique pour permettre l'évaluation de la vérité des formules.
- Un système d'entreposage des règles et des connaissances afin de permettre la gestion de leur qualité et leur évolution. Il est doté d'un processus ETC que nous avons conçu spécifiquement pour l'entreposage des règles.

- Un système d'acquisition des connaissances par élicitation afin de permettre l'incorporation de la connaissance du domaine et/ou experte dans l'amélioration de la qualité.

2. Rappel de quelques concepts et propriétés de la logique

Dans cette section, nous rappelons premièrement les différentes notions et propriétés de la théorie de la logique d'ordre 1 (dite aussi logique des prédicats) que nous avons retenus pour la conception du système d'entreposage des règles et, plus spécifiquement, pour la définition de formalisme unifié de représentation des règles. Ces notions sont les bases de l'évaluation de la syntaxe et la sémantique d'une formule dans la logique. Nous les avons choisis car dans ce travail, nous considérons une règle (respectivement une connaissance) comme étant une formule. Les concepts que nous avons introduits sont la structure et la forme normale d'une formule. Nous avons adopté le concept de structure pour permettre l'évaluation de la qualité d'une règle et le concept de forme normale pour la mise sous forme normale de la condition et conclusion des règles afin de faciliter leur gestion. Les références [85-87] nous y rapporteront plus de détails à propos de ces concepts.

a) Notion de structure

Nous appelons structure tout triplet (D, F_D, R_D) où D est un ensemble non vide appelé domaine ou discours, F_D est un ensemble de fonctions réalisables sur D et R_D est un ensemble de relations réalisables sur D.

Exemple le triplet (N, {+, *, 0, successeur}, {=, <=}) est une structure connue comme structure de l'arithmétique de Piano. C'est la structure dans laquelle on peut interpréter les expressions arithmétique. Elle est la base de la preuve par récurrence.

La notion de la structure est très importante dans la logique car elle permet de calculer la valeur de vérité des formules dans un environnement donnée. C'est-à-dire que la vérité d'une formule est relative et différente d'une structure à une autre. L'évaluation des formules sera déterminée si et seulement si on dispose des interprétations des fonctions et des relations.

Dans ce travail, nous avons retenu ce concept de structure car l'évaluation de la règle est aussi relative à la source des données où elle est appliquée.

b) Forme normale d'une formule

Nous donnons ici tous les concepts liés à la mise sous forme normale d'une formule

- Littéral : C'est une formule atomique ou sa négation (noté l_i)
- Clause : C'est une formule sous forme de disjonction de littéraux :

$$l_1 \vee l_2 \dots \dots \vee l_i \dots . . \vee l_n = \bigvee_{i=1}^{i=n} l_i \qquad \text{Où } l_i \text{ est un littéral.}$$

- Une formule F est dite sous Forme Normale Conjonctive (FNC) si elle est de la forme : $D_1 \wedge D_2 \dots \dots . \wedge D_i \dots . \wedge D_n = \bigwedge_{i=1}^{i=n} D_i$ où Di est une clause. Toute formule peut être mise sous FNC.

- Une conjonction élémentaire est une formule de la forme :

$$l_1 \wedge l_2 \ldots\ldots \wedge l_i \ldots\ldots \wedge l_n = \bigwedge_{i=1}^{i=n} l_i$$ où l_i est un littéral.

- Une formule F est dite sous Forme Normale Disjonctive (FND) si elle est de la forme : $D_1 \vee D_2 \ldots\ldots \vee D_i \ldots\ldots \vee D_n = \bigvee_{i=1}^{i=n} D_i$ où Di est une conjonction élémentaire. Toute formule peut être mise sous FND.

3. Formalisme unifié de représentation des règles

Comme nous l'avons évoqué dans le chapitre 1, il existe plusieurs types de règles et pour chaque type, il existe plusieurs formalismes de représentation. Par la suite, chaque système ou langage à base de règles est dédié généralement à un type donné de règle. Pour pallier ce déficit, nous avons introduit la théorie de la structure dans notre travail pour la création d'un formalisme unifié, uniforme et général permettant la représentation de tout type de connaissance modélisable sous forme de règle.

3.1. Motivation

Dans le tableau 10, nous avons défini les différents composants de chaque type de règles qui sont : Condition, Conclusion, PostCondition, Événement et Action. Pour décrire la forme de chaque règle, nous avons utilisé les coefficients (0, 0+, 1, 1+) pour indiquer la présence des composants dans la règle. Par exemple, la règle de réaction a au moins (1+) une condition, une (1) conclusion et une éventuelle (0) Post-Condition. Notons bien que la présence du composant action dans les règles du nettoyage des données est indispensable.

Type	Condition	Conclusion	Post Condition	Event
Intégrité	0	1+	0	0
Réaction	1+	1	0+	1
Production	1+	1	0	0
Dérivation	1+	1	0	0
Transformation	0+	1+	0	0
Unification	0+	1+	0+	0+

x=0, 1. x: exactement x x+=au moins x

Tableau 10 : représentation et unification des différents types de règles.

Le tableau 10 montre que la création d'un formalisme unifié permettant la représentation de toutes les caractéristiques des différents types de règles est faisable.

Afin de concevoir un formalisme robuste, il est crucial de construire le métamodèle de système de nettoyage de données à base de règles à partir duquel nous pouvons identifier tous les composants nécessaires à la réalisation du formalisme.

3.2. Conception du métamodèle générique du système de nettoyage des données à base de règles

La figure 16 que nous avons décrite selon le langage UML présente notre métamodèle conceptuel relative au système du nettoyage des données. Ce métamodèle nous a permit de déterminer toutes les caractéristiques essentielles à la gestion de la règle et de sa qualité. Nous l'avons conçu d'une manière générique afin d'être applicable dans n'importe quel système d'amélioration de la qualité à base de règles. Il peut être même utilisé pour exécuter d'autres tâches telles que la personnalisation des entrepôts des données.

L'objectif attendu de la conception de ce métamodèle générique est la détermination de toutes les caractéristiques des règles ainsi que leurs interactions. L'interprétation de ce modèle est la suivante. Une règle est caractérisée par une condition qu'est un ensemble des conditions élémentaires, une conclusion qu'est aussi un ensemble des conclusions élémentaires et un environnement dans lequel la qualité de la règle est évaluée. Nous avons introduit le concept d'environnement, qui est inspiré de celui de structure, afin de représenter les caractéristiques des règles et surtout celles qui conditionnent leur qualité. L'environnement de la règle comporte deux parties l'Univers et les Propriétés. L'Univers est introduit afin de permettre la représentation des sources de données dans lesquelles la règle est applicable. Ce concept est essentiel car nous avons constaté que lorsqu'on exécute le processus d'ECD sur des données intégrées à partir des différentes sources, le processus élague des connaissances à cause de leur qualité (contradictoire par exemple). Cependant, si on applique ces règles sur chaque source, nous constatons qu'elles sont valides dans certaines sources et contradictoires dans d'autres. Cet aspect n'est pas tenu en compte par les travaux de recherche existants. Chaque source de données peut nécessiter une action spécifique. Par conséquent, une règle peut comporter plusieurs actions. Le concept propriété est introduit pour décrire les différentes dimensions de qualité d'une règle et leur métrique. Ce concept peut contenir n'importe quelle caractéristique utile pour la gestion de la règle et sa qualité telles que l'événement qui déclenche la règle et le post condition.

L'avantage majeur de ce système est qu'elle considère l'évolution de la règle et de sa qualité. L'évolution de la qualité d'une règle est un aspect important qui n'est pas considéré dans les travaux de recherche portant sur la qualité des données et des connaissances. Par conséquent, nous devons définir le cycle de vie d'une règle.

3.2.1. Détermination des principales caractéristiques des règles

La recherche bibliographique que nous avons effectuée, nous a permis de collecter l'ensemble des caractéristiques nécessaires pour la gestion des règles et de leur qualité. En outre, nous avons ajouté d'autres des caractéristiques utiles à notre système et qui ne sont pas

considéré dans les travaux de recherche liés à la formalisation des règles. Ces caractéristiques concernent généralement la qualité de la règle.

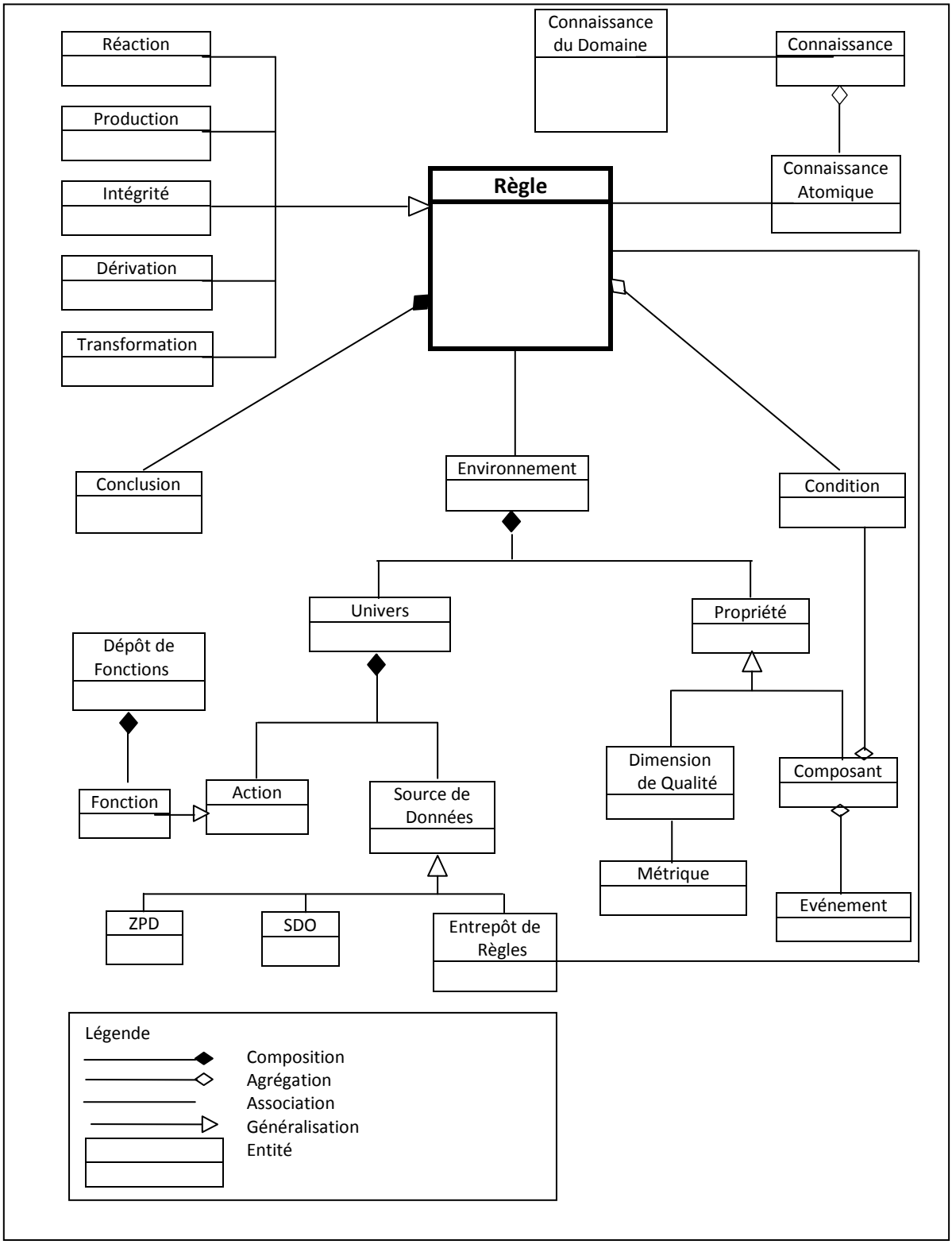

Figure 16 : Métamodèle générique du système de nettoyage de données à base de règles.

Le tableau 11 présente toutes les caractéristiques que nous qualifions bénéfiques à notre système. Nous avons distribué ces caractéristiques dans deux catégories disjoints. La première catégorie concerne les caractéristiques indépendantes des sources des données. Ces caractéristiques sont identiques pour toutes les sources. Elles permettent généralement de décrire le nom et le corps d'une règle donnée. Par contre la deuxième catégorie contient les caractéristiques dépendantes des sources des données. Ces caractéristiques se varient d'une source à une autre. Notons bien que les caractéristiques sont aussi valables pour les connaissances avant leur transformation sous forme de règles. La deuxième catégorie montre que l'évaluation de la qualité des connaissances et plus spécifiquement des règles est relative.

	Caractéristique	Définition
Caractéristiques indépendantes source	Code_Règle	Chaque règle est caractérisée par une clé unique.
	Expression_Règle	Elle définit le corps de la règle.
	Texte_Règle	Elle décrit la règle ou la connaissance avant leur transformation.
	Sources_Règle	Elle contient l'ensemble des sources où la règle est applicable. Une source peut être une SDO_i, une ZPD ou entrepôt de règles (ER). $Sources(Règle(i)) = \{S_j, j \in N \text{ où } S_j \subset \{SDO_i, i \in N\} \cup \{ZPD\} \cup \{ER\}$
	Date_Création	C'est la date d'insertion de la règle dans l'entrepôt de règles. Cette information peut être donnée par l'utilisateur ou déterminé par le système.
Caractéristiques dépendantes source	Statut_Règle	Elle décrit l'état actuel de la règle. elle prend l'une des valeurs suivantes : proposée, approuvée, acceptée, rejetée, valide, contradictoire, théorème, satisfiable ou archivée.
	Univers_Règle	Elle permet de définir si une règle est singulière, multiple or agrégat. Une règle est singulière s'elle concerne une seule source et ne peut être jamais applicable à une autre source. Une règle est multiple si elle peut être applicable dans toutes les sources. Notons qu'une règle multiple peut être applicable dans une seule source mais on ne dit pas qu'elle est singulière. Une règle agrégat est une règle applicable uniquement aux données agrégées.
	Active_Règle	Elle indique si une règle est active ou inactive.
	Date_Effective	Elle exprime la date de la mise en application d'une règle dans une source donnée.
	Date_Réveil	Si une règle est inactive, cette information indique la date de réactivation de la règle.
	Durée_Validité	Elle indique le temps pendant lequel la règle est valide.
	Effective_Règle	Elle exprime le nombre de fois de l'application d'une règle sur les données dés sa mise en application.
	Effective_Règle_Dernier	Elle permet de connaître le nombre de fois de l'application de la règle pendant le dernier entreposage des données ou des règles $\text{Effective\_Règle}(Ri, Sj) = \sum_{i=1}^{i=n} \text{Effective\_Règle\_Dernier}(Ri, Sj)$
	Action_Règle	Elle indique les actions à exécuter si la règle est violée.
	Echec_Règle	Elle indique le nombre de fois de l'échec d'une règle.
	Echec_Action	Elle indique le nombre des corrections faites par l'action d'une règle violée mais validées incorrectes par les utilisateurs.

Légende : R_i : Règle donnée, S_j : Source donnée

Tableau 11: Caractéristiques des règles.

Certaines caractéristiques peuvent être considérées comme des métriques et d'autres sont calculables à partir des différentes métriques. Dans le tableau 11, nous avons expliqué en

détail les différentes caractéristiques et nous avons montré leur importance dans la gestion de la qualité des règles.

3.2.2. Cycle de vie d'une règle

Afin d'assurer l'évolution de la qualité de la règle, nous avons conçu son cycle de vie. Ce cycle de vie dépend étroitement des processus d'entreposage de données et des règles. Le statut de la règle se calcule automatiquement par le système d'une manière continue. La figure 17 décrit le cycle de vie d'une règle et dans le tableau 12 nous donnons les interprétations détaillées des valeurs qui peuvent être affectées au statut d'une règle. Le statut d'une règle est la caractéristique fondamentale d'évaluation et d'amélioration de la qualité de la règle.

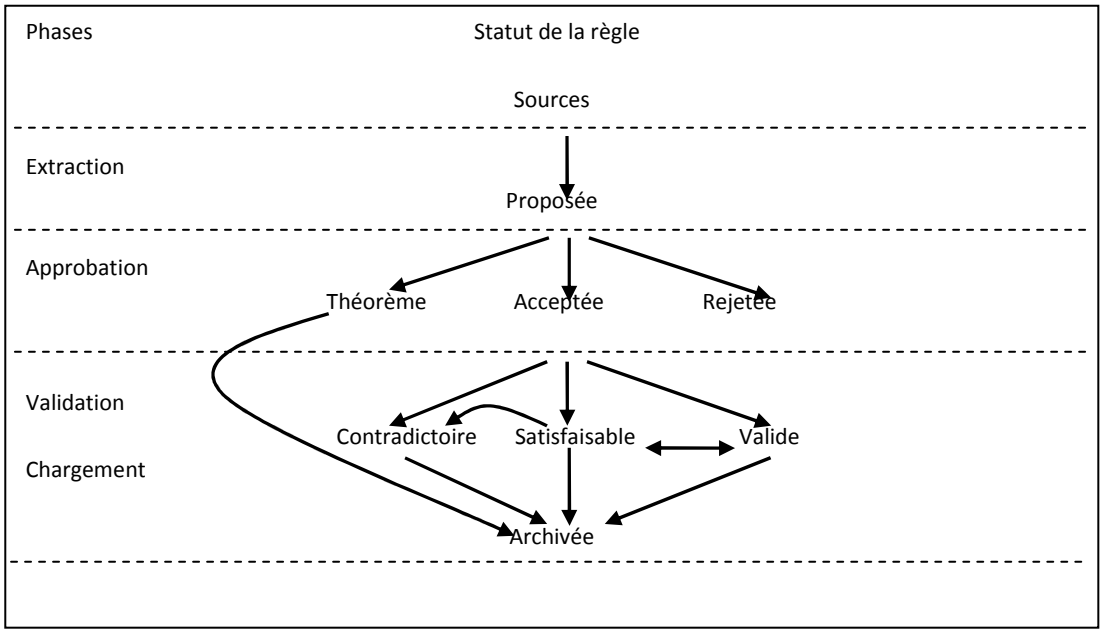

Figure 17: Cycle de vie d'une règle.

Dans notre proposition, une règle archivée ne sera jamais réutilisée par le système car si l'utilisateur veut bloquer une règle, il doit la désactiver (Active_Règle=Inactive) ou la suspendre pendant un certain temps (en indiquant la date de Date_Réveil). Cependant, le statut d'une règle est relatif à la source. On peut trouver une règle valide dans une source et archivée dans une autre source par exemple.

3.3. Description du formalisme unifié de représentation des règles

Nous décrivons dans cette partie le formalisme que nous proposons à utiliser pour représenter formellement les connaissances acquises automatiquement ou par élicitation. Ce formalisme s'inspire de la théorie de la structure de la logique d'ordre 1 décrite dans la section 2 de ce chapitre. Il repose sur un langage formel fondé sur un ensemble des symboles et règles de production permettant la représentation de tous les types de règles. Il permet à la fois de gérer

les règles et leur qualité à travers quelques propriétés des caractéristiques. La mise en application de ce formalisme nécessite de redéfinir et adapter le concept d la règle au sujet du nettoyage des données.

Valeur	Interprétation
Proposée	Lorsque une connaissance ou une règle est créée pour la première fois, nous affectons la valeur proposée à son statut Une règle proposée est par défaut inactive.
Rejetée	Si une règle ou connaissance est jugée de mauvaise qualité alors elle sera rejetée (elle ne s'insère pas dans l'entrepôt des règles). Cette opération est généralement faite par des experts.
Acceptée	Une règle ou une connaissance jugée de bonne qualité prend la valeur acceptée. Toute règle acceptée est activée automatiquement pourqu'elle soit utilisable par le système.
Théorème	Toute règle démontrée valide théoriquement ou a priori prend la valeur théorème. Ce type de règle n'est pas concerné par toutes les dimensions de qualité (la règle de la résolution de l'équation du $2^{ème}$ degré par exemple)
Contradictoire	Une règle qui échoue toujours prend la valeur contradictoire et elle sera automatiquement archivée.
Valide	Une règle qui réussit toujours prend la valeur valide.
Satisfaisable	Une règle qui réussit partiellement est dit satisfaisable.
Archivée	Une règle peut être archivée par le système, ou par l'utilisateur même si elle est de bonne qualité.

Tableau 12 : Interprétation des statuts d'une règle.

3.3.1. Définition d'une règle

Dans ce travail, nous considérons une règle comme étant une connaissance qui définit et contraigne une (les) donnée(s) et leur qualité dans une source de données bien déterminée (sources des données opérationnelles : SDO, zone de préparation des données: ZPD ou entrepôt des données ED) ou une (des) règle(s) et leur qualité dans l'entrepôt de règles que nous décrirons dans la suite de chapitre.

Cette définition nous a conduit à regrouper les règles dans trois classes :

- Règles mono source: Ce sont des règles applicables uniquement dans une seule source de données opérationnelle,
- Règles multi sources: Ce sont des règles applicables au moins dans une SDO et/ou dans une ZPD, et
- Règles agrégats: Ce sont des règles applicables aux données agrégées.

Dans le tableau 11, nous avons spécifié la caractéristique Univers_Règle pour indiquer le type de règle.

Les règles peuvent être définies avant l'entreposage initial des données par les processus d'élicitation des connaissances et/ou d'extraction automatique des connaissances, ou pendant les processus d'entreposage des données et des règles.

3.3.2. Forme de la règle

Avant de présenter la forme que nous proposons dans ce travail pour la représentation des règles, nous devons premièrement définir l'alphabet que nous utilisons pour construire les règles.

L'alphabet contient:

- Un ensemble fini dénombrable des symboles dénotent les sources des données (SDO_i, ZPD, ED et ER) et de leurs tables et attributs, et les domaines de ces attributs.
- Un ensemble fini des séparateurs suivants : ( ), < >
- Un ensemble fini des mots réservés suivants : SI, ALORS, DANS
- Un ensemble fini des opérateurs : logique ($\neg$ (Négation), $\wedge$ (Conjonction), $\vee$ (Disjonction)), opérateurs et fonctions arithmétiques (+,*,-, /,…, racine, exponentielle,….), opérateurs ensembliste ($\cup$ (Union), $\cap$(Intersection), $\notin$ (Inclusion), $\in$ (Appartenance),….), etc.
- Un ensemble fini des fonctions possibles sur les attributs tels que : Sum, Moyen, Maximum, Minimum, …. etc.

a) Calcul des ensembles complets de connecteurs

Afin de minimiser l'ensemble des opérateurs dénoté E, nous calculons pour chaque groupe d'opérateurs dénoté G l'ensemble complet d'opérateurs dénoté G' qui lui correspondant. Notons bien que un ensemble complet d'opérateurs G' est un sous ensemble de G tel que pour toute expression Exp, il existe une expression Exp' équivalente à Exp formalisée uniquement à l'aide des opérateurs de G'. Autrement dit, tout opérateur peut se définir au moyen des opérateurs de l'ensemble complet. Formellement nous le définissons comme suit :

$$\text{Soient } G, G' : \text{deux ensembles d'opérateurs tel que } G' \subset G \text{ et Exp} : \text{Expression } \exists\ T : \text{Transformations et } \exists\ Exp' : \text{Expression tel que:}$$

$$Exp' = T(Exp)$$

(Dans notre travail l'expression peut être soit une condition élémentaire, soit une conclusion élémentaire).

T est une suite de transformations applicables sur l'expression où chaque transformation concerne un opérateur appartient à l'ensemble résultant de la différence de G et G' dénoté G-G' (c'est-à-dire les opérateurs qui n'appartiennent pas à G'). Ces transformations sont généralement présentées par des règles de passage dans les théories concernant ces opérateurs.

Dans le tableau 13, nous donnons des exemples sur quelques ensembles complets des opérateurs. Notons bien que pour chaque ensemble des opérateurs, nous pouvons trouver plusieurs ensembles complets. De ce fait, nous proposons de choisir l'ensemble qui contient le moins d'opérateurs (c'est-à-dire $G' = \text{Minimum}\,(\text{cardinalité}(G'_i))$ oùG'_i sont les différents ensembles complets d'un groupe d'opérateurs donné)

Opérateurs	Ensemble d'opérateurs	ensemble complet d'opérateurs
Logiques	$\neg, \vee, \wedge, \rightarrow, \leftrightarrow$	$\neg, \vee, \wedge$
Ensemblistes	$\in, \notin, \cap, \cup, \subset, \not\subset, \subseteq$	$\neg, \in, \cap, \cup$
Relationnels	$\neq, \leq, \geq, <, =, >$	$\neg, <, =$

Tableau 13 : Exemples de quelques ensembles complets d'opérateurs.

Nous avons proposé l'utilisation des ensembles complets d'opérateurs dans notre travail parce qu'ils facilitent la réalisation de certaines opérations sur les règles, telles que la déduplication des règles.

b) Présentation et description du formalisme proposé

Après avoir présenté l'alphabet que nous utilisons pour la présentation des connaissances sous la forme de règles, nous décrivons ici le formalisme que nous avons proposé dans [127, 191] et qui permet de représenter les différents concepts manipulés dans la mise en œuvre de notre système de gestion de la qualité à base de règle. La forme générale d'une règle est définie comme suit:

SI Condition **ALORS** Conclusion **DANS** Environnement

Dans cette forme, nous avons introduit le concept *Environnement* qu'est inspiré de celui de structure de la logique afin d'assurer la relativité, l'évolution et la flexibilité des règles et de leur qualité. Ce formalisme que nous avons développé tient compte de toutes les caractéristiques des règles et de leurs dépendances nécessaires à la gestion des règles et de leur qualité. Cette forme permet la construction des règles d'une manière incrémentale et itérative.

Nous donnons ci-dessous les détails concernant les différents composants de la forme que nous proposons:

- **Condition** : A la différence des autres formalismes, où la condition d'une règle de la forme "SI condition ALORS conclusion " est définie comme étant la conjonction de clauses, nous le définissons dans ce formalisme comme étant la disjonction des conjonctions élémentaires où la conjonction élémentaire est une conjonction de littéraux. Chaque littéral est une expression logique. L'expression logique est formée à partir des expressions relationnelles à l'aide des opérateurs logiques. Ainsi, la forme d'une condition est définie comme suit:

  $\text{Condition} := \bigvee_{i=1}^{i=n} C_i$, C_i est une conjonction élémentaire

  $C_i := \bigwedge_{i=1}^{i=m} l_i$, l_i est un littéral

  $l_i := \text{Expression}$

  Les raisons pour lesquelles nous avons décidé d'utiliser la disjonction au lieu de la conjonction dans la condition sont:

  1. Rapidité et simplicité de l'évaluation de la valeur de vérité (vraie ou fausse) de la condition: Ce formalisme donne le meilleur temps de réponse de l'évaluation de la condition car:
     - Dans le cas des autres formalismes où la condition est la conjonction des clauses, nous devons calculer la valeur de vérité de chaque clause. Cela veut dire que : si nous avons n clauses dans la condition ($=\bigwedge_{i=1}^{i=n} C_i$), l'algorithme de l'évaluation de la vérité de la condition doit parcourir le n

clauses. Si t est le temps moyen nécessaire pour évaluer une clause alors le temps nécessaire pour l'évaluation de la condition est n*t.

- Dans le cas de notre formalisme, si nous avons n conjonctions élémentaires dans la condition (=$\bigvee_{i=1}^{i=n} C_i$), l'algorithme de l'évaluation de la vérité de la condition s'arrête dés qu'il évalue une conjonction élémentaire quelconque vraie. Cela veut dire que l'algorithme parcourt généralement m conjonctions élémentaires tel que m est inférieur ou égale à n. de ce fait, si t est le temps moyen nécessaire pour évaluer une clause alors le temps nécessaire pour l'évaluation de la condition est m*t.

Comme m est inférieur ou égale à n (m<=n), nous déduisons que le temps d'évaluation de la vérité (vraie) de la condition pour notre formalisme est inférieur ou égale à celui des autres formalismes (m*t<=n*t)

2. Unification d'un ensemble règles dans une règle résultante: La forme que nous avons utilisée permet d'unifier plusieurs règles qui ont la même conclusion et le même environnement dans une même règle (règle résultante). Cela veut dire que si nous avons un nombre dénoté n de règles :

$$\{\mathrm{Ri} = \mathrm{SI\ Ai\ ALORS\ Conclusion\ DANS\ Environnement}, 1 <= i <= n\ \}$$

Alors la règle résultante de ces règles R_i est écrite de la manière suivante :

$$\mathrm{R} = \mathrm{SI} \bigvee_{i=1}^{i=n} \mathrm{A_i}\ \mathrm{ALORS\ Conclusion\ DANS\ Environnement}$$

Cette opération d'unification qui caractérise notre formalisme permet de réduire le nombre de règles, ce qui implique l'optimisation des performances en termes d'une espace mémoire (disque) et le temps d'accès à ces (réduction de l'espace disque et accélération de l'accès aux règles).

Dans la figure 18, nous illustrons l'unification des règles par un exemple où nous avons calculé la règle résultante dénotée R_{12} de deux règles dénotées R_1 et R_2 extraites à partir d'une table T formée de trois attributs et sept enregistrements. . Cet exemple montre l'importance de l'utilisation de la disjonction par rapport à la conjonction.

Ces deux raisons que nous représentons, montrent l'avantage du formalisme proposé dans ce travail par rapport aux formalismes existants.

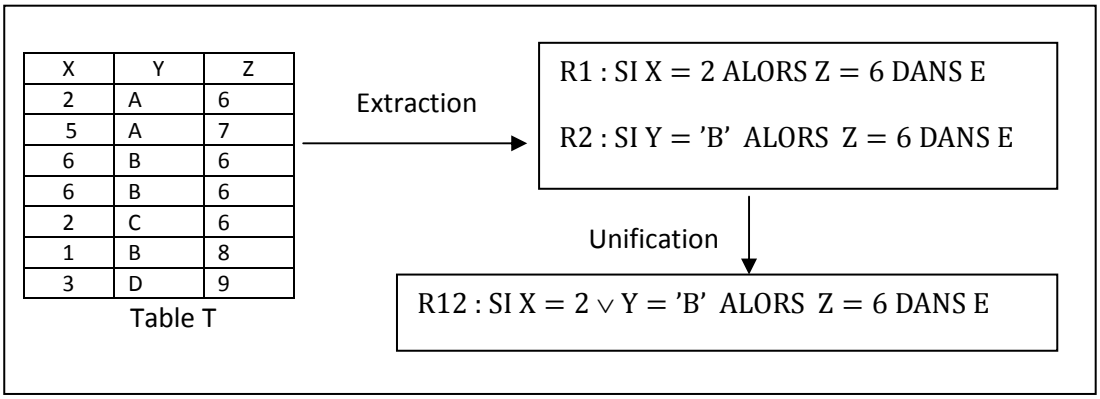

Figure 18 : Exemple d'unification de deux règles.

- **Conclusion :** Une Conclusion est un littéral (l) désignant une expression logique.

$$l \coloneqq \text{Expression}$$

- **Environnement :** l'évaluation et l'amélioration de la qualité d'une règle est fortement corrélée à certaines caractéristiques qui varient d'une source à une autre. Cet aspect n'est pas été considère dans les travaux actuels portés sur les systèmes à base de règles. Cela est dû principalement à la considération suivante : la majorité des outils de l'amélioration de la qualité s'intéressent aux transformations sur les données -telles que la normalisation, standardisation, formatage, ..., et la déduplication- et non au nettoyage des données de mauvaise qualité. La correction de ces données doit tenir compte de l'environnement original où cette donnée est créée (stockée). De ce fait, dans ce travail, nous avons introduit la notion d'environnement dans la formalisation de la règle. L'environnement est représenté de la manière suivante:

$$\text{Environnement} \coloneqq (\text{Univers}, \text{Dimension}, \text{Composant})$$

Le concept univers est l'ensemble des sources où la règle est applicable. Le concept dimension désigne l'ensemble des dimensions qualité nécessaires à l'évaluation de la qualité de la règle. Le concept composant est introduit pour contenir les caractéristiques autres que celles de l'univers et dimension. Vu l'importance de ce concept environnement dans la gestion des règles et de leur qualité, nous détaillerons ces éléments dans la section suivante.

3.3.3. Composants de l'environnement d'une formule

Comme nous l'avons présenté dans la section précédente, l'environnement comporte trois éléments qui sont:

1. **Univers**. C'est un ensemble non vide des sources où la règle est applicable.

$$\text{Univers} := \{S_j, j \in N \text{ où } S_j \text{est un source} \subset \{SDO_i, i \in N\} \cup \{ZPD\}$$

ou (exclusif)

$$\text{Univers} := \{ER\}$$

Cela veut dire que l'univers peut contenir soit l'entrepôt des règles que nous décrirons dans la suite de ce chapitre, soit un ensemble formé des sources des données opérationnelles (SDOi) et/ou la zone des préparation des données (ZPD) mais pas les deux car les règles applicables aux données ne sont pas applicables aux règles et vice versa.

2. **Dimension.** Ce concept représente l'ensemble des dimensions de qualité applicables à une règle. A la différence des autres travaux qui s'intéressent à la mesure de la qualité des règles que nous avons présenté dans les deux premiers chapitres, le formalisme proposé permet la construction des dimensions de qualité d'une manière incrémentale, itérative, interactive et relative. La relativité signifie que l'environnement permet de définir les dimensions de qualité par règle et par source. Cela veut dire qu'une dimension peut concerner une seule règle, un groupe de règles ou toutes les règles. Comme, elle peut concerner la règle uniquement dans un sous ensemble des sources de l'univers. Les dimensions les plus fréquemment utilisées dans la plupart des travaux de recherche en pratique et en théorie qui sont : généralité, validité, lisibilité, nouveauté, surprise et utilité nous les avons considérées valables pour toutes les règles. Cela veut dire que le composant dimension de l'environnement contient toujours ces six dimensions qualité (indépendamment des sources des données, des applications et des entreprises).

   De ce fait, nous avons divisé l'environnement en trois ensembles disjointes : Environnement initiale (Envir_Init), Environnement globale (Envir_Glob) et Environnement règle (Envir_Règle). Pour chaque dimension qualité, nous devons indiquer l'ensemble des métriques nécessaires pour l'évaluation de chaque dimension qualité. Nous détaillons la construction et l'utilité de ces trois ensembles dans la suite de ce chapitre.

3. **Composant.** Comme nous l'avons évoqué, ce concept est introduit pour contenir les caractéristiques autres que celles décrites dans l'univers et dimension. Ces caractéristiques sont généralement : Action qu'est l'ensemble d'actions (programmes) élémentaires à exécuter si la règle est violée, Evénement qu'est le déclencheur de la règle, et PostCondition qu'est une condition à vérifier après l'exécution d'une action atomique par une règle donnée. Une PostCondition est décrite de la même que la condition d'une règle (voir 3.3.2).

   Une action atomique peut être exécutée par un groupe de règle dans des environnements différents. Cela veut dire qu'une règle peut avoir plusieurs actions atomiques où chaque action atomique est applicable dans un groupe de sources. Cependant, une source est concernée uniquement par une seule action atomique pour une règle donnée.

3.3.4. Métamodèle des règles du formalisme proposé

Après avoir définir les constituants de base de notre formalisme, nous présentons dans la figure 19 le métamodèle décrit en langage UML d'une règle construit selon la description du formalisme proposé que nous avons donnée dans les sections précédentes de ce chapitre. Ce métamodèle est le résultat de l'adaptation du formalisme du métamodèle d'une règle donné dans le chapitre 1 pour qu'il soit adéquat à notre formalisme.

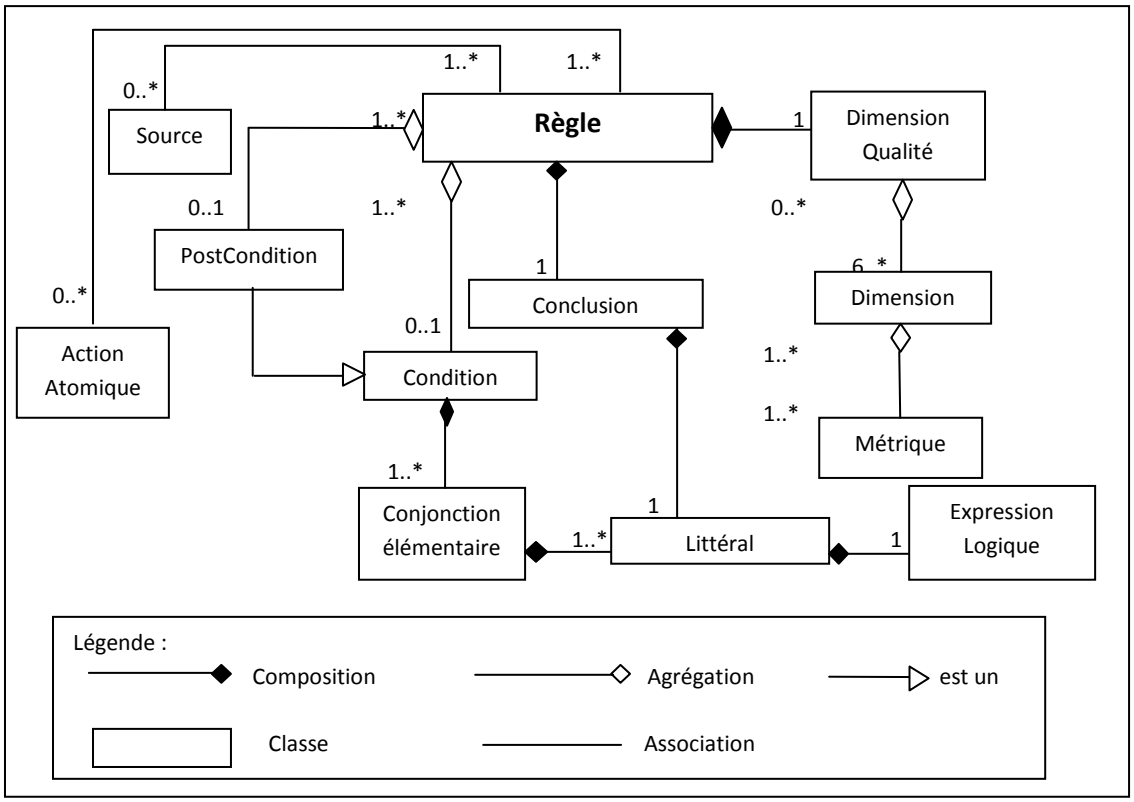

Figure 19 : Métamodèle UML d'une règle

A la différence du métamodèle décrite dans le premier chapitre, le métamodèle que nous décrivons dans la figure 19 n'est pas lié au type de règle et tient compte de la qualité.

3.3.5. Détermination des propriétés des règles

Il est nécessaire de déterminer les propriétés des règles et de leurs dépendances dans ce nouveau formalisme. Dans le tableau 14, nous avons élaboré un ensemble des propriétés (cet ensemble n'est pas exhaustif) que nous considérons utiles pour la gestion des règles et de leur qualité. Nous illustrons leur avantage par le biais de l'exemple suivant:

Soient R_1, R_2 and R_3 trois règles définies comme suit:

R_1: SI $\varnothing$ ALORS $x > y$ *DANS* E ,

R_2: SI $\big(S(x,2) \vee R(x,z)\big) \wedge A(x,y,z)$ ALORS $y = x$ DANS E ,

R_3: SI $B(x,y)$ ALORS $x = y$ DANS E

Soient
$C1 = \bigvee_{i=1}^{i=n} C_i$ *et* $C2 = \bigvee_{i=1}^{i=k} C'_i$: 2 *Conditions* , L_i *et* L_2: *Conclusions*, l_1*et* l_2: *littéraux et*, E *et* E': *Environnement*

Propriété	Expression	Description
Satisfiabilité	Si $\exists j\ tel\ que\ Cj \in C1\ \ telque$ $SI\ CJ\ ALORS\ L_1 DANS\ E\ est\ vérifié$ Alors $SI\ C1\ ALORS\ L_1 DANS\ E\ est\ vérifié$	Il suffit qu'une conjonction élémentaire et la conclusion soient vraies pour que la règle soit vraie
Unification	Si $SI\ C1\ ALORS\ L_1 DANS\ E$ et $SI\ C2\ ALORS\ L_1 DANS\ E$ Alors $SI\ C1 \vee C_2\ ALORS\ L_1 DANS\ E$	Si pour une même Conclusion appartient à deux règles différentes dans la même structure alors nous unifions ces règles.
Consistance des conclusions	$SI\ C1\ ALORS\ L_1 DANS\ E$ et $SI\ C1\ ALORS\ L_2 DANS\ E$ sont valides si et seulement si L_1 et L_2 ne sont pas contradictoire	Vérification du principe de la non contradiction
Expiration	Si R1 = SI C1 ALORS L_1DANS E , R2 = SI C2 ALORS L_1DANS E , C2 ⊂ C1 et R1 expire avant R2 ALORS le système désactive R2 jusqu'à expiration de R1	L'expiration se calcule à partir de la caractéristique Durée_Validité. R2 se réveille dés l'expiration de R1 (calcul de la date Date_Réveil)
Consistance des littéraux	Si $R1 = SI\ C1\ ALORS\ L_1 DANS\ E$ et $l_1, l_2 \in C_i\ /\ l_1 = \neg l_2$ Alors $R1 = SI\ C1 - \{C_i\}\ ALORS\ L_1 DANS\ E$	Si deux littéraux sont contradictoires alors la conjonction élémentaire qu'il les contient est contradictoire
Consistance des conjonctions élémentaires	Si $R1 = SI\ C1\ ALORS\ L_1 DANS\ E$ et $C_1, C_2 \in C_i\ /\ C_1 = \neg C_2$ Alors $R1 = SI\ \varnothing ALORS\ L_1 DANS\ E$	Si deux conjonctions élémentaires sont contradictoires alors la condition qu'il les contient est valide toujours. La règle sera donc définie uniquement par la conclusion.
Distinct	$C_1 \cap L_1 = \varnothing$	La conclusion d'une règle ne doit pas identique à une conjonction élémentaire de la règle ou même à un littéral
Boucle	Si R1 = SI C1 ALORS L_1DANS E , R2 = SI C2 ALORS L_2DANS E alors L1 ⊄ C2 et L2 ⊄ C1	Si L1 ⊂ C2 et L2 ⊂ C1, alors La R1 déclenche la R2 et R2 déclenche R1. Donc le système ne s'arrête pas.

Tableau 14 : Quelques propriétés de base des règles.

Par application de l'ensemble complet d'opérateurs relationnels du tableau 13, la règle R_1 sera réécrite comme suit : $\mathrm{R_1}$: SI $\varnothing$ ALORS $\neg(\mathrm{x} <= y)$ DANS E ,

Par application de la propriété de l'unification des règles sur R_2 et R_3, nous obtenons une règle résultante R_{23} écrite comme suit :

$$\mathrm{R_{23}}: \mathrm{SI}\ (\mathrm{A(x,y,z)} \wedge \mathrm{R(x,z)}) \vee (\mathrm{A(x,y,z)} \wedge \mathrm{S(x,2)}) \vee \mathrm{B(x,y)}\}\ \mathrm{ALORS}\ \mathrm{x} = \mathrm{y}\ \mathrm{DANS}\ \mathrm{E}$$

Les propriétés que nous proposons dans le tableau 14 sont inspirées des propriétés de la logique telles que la consistance et satisfiabilité. . Les références [85-87] nous y rapporteront plus de détails à propos de ces propriétés.

4. Système de gestion de l'entrepôt des règles

A ce stade, nous avons uniquement défini le formalisme unifié de représentation des règles et de leur qualité. Pour atteindre l'objectif principal de ce chapitre qu'est la gestion des règles et de leur qualité, nous proposons la conception d'un système de gestion de l'entrepôt des règles dénoté SGER.

Dans la figure 20, nous montrons l'architecture du système proposé. Ce système comporte deux sous-systèmes : un pour la gestion des règles et l'autre pour la gestion de la qualité des règles, un processus d'entreposage des règles et un module pour le rafraîchissement de l'entrepôt des règles.

Ce système s'inspire des architectures des systèmes à base de règle et à base de connaissances, d'entreposage des données et d'acquisition des connaissances et utilise l'approche GQM pour la collection des informations nécessaires à la gestion des règles et de leur qualité.

4.1. Identification des structures des données du SGER

Nous avons identifié trois types des structures : structures d'alimentation ou structures en entrée, structures intermédiaires et structures en sortie. Dans ce système, nous définissons une structure comme étant une base qui peut contenir soit des données, soit des règles, soit des connaissances. Toutes ces sources suivent le schéma relationnel.

a) Structures d'alimentation

Ce sont les sources à partir des quelles le système extraient les connaissances et les règles par élicitation ou automatiquement. Ces sources sont:

1. les connaissances du domaine qui sont toute source autre que les bases des données. il peut être : un humain (expert, utilisateur, ..), livre, revues périodique, les modèles des données, etc... Comme l'acquisition des connaissances du domaine est coûteuse en termes de temps, nous avons décidé d'utiliser ces sources d'une manière incrémentale.
2. Les programmes peuvent être des sources des connaissances extraites par l'utilisation des techniques de découpage des programmes[22] . La référence [192] nous y rapportera plus de détails à propos de ces techniques.
3. Les actions atomiques : comme une règle fait appel à des différentes actions atomiques pour faire les corrections des erreurs détectées, il est nécessaire de les stocker dans une base approprié. Notons bien que dans ce travail, nous supposons que cette base existe et que le système accède à ces actions sans contraintes.

[22] En anglais Program Slicing

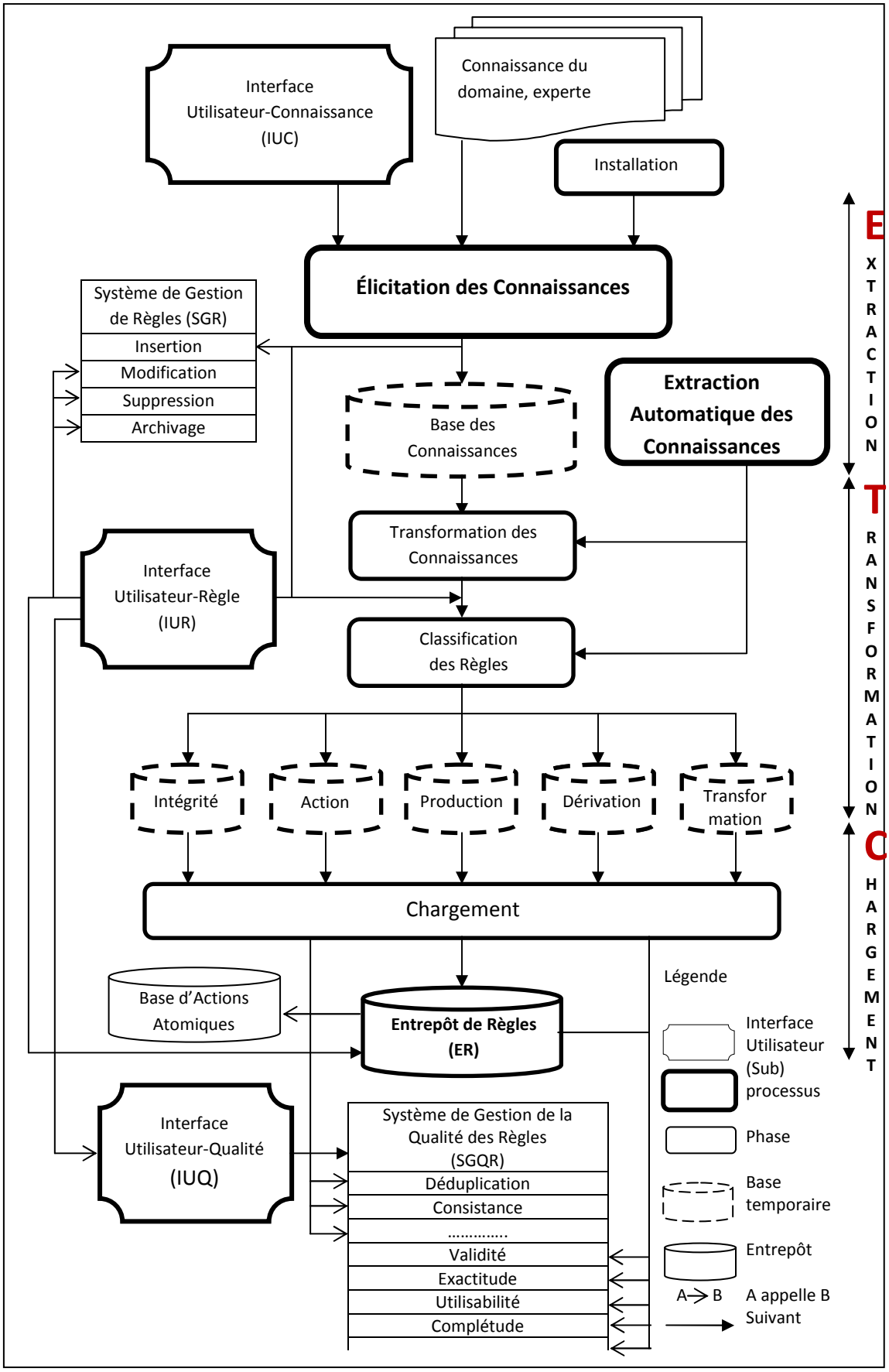

Figure 20 : Système de gestion de l'entrepôt de règles.

Ces structures sont les sources opérationnelles du processus d'entreposage des règles proposé.

b) Structures intermédiaires

Nous définissons les structures intermédiaires comme étant les bases de connaissances (BC) et les bases de règles temporaires similaires à la ZPD du système d'entreposage des règles. Dans ces structures, le système proposé effectue des transformations sur les règles et les connaissances ainsi qu'il évalue et améliore leur qualité. Notons toutefois que les contenus de ces structures seront détruits après leur chargement dans l'entrepôt des règles. Comme le montre la figure 20, nous avons introduit les structures intermédiaires suivantes: Base de Connaissance (BC) et cinq bases des règles. La base de connaissance est utilisée pour stocker les informations sur les connaissances. Comme nous avons souligné dans le chapitre 1 qu'il existe cinq types de règles, nous avons construit pour chaque type de règle une base correspondante. Ceci est important car les règles sont écrites et sont extraites dans des différentes formes. De ce fait, notre système autorise l'introduction des différents types de règles puis il les transforme à l'aide d'un ensemble des règles que nous présenterons dans la suite de ce chapitre (ces règles sont écrites sous selon notre formalisme).

c) Structure en sortie

La structure en sortie est l'entrepôt des règles dénoté ER qu'est la base physique externe où seront stockées les règles et les métarègles. La séparation des règles des données permettent leur réutilisation et leur partage. Les règles entreposées ont les mêmes caractéristiques que les données entreposées (intégrées, orientées sujets, historisées) sauf que les règles sont volatiles. Nous avons voulu des règles volatiles afin de permettre l'ajout des informations concernant à ces règles après leur chargement dans l'ER. Cela est important car dans la plupart des cas, nous n'arrivons pas à collecter toutes les informations d'une règle avant leur chargement dans l'ER.

4.2. Description des composantes du système de gestion de l'entrepôt des règles

Dans cette section, nous décrivons en détail les différentes fonctionnalités du SGER proposé. Dans la figure 20 qui décrit l'architecture conceptuelle du système SGER, nous avons présenté les différentes composantes fonctionnelles (processus et tâches) du système proposé qui sont : installation du système, construction de l'environnement de gestion des règles, processus d'entreposage des règles et gestion de la qualité des règles et des connaissances. Nous détaillerons ces fonctionnalités dans la suite de ce chapitre.

4.2.1. Installation du système

Nous avons introduit cette tâche, pour permettre à l'utilisateur de définir les informations et conditions nécessaires au bon fonctionnement du système avant sa mise en œuvre. L'ensemble des informations sont (mais pas exhaustif) :

- Les chemins des sources des données et des programmes,
- Les personnes autorisées à utiliser les systèmes,

- Les sources des connaissances du domaine (humain, livre, …),
- Les dimensions et métriques de qualité le plus fréquemment utilisées au niveau de l'entreprise autres que celles que nous avons proposé dans la section 3.3.3. qui sont les dimensions qualité le plus fréquemment utilisées dans les travaux de recherche,
- Les mots clés des connaissances du domaine afin de regrouper les connaissances extraites pendant l'élicitation des connaissances.
- Les sources des données, attributs et tables liées à chaque mot clé.
- …..

La tâche d'installation permet aussi la création de l'environnement initial (Envir_Init). Ces informations sont accessibles et utilisables par toutes les règles, plus particulièrement ce sont les dimensions qualité ainsi que leurs métriques définies par le système et celles définies par les utilisateurs (experts).

4.2.2. Construction de l'environnement des règles

Comme nous l'avons évoqué auparavant, l'environnement comporte trois sous ensembles : Envir_Init, Envir_Glob et Envir_Règle. Nous avons déjà présenté comment se fait la construction de la Envir_Init au cours de la description de la tâche d'installation du système. Nous décrivons dans cette partie la manière de construction des deux autres sous ensembles de l'environnement.

a) Environnement global

Cette partie de l'environnement se construit au cours et/ou pendant l'entreposage des règles. Elle permet à l'utilisateur de créer et définir certaines dimensions de qualité ainsi que leurs métriques pour un groupe donné des règles et les sources où ces règles seront applicables. Notons bien que les règles sont regroupées selon les mots clés ou par l'utilisateur. Par exemple dans une administration nous regroupons les règles en trois groupes : règles sur l'avancement et promotion des employés, règles sur la comptabilité et règles sur la gestion stages des employés.

b) Environnement de la règle

Nous avons proposé de créer un environnement spécifique à chaque règle pour permettre la gestion de l'évolution des règles et de leur qualité ainsi que de permettre leur formation incrémentale (mise à jour). Ces informations concernent généralement la précision:

- Des chemins d'actions,
- Des sources où les règles seront applicables, et
- Des dimensions de qualité pour chaque règle (autre que celles indiquent précédemment dans les environnements initial et global).

4.2.3. Interfaces utilisateurs

Afin d'assurer une interactivité forte entre le système SGER et l'utilisateur, nous avons doté le système proposé de quelques interfaces utilisateurs. Comme le montre la figure 20, nous avons proposé trois types d'interfaces : Interface Utilisateur-Règle (IUR), Interface Utilisateur-Connaissance (IUC) et Interface Utilisateur-Qualité (IUQ). L'interface utilisateur-connaissance est utilisée afin de permettre la manipulation les mots clés et les connaissances avant leur transformation. L'interface utilisateur-qualité permet la manipulation des dimensions qualité ainsi que leurs métriques. L'interface utilisateur-règle permet la manipulation de certaines caractéristiques des règles telles que la date de validation, réveil, effectivité, activation, statut, et regroupement des règles. Cette interface permet aussi à l'utilisateur d'introduire des nouvelles règles. Comme notre objectif est la gestion des règles et de leur qualité, cette interface fait appel à celle de l'utilisateur-qualité afin de gérer la qualité des règles pendant leur introduction.

4.2.4. Entreposage des règles

Pour une meilleure gestion des règles et de leur qualité, nous avons proposé l'entreposage des règles. Comme le montre la figure 2O, le processus d'entreposage des règles est inspiré de celui d'entreposage des données. Cela veut dire que l'entreposage de règles est basé sur un processus d'Extraction, Transformation et Chargement des règles (ETC) mais son fonctionnement est différent de celui d'entreposage des données. Le processus ETC des règles acquise les connaissances par élicitation et automatiquement. Puis, il les transforme sous forme des règles en appliquant une suite des transformations élémentaires sur les connaissances acquises. Finalement, il les charge dans l'entrepôt de règles. Le système offre d'autres fonctionnalités telles que le rafraîchissement de l'ER.

4.2.4.1. Extraction des connaissances et des règles

Partant du principe général de notre proposition qu'est la personnalisation de l'amélioration de la qualité des données, ce système comprend un module d'élicitation des connaissances pour permettre à l'utilisateur d'intégrer ces connaissances expertes et du domaine dans la gestion des règles et, plus spécifiquement, de leur qualité. Nous avons conçu le système indépendamment du système de gestion de la qualité des données afin de permettre l'incorporation des règles autres que celles orientées qualité. De ce fait, le système autorise l'entreposage des règles qui peuvent être utilisées pour d'autres fonctionnalités autres que la qualité en assurant leur (règles) qualité.

a) Élicitation des connaissances

L'objectif de ce module est de nous permettre l'extraction des connaissances à partir des sources autres que celles des systèmes opérationnelles des données (données et programmes). L'élicitation est coûteuse en termes de temps car elle est quasiment manuelle. De ce fait, nous avons conçu l'entreposage des règles séparément des processus d'entreposage des données et d'extraction des connaissances à partir des données (ECD et ED). Cela veut dire que l'extraction

des connaissances peut se faire périodiquement (à des différents temps) et d'une manière incrémentale.

L'élicitation des connaissances est itérative et interactive et se fait en deux phases : la phase de collection des informations et la phase de préparation des informations. La collection des informations est basée sur l'approche GQM.

- **Collection des informations**

  La construction des connaissances nécessite la collection des informations à partir des sources telles que les experts, livres ou questionnaires. Afin de réaliser cette tâche, nous avons conçu un formulaire questionnaire reposant sur le principe de l'approche GQM que nous avons décrit dans le chapitre 2. Ainsi, ce questionnaire comporte trois parties : objectif, questions et réponses. Comme les informations collectées des utilisateurs peuvent être de mauvaise qualité ou mal interprétées, nous avons regroupé les utilisateurs dans trois groupes: employé, spécialiste et expert. Ce regroupement permet l'assurance de la qualité des informations collectées surtout en cas de conflit entre les réponses des utilisateurs. Pour assurer la qualité des réponses, nous avons utilisé deux règles dites règles de priorité des réponses qui sont :

  Règle 1 : L'expert est prioritaire par rapport à l'employé et le spécialiste.

  Règle 2 : Le spécialiste est prioritaire par rapport à l'employé.

  Cependant, notons bien que l'expert peut être externe à l'entreprise (n'est pas un employeur de l'entreprise) où le système sera installé. Nous avons aussi regroupé les utilisateurs en deux groupes: utilisateur monoposte et utilisateur multipostes. Utilisateur monoposte est un utilisateur qui travaille dans un poste (fonction) bien déterminé. Cet utilisateur peut donner des informations propres uniquement à son poste de travail. Cependant, l'utilisateur multipostes peut donner des informations liées aux différentes postes occupés avec leurs dépendances.

  Le système regroupe les informations qui expriment des connaissances par rapport aux mots clés afin de faciliter leur préparation dans la phase suivante. Dans la figure 21, nous donnons la forme générale d'un formulaire questionnaire de collections des informations avec un exemple.

  Une fois les remplissages des questionnaires sont faits par les utilisateurs, nous passerons à leur saisie et leur stockage dans la base de connaissances (voir figure 20). La saisie se fait via l'interface IUC. Le système autorise à l'utilisateur de répondre directement au questionnaire via l'interface IUC sans passer par le remplissage de formulaire. Notons bien que dans ce système, nous considérons chaque réponse à une question du formulaire comme étant une connaissance brute qui nécessite une préparation.

volet	Description avec exemple
Objectif	Il comporte un mot clé principal et mots clés secondaires Exemple : Mot clé principale: Avancement Mots Clés secondaires associés : Échelon, Promotion, Retraite, Congés, Sanctions, Validation Expérience,.., etc.,
Questions	Question 1 : Description du mot clé principal. Question 2 : Relations entre le mot clé principal et les mots clés secondaires Quelle est la relation entre Avancement, Échelon et Congé. Question 3 : Relations entre les mots clés secondaires. Quelle est la relation entre la validation de l'expérience et échelon.
Métriques	Les métriques seront déterminées après remplissage des réponses du formulaire par un ingénieur de connaissances (Informaticien dans notre cas) Exemple Réponse Question 2 : Si l'employé n'est pas en congé alors il a droit à l'avancement. L'employé peut aller jusqu' à 12 échelon Métrique déduite: les Échelons sont de 0 à 12 (Domaine de valeurs)

Figure 21 : Forme générale du formulaire-questionnaire

- **Préparation des informations**

  La phase de préparation des informations est la première opération de l'amélioration de la qualité des informations stockées dans la base des connaissances. Cette phase comporte un ensemble des opérations, le plus souvent manuelles, qui seront appliquées sur les informations collectées par le responsable de la collection des connaissances (ingénieur de connaissance, informaticienne ou autre). Parmi ces transformations que nous proposons, nous citons :

  1. Élimination des connaissances brutes de mauvaise qualité : Cette tâche consiste à mesurer l'utilité et l'exactitude des connaissances par les spécialistes (réponse contradictoire, n'a pas d'importance, incompatible avec la question).
  2. Sélection des connaissances pertinentes qui peuvent être transformées sous forme de règles.
  3. Ajout des informations manquantes telles que la description de la connaissance.
  4. Regroupement des connaissances selon les mots clés principaux (cette opération est automatique).

Pour chaque groupe, nous appliquons les opérations (5) et (6) suivantes :

5. Élimination des connaissances redondantes (semi automatique).
6. Résolution des conflits entre connaissances brutes par application des règles des priorités ou par l'intervention de spécialiste.
7. Répétition de l'opération 4 de regroupement mais cette fois ci par rapport aux mots clés secondaires, puis l'opération (5) et (6).

Notons bien que l'élicitation des connaissances est coûteuse en termes de temps pendant la construction initiale de l'entrepôt des règles.

b) Extraction automatique des connaissances

Ce module permet l'application des outils d'extraction automatique des connaissances à partir des systèmes opérationnels des données (données et programmes). A cette fin, nous utiliserons le processus d'ECD et les outils e découpage des programmes. Cependant, le problème majeur de ces méthodes est qu'elles ne permettent pas l'interopérabilité des résultats. Dans ce travail, nous n'avons pas traité ce problème car il est possible de réaliser des outils d'extraction des connaissances, le plus souvent sous forme de règles d'association, propres aux entreprises en assurant l'interopérabilité et l'échange des connaissances.

4.2.4.2. Transformation des connaissances et des règles

La phase de transformation des connaissances et des règles sert à effectuer des opérations sur les connaissances et les règles extraites pendant la phase précédente. Elle comporte deux étapes : la transformation des connaissances et la classification des règles.

a) Transformation des connaissances

A la différence des systèmes à base de connaissances et règles qui réalisent les transformations pendant l'acquisition des connaissances, nous les réalisons dans ce système pendant la phase de transformation des connaissances et règles afin de permettre d'appliquer les transformations sur les règles extraites automatiquement et les connaissances acquises par élicitation dans en une seule phase. Ceci est important surtout pour certaines opérations telles que le déduplication des connaissances et la résolution de l'inconsistance et conflits entre les connaissances extraites automatiquement et celles acquises par élicitation. L'intervention de l'expert est indispensable pendant cette phase. Parmi les transformations que nous proposons de réaliser pendant cette phase, nous citons:

1. Décomposition des connaissances en connaissances atomiques. Cette opération est faite manuellement par le spécialiste de gestion des connaissances et elle concerne uniquement les connaissances acquises par élicitation.
2. Déduplication des connaissances atomiques.
3. Transformation des connaissances atomiques sous forme de règles ou de modèle mathématique en utilisant les symboles de désignation des sources des données, tables et attributs associées à la connaissance (cette opération concerne uniquement les connaissances acquises par élicitation).
4. Ajout de quelques informations aux connaissances et règles si nécessaire.
5. Détermination des dimensions qualité correspond à chaque connaissance.

b) Classification des règles

Cette étape concerne toutes les règles. Elle consiste à affecter chaque règle à la base correspondante (intégrité, action, production, dérivation et transformation). Nous avons séparé

cette opération des transformations des connaissances car elle consiste à calculer les mesures de qualité pour chaque règle avant leur réécriture selon le formalisme proposé. Pendant cette phase, nous effectuons les tâches suivantes :

1. Identification du type des règles: le système identifie le type des règles en les comparant aux modèles de chaque type de règle qui nous avons présenté dans le métamodèle de la figure 19,
2. Affectation de chaque règle à la base correspondante. Les modèles mathématiques sont par défaut traités comme étant des règles d'intégrité.
3. Transformations des mesures de qualité : Cette opération concerne les règles extraites automatique, plus spécifiquement les règles d'association, par les outils de fouille de données qui calculent certaines métriques de qualité des règles pendant l'extraction telles que le support et confiance. Ces métriques seront utilisées par des algorithmes de transformations de métriques que nous présenterons dans la suite de cette thèse afin de calculer les valeurs initiales de certaines métriques de qualité des règles après leur réécriture selon notre formalisme. Par exemple le support et la confiance seront combinés pour calculer la valeur de l''exactitude de la règle.

4.2.4.3. Chargement des règles

L'objectif de cette phase est le chargement des règles dans l'ER à partir des bases intermédiaires des règles. Chaque règle doit subir trois opérations que nous avons élaborées pour l'amélioration de la qualité des règles pendant leur chargement. Ces opérations sont: Transformation de règle, calcul des dimensions et métriques de qualité, et vérification de la qualité de l'entrepôt des règles.

a) Transformation des règles

Elle permet la réécriture de la règle selon le formalisme que nous avons décrit dans ce chapitre. Pour toutes les règles de toutes les bases intermédiaires, le système applique une série de transformations élémentaires. Ces transformations sont :

1. **Transformation de la condition**: Elle concerne la transformation de la condition d'une règle R en condition écrite selon notre formalisme. La forme générale de cette transformation est comme suit :

$$\text{Transformation}\left(\text{Condition}(\text{R : règle temporaire})\right) = \bigvee_{i=1}^{i=n} C_i$$

2. **Transformation de la conclusion** : Elle sert à réécrire la conclusion d'une règle R selon notre formalisme. La forme générale cette transformation est comme suit :

$$\text{Transformation}\left(\text{Condition}(\text{R : règle temporaire})\right) = \text{Conclusion}$$

Ces deux transformations permettent la réécriture de la condition et la conclusion des règles des bases temporaires à l'aide des systèmes complets d'opérateurs et leurs mises sous forme normale disjonctive.

3. **Transformation de l'univers**: Elle sert à déterminer l'univers de la règle. Ce dernier est généralement indiqué dans les bases de règles intermédiaires. S'il est vide, le système suppose que la règle est applicable dans toutes les sources.

$$\text{Sources\_Règle} := \text{Transformation (Univers)}$$

4. **Transformation du Composant**: Elle permet de déterminer certaines caractéristiques autres que celles transformées par les trois transformations précédentes. Généralement, elle extrait l'événement, l'action, la postcondition et les informations concernant les caractéristiques décrites dans le tableau 11 (Expression_Règle, Règle-Code, Date_Création, Texte_Règle). L'évènement et les actions s'extraient directement des bases des règles intermédiaires. Cependant la postcondition sera transformée à l'aide de la première règle transformation de la condition décrite ci-dessus. La Date_Création est la date de chargement de la règle dans l'ER. Le code de la règle (Code_Règle) est attribué automatiquement par le système.

Cependant, le système ne transforme que les règles qui possèdent au moins une conclusion. Le reste des informations peuvent être compléter après le chargement des règles dans l'ER via l'interface utilisateur IUR.

b) Calcul des dimensions et métriques de qualité

Nous avons créé cette opération afin de déterminer les valeurs initiales des dimensions et des métriques de qualité. Pour atteindre cet objectif, nous avons établi l'algorithme présenté dans la figure 22 qui est l'algorithme général de la transformation des dimensions de la qualité. Cela veut dire que si les règles avant leur chargement ont des mesures de qualité, il est essentiel d'établir des fonctions algorithmiques permettant de calculer les mesures prédéfinies par le système pendant la phase d'installation à partir de ces mesures de qualité des règles dans les bases intermédiaires. Dans la figure 22, nous avons utilisé F pour désigner la fonction permettant de calculer l'exactitude à partir de support et confiance. Cette opération concerne uniquement les règles extraites automatiquement par les outils de FD qui les valident sur un échantillon de données en calculant certaines mesures de qualité.

Le principe de l'algorithme est basé sur la méthodologie TDQM que nous avons détaillée dans le premier chapitre. Cela veut dire que la création des dimensions et de leurs métriques de qualité d'une règle passe par les quatre étapes proposées dans TDQM : définition (description de la dimension), mesure (détermination des métriques nécessaires pour évaluer la dimension), analyse (détermination des seuils aidant le système à décider sur la qualité de la règle) et amélioration (quelles sont les actions à faire en cas où la règle est évaluée de mauvaise qualité).

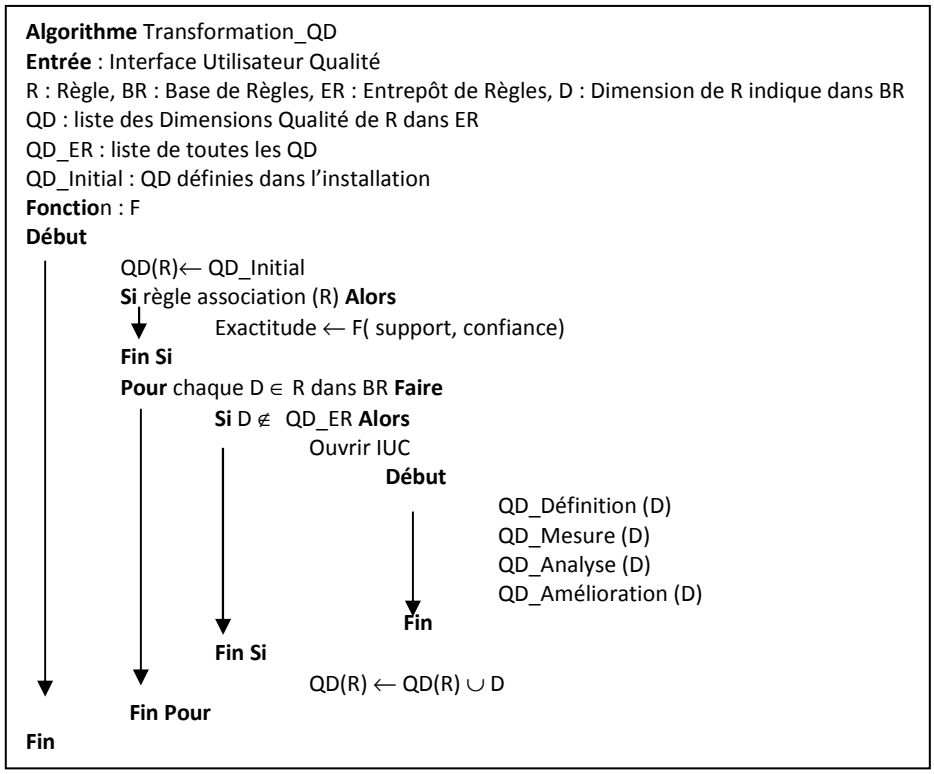

Figure 22 : Algorithme de transformation des dimensions de la qualité

Comme la complexité est un point sensible pour les algorithmes, nous avons étudié la complexité temporelle de l'algorithme que nous avons proposé pour la transformation des dimensions de la qualité. Dans cette étude nous avons tenu compte de deux paramètres:

1. le nombre de règles dénoté n parce que notre algorithme est applicable sur toutes les règles pendant la construction initiale de l'entrepôt des règles qui est défini comme étant le cardinal de l'union des cinq bases intermédiaires des règles. La cardinalité signifie le nombre de règles.
2. Le nombre de dimensions qualité dénoté m

De ce fait, l'algorithme est de complexité quasi linéaire $O(n (\ln n))$ dans le pire des cas et est de complexité linéaire $O(m)$ dans le meilleur des cas. Notons bien que le pire des cas représente le cas de la construction initiale de l'ER et le meilleur des cas représente le cas de l'insertion d'une règle dans l'ER

c) Vérification de la qualité de l'entrepôt des règles

Avant le stockage des règles dans l'ER, le système doit vérifier la cohérence et la consistance de l'ER avec la règle à entreposer. La qualité de l'ER nécessité la vérification des propriétés des règle décrites dans le tableau 14 sauf celle de satisfiabilité. Une fois la règle est validée cohérente avec les règles de l'ER, elle sera stockée dans l'ER et automatiquement supprimée de la base intermédiaire. Comme nous l'avons évoqué auparavant, la date de création de la règle est la date de leur chargement dans l'ER.

4.2.5. Système de gestion des règles

A la différence des entrepôts des données, les entrepôts des règles peuvent être mis à jour via les interfaces IUR et IUC. Par exemple, l'ajout de quelques informations manquantes ou l'archivage d'une règle. Pour cela, nous avons doté le SGER par un système de gestion des règles (qu'est un sous système pour le système de SGER) dans l'ER (voir figure 20). L'objectif fondamental de ce système de gestion de règle est de permettre:

- aux spécialistes d'insérer des règles qui respectent le formalisme que nous avons proposé dans l'ER. Dans ce cas, le système doit uniquement vérifier la cohérence de l'ER et la règle.
- la mise à jour de certaines informations des règles dans l'ER.
- l'archivage et l'activation des règles.
- L'ajout, suppression et modification des règles pendant la phase de transformation.
- L'ajout des informations manquantes des règles (action, événement, sources des données, etc.)

4.2.6. Système de gestion de la qualité.

Nous avons ajouté ce système (qu'est un sous système pour le système de SGER) de gestion de qualité au SGER afin de permettre une gestion continue des règles par l'utilisateur. La plupart de ces fonctionnalités sont opérables via l'interface utilisateur qualité IUQ. Ce système permet :

- La gestion des dimensions et métriques de qualité : ajout, modification,.., etc.
- La vérification de la qualité des règles séparément du processus de l'entreposage des règles.
- La vérification de la qualité de l'ER périodiquement (Déduplication, cohérence, déterminisme).
- La vérification de l'expiration, réveil, (dé)activation des règles automatiquement à chaque lancement du système.

Dans ce système, nous avons proposé de créer un échantillon des données afin de permettre à l'utilisateur si nécessaire d'évaluer automatiquement la qualité de certaines règles dans l'ER avant leur mise en œuvre.

5. Conclusion

Dans ce chapitre, nous avons présenté notre système de gestion de l'entrepôt des règles. Ce système nous l'avons conçu afin de permettre la gestion des règles et de leur qualité. Afin d'atteindre cet objectif, l'architecture globale de notre système comprend des modules permettant l'acquisition des connaissances par élicitation et par des outils d'extraction automatique des connaissances tels que l'ECD. L'élicitation des connaissances est importante car

elle permet l'exploitation des connaissances du domaine et expertes pour des fins diverses telles que l'amélioration de la qualité des données dans notre cas.

Cette architecture est soutenue par notre formalisme unifié de représentation des règles que nous l'avons conçu pour permettre la représentation de tous les types de règles. Ce formalisme est composé de trois parties complémentaires. Il permet la représentation des règles et de leur qualité. Son avantage majeur est qu'il permet la construction de la règle d'une manière incrémentale et interactive. Il tient aussi compte de la relativité de l'évaluation de la qualité des règles par rapport aux sources où ces règles sont opérables. Ce système repose sur un processus d'ETC adapté aux règles.

L'objectif majeur de la conception de ce système de gestion de l'entrepôt des règles est qu'il sera utilisé dans un processus d'extraction des connaissances à partir des données pour assurer la qualité des données et des connaissances. L'intégration du système proposé et du processus d'extraction des connaissances à partir des données nécessite l'adaptation de ce dernier (ECD).

De ce fait, nous abordons dans le chapitre suivant l'adaptation de l'ECD.

Chapitre 4: Adaptation des processus d'ECD et d'entreposage des données

« Le principe de l'évolution est beaucoup plus rapide en informatique que chez le bipède.»
Jean Dion

1. Introduction

Un système d'évaluation et d'amélioration de la qualité des données repose généralement sur le processus du nettoyage des données qui est la tâche la plus coûteuse en termes du temps dans les processus d'ECD et d'entreposage des règles. Les travaux de recherche que nous avons présenté dans le deuxième chapitre ont montré la complémentarité et la synergie de ces deux processus, et que l'ED est un élément essentiel du processus ECD. Par conséquent, Il est important de tenir compte ces spécificités dans la conception de ces processus.

Même si les travaux de recherche actuels dans le domaine de la qualité des données et des connaissances dans l'ECD s'inspirent des travaux réalisés dans celui des bases des données et des bases de connaissances, il n'en demeure pas moins que les ECDs ont des spécificités qu'il faut prendre en compte. Dans ce travail, nous avons concentré sur les spécificités suivantes : (1) l'ED doit être considéré comme étant le point de départ du processus d'ECD, (2) la qualité doit être gérée tout au long du processus ECD, (3) la qualité des données (sources des données opérationnelles des données et ED) conditionne la qualité des connaissances extraites par l'ECD, (4) le nettoyage des données à base de règles est plus adopté pour les ECD car il permet la personnalisation (incorporation de l'utilisateur dans l'amélioration de la qualité) et est systématique, (5) les connaissances après leur transformation sous forme de règles peuvent être utilisées pour l'améliorations de la qualité des données à partir des quelles sont formées ces connaissances.

Après la proposition d'un système de gestion de l'entrepôt des règles dans le chapitre précédent qui tient compte de certaines spécificités, nous allons présenter dans ce chapitre notre le processus d'ECD que nous avons adapté pour permettre une meilleure prise en charge de la qualité des connaissances et des données. Ainsi, l'adaptation de l'ECD, nous a conduit à la proposition des solutions suivantes:

- L'adaptation de l'ED qui repose sur l'extension du processus d'ETC afin d'améliorer les performances d'entreposage des données -en termes de temps d'exécution et de qualité des données- et par la suite celles de l'ECD.
- La conception d'un système de propagation des corrections faites sur les données évaluées de mauvaise qualité vers leurs sources originales (l'amélioration de la qualité doit être faite aux niveaux des sources originales et cibles des données afin d'éviter de refaire les mêmes opérations d'améliorations de la qualité à chaque entreposage des données).
- La réalisation d'un processus de traçabilité des données corrigées pour faciliter la propagation des corrections des données vers leurs sources et, par conséquent, leur validation par les utilisateurs.

- La proposition d'un système de capture des changements des données corrigées après leur validation par l'utilisateur. Cela est important car il permet d'évaluer la qualité des règles et de recorriger les données si nécessaires.
- L'adaptation du processus ECD afin de permette l'intégration du système de gestion de l'entrepôt des règles que nous avons présenté dans le chapitre précédent et aussi les propositions décrites ci-dessus.

Notons bien que l'utilisateur occupe une place centrale dans les travaux que nous avons menés dans cette thèse et que le terme utilisateur désigne à la fois l'utilisateur final et l'administrateur.

2. Objectifs et apports de l'adaptation du processus d'entreposage des données

Comme nous l'avons évoqué, l'adaptation du processus d'ECD que nous avons proposé dans [127, 191, 193] nécessite la personnalisation du système de gestion de la qualité des données. Cette personnalisation devient plus systématique par le système à base de règles que nous l'avons conçu dans le chapitre précédent. L'intégration de ce système dans le processus ECD requiert l'adaptation du processus d'entreposage des données. Cette adaptation permet de garantir des bonnes performances en termes de qualité des données et de temps d'entreposage des données et par conséquent le temps d'extraction des connaissances à partir des données

2.1. Objectifs du processus adapté

En outre l'objectif principal qu'est l'adaptation de l'ECD afin de permettre une meilleure prise en charge de la qualité, le processus d'entreposage que nous adaptons vise d'autres objectifs que nous qualifions nécessaires à notre travail. Ces objectifs sont:

1. L'utilisation du parallélisme pendant l'entreposage des données (construction initiale et rafraîchissement des EDs), ce qui implique un temps d'entreposage réduit et une bonne qualité des données.
2. L'assurance de la cohérence mutuelle des données des SDO_i et l'ED afin de permette l'amélioration de la qualité des sources des données opérationnelles (SDO) et de règles.
3. L'implication des utilisateurs des SDO dans l'amélioration de la qualité des données et des règles.

2.2. Apports du processus adapté

L'adaptation du processus d'entreposage des données que nous proposons s'articule sur les apports suivants :

1. L'ajout de deux phases au processus ETC classique.
2. La répartition et la séparation des données en trois sous ensembles avant leur transformation afin de permettre l'application du parallélisme et le nettoyage des données à base de règles.
3. L'ajout d'un processus de propagation des données corrigées vers les SDO_i avant leur chargement dans l'ED afin d'assurer la cohérence mutuelle des données.

4. L'adaptation du processus de nettoyage de données au nettoyage et à l'analyse par les techniques de FD. Ce processus utilise le système de gestion de l'entrepôt des règles que nous avons proposé dans le troisième chapitre.

3. Présentation du processus adapté d'entreposage des données

Dans la figure 23, nous présentons le processus d'entreposage des données que nous avons adapté. Son avantage majeur est qu'il entrepose les données sans pour autant influencer le processus d'entreposage des règles.

3.1. Composantes du processus adapté d'entreposage des données

Le processus proposé qu'est un n-uplets de structures des données et des processus est défini comme suit :

Processus ED adapté = <SDO, ZPD, ODS, ETCTC, SGER, PDC, TDC, CCDC, Utilisateur>

où

- SDO est l'ensemble des sources des données opérationnelles à partir des quelles sont formées l'ED et les connaissances: $\mathrm{SDO} = \{\mathrm{SDO_i}, \mathrm{i} \geq 1\}$
- ZPD est l'ensemble des zones de préparation des données où les transformations des données seront opérables: $\mathrm{ZPD} = \{\mathrm{ZPD}, \mathrm{mZPD_i}, \mathrm{i} \geq 1\}$
  Dans ce travail, nous avons proposé de créer pour chaque SDO_i une zone de préparation des données dénotée $mZPD_i$. C'est dans ces zones où seront opérées les règles mono source et multi sources. La ZPD est la zone globale de préparation des données après leur chargement des $mZPD_i$. C'est dans cette zone où seront opérées les règles agrégats.
- ODS est une base des données opérationnelles utilisée dans certaines architectures d'entreposage des données (pour plus d'informations sur l'ODS, nous y rapporte à la référence [194]) pour stocker des données proche des données opérationnelles pour le rafraîchissement de l'ED. Ce travail exploite cette base pour stocker les données corrigées et leurs métadonnées.
- ETCTC est le processus que nous avons conçu pour l'extraction, transformation et chargement des données. Comme le montre la figure 23, ce processus comporte deux nouvelles phases par rapport au processus ETC classique qui sont la monotransformation et le monochargement.
- SGER qu'est le système de gestion de l'entrepôt des règles que nous avons présenté dans le troisième chapitre.
- PDC est le processus de Propagation des Données Corrigées[23] que nous avons proposé dans [105].
- TDC est le processus de la Traçabilité des Données Corrigées[24] que nous avons proposé dans [128].

[23] En anglais : Cleaned Data Backflow

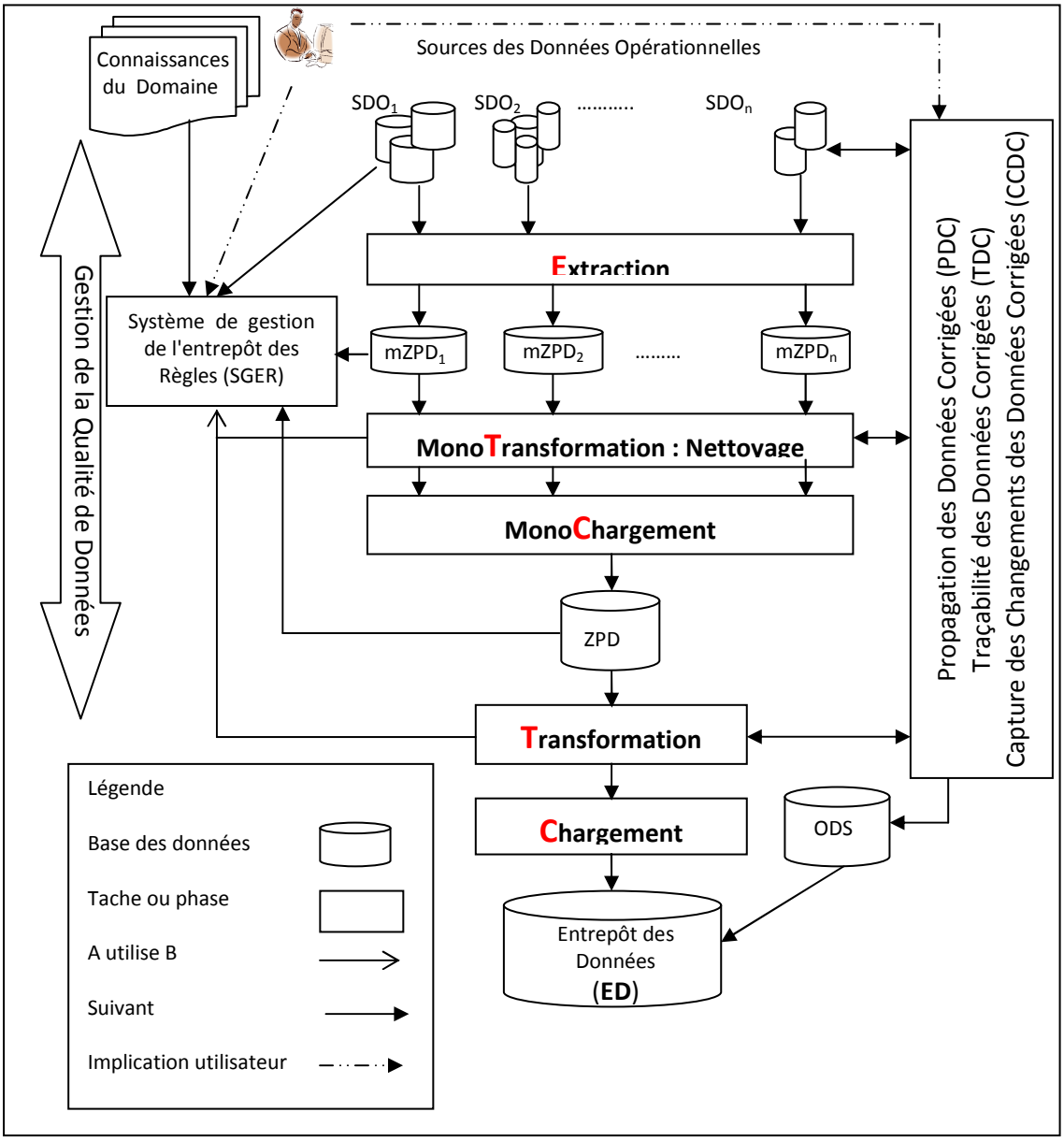

Figure 23 : Processus adapté d'entreposage des données.

- CCDC est le processus de Capture de Changement des Données Corrigées[25] que nous avons proposé dans [105].

Dans ce travail, nous définissons une donnée de mauvaise qualité comme étant la donnée elle-même et les données formées à partir de cette donnée. De ce fait les trois processus (PDC, TDC, CCDC) concernent les données corrigées et les données qui utilisent ces données.

Ce processus est conçu de façon à permettre une gestion totale et continue de la qualité des données et des règles. A la différence des autres travaux de recherche que nous avons présentés dans le deuxième chapitre qui réservent une phase unique pour le nettoyage des

[24] En nglais : Cleaned Data Lineage Tracing
[25] En anglais : Capture Cleaned Data Change

données, le processus adapté gère la qualité données tout au long du processus et d'une manière bidirectionnelle (c'est-à-dire à partir source des données opérationnelles jusqu'à leur chargement dans l'ED et vice versa).

Dans la suite de chapitre, nous détaillerons chaque composant du processus adapté d'entreposage des données.

3.2. Description du processus ETCTC

Le processus adapté d'entreposage des données se repose sur un processus d'extraction, chargement et transformation que nous avons conçu dans [197] spécialement pour permettre une meilleure prise en charge de la qualité des données et des règles. Ce processus est dénoté ETCTC pour Extraction, monoTransformation, monChargement, Transformation et Chargement des données. Il est une extension et adaptation du processus ETC classique. Nous détaillons ici le processus ainsi que les définitions de concepts de base et la règle d'évaluation de la qualité que nous avons introduites pour les performances de la qualité.

3.2.1. Définitions de quelques concepts de base

Nous définissons, dans cette partie, quelques concepts que nous avons introduits pour évaluer et garantir les performances de l'ETCTC en termes de qualité. Ces concepts concernent les données.

Concept 1 *:* Support des valeurs corrigées d'un attribut A

C'est le rapport de nombre des valeurs corrigées de l'attribut A dénoté Vc(A) et le nombre d'enregistrements de la table contenant l'attribut A dénoté m. On le note Svc(A) et il est défini comme suit :

$$\mathrm{Svc(A)} = \mathrm{Vc(A)} / \mathrm{m}$$

Propriétés*:*

- $0 \leq \mathrm{Svc(A)} \leq 1$.
- $\mathrm{Svc(A)} = 1$: toutes les valeurs de A sont de mauvaise qualité.
- $\mathrm{Svc(A)} = 0$: toutes les valeurs de A sont correctes.

Concept 2 *:* Support des valeurs corrigées d'un enregistrement E

C'est le rapport de nombre des valeurs corrigées dans un enregistrement dénoté Vc(E) d'une table quelconque et le nombre total des attributs dénoté Ne. On le note Tvc(E) et il est défini comme suit:

$$\mathrm{Tvc(E)} = \mathrm{Vc(E)} / \mathrm{Ne}$$

Propriétés *:*

- $0 \leq \mathrm{Tvc(E)} \leq 1$.
- $\mathrm{Tvc(E)} = 1$: toutes les valeurs de E sont de mauvaise qualité.

- Tvc(E) = 0 : toute les valeurs de A sont exactes.

Concept 3 : Ensemble Corrigé

Cet ensemble dénoté E_c contient uniquement les données corrigées. Ces données ont les supports précédents positives (Tvc(E) > 0 et Svc(A) > 0).

Concept 4 : Ensemble modélisable

Cet ensemble dénoté E_m contient les enregistrements qui ont un Tvc(E) = 0. Cet ensemble sera utilisé pour construire les connaissances et comme échantillon des données pour tester la qualité des règles.

Concept 5 : Ensemble valide

Cet ensemble dénoté E_v contient les données recorrigées par l'utilisateur. Ces données sont corrigées pendant les phases de transformation et monotransformation du processus ETCTC puis validées incorrectes par l'utilisateur.

3.2.2. Règle d'évaluation de la qualité

Nous avons formalisé une règle dite règle d'évaluation de la qualité qu'est stockée dans l'entrepôt des règles du SGER. Cette règle audite la qualité des données sur demande de l'utilisateur, plus spécifiquement l'administrateur de l'entrepôt des règles. L'idée derrière cette règle est de permettre à l'administrateur de s'informer de la qualité des données à entreposer à n'importe quel moment et de décider sur le processus d'entreposage (continuer ou arrêter). En pratique, l'évaluation de la qualité des EDs est faite généralement après leur construction et l'exploitation de ces données par des outils d'analyse telle que la fouille des données, requêtes SQL pour des fins d'analyses. Cependant la construction d'ED est coûteuse en termes de financement et de temps. D'où, Il est inutile de juger les EDs après leur construction. De ce fait, nous avons établi cette règle intégrée dans l'ER. La forme de la règle est la suivante :

$$\text{SI } \emptyset \text{ ALORS Conclusion DANS } (< \text{ZPD}, \text{mZPDi}, \neg(\text{i} < 1) >, \emptyset, < On - demande, \text{Action} >$$

$$\text{Conclusion} := \{\text{Tvc(x)} > \propto, \quad Svc(y) >\}$$

$$\text{Action} := \text{Choisir (Supprimer, Arrêter, Continuer)}$$

La Règle utilise les supports Tvc et Svc (1 et 2) décrites précédemment et deux seuils d'évaluation de la qualité (α etβ) donnés par les utilisateurs afin de leur permettre de prendre une décision à propos des données et/ou du processus d'entreposage des données. L'action de la règle offre à l'utilisateur trois choix : supprimer les données, arrêter le processus d'ER ou continuer. Il est préférable d'exécuter cette règle pendant la phase de monotransformation d'ETCTC. La règle est formalisé selon le formalise que nous avons proposé dans le chapitre précédent.

3.2.3. Description des différentes phases du processus ETCTC

Comme le montre la figure 23, L'ETCTC comporte cinq phases : extraction, monotransformation, monochargement, transformation et chargement. La première et la

dernière phase sont similaires à celles de l'ETC que nous avons détaillées dans le deuxième chapitre.

3.2.3.1. Phase d'extraction des données

L'extraction des données se déroule de la même manière décrite dans le chapitre 2 (section 1.3.3.1). Cependant, la différence avec l'ETC est que les données extraites de chaque SDO_i seront stockées dans la zone de préparation des données $mZPD_i$ correspondante à cette SDO_i. Notons bien que pendant la transformation le système crée des clés primaires pour les sources des données opérationnelles, les tables.

3.2.3.2. Phase de monotransformation des données

Pendant cette phase, le système applique les règles mono source et multi sources aux données des différentes zones intermédiaires $mZPD_i$. A la différence des travaux de recherche qui n'exploitent pas le parallélisme pendant la phase de transformation, la distribution des données sur ces différentes zones que nous avons créées permet l'exploitation du parallélisme des traitements et par conséquence nous atteindrons une haute performance en termes de temps des transformations des données. En outre, la spécification des règles applicables dans chaque zone est intéressante car elle permet aussi l'amélioration de temps de transformations des données. Les tâches principales réalisées pendant cette phase sont :

- Le système applique les règles correspondantes à chaque $mZPD_i$ aux données afin de détecter et corriger les erreurs mono-source. Les zones où la règle est applicable sont extraites à partir de l'univers de la formule.
- Les données corrigées seront stockées dans l'ensemble corrigé (E_c) afin de les propager vers leur source (SDO_i) en utilisant le processus de propagation des données corrigées (PDC) que nous allons détailler dans la suite de ce chapitre.

Notons bien que dans ce travail, toutes les transformations sur les données sont représentées sous forme de règles.

3.2.3.3. Phase de monochargement des données

La création de cette phase est une conséquence de la création de la phase précédente. Elle permet le chargement parallèle des données à partir des zones intermédiaires $mZPD_i$ vers la zone globale ZPD. Toute donnée chargée dans la ZPD sera automatiquement effacée de la $mZPD_i$.

3.2.3.4. Phase de transformation des données

Comme nous l'avons souligné précédemment, cette phase est similaire à celle du processus ETC classique. Les transformations s déroulent dans la zone ZPD. Pendant cette phase, nous nous intéressons uniquement aux règles multi sources et agrégats. Les opérations de base réalisées pendant cette phase sont :

- L'enrichissement des données.

- L'application des règles multi sources (standardisation, normalisation, conversion, discrétisation, réduction, fusion, filtrage, etc.) aux données.
- L'agrégation des données.
- L'application des règles agrégats sur les données agrégées.
- L'insertion des données agrégées corrigées dans l'ensemble corrigé (E_c): chaque donnée corrigée sera automatiquement enregistrée dans cet ensemble. Comme les données agrégées sont difficiles d'être analysées par les utilisateurs des sources des données opérationnelles, nous avons autorise l'administrateur d'accéder à ces données par le biais d'une interface utilisateur afin de décider sur les procédures à prendre face à ces données. Cependant le processus de propagation des données corrigées (PDC) envoie cette donnée aux utilisateurs des SDO ainsi que les données opérationnelles à partir desquelles est formée cette donnée (agrégée et corrigée). Théoriquement, nous pouvons déduire que la quantité des données agrégées de mauvaise qualité est négligeable devant celle des données opérationnelles de mauvaise qualité corrigées pendant la phase de monotransformation des données. Cela s'explique par le fait que la plupart des données sont généralement corrigées pendant la monotransformation des données.
- Création des clés de substitution.

3.2.3.5. Phase de chargement des données

Cette phase est similaire à celle du processus ETC classique qui permet de charger les données obtenues à l'issue de la phase de transformation vers l'ED. Une fois, une donnée est chargée dans l'ED, elle sera automatiquement effacée de la ZPD.

3.3. Processus de propagation des données corrigées

A la différence des travaux de recherche sur le rafraîchissement de l'ED qui s'intéressent uniquement à la propagation des données des SDO_i après leur mise à jour, dans ce travail nous proposons la propagation des données corrigées pendant leur entreposage, plus spécifiquement, pendant les phases de transformation et de monotransformation des données.

3.3.1. Motivation et apports de la propagation des données corrigées

Comme nous l'avons souligné dans cette thèse, l'amélioration de la qualité des données et, plus spécifiquement, le nettoyage des données peut introduire des erreurs. De ce fait, il nous semble intéressant de placer l'utilisateur au cœur du système d'amélioration de la qualité afin de vérifier et valider les données corrigées. Pour atteindre cet objectif, nous avons développé un processus de propagation des données corrigées (PDC). Cela est important car il porte beaucoup d'apports à notre système. Parmi ces apports, nous citons:

- L'amélioration de la qualité des sources des données opérationnelles,
- L'évitement de refaire les mêmes opérations de nettoyage de données pendant le rafraîchissement de l'ED et les futures entreposages et intégrations des données.

- La validation des données corrigées par les utilisateurs des SDO et l'administrateur de l'ED.
- L'évaluation, la vérification et l'amélioration de la qualité des règles. Si par exemple, nous constatons que les données corrigées à travers une règle R sont validées incorrectes par l'utilisateur, alors cela met en question la qualité de la règle. De ce fait, nous avons introduit les caractéristiques de mesure de la règle dans le métamodèle de la règle (comme nous l'avons présenté dans le chapitre précédent). Ces caractéristiques sont mises à jour à chaque application de la règle.

3.3.2. Description des différentes phases du processus de propagation des données corrigées

Comme le montre la figure 24, le processus de propagation des données corrigées comprend trois phases : envoi des l'ensemble corrigé E_c, évaluation et amélioration de la qualité des données corrigées et propagation des données validées.

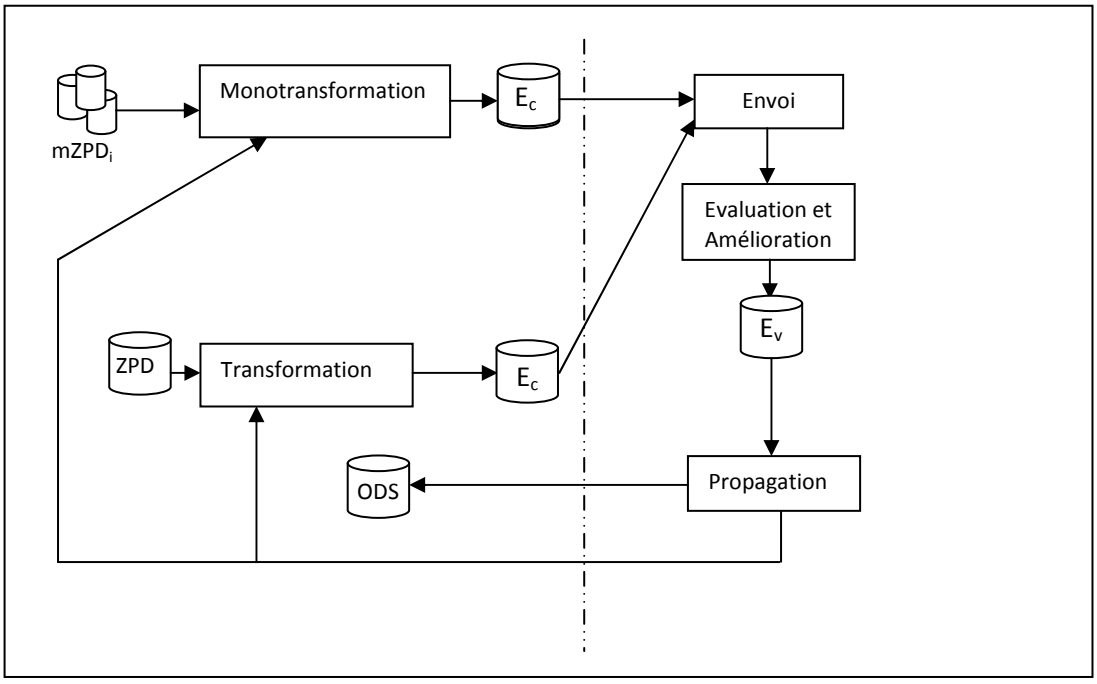

Figure 24 : Processus de propagation des données corrigées.

La figure 24 montre que le processus de propagation des données corrigées se déclenche pendant les phases de monotransformation et de transformation des données du processus ETCTC que nous avons détaillée précédemment. Comme les sources des données opérationnelles sont dispersées et indépendantes, il nous semble intéressant d'exploiter le parallélisme lors de la propagation des données corrigées. Cela peut améliorer le temps nécessaire à la propagation. La propagation se déclenche dés que les opérations de monotransformation et/ou des transformations des données dans une source mZPD$_i$ ou ZPD sont terminées.

3.3.2.1. Envoi de l'ensemble corrigé

Comme les SDO_i sont hétérogènes, l'envoi des données corrigées consiste pour chaque source des données à :

- La réécriture (Transformation) des données à stocker dans l'ensemble corrigé (E_c) selon le schéma de données des SDO_i correspondante. Les données corrigées doivent être accompagnées des valeurs initiales.
- La construction de l'ensemble à envoyer.
- Envoi de chaque ensemble corrigé vers le système opérationnel correspondant. L'envoi peut se faire à travers des messages électroniques, à travers un outil spécialisé ou les utilisateurs accèdent directement à ces données si l'entreprise possède des réseaux.

3.3.2.2. Evaluation et amélioration de la qualité des données corrigées

Cette phase est la clef de voûte du processus de propagation des données corrigées. Elle implique l'utilisateur dans l'évaluation et l'amélioration de la qualité des données corrigées car comme nous l'avons précisé dans le premier chapitre des erreurs peuvent être commises pendant l'amélioration de la qualité des données au cours des phases de monotransformation et transformation des données du processus ETCTC. Les opérations de cette phase sont faites via une interface utilisateur qui nous proposons à établir spécialement pour l'évaluation et l'amélioration de la qualité des données corrigées. Cette interface doit permettre à l'utilisateur de visualiser toutes les données corrigées avec leurs métadonnées (date d'extraction de la donnée, dernière date modification de la donnée au niveau de la source SDO). Ces métadonnées sont nécessaires car ces données souvent se changent.

Nous avons identifié trois cas que l'utilisateur peut fréquenter pendant cette phase. Ces cas dépendent des trois valeurs suivantes :

- V_i qu'est la valeur initiale d'une donnée avant son extraction des sources opérationnelles des données SDO_i,
- V_c qu'est la valeur de la donnée après sa correction par les règles du nettoyage des données. Dans ce cas, nous avons : $V_i \neq V_c$, et
- V_u qu'est la valeur de la donnée affectée par l'utilisateur s'il refuse les deux valeurs précédentes (les deux valeurs (V_i, V_c) sont incorrectes). Dans ce cas, nous avons: $V_u \neq V_i$ et $V_u \neq V_c$

Ainsi, les cas fréquentés par les l'utilisateur sont :

- Cas 1 : Si la valeur V_c est validée exacte par l'utilisateur, alors l'utilisateur met à jour V_i dans le SDO_i correspondante.
- Cas 2 : Si la valeur initiale V_i est validée exacte par l'utilisateur, alors la valeur de la donnée corrigée V_c doit être mise à jour au niveau ZPD, $mZPD_i$ ou ED si possible.

- Cas 3 : Si la valeur initiale V_i et la valeur de la donnée après la correction V_c sont validées fausses et l'utilisateur met une valeur V_u, alors les valeurs V_i et V_c doivent être mises à jour.

De ce fait nous avons proposé un algorithme générique pour la prise en charge de l'évaluation et l'amélioration de la qualité des données corrigées. Cet algorithme est détaillé dans la figure 25.

La correction de la valeur de la donnée au niveau des SDO_i ($1^{èr}$ et $3^{ème}$ cas) ne pose pas de problème car l'utilisateur peut faire la mise à jour de cette donnée directement dans les sources des données opérationnelles correspondantes. En revanche, si la valeur est au niveau des zones de préparation des données (ZPD, $mZPD_i$) ou ED, la mise à jour dépend de l'état d'avancement du processus d'entreposage des données.

Dans l'algorithme décrit dans la figure 25, nous avons introduit deux procédures pour la mise à jour des données (Mise_à_jour_utilisateur et Propose_valeur) et une procédure baptisée Évaluer_règle pour mettre à jour les caractéristiques de qualité des règles appliquées sur les données si la valeur de la donnée corrigée est refusée par l'utilisateur. La procédure Mise_à_jour_utilisateur permet à l'utilisateur de mettre à jour la valeur d'une donnée corrigée pendant les phases de monotransformation et transformation. La procédure Propose_valeur permet à l'utilisateur d'introduire la valeur de l'utilisateur V_u si les deux autres valeurs sont évaluées invalides ($3^{ème}$ cas). Dans le $2^{ème}$ et $3^{ème}$ cas, le système insère la donnée dans l'ensemble valide E_v.

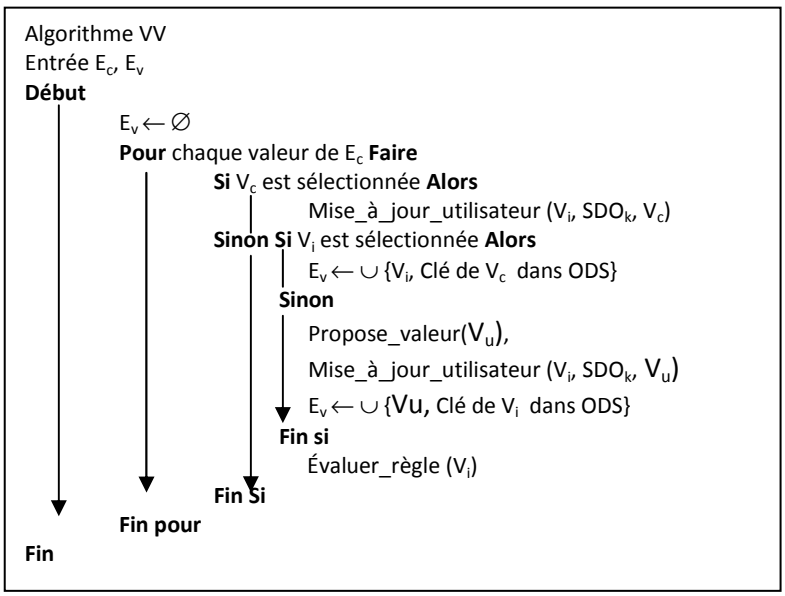

Figure 25 : Algorithme d'évaluation et amélioration de la qualité des données corrigées.

La procédure Évaluer_règle que nous avons conçue pour gérer la qualité des règles est mise en œuvre pour déterminer la cause de l'échec de la règle. L'échec peut être à cause du corps (SI Condition ALORS Conclusion) ou bien à cause de l'action. La figure 26 présente la

procédure algorithmique de l'évaluation de la qualité d'une règle. La procédure utilise l'ODS afin de chercher la donnée V_c en utilisant sa clé. Puis, elle récupère la clé de la règle appliquée sur la donnée à partir de TDC (V_c) où le système stocke les données corrigées avec leurs métadonnées associées telles que les règles appliquées sur la donnée. La TDC désigne la traçabilité des données corrigées que nous allons présenter dans la suite de ce chapitre. Si la valeur initiale V_i de donnée corrigée est validée fausse, alors le système incrémente la métrique Echec_Action par 1 car l'action est la l'origine de l'erreur. Par contre si la valeur initiale V_i est validée correcte, alors le système incrémente la métrique Echec_Règle par 1 car le corps de la règle est l'origine de l'erreur. Ces deux métriques que nous avons présentées dans le tableau 11 du deuxième chapitre seront utilisées par le système pour évaluer et améliorer la qualité des règles. Le chapitre 5 donne des explications sur l'exploitation de ces métriques de qualité.

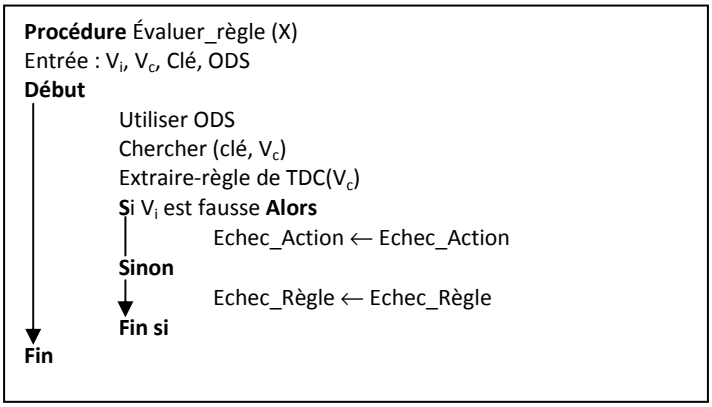

Figure 26 : Algorithme d'évaluation de la qualité d'une règle.

De point du vue de la complexité, nous avons étudié la complexité temporelle de l'algorithme d'évaluation et d'amélioration de la qualité des données corrigées par rapport à la cardinalité de l'ensemble corrigé E_c qu'est le nombre des données corrigées et sans tenir compte de la complexité temporelle des procédures et fonctions utilisées dans l'algorithme. Nous avons constaté que l'algorithme est de complexité linéaire O(n) dans le pire des cas et est (où n est la cardinalité de E_c) et de complexité constante O(1) dans le meilleur des cas. Pour le deuxième algorithme, il est de complexité constante O(1).

3.3.2.3. Propagation des données validées

La propagation de l'ensemble valide E_v vers le processus d'ED passe toujours par la ZPD et $mZPD_i$ et nécessite quelques opérations de transformations. Cette propagation ne peut être faite à tout moment car elle est coûteuse en termes de temps. De ce fait, nous distinguons quatre choix pour la propagation d'E_v : immédiate, périodique, reconstructive ou sur demande.

- Propagation immédiate: si les phases de monotransformation et transformation des données sont encore en cours d'exécution, le système est doté d'un processus de capture des données corrigées, que nous allons détailler dans la suite de chapitre, qui peut récupérer certaines données

améliorées et les mettre à la place des données corrigées au niveau des zones de préparations des données $mZPD_i$ et ZPD;

- Propagation périodique : La propagation est faite pendant le rafraîchissement de l'ED.
- Propagation reconstructive : On calcule le rapport de la capacité des données validées et la capacité des données intégrées dans l'ED. Si ce rapport est significatif, alors on reconstruit l'ED.
- Propagation sur demande : Dans ce cas, nous faisons uniquement l'intégration des données validées sur demande de l'administrateur de l'ED.

3.3.3. Capture des changements des données corrigées

Le système opérationnel et le système analytique qu'est basé sur les processus d'entreposage des données et d'extraction des connaissances des données- sont deux systèmes séparés. De ce fait, il nous demeure intéressant d'établir un outil permettant de capturer les données validées (s'elles existent) dans l'ensemble valide (E_v). Ces données validées remplacent les données corrigées correspondantes au niveau de mZPDi et ZPD avant leur chargement dans l'ED. Notons qu'une donnée validée est une donnée corrigée après leur nettoyage (par une règle) mais l'utilisateur l'évalue fausse et la remplace par une autre donnée.

La capture des changements des données corrigées se déroule d'une manière simple et facile. Il est applicable uniquement dans le cas où les sources des données opérationnelles appartiennent au même organisme. L'ensemble valide (E_v) est un ensemble vide au début de l'entreposage (cardinalité(E_v)=0). Si le processus CCDC constate que l'ensemble valide contient des données (cardinalité($E_v \neq 0$), alors il récupère les données pendant les phases de monotransformation et transformation des données. Le CCDC s'applique lorsque la propagation est immédiate ou on-demande. Le problème majeur que nous avons rencontré dans la conception de CCDC est que la propagation des données validées nécessite l'interruption du processus d'entreposage des données afin de permettre la mise à jour de ces données au niveau des zones de préparation des données et de refaire les transformations (règles) appliquées sur ces données au cours de l'entreposage sur ces données.

3.4. Processus de la traçabilité des données corrigées

La traçabilité des données corrigées est très importante car elle facilite leur intégration dans les zones de préparation des données au cours de l'entreposage des données. La figure 27 décrit l'algorithme que nous avons conçu pour permettre la traçabilité des données corrigées (y compris ses données dérivées).

3.4.1. Algorithme de la traçabilité des données corrigées

L'algorithme que nous proposons pour la traçabilité se repose sur quelques fonctions algorithmiques. Ces fonctions reflètent les principales opérations de l'algorithme que nous résumons dans les points suivant :

- Création de la structure des données de TDC
- Initialisation de la structure de TDC
- Traçabilité des données corrigées
- Traçabilité des données dérivées.

Fonction de la création de structure des données
Structure TDC
Début
Trace (d_j)=< id_{global} : code, id_{DSO} : code, v_i, v_c : valeurs,
List-R : Règles, X : trace >
// List_R : les règles appliquées sur d_j
Fin

Fonction d'initialisation des variables de la structure
Initialisation
Début
TDC ←∅
List-R←∅
Fin

Fonction de suivi de la traçabilité des données corrigées
Déclenche_Traçabalité
Début
TDC(d_j) ← Trace(d_j)
List-R←List-R∪ { Rj}
TDC← TDC∪ TDC(dj)
Fin

Fonction de suivi de la traçabilité des données dérivées
Trace_Donnés_Dérivées
Début
Si dj utilise di **Alors**
TDC (di) ← TDC (di) ∪ Trace (dj)
Fin si
Fin

Figure 27 : Algorithme de la traçabilité des données corrigées

Nous pouvons montrer que la complexité temporelle de l'algorithme de la traçabilité des données est de complexité constante O(1).

3.4.2. Création de la structure des données de TDC

Comme les données dérivées sont formées à partir soit des données corrigées, soit des données dérivées, nous avons choisi une structure dynamique des données pour la structure des données corrigées. Cette structure permet de manipuler de façon simple des données volumineuses comme notre cas des données corrigées dans les ED. Elle permet de stocker en premier lieu les données corrigées. Puis celle-ci pointe vers les données dérivées. Chaque donnée dérivée est aussi traitée comme étant un pointeur pour pointer vers ses données dérivées et ainsi de suite. La figure 28 montre la structure des données retenue pour les données corrigées dans l'algorithme TDC.

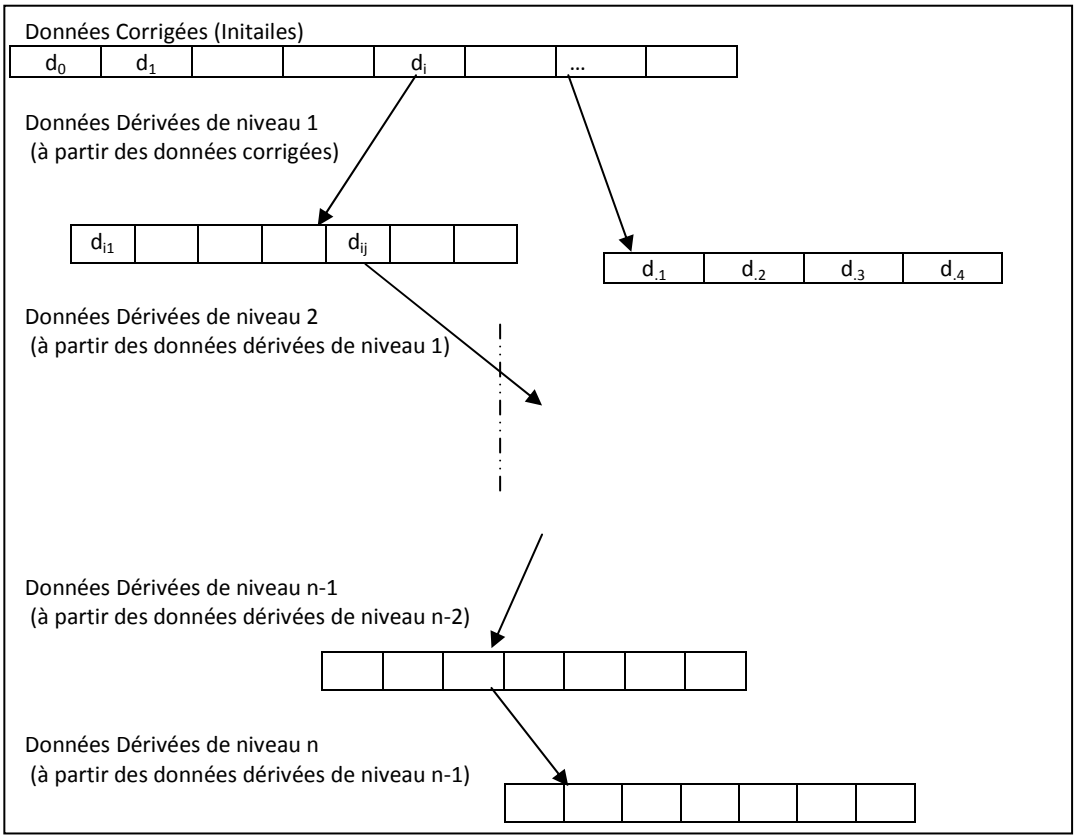

Figure 28 : Structure des données corrigées

Une structure de données TDC est un 6-uplets défini comme suit:

$$TDC =< Id_{globale}, id_{ods}, V_i, V_c, List\_R, TDC> \text{ où}$$

Cela veut dire que le pointeur est un enregistrement composé de six attributs : $Id_{globale}$ qu'est la clé de substitution donnée par le système, id_{ods} qu'est la clé de la donnée dans la base, la valeur initiale V_i, la valeur corrigée V_c, List_R qu'est la liste des règles appliquées sur la donnée, et une variable de type TDC.

3.4.3. Initialisation de la structure

L'initialisation sert à initialiser le 6-uplets de la structure des TDC. Cette initialisation de déclenche automatiquement au début de l'entreposage des données. Elle crée le pointeur de la structure des données corrigées et l'initialise à Nil.

3.4.4. Traçabilité des données corrigées

A chaque détection d'une valeur de mauvaise qualité, le système crée un enregistrement dans lequel seront stockées les valeurs de cinq attributs de l'uplets décrite ci-dessus après l'amélioration de la valeur de la donnée.

3.4.5. Traçabilité des données dérivées

Comme le montre la figure 28, il existe plusieurs niveaux des données dérivées. De ce fait, le système crée un pointeur des données corrigées qui lui-même contient une variable de type pointeur afin de contenir les informations sur les données dérivées et ainsi de suite. De cette manière, nous assurerons la gestion des données corrigées à différents niveaux.

4. Présentation du processus ECD adapté

Les contributions que nous avons présenté jusqu'à ce point sont faites pour permettre l'adaptation du processus d'extraction des connaissances à partir des données afin de permettre une meilleur prise en charge de la qualité des données et des connaissances plus spécifiquement les règles de nettoyage des données. Dans la figure 29, nous avons décrit le processus adapté d'extraction des connaissances à partir des données après son adaptation.

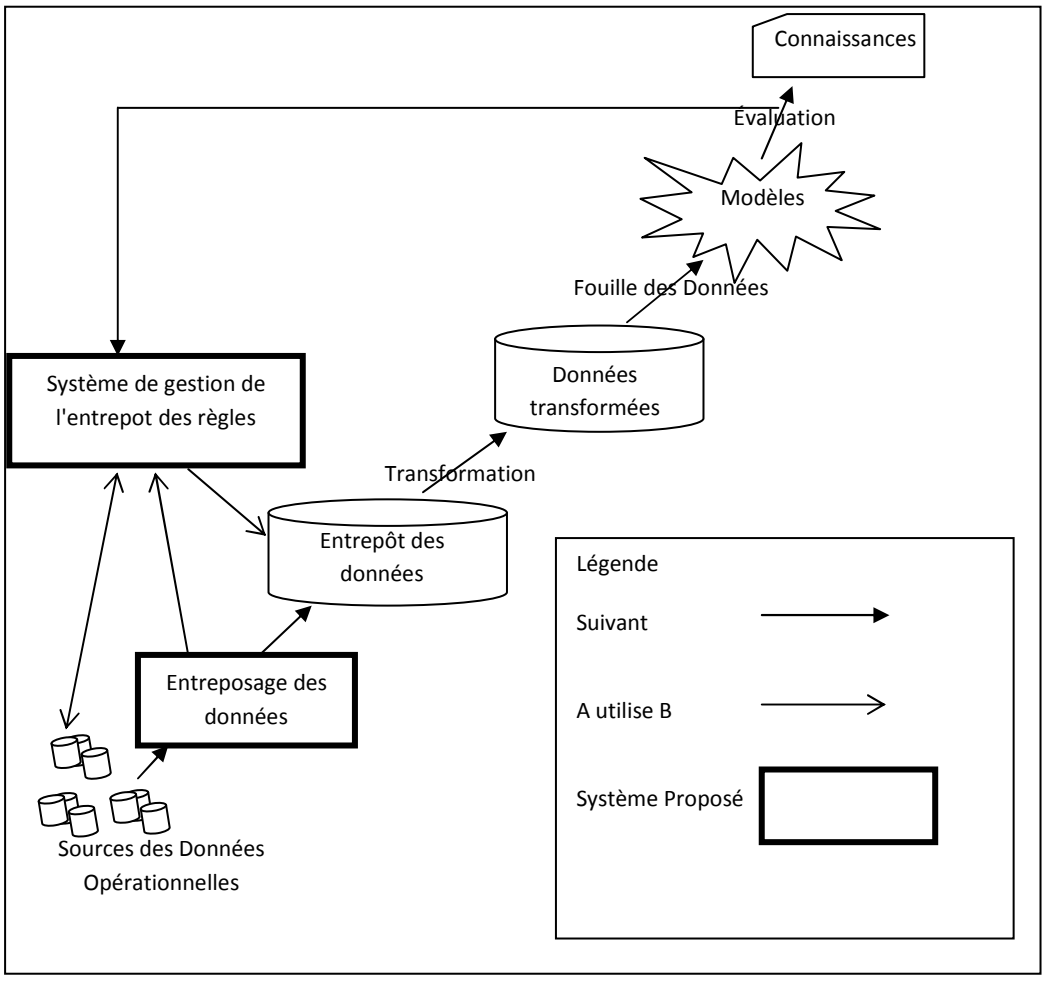

Figure 29 : Processus adapté d'ECD à base de règles.

Pour faire la comparaison entre les travaux de recherche actuels sur l'ECD et le processus que nous avons conçu, nous nous rapportons aux figures 12 et 14 d'ECD présentés dans le chapitre 2. L'avantage majeur du processus que nous avons adapté qu'est qu'il permet l'évaluation et l'amélioration de la qualité des données à bases de règles des sources des données opérationnelles séparément des processus d'ECD et d'ED. Ce qui permet de garantir une haute

performance en termes de temps d'entreposage des données et d'ECD. En outre, nous constatons que le système de gestion de l'entrepôt des règles peut être exploité pour la gestion de la qualité des sources des données opérationnelles indépendamment des processus d'ECD et d'ED.

Dans ce cas l'amélioration de la qualité des données opérationnelles s'effectue sur des copies des SDO_i afin de ne pas perturber le fonctionnement de leurs systèmes opérationnels. Puis le système propage les données corrigées vers leurs SDO_i en exploitant le processus de propagation des données corrigées. Finalement les responsables de ces systèmes décident sur les mesures à entreprendre (accepter ou refuser les corrections). Comme on peut effectuer l'amélioration de la qualité des données opérationnelles directement dans les sources SDO_i si les responsables l'autorisé et lorsque systèmes opérationnels sont offlines.

Le processus d'extraction des connaissances à partir des données que nous avons adapté à la qualité des données s'est réduit c'est-à-dire qu'il se déroule uniquement en quatre phases : Entreposage des données, Transformation, Fouille des données et Évaluation des modèles et une phase indépendante qu'est l'entreposage des règles.

4.1. Entreposage des données

Pour l'ECD, l'entreposage sert à sélectionner, préparer et charger les données dans l'ED selon le processus d'ED que nous avons proposé dans ce chapitre. Le processus d'entreposage des données fait aussi appel au système de gestion de l'entrepôt des règles (SGER) pour appliquer les différentes transformations, qui sont stockées sous forme des règles dans l'entrepôt des règles du SGER, sur les données.

Le système de gestion de l'entrepôt des règles peut utiliser l'ED pour extraire des règles agrégats pour les appliquer sur les données agrégées dans les futurs entreposages des données sans influencer le processus d'ECD. Dans la figure 29, nous avons utilisé une flèche bidirectionnelle entre l'ED et SDO_i car l'ED propage les données corrigées par le biais du processus de propagation des données corrigées (PDC) que nous avons conçu spécialement pour le processus d'entreposage des données.

4.2. Transformation des données

Nous avons maintenu cette phase dans le processus adapté d'ECD afin de permettre certaines transformations propres aux techniques de fouille de données et qui ne sont pas transformables en règles.

4.3. Évaluation des données

Après l'extraction des modèles par les techniques de fouille des données, le système d'entreposage des règles les récupère afin de les valider et transformer sous forme de règles selon le formalisme que nous avons présentée dans le chapitre 3. Généralement, nous nous intéressons aux modèles qui sont soit des modèles mathématiques, soit des règles.

5. Conclusion

Dans ce chapitre, nous avons présenté notre processus adapté d'extraction des connaissances à partir des données afin de permettre l'amélioration de la qualité à base des règles. Nous avons adapté certaines composantes de ce processus et nous avons ajouté d'autres composantes afin de garantir les performances en termes de la qualité des données et des connaissances, et de temps d'exécution du processus et de leur composante.

Nous avons démontré que la réalisation d'un processus ECD orienté qualité (pour une meilleure prise en charge de la qualité des données et des connaissances) nécessite son adaptation et l'ajout d'un processus de propagation des données corrigées vers leurs sources. Nous avons conçu un nouveau processus ETCTC de manière à être compatible avec le système de gestion de l'entrepôt des règles que nous avons proposé et détaillé dans le chapitre 3. Les phases que nous avons ajoutées au processus classique d'ETC ont permis d'exploiter le parallélisme des traitements et la répartition des données pendant leur entreposage. Cela permet un gain de performance en termes de temps. Le processus de propagation des données corrigées que nous avons ajouté au processus adapté ETCTC a permis d'impliquer les utilisateurs finaux des sources des données opérationnelles dans l'évaluation et l'amélioration de la qualité des données et des règles. Afin de permettre la propagation des données corrigées, nous avons conçu un processus de traçabilité et d'une méthode de capture des changements spécifiques aux données corrigées.

De point de vue de la complexité temporelle des algorithmes que nous avons proposés dans ce chapitre, nous avons démontré que la plupart des algorithmes sont de complexité constante O(1).

Dans le chapitre suivant, nous présentons une validation théorique et expérimentale à travers une étude de cas de certains composants de notre proposition.

Chapitre 5: Validation et performances

« le critère de la scientificité d'une théorie réside dans la possibilité de l'invalider, de la réfuter.»

Karl Popper

1. Introduction

Comme nous l'avons évoqué dans l'introduction générale, nos contributions sont plus conceptuelles que pratiques. Par conséquence, dans ce chapitre, nous proposons de présenter une validation théorique et expérimentale de quelques composants du système que nous avons proposé pour l'adaptation du processus d'extraction des connaissances à partir des données afin de permettre une meilleure prise en charge de la qualité des données et des connaissances. Cette validation à pour but de montrer les performances de notre système en termes de temps et de qualité.

De point de vue pratique, nous présentons l'analyse des résultats obtenus suite à l'expérimentation que nous avons réalisée dans un établissement sanitaire afin d'évaluer les impacts reliés à l'application de notre proposition. Cette expérimentation est basée sur un outil ETCTC -supportant tous les systèmes de gestion des bases des données relationnelles- que nous avons développé avec la langage de programmation JAVA pour permettre l'entreposage des données selon notre processus adapté d'ECD. Nous décrivons aussi notre méthodologie d'acquisition des connaissances spécifique à cet établissement. Ensuite nous détaillerons les algorithmes utilisés et développés lors de cette expérimentation.

De point de vue théorique, nous donnons une démonstration mathématique qui preuve que le temps d'entreposage des donnée à l'aide du processus que nous avons adapté est amélioré (réduit) par comparaison à celui du processus classique d'entreposage des données.

2. Expérimentation: étude de cas et implémentation

Dans cette section, nous allons valider quelques composants de notre proposition par des résultats expérimentaux.

2.1. Présentation de l'étude de cas

Nous avons réalisé une expérimentation au sein de l'établissement hospitalier de la santé publique de Tébessa. Cet établissement comporte plusieurs sous directions dont celle des services sanitaire est la plus importante. Cette sous direction comporte plusieurs services. Parmi ces services, nous citons : pharmacie, laboratoires d'analyses médicales et bureau des entrées qu'est responsable de la gestion des patients hospitalisés.

Nous avons conçu et développé trois mini entrepôts des données que nous avons appelé respectivement ED_1, ED_2 et ED_3. Ces entrepôts comportent des données nécessaires au calcul et analyse du :

- Coût d'hospitalisation des malades,

- Décès, et
- Occupation des lits

Ils sont formés à partir des sources des données opérationnelles des trois services décrits ci-dessus.

Pour réaliser ces entrepôts, nous avons réalisé un outil baptisé ETCTC_ED pour l'extraction, transformation et chargement des données selon notre système proposé dans le quatrième chapitre.

Au cours de cette expérimentation, nous avons réalisé deux tests : test 1 qu'est basé sur l'utilisation de l'ETL classique et test 2 qu'est basé sur l'utilisation de l'ETCTC_ED que nous avons proposé dans ce travail. L'objectif est de déterminer l'amélioration des performances de notre système ETCTC en termes de qualité et de temps par rapport à celles de l'ETC.

2.2. Démarche de l'expérimentation

La réalisation de tous les composants du système que nous avons proposé est coûteuse en termes de temps et moyens (un projet d'entreposage des données peut durer plus de 03 ans) à cause des raisons suivantes :

- La plupart des établissements ne s'intéresse pas à la qualité de leurs données malgré que les problèmes liés à la mauvaise gestion au sein de ces établissements et sans doute dues à la mauvaise qualité des données et des connaissances qui sont généralement incomplètes, erronées, redondantes, etc.
- Le manque des spécialistes en connaissances et en informatique

De ce fait, nous avons réalisé une expérimentation pour tester certains composants de notre système et, plus spécifiquement, l'ETCTC, l'élicitation et le nettoyage des données à base e règles que nous avons proposés dans cette thèse. Pour atteindre ces objectifs, nous avons suit les étapes suivantes:

- Etape 1: Elicitation des connaissances selon le processus que nous avons présenté dans le troisième chapitre.
- Etape 2: Identification des sources des données opérationnelles (tables et attributs) et leur modèle physique: Cela est nécessaires afin de corriger les problèmes liés aux schémas des données de ces sources.
- Etape 3: Injection et identification des données de mauvaise qualité dans les différentes sources des données opérationnelles: Cela nous permet de calculer la performance du système en termes de qualité des données.
- Etape 4: Extraction des tables nécessaires à la construction de l'entrepôt des données à partir de chaque source opérationnelle des données et leur stockage dans les mono-zones de préparation des données ($mZPD_i$).

- Etape 5: Application des règles multi sources et mono source de nettoyage des données sur les données de chaque mono zone de préparation des données ($mZPD_i$).
- Etape 6: Chargement des données des différentes mono zones de préparation des données ($mZPD_i$) dans la zone de préparation des données (ZPD).
- Etape 7: Application des règles agrégats de nettoyage des données sur les données de la zone de préparation des données (ZPD).
- Etape 8: Chargement des données de la zone de préparation des données (ZPD) dans l'entrepôt des données.
- Etape 9: Calcul et analyses des performances du système proposé en termes de temps et de qualité.

Notons bien que notre expérimentation est basée sur l'extraction des tables des données.

2.3. Sources des données opérationnelles

Come nous l'avons évoqué, les EDs sont formés à partir de trois sources des données opérationnelles qui sont: SDO_1: Laboratoire des analyses biologiques, SDO_2 : Pharmacie centrale et SDO_3 : Bureau des entrées.

Dans le tableau 15, nous présentons les caractéristiques informatiques des sources des données opérationnelles en termes de système de gestion de base de données (SGBD) utilisé, nombre d'ordinateurs exploités et le personnel informaticien existant. Cependant nous avons noté l'absence du personnel spécialisé dans l'ingénierie des connaissances. Tous les SGBD utilisés sont relationnels.

Source	SGBD	Nombre d'ordinateurs	Personnel Informaticien
SDO_1	Dbase IV	6	néant
SDO_2	Microsoft Access	3	Technicien Supérieur
SDO_3	Paradox	12	Ingénieur d'application et 5 techniciens

Tableau 15 : Caractéristique informatique des sources des données opérationnelles de l'établissement sanitaire.

Dans notre expérimentation, nous avons exploité un seul ordinateur pour réaliser le test 1 basé sur l'ETL classique et trois ordinateurs (un par chaque source) pour réaliser le test 2 basé sur l'outil que nous avons développé ETCTC_ED.

La figure 30 décrit le modèle conceptuel de la base des données au niveau de la pharmacie.

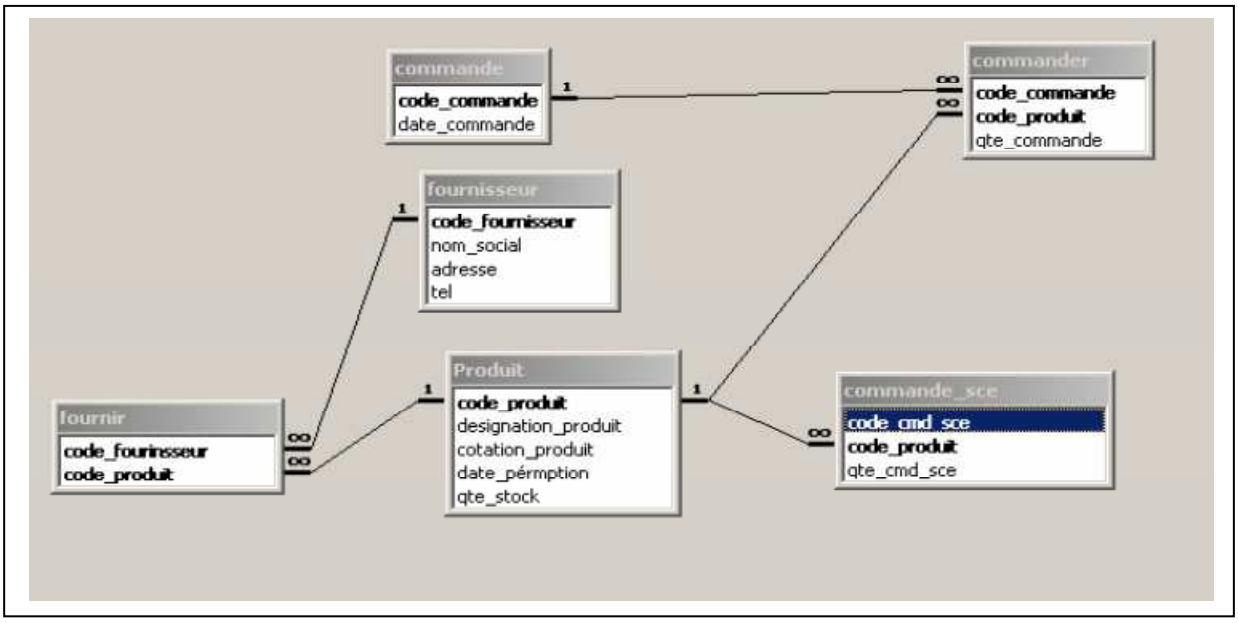

Figure 30 : Modèle conceptuel de la base de données de la pharmacie.

La figure 31 décrit le modèle conceptuel de la base des données du bureau des entrées.

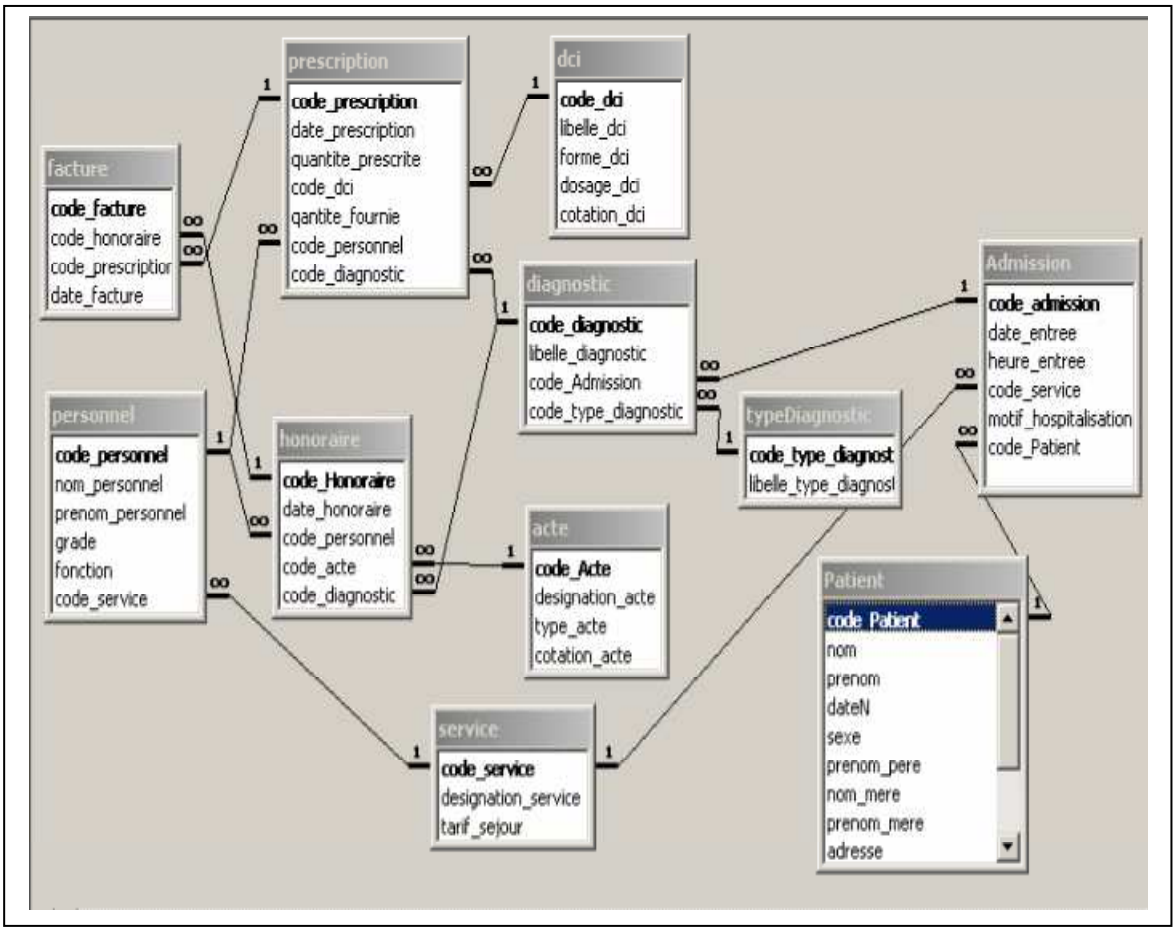

Figure 31 : Modèle conceptuel de la base des données du bureau des entrées.

Dans la figure 32, nous présentons le modèle physique de la base des données du laboratoire d'analyse biologique après la rectification de celui existant qui ne répond pas à nos besoins d'expérimentation.

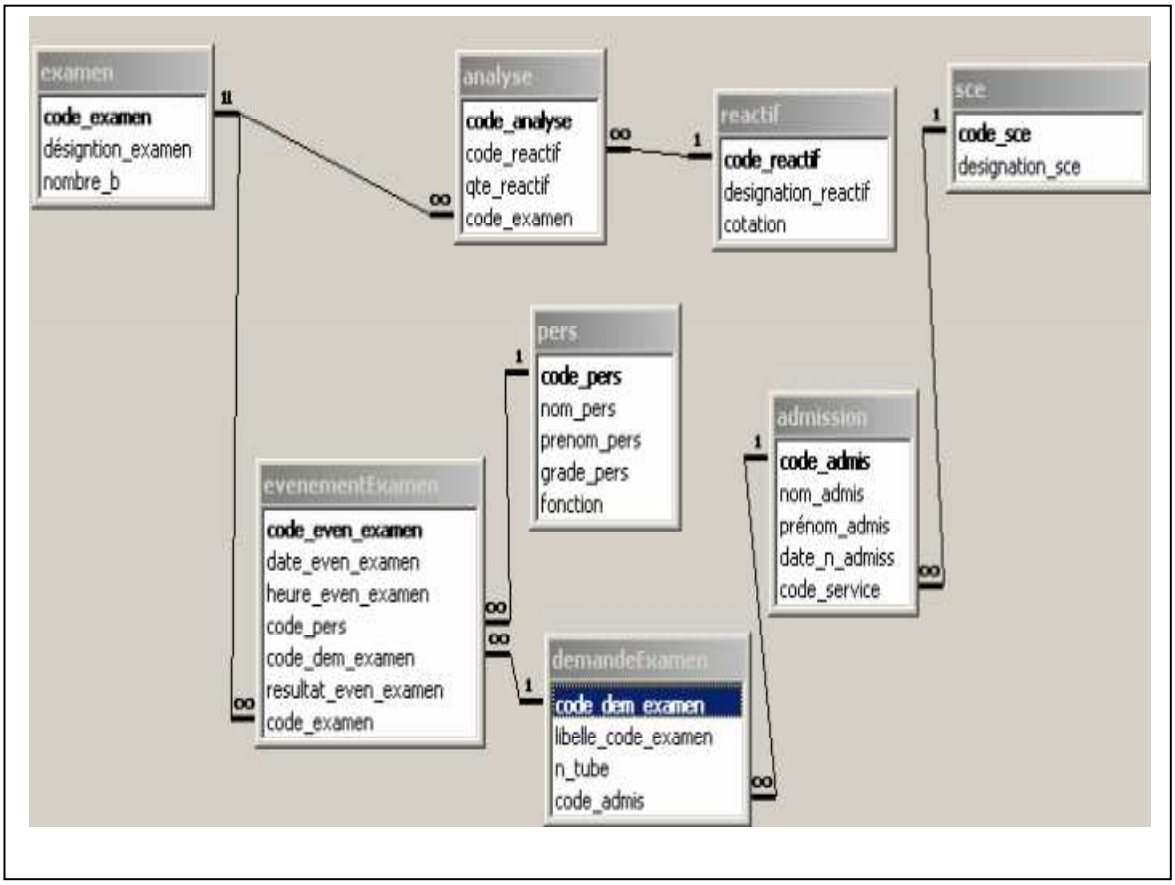

Figure 32 : Modèle conceptuel de la base des données du laboratoire.

La figure 33 décrit la métabase des données selon la notation UML. La métabase consisté à stocker les métadonnées sur les attributs, tables et bases de données.

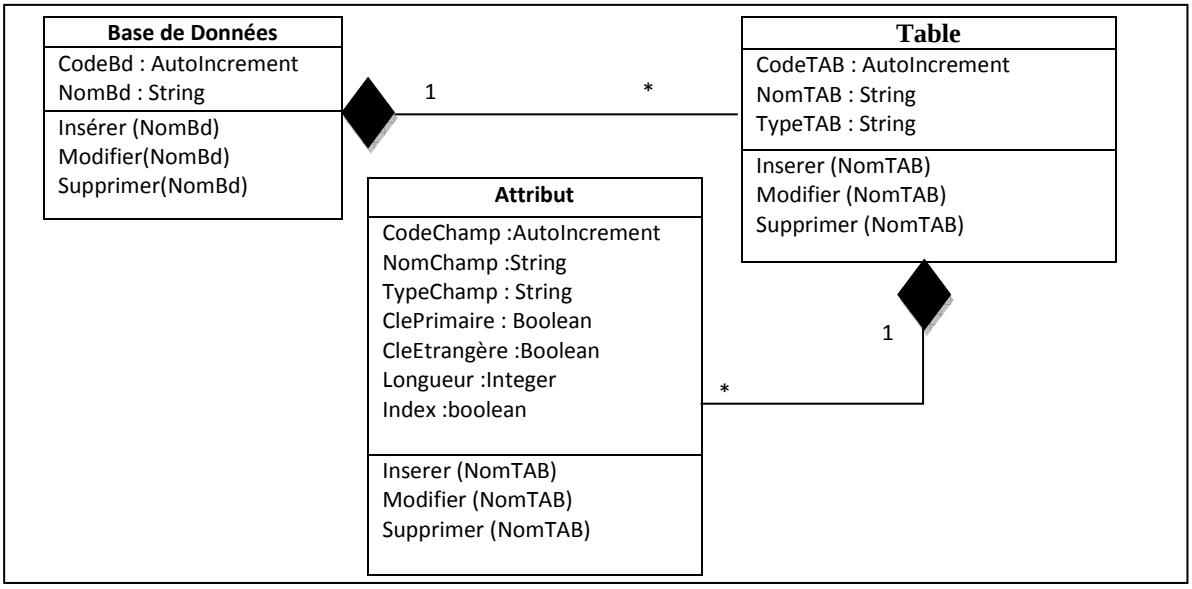

Figure 33 : Diagramme de classe UML de métabase.

La figure 34 décrit la représentation des tables sous MySQL des bases de données, des tables et des champs décrits dans la figure 33 qui contiennent leur (table) métadonnées.

	Champ	Type	Attributs	Null	Défaut	Extra	Action
☐	codeBD	int(11)	UNSIGNED	Non		auto_increment	
☐	nomBD	varchar(30)		Non			

La table *base de données* de la figure 33 sous MySQL

	Champ	Type	Attributs	Null	Défaut	Extra	Action
☐	codetab	int(11)	UNSIGNED	Non		auto_increment	
☐	nomtab	varchar(30)		Non			
☐	codebd	int(11)		Non	0		
☐	typetab	varchar(30)		Non			

La table *Table* de la figure 33 sous MySQL

	Champ	Type	Attributs	Null	Défaut	Extra	Action
☐	codeChamp	int(11)		Non	0		
☐	nomChamp	varchar(30)		Non			
☐	TypeChamp	varchar(30)		Non			
☐	longueurChamp	int(11)		Non	0		
☐	indexChamp	int(11)		Non	0		
☐	clePrimaire	int(11)		Non	0		
☐	cleEtrangere	int(11)		Non	0		
☐	codeTable	int(11)		Non	0		

La table *Champ* de la figure 33 sous MySQL

Figure 34 : Implémentation des bases et tables des données et leurs champs sous MySQL.

Après l'injection des données de mauvaise qualité dans les tables des différentes sources des données opérationnelles, nous avons mesuré le taux de ces données par rapport à la totalité des données. Ce taux était de 12%. Il est possible qu'il existe d'autres données de mauvaise qualité mais dans ce travail nous nous concentré sur cet ensemble de 12% afin de mesurer les performances de notre système. Ces problèmes de mauvaise qualité sont : des données manquantes, incomplètes, invalides et incorrectes.

2.4. Elicitation des connaissances

C'est l'étape la plus difficile et la plus coûteuse en termes de temps et de personnel. Elle est réalisée en collaboration avec le personnel des différents services de l'établissement. Afin de permettre la collection des informations utiles à la construction des connaissances, nous avons réalisé le questionnaire décrit dans la figure 35.

Le formulaire comporte cinq parties. La première partie pour l'identification de source des informations. Ces informations concernent l'employé à questionner et ses responsables. La deuxième partie contient le mot clé principal et les mots clés secondaires à propos des connaissances. Nous avons ajouté une rubrique aide pour donner des explications aux employés pour faciliter leur tâche. Cette partie est remplie par le responsable de la collection des informations. Les troisième et quatrième parties concernent l'employé afin de répondre à deux questions qui portent sur la description de mot clé principal et les mots clés secondaires et leur dépendance. La dernière partie est réservée au responsable de collection des connaissances afin d'extraire les connaissances, les règles et les facteurs de la qualité.

Après la collection des informations, nous avons passé à l'amélioration de leur qualité en suivant les opérations décrites dans la phase de transformation du processus ETC des règles proposé dans le troisième chapitre. Nous avons transformé certaines connaissances sous forme de règles selon notre formalisme. Dans ces règles, nous avons présenté les littéraux et la conclusion sous la forme suivante:

$$(expression(at_i, 1 < i < n))opérateur\ (valeur)\ \text{où}, at_i\ est\ un\ attribut\ quelconque$$

Une action dans cette expérimentation peut être l'une des formes suivantes :

- Suppression des données de mauvaise qualité,
- Ignore des données de mauvaise qualité, et
- Suite d'instructions où chaque instruction est écrite sous la forme suivante:

  $$attribut = (expression(at_i, 1 < i < n))\ \text{où}\ at_i\ est\ un\ attribut\ quelconque$$

  La règle suivante est écrite selon la forme décrite ci-dessus:

  SI service="pédiatrie" **ALORS** âge <=14 ans **DANS** (<bureau entrée>,∅, <âge=date_actuelle-date_naissance, supprimer (malade)>)

  Cela veut dire que dans le service pédiatrie, si nous trouvons des malades dont l'âge est supérieur à 14 ans, alors nous recalculons l'âge à partir de la date actuelle et la date de naissance du malade. Si on constate que cette date est toujours supérieur à 14 ans alors

nous supprimons la malade de la base. Cela veut dire que cet hospitalier n'est qu'un garde malade.

Nous avons choisi ces formes pour les règles lors de cette expérimentation pour les composants de la règle : condition, conclusion et action afin de faciliter leur implémentation avec des requêtes SQL simples: Replace or Delete.

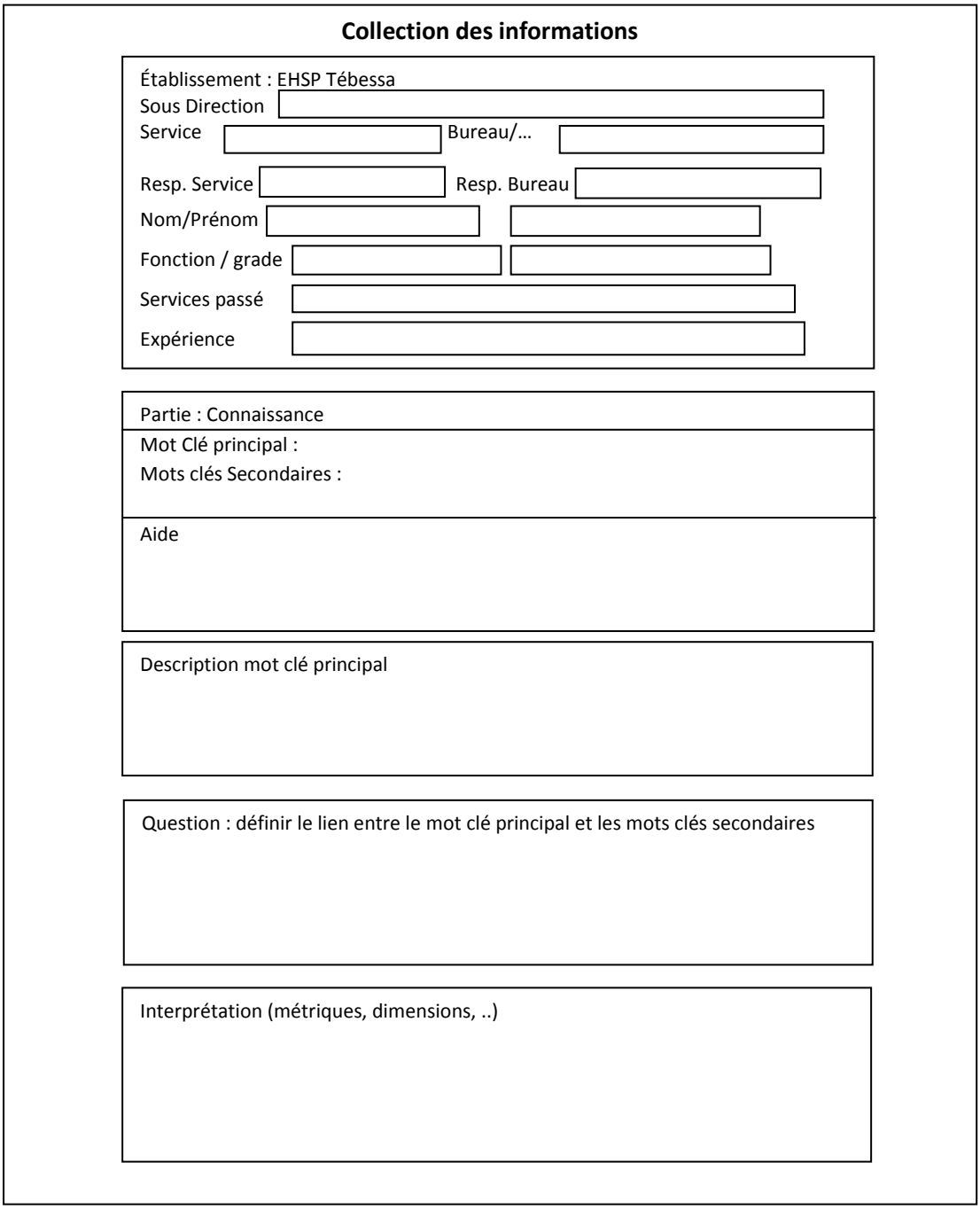

Figure 35 : Formulaire-Questionnaire de la collection des informations.

2.5. Fonctionnalités de l'outil ETCTC_ED

L'ETCTC_ED est un outil que nous avons développé avec le langage de programmation JAVA pour permettre l'extraction, la transformation et le chargement des données selon le processus que nous avons détaillé dans les chapitres 3 et 4. Les entrepôts des données sont des systèmes de gestion bases de données relationnelles MySQL. Cet outil comporte trois composants de base qui sont: création des connexions, création des bases des données, et ETC des données. Nous détaillerons ces composants dans la suite de cette section.

2.5.1. Création des connexions

La figure 36 montre l'interface graphique de l'outil ETCTC_ED que nous avons développé pour permettre à l'utilisateur de se connecter aux sources des données opérationnelles ($SDO_{i)}$ via divers connecteurs : ODBC, JDBC, SQL natif, Fichiers plats ou encore avec des connecteurs spéciaux.

Figure 36: Interface d'ajout des connexions

Après l'indication de source des données opérationnelles à travers l'interface de connexions présentées dans la figure 36, le système ouvre une fenêtre qui demande à l'utilisateur d'indiquer les paramètres nécessaires à mettre en œuvre la connexion à la source des données opérationnelles. Dans cette fenêtre L'ETCTC_ED affiche des messages lorsqu'un événement ou une erreur se produit. La figure 37 présente cette fenêtre.

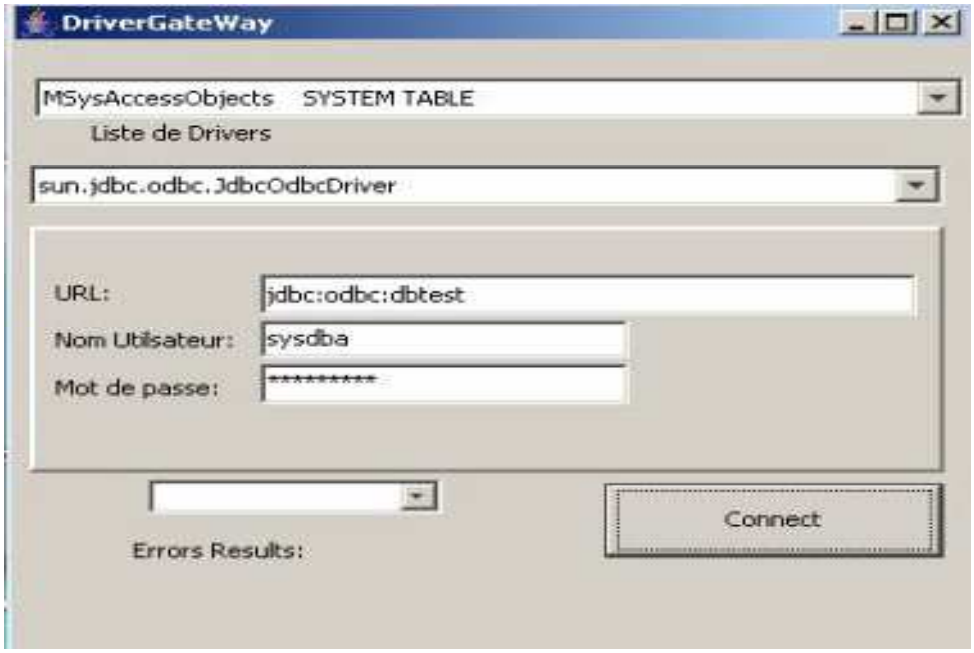

Figure 37: Interface DriverGateWay

2.5.2. Création des bases des données

Nous avons doté l'outil ETCTC_ED par des modules génériques permettant de créer les mono zones et zone de préparation des données, et les EDs des données. Nous avons développé trois interfaces graphiques pour la mise en œuvre de la création des bases des données.

La figure 38 présente l'interface de création de la base de données qu'est dans notre cas les bases de préparation des données et l'entrepôt des données.

Figure 38: Interface de création des bases des données

La figure 39 présente l'interface graphique que nous avons développée pour permettre la manipulation des tables dans les bases des données. Il permet de donner le nom de la table et la réalisation de certaines opérations sur les champs telles que : l'ajout, la modification et la suppression des champs.

Figure 39: Interface de création des tables

L'interface graphique présentée dans la figure 40 illustre l'opération de la création des champs d'une table lancée à partir de l'interface de création des tables.

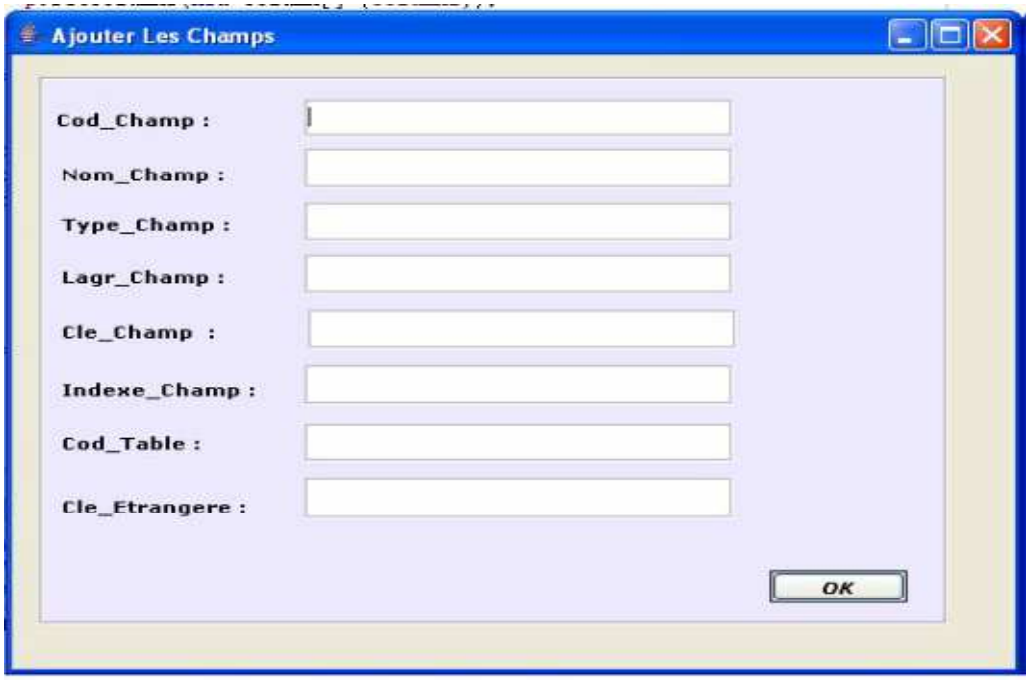

Figure 40: Interface de création des champs

2.5.3. Extraction, transformation et chargement des données

Dans cette expérimentation que nous avons réalisée afin de valider certaines composants de notre système, nous avons exécuté les opérations de:

- Extraction des données à partir des sources opérationnelles des données,
- Transformation des données plus spécifiquement le nettoyage des données, et
- Chargement des données dans les bases des données ($mZPD_i$, ZPD et ED)

Par le biais des requêtes SQL.

2.6. Algorithmes développés lors de l'expérimentation

Dans cette section, nous présentons les algorithmes développés pour l'implémentation de certains composants de notre proposition.

2.6.1. Algorithmes de gestion de la qualité des règles.

Nous décrivons les principaux algorithmes que nous avons développés pour la gestion de la qualité des règles. Nous présentons premièrement la démarche que nous avons suivi pour la collection des dimensions et métriques de la qualité lors de cette expérimentation.

2.6.1.1. Démarche de collection des dimensions et des métriques de qualité

Nous avons établi une démarche inspirée de la méthodologie TDQM décrite dans le chapitre 1 pour la collection des dimensions et métriques de qualité. La figure 41 présente le déroulement de la démarche avec la dimension exactitude. Cette démarche comporte quatre étapes (identique à TDQM).L'étape de définition qui permet de décrire la dimension. L'étape de mesure où nous devons déterminer les métriques nécessaires à la mesure de la qualité dimension règle. L'étape d'analyse qui calcule la valeur de la dimension qualité en combinant ses métriques. Et finalement l'étape d'amélioration où nous devons indiquer les actions à entreprendre sur la règle si la dimension est jugée de mauvaise qualité.

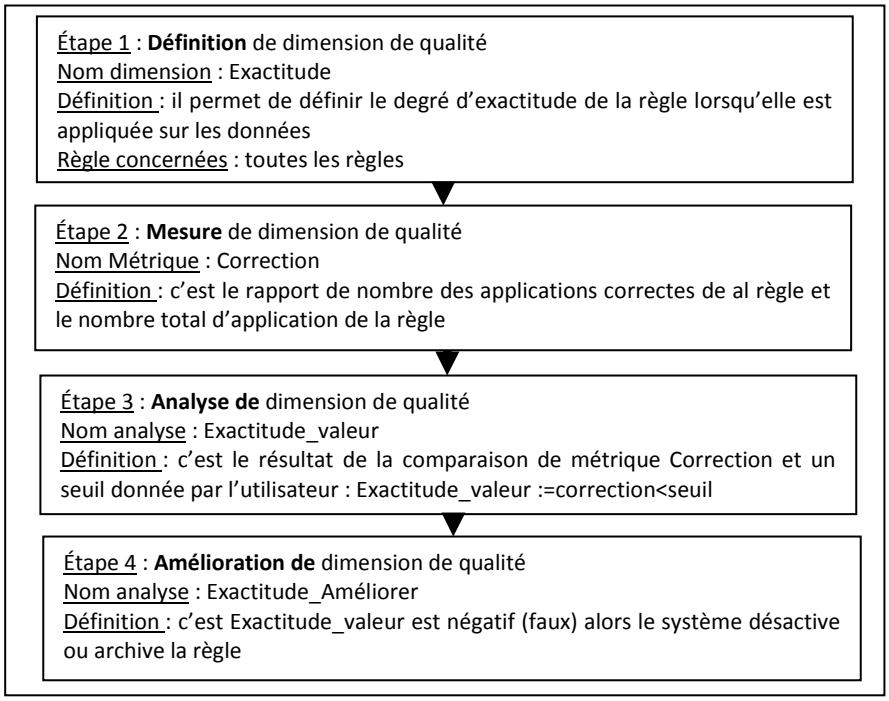

Figure 41 : Gestion de la dimension de qualité : Exactitude

2.6.1.2. Algorithme d'initialisation des valeurs des métriques de qualité des règles

L'algorithme val_initial que nous présentons dans la figure 42 est développé pour permettre l'initialisation des valeurs des métriques de qualité des règles avant leur chargement dans l'entrepôt des règles.

Algorithme val_initial
R : Règle, S:structure
Début
Code_Règle←clé(R) // clé fonction affecte une clé primaire à la règle R
Sources_Règle ← S // affectation de la source (SDO_i, ZPD)
Si S(R)+C(R)≠0 **Alors**
$$\text{Effective\_Règle}(R,S) \leftarrow \frac{S(R)}{C(R)} * \left| \cup_{i=0}^{i=n} T_i \right|$$
$$\text{Effective\_Règle\_Dernier}(R,S) \leftarrow \frac{S(R)}{C(R)} * \left| \cup_{i=0}^{i=n} T_i \right|$$
Sinon
$$\text{Effective\_Règle}(R,S) \leftarrow 0$$
$$\text{Effective\_Règle\_Dernier} \leftarrow 0$$
Fin SI
$$\text{Echec\_Règle} \leftarrow 0$$
$$\text{Date\_Création}(R,S) \leftarrow \text{date}()$$ // Date système
$$\text{Statut\_Rule}(R,S) \leftarrow \text{proposée}$$
Fin

Figure 42 : Algorithme d'initialisation des valeurs d'une Règle

L'algorithme Val_initial commence par calculer la clé primaire de la règle. Puis il affecte les sources de données où la règle sera applicable dans la variable Sources_Règle. Comme certaines règles comportent des valeurs des métriques concernant le support d'une règle (S(R)) et de confiance d'une règle(C(R)), le système calcule les valeurs des attributs suivants : Effective_Règle, et Effective_Règle_Dernier -que nous avons détaillé dans le chapitre 2- à partir de ces deux métriques. Pour le reste des règles, le système affecte la valeur nulle à tous les attributs de qualité de la règle.

Nous démontrons ci-dessous comment nous avons évalué dans cette expérimentation les valeurs des deux métriques: Effective_Règle, et Effective_Règle_Dernier - à partir de support et confiance :

Nous avons les hypothèses suivantes (R, S, S(R) et C(R) dénotent respectivement les concepts suivants : règle, structure, support de R et confiance de R) :

S(R) = n_{ab}/n (1) Il s'agit de la proportion d'individus qui vérifient la règle (n_{ab}) dans le jeu de données (n)

C (R) = n_{ab}/n_a (2) Il s'agit de la proportion d'individus qui vérifient la conclusion (n_{ab}) parmi ceux qui vérifient la prémisse (n_a)

Pour plus d'informations sur le support et la confiance d'une règle, nous y rapportons au chapitre 1. Les symboles a et b dénotent la condition et la conclusion d'une règle.

Dans notre cas, il sert à calculer les attributs de qualité : Effective_Règle, Effective_Règle_Dernier à partir des ces valeurs. Cependant le jeu des données est l'ensemble des tables (Ti) où la règle est applicable dans une SDO_i donnée. Cela veut dire que:

La taille du jeu des données = cardinalité des Ti= $|\bigcup_{i=0}^{i=n} T_i|$=X

De (1) et (2), nous déduisons que le jeu des données où la règle est applicable est n_a qu'est une portion de n. nous appliquons la règle de trois pour calculer Effective_Règle_Dernier (R, S) qu'est une portion de X, nous obtenons:

$\text{Effective\_Règle}(R,S) = \frac{X}{n}(n_{ab} + n_{a,nonb})$ (3) où $n_{a,nonb}$ dénote le nombre d'enregistrements qui vérifient la condition et ne vérifient pas la conclusion.

On a aussi : $n_a = n_{ab} + n_{a,nonb}$ donc : $n_{a,nonb} = n_a - n_{ab}$ (4)

De (1) et (2), nous obtenons : $n_a = \frac{S(R)}{C(R)} * n$ (5)

De (1), (4) et (5), nous obtenons $n_{a,nonb} = n * S(R)(\frac{1}{C(R)} - 1)$ (6)

Nous remplaçons dans (3), nous obtenons : $\text{Effective}_{\text{Règle}(R,S)} = \frac{X}{n} * n_a = \frac{S(R)}{C(R)} * X$

Cependant, notons bien que pendant le chargement nous avons :

$\text{Effective\_Règle}(R,S)$ = Effective_Règle_Dernier (R, S)

Il est clair que de l'algorithme est de complexité temporelle constante O(1).

2.6.1.3. Algorithme de calcul des métriques par échantillonnage

Comme le système d'entreposage est séparé de ceux d'entreposage des données et d'extraction des connaissances, nous avons développé au début de l'expérimentation un algorithme pour tester les règles sur un échantillon de données. La figure 43 décrit cet algorithme intitulé test_Règle.

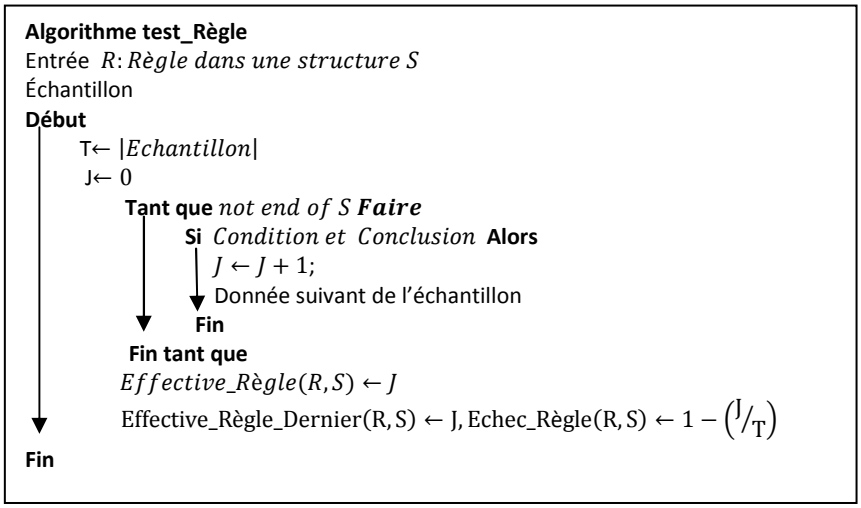

Figure 43 : Algorithme de calcul des métriques par échantillonnage

Test_Règle permet aussi de déterminer les valeurs des attributs : Effective_Règle, Effective_Règle_Dernier et Echec_Règle. Cependant l'algorithme est de complexité temporelle linéaire O(cardinalité (jeu de données où la règle est applicable)).

2.6.1.4. Algorithme de statut de règle

Comme le montre la figure 44, la l'algorithme Statut_R permet de calculer le statut de la règle.

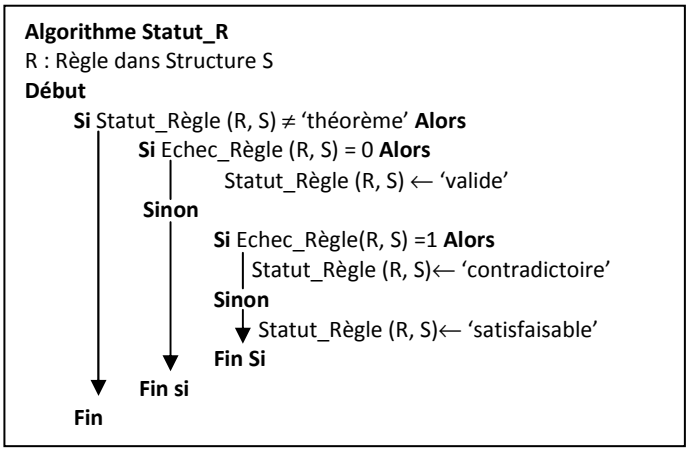

Figure 44 : Algorithme de calcul du statut d'une règle

Cet algorithme s'applique pour vérifier la qualité de l'entrepôt des règles avant son utilisation par les processus ED et ECD. L'utilisateur peut aussi changer le statut de la règle en la mettre en archive via l'interface utilisateur règle. L'algorithme est de complexité temporelle constante O(1).

2.7. Evaluation de l'expérimentation

Afin d'illustrer la pertinence de notre proposition dans cette première expérimentation, nous comparons les résultats obtenus avec notre outil ETCTC_ED avec ceux obtenus en utilisant l'approche ETC. Nous avons formé les trois mini entrepôts des données (ED_1, ED_2 et ED_3) décrits au début de ce chapitre en utilisant les processus ETL et ETCTC avec trois ordinateurs. Comme l'objectif de cette expérimentation est la démonstration de l'amélioration des performances en termes de qualité des données et de temps, nous avons mesuré le taux des données de mauvaise qualité dans les entrepôts des données et les sources des données opérationnelles et temps de transformations des données. Notons bien que cette expérimentation est basée sur la mesure des taux des données améliorée ou des données restant de mauvaise qualité dans l'ensemble des données de mauvaise qualité que nous avons injecté dans les sources des données opérationnelles.

Les figures 45 et 46 présentent les résultats obtenus au cours de cette expérimentation. La figure 45 montre que la performance en termes de la qualité des données par contre la figure 46 montre la performance en terme de temps d'exécution des transformations.

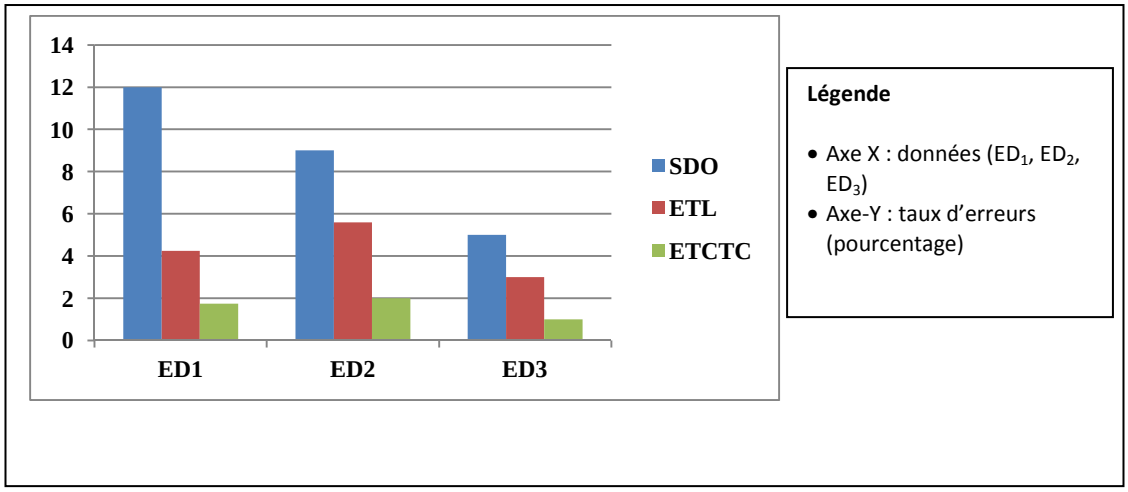

Figure 45 : Histogramme de la qualité des entrepôts des données et SDO.

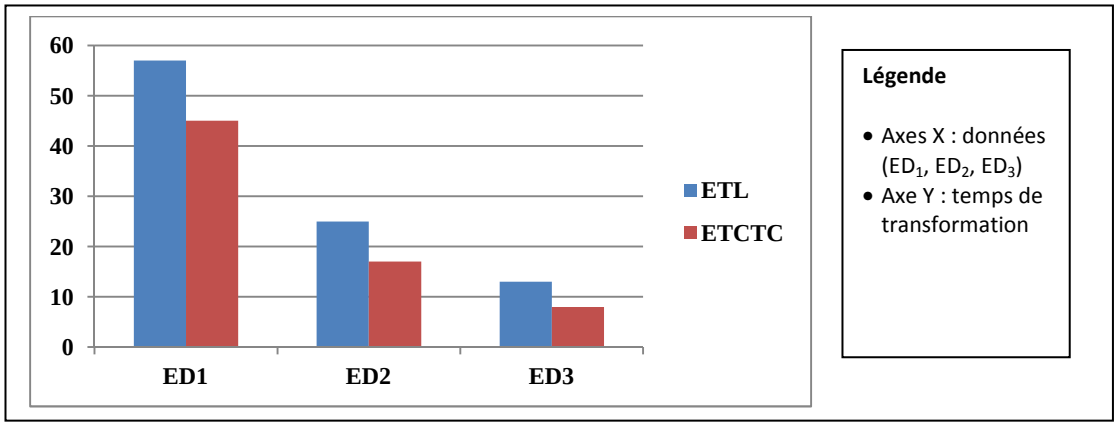

Figure 46 : Histogramme du temps estimé de transformations des données.

La figure 45 montre que:

- ✓ Le taux des données de mauvaise qualité est réduit dans les trois entrepôts des données et les SDO_i.
- ✓ Le taux des données de mauvaise qualité des SDO_i pour le système ETC est toujours de 12 %. Par contre, il est de 5 % pour le système ETCTC. Cela est le résultat de l'application du principe de l'amélioration de la qualité des données dans les sources originales et cibles par notre système à l'aide du processus de propagation des données corrigées vers les sources des données opérationnelles. Ce processus a permis l'amélioration de la qualité des SDO_i (le taux d'erreurs est diminue de 12 à 9 %) pendant la formation de ED_1, et par la suite, nous a permis d'éviter les mêmes opérations de nettoyage des données faites pendant la construction de ED_2 et ED_3.

La figure 46 montre un gain de performance considérable en termes de temps de transformation avec notre système par rapport à l'ETC. La différence entre les temps des deux systèmes est entre 20 et 38% du temps de l'ETC. Ceci est dû principalement à l'application du parallélisme pendant les deux phases de transformation et monotransformation des données que nous avons ajoutées au processus ETC classique et la mise en œuvre du principe de l'amélioration de la qualité des données des SDOi.

De ce fait, les résultats de cette expérimentation montrent que les meilleures performances ont été obtenues par le système ETCTC que nous avons proposé.

Notons bien que nous avons constaté des changements des données après la comparaison des tables avant leur nettoyage et après leur nettoyage.

3. Validation théorique

Dans cette section, nous montrons l'amélioration du temps d'entreposage des données du système ETCTC par rapport à celui d'ETC d'une manière théorique. Le facteur du temps est un facteur très important et considéré comme un critère de qualité pour l'évaluation des processus d'extraction des connaissances à partir des données et d'entreposage des données.

Cette validation théorique consiste à estimer le temps nécessaire à l'entreposage via l'ETCTC et ETC. Puis, nous comparons la performance du chaque système. Cette validation s'intéresse uniquement au temps nécessaires aux transformations opérables sur les données car les temps nécessaires pour l'extraction et chargement sont supposés les mêmes pour les deux processus ETCTC et ETC.

Le calcul du temps estimé est basé sur les paramètres suivants :

- Le nombre de transformations (règles) dénoté n
- Le temps estimé pour réaliser chaque transformation par les règles mono source et multi sources dénoté T_i
- Le nombre d'ordinateurs affectés à chaque $mZPD_i$ dénoté m_j
- Le nombre d'ordinateurs affectés au ZPD dénoté m
- Le temps nécessaire pour réaliser les transformations par les règles agrégats dans la ZPD avec un seul ordinateur dénoté T
- Le nombre de sources des données opérationnelles $mZPD_i$ dénoté k

Ces notations nous permettent d formaliser les hypothèses suivantes :

- Le temps moyen nécessaire à chaque transformation : $T_{moy} = \sum_{i=1}^{i=n} T_i / n$
- Le nombre total des ordinateurs est : $m_t = m + \sum_{j=1}^{j=k} m_j$
- Le temps nécessaire à la phase de transformation est : T/m

a) Le Temps estimé de transformation du processus ETC

Nous calculons dans cette partie, le temps nécessaires à la réalisation de la phase de transformation pour le processus ETC dénoté $\mathbf{T_{ETC}}$. Nous avons déjà souligné dans le chapitre 2 que la plupart des outils d'ETC existants n'applique pas le parallélisme pendant la phase de transformation. De ce fait, nous avons:

$$T_{etl} = T + (n * T_{moy})$$

Cela veut dire que le temps T_{ETC} est la somme de temps nécessaire à la transformation dans la ZPD. Notons bien que l'ETC extrait les données à partir des sources des données opérationnelles et les stocke dans la zone de préparation des données ZPD.

b) Le temps estimé de transformation du processus ETCTC

Nous calculons dans cette partie, le temps nécessaires à la réalisation des phases de transformation et monotransformation du processus ETCTC que nous avons proposé. Ce temps est dénoté $\mathbf{T_{ETCTC}}$.

Comme dans les zones ZPD et mZPDi, les données sont partagées en trois ensembles de données (valide : E_v, corrigé : E_c, et modélisé : E_c) et Pour chaque $mZPD_i$ on a m_j processeurs ($1<=m_j<=3$) donc :

Le temps estimé pour un $mZPD_i$ est : $\frac{1}{n}\sum_{i=1}^{i=n} \mathrm{Ti}/_{\mathrm{mj}}$

Par conséquent, nous déduisons le temps moyen nécessaire pour une transformation est :

$$\sum_{i=1}^{i=n} T_i \Big/ \left(\frac{\sum_{j=1}^{j=k} m_j}{k}\right)$$

Cette quantité $\left(\frac{\sum_{j=1}^{j=k} mj}{k}\right)$ représente le nombre moyen des ordinateurs par $mZPD_i$.

De ce fait, le temps total estimé pour la phase monoTransformation est :

$$\underline{\mathrm{p} + (\mathrm{n\ div}\ (m_t\text{-}\mathrm{m})\ )} \left(\sum_{i=1}^{i=n} T_i \Big/ \left(\frac{\sum_{j=1}^{j=k} mj}{k}\right)\right)$$

avec $\mathrm{p} + \left(\mathrm{k\ div}\ (m_t\text{-}\mathrm{m})\ \right)$ représente le nombre des sessions séquentielles de transformations à réaliser où p est défini comme suit :

$$p = \begin{cases} 0\ si\ (n\ mod\ (m_t - m)\ ) = 0 \\ 1\ si\ \left(n\ mod\ (m_t - m)\right) \neq 1 \end{cases}$$

Finalement, le temps total estimé de transformation et monotransformation est défini comme suit :

$$\mathrm{T_{ETCTC}} = \mathrm{T} + \frac{\mathrm{p} + (\mathrm{N\ div}\ (m_t\text{-}\mathrm{m}))}{} \left(\sum_{i=1}^{i=n} T_i \Big/ \left(\frac{\sum_{j=1}^{j=k} mj}{m_t - m}\right)\right)$$

Il est démontrable que $\mathbf{T_{ETCTC}} < \mathbf{T_{ETC}}$ car le temps T_{ETCTC} est égale à T_{ETC} divise par le nombre moyen des ordinateurs utilisés par source des données opérationnelles.

De ce fait, l'adaptation du processus d'ETC implique une amélioration des performances en termes de temps d'exécution du processus d'entreposage des données.

4. Performances du système proposé

L'expérimentation réalisée a permis de constater des bonnes performances de notre proposition en termes de qualité des données et de temps. Cependant, il existe d'autres résultats qui montrent la performance du notre système. Parmi ces résultats, nous citons:

1. Optimisation de l'espace de stockage des règles et de leur temps d'accès: C'est le résultat de l'opération de l'unification des règles que nous avons proposé dans le troisième chapitre. L'unification a permis de réduire le nombre des règles, par conséquence, l'espace de stockage te le temps d'accès à ces règles ont été performés.
2. Récupération des règles: L'incorporation des connaissances des utilisateurs a permis de récupérer des règles importantes élaguées par les outils d'extraction des connaissances à partir des données automatiquement.
3. Réduction du nombre règles rares et les règles incertaines: L'utilisation de l'environnement et, plus spécifiquement, les sources des données concernées par les règles a permis de déterminer le jeu exact des données où les règles sont applicables. Par conséquent, le nombre des règles rares et incertaines est réduit.

5. Conclusion

Dans ce chapitre, nous avons réalisé une première expérimentation pour valider certains composants du notre système et une validation théorique pour montrer les performances du système proposé en termes de la qualité des données et de temps.

L'expérimentation qu'est effectué dans un établissement sanitaire est basée sur le développement d'un outil d'extraction, transformation et chargement des données que nous avons implémenté selon le processus ETCTC proposé. Elle a montré l'efficacité de l'adaptation du processus d'ECD et, plus spécifiquement, du processus d'ETCTC et de propagation des données corrigées vers les sources des données opérationnelles. Nous avons aussi déduit certaines performances en termes d'espace de stockage, de temps d'accès; de réduction des règles rares et incertaines et de récupération des règles élaguées automatiquement.

Cependant, nous avons constaté la difficulté de l'extraction des connaissances à partir des employées qu'est coûteuse en termes de temps et de qualité.

Conclusion générale et perspectives

Conclusion générale et perspectives

Le travail que nous avons mené dans cette thèse entre dans le cadre des travaux qui s'intéressent à l'évaluation et l'amélioration de la qualité des données et des connaissances dans le processus d'extraction des connaissances à partir des données. L'étude bibliographique que nous avons effectuée, nous a permis de constater que les processus d'extraction des connaissances à partir des données ont des spécificités qu'il faut prendre en compte à des fins de qualité. La première spécificité que nous avons constatée est que l'entrepôt des données est l'élément fondamental du processus d'ECD. La deuxième spécificité est que l'amélioration de la qualité est centrée utilisateur. La troisième spécificité est la synergie des données et des connaissances. La quatrième spécificité est que la qualité concerne les données et les connaissances. De ce fait nous avons proposé l'adaptation de processus d'ECD pour une meilleure prise en charge de la qualité des données.

L'implication de l'utilisateur en tant que utilisateur final ou administrateur dans la gestion de la qualité demeure importante car les utilisateurs possèdent des connaissances qui peuvent être exploitée à des fins d'évaluation et d'amélioration de la qualité et, plus spécifiquement, à des fins du nettoyage des données. Nous avons montré que l'exploitation des connaissances expertes et du domaine est conditionnée par la gestion de leur qualité. Cela nous a conduit à proposer la formalisation et l'incorporation des connaissances expertes dans le processus d'ECD comme un moyen de combiner les approches expertes basées sur l'élicitation des connaissances et les approches basées sur l'extraction automatique des connaissances. De ce fait, nous avons développé dans ce travail un système de gestion d'entrepôt des règles que nous généralisons pour fournir un cadre global de représentation des connaissances. Ainsi ce système permet :

- La représentation de tout type de règle: à la différence des systèmes existants à base de règles qui permettent uniquement la représentation d'un seul type de règle, le système que nous avons proposé est doté d'un formalisme que nous avons développé pour permettre la représentation de tout type de règles. Ce formalisme permet la représentation des règles et de leur qualité. L'avantage majeur de ce formalisme est qu'il tient compte de la relativité de l'évaluation de la qualité et de l'évolution de la règle.
- L'entreposage des règles : Le système que nous avons développé propose un processus dédié à l'entreposage des règles. Dans ce processus nous avons adapté le processus d'extraction, transformation et chargement des données aux règles. A la différence du processus ECD qui permet uniquement l'extraction automatique des connaissances, nous avons conçu l'ETC des règles de manière à permettre l'acquisition des connaissances par élicitation. Au début de ce travail, nous avons voulu ajouter l'élicitation des connaissances au processus ECD mais, par la suite, elle nous s'est avérée coûteuse en termes de temps et de moyens humains. Par conséquent, nous l'avons intégrée dans le processus d'entreposage

des règles. De cette façon, nous garantissons la séparation des processus d'entreposage des règles et d'ECD. Cela veut dire qu'aucun processus ne perturbe l'autre.

- La gestion de la qualité des connaissances et, plus spécifiquement, des règles: Nous avons doté le système proposé par des différentes opérations afin de permettre une gestion totale et complète de la qualité des règles tout au long du processus d'entreposage des règles. Pendant l'extraction des connaissances, nous avons développé une méthode basée sur la méthodologie But-Question-Métrique pour la collection des informations à propos des connaissances du domaine. La qualité des ces informations sera auditée au niveau de la base des connaissances afin de détecter les anomalies. Au cours de la transformation, nous avons crée pour chaque type de règle un base intermédiaire afin de permettre aussi la vérification de leur qualité avant leur chargement dans l'entrepôt des règles. Pour cela, nous avons mis en œuvre des différents algorithmes orientés qualité. même pendant le chargement, le sytème vérifie la consistance et la redondance des règles de l'entrepôt à chaque insertion d'une nouvelle règle.
- Le rafraîchissement de l'entrepôt des règles: le système est doté de deux sous systèmes permettant le rafraîchissement de l'entrepôt des règles. Cela est important car l'entreposage des règles ne peut être réalisé en une seule fois. Nous l'avons conçu de manière à permettre l'implication de l'utilisateur à tout moment pour mettre à jour la base des connaissances et les bases intermédiaires des règles. A la différence du processus classique d'entreposage des données, notre système autorise l'utilisateur à compléter certaines informations des règles. Pour cette raison, nous avons dit que les règles de l'entrepôt sont volatiles (contrairement à l'entrepôt des données où les données sont non volatiles). Cette opération concerne uniquement certaines informations.

La prise en charge des spécificités de la complémentarité et la synergie des processus d'extraction des connaissances à partir des données et de l'entreposage des données dans l'amélioration de la qualité dans l'ECD, nous a conduit à la proposition de l'adaptation du processus d'entreposage des données. cette adaptation est portée sur le processus d'extraction, transformation et chargement des données qui assure 80% du travail de l'entreposage des données. Nous avons étendu le processus ETC par deux phases : monotransformation et monochargement. Le processus adapté que nous avons appelé ETCTC pour extraction, monotransformation, monochargement, transformation et chargement des données a permis:

- L'exploitation du parallélisme pendant la phase de monotransformation et de monochargement des données. Cela nous a permis de garantir une haute performance en termes de temps d'entreposage des données.
- L'amélioration de la qualité des données des sources des données opérationnelles en propageant les données évaluées de mauvaise qualité et corrigées vers leurs sources opérationnelles. Cet aspect n'est pas été pris en charge par les travaux de

recherche actuels menés sur la qualité des données de l'entrepôt. Cette propagation a permis d'éviter de refaire les mêmes opérations d'amélioration de la qualité pendant des futurs entreposages des données.

- L'implication des utilisateurs des sources des données opérationnelles pour valider les corrections faites pendant le nettoyage des données a permis de détecter et corriger certaines erreurs introduites pendant le nettoyage des données.

De ce fait, dans le processus ETCTC que nous avons développé, nous avons considéré les données corrigées comme étant un cas particulier des données. La prise en charge de cette particularité n'a pas permis uniquement la validation des données corrigées mais aussi d'évaluer et d'améliorer la qualité des règles qui ont été appliquées sur les données corrigées.

Pour une meilleure prise en charge des données corrigées, nous avons doté le processus adapté d'ETC par trois processus. Le premier est établi pour la traçabilité des données corrigées. Il permet de pointer directement sur les données corrigées et ses données dérivées si l'utilisateur des sources des données opérationnelles évalue les corrections faites sur ces données de mauvaise qualité. Le deuxième processus permet de capturer les changements des données corrigées au niveau de leur sources originales permet par les utilisateurs et de les récupérer en cas où les utilisateurs les refusent et les changent. Le troisième processus est conçu afin de permettre la propagation des données corrigées vers leurs sources. C'est un processus très important car à notre connaissance, il traite un problème qui n'a pas été considéré dans les travaux actuels est que la qualité concerne les données dans les bases sources et cibles. De ce fait, l'objectif de ce processus est de propager les données corrigées vers les sources des données opérationnelles afin de permettre leur amélioration au niveau de leurs sources. Il permet aussi de mesurer la qualité des règles.

Finalement, nous avons présenté le processus d'extraction des connaissances à partir des données après son adaptation. Cette adaptation est logique car la plupart des opérations de transformations et d'améliorations de la qualité des données sont faites par le processus d'entreposage des données. Donc nous avons supprimé la phase de préparation des données. Il reste uniquement quelques transformations qui ne peuvent pas être transformées sous forme de règles. Aussi, nous avons ajouté une phase pour l'insertion des connaissances extraites dans la base des règles afin de les transformer sous forme de règles et de les utiliser pendant le nettoyage des données.

L'expérimentation effectuée nous a permis de confirmer les avantages et les performances offertes par le système proposé qui sont principalement la gestion totale et continue de la qualité des données et des connaissances et l'amélioration du temps d'entreposage des données.

Ce travail est d'autant moins évident, qu'il est difficile pour les utilisateurs d'exprimer de façon exhaustive leurs connaissances. Cela, implique, lors de la collection des informations, d'être capable de creuser par des questions, pour être sûr qu'aucun détail ne nous a échappé. La plus

grande satisfaction que l'on peut avoir par rapport à un travail de développement de ce type est bien sûr de constater la positivité des résultats de la première expérimentation.

Cependant, de nombreuses perspectives de développement émergent de ce travail. Du point du vue informatique, un des points cruciaux qu'il nous reste à explorer dans le cadre de notre recherche est l'implémentation de la totalité du système. C'est important de chercher une entreprise intéressée à la qualité de ses données pour l'implémentation de cette proposition car elle est coûteuse en termes de temps et de financement.

D'autre part il est nécessaire de penser à l'automatisation du processus d'élicitation des connaissances. Les travaux de recherche ont démontré l'utilité et l'efficacité des connaissances du domaine dans des différents domaines et différentes application. Cependant, ce problème reste ouvert.

En outre, il est nécessaire d'étudier comment tirer profit au mieux des ontologies dans le processus d'extraction des connaissances à partir des données, et notamment pour l'évaluation et l'amélioration des règles extraites. Une des applications privilégiées sera le domaine de la santé où les ontologies sont très largement répandues et utilisées pour décrire les connaissances des différents domaines. Il s'agira donc dans cette perspective d'étudier: comment optimiser la qualité des données et des connaissances en s'appuyant sur les ontologies.

Une autre perspective qui nous semble la plus prometteuse est la fouille des connaissances. Partant du constat que les connaissances peuvent êtres considérées comme des données à structure complexe, cette perspective consiste à développer des algorithmes spécifiques de fouille de données dans des bases de connaissances (composées d'ontologies OWL, de règles, . . .), afin de découvrir des relations (méta connaissances ?) au sein des connaissances stockées.

Finalement, le développement de nouvelles technologies comme le Web sémantique, le calcul sur grille, les services web ouvrent de nouvelles perspectives et défis pour une nouvelle génération des systèmes de gestion de la qualité et d'extraction des connaissances à partir des données. Cette nouvelle génération de ces systèmes peut être conçue comme services distribués, pervasifs, contextualisées, services web ou grille.

Bibliographie

Références bibliographiques

1. K. J. Cios, W. Pedrycz, R. W. Swiniarski, L. A. Kurgan, "Data Mining: A Knowledge Discovery Approach",1st edition, Springer, 2007.
2. G. Felici, C. Vercellis, "Mathematical Methods for Knowledge Discovery and Data Mining", Hershey, New York, Information science reference, 2007.
3. F. Pennerath, "Méthodes d'extraction de connaissances à partir de données modélisables par des graphes : Application à des problèmes de synthèse organique", Thèse de Doctorat, Université de Nancy –Poincaré, France, 2009.
4. J.C. Ralaivao, J. Darmont, "Knowledge and Metadata Integration for Warehousing Complex Data", In *Proceedings of 6th International Conference on Information Systems Technology and its Applications (ISTA 07),* Kharkiv, Ukrain, Lecture Notes in Informatics, Bonn, Germany, Vol. P-107, pp. 164-175, GI-Edition, May 2007.
5. F. Atigui, F. Ravat, R. Tournier, G. Zurfluh, "A Unified Model Driven Methodology for Data Warehouses and ETL Design", In *Proceedings of the 13th International Conference on Entreprise Information Systems, (ICEIS)*, Beijing, China, June 8-11 2011, Vol. 1, pp. 247-252, ScitePress, 2011.
6. R. Wrembel, C. Koncilia, "Data Warehouses and OLAP: Concepts, Architectures and Solutions", editors IRM Press, Idea Group Inc., 2007.
7. F. Bentayeb, C. Favre, O. Boussaid, "Dynamic Workload for Schema Evolution in Data Warehouses: a Performance Issue", Chapter 2 in book of "*Complex Data Warehousing and Knowledge Discovery for Advanced Retrieval Development: Innovative Methods and Applications", Advances in Data Warehousing and Mining(Adwm), Book Series*, pp. 28-46, IGI Publishing, 2010.
8. M. P. Angeles, F. García-Ugalde, "A Data Quality Practical Approach", *International Journal on Advances in Software*, Vol. 2 , No. 2&3, pp. 259-273, 2009.
9. L. Berti-Equille, "La qualité des données comme condition à la qualité des connaissances : un état de l'art", *Revue Nationale des Technologies de l'Information,* RNTI-E, Cépaduès-Editions, 2004.
10. J. Blanchard, F. Guillet, H. Briand, "Une Visualisation orientée qualité pour la fouille anthropocentrée de règles d'association", *Journal Cognito –Cahiers Romans de Sciences Cognitives-,* Vol. 1, No. 03, pp. 79-100, 2003.
11. D. A. Zighed, G. Venturini, "Fouille de données d'opinions", *Revue des Nouvelles Technologies de l'Information,* RNTI-E-17, Cépaduès-Éditions, 2009.
12. C. Favre, F. Bentayeb, O. Boussaid, "A Rule-based Data Warehouse Model", In *Proceedings of 23th British National Conference on Databases (BNCOD'06*), Belfast, Northern Ireland, July 18-20 2006, LNCS, Vol. 4042, pp. 274-277, 2006.

13. W. Ben Ahmed, "SAFE-NEXT : Une approche systémique pour l'extraction de connaissances de données: Application à la construction de scenarios d'accidents de la route", Thèse de Doctorat, École Centrale Paris, France, 2005.
14. F. Bentayeb, C. Favre, O. Boussaid, "A User-driven Data Warehouse Evolution Approach for Concurrent Personalized Analysis Needs", *Journal of Integrated Computer-Aided Engineering (ICAE)*, Vol.15, No. 1, pp. 21-36, 2008.
15. K. Aouiche, J. Darmont, "Data mining-based materialized view and index selection in data warehouses", *Journal of Intelligent Information Systems (JIIS), Special Issue: Computational Techniques for Intelligent Information Systems*, Vol. 33, No. 1, pp. 65-93, August 2009.
16. J. Barateiro, H. Galhardas, "A survey of data quality tools", *Datenbank-Spektrum*, No. 14, 2005.
17. P. Vassiliadis, "A Survey of Extract–Transform–Load Technology", *International Journal of Data Warehousing & Mining*, Vol. 5, No. 3, pp. 1-27, July –September, 2009.
18. H. Briand, M. Sebag, R. Gras, F. Guillet, "Mesures de Qualité pour la Fouille de Données", *Revue des Nouvelles Technologies de l'Information*, RNTI-E-1*,* Cépaduès-Éditio*ns*, 2004.
19. R. Y. Wang, M. Ziad, Y. W. Lee, "Data Quality", *Kluwer Academic Publishers*, 2002.
20. S. Stumpf, j. Mcdonnell, "'Data, Information and knowledge Quality in retail security decision making", In *Proceedings of* 3*rd International Conference on Knowledge Management (IKNOW'03),* Graz, Austria, July 2-4 2003.
21. A. Napoli, "Formalisation des connaissances et contribution du langage de modélisation ML: Application à l'aide à la modélisation du comportement des incendies de forêt", Chapitre dans le livre : *"Systèmes d'information et risques naturels", édité par F., Guarnieri,* E. Emmanuel, Ecole des mines de Paris, 2003.
22. M. H. Haddad, "Knowledge extraction and impact on information retrieval systems", A Thesis, University of Joseph Fourier – Grenoble, France, 2002.
23. S. Kambhampaty, "Árchitecting knowledge management systems", chapter in book: "Strategic knowledge management in multinational organizations", *edited by O., O'Sullivan,* pp.119-125, IGI Global, USA, 2008.
24. C. Guerra-García, I. Caballero, L. Berti-Équille, M. Piattini, "DAQ_UWE: A Framework doe Designing Data Quality Aware Web Applications", In *Proceedings of the 16th International Conference on Information Quality (ICIQ),* Adelaide, Australia, November 18-20 2011, pp. 115-129, 2011.
25. S. Bard, "Méthode d'évaluation de la qualité de données géographiques généralisées Application aux données urbaines", Thèse de Doctorat, Université De Paris 6, 2004.
26. H. Hinrichs, T. Aden, "An ISO 9001:2000 Compliant Quality Management System for Data Integration in Data Warehouse Systems*",* In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2001),* Interlaken, Suisse, June 4 2001, Vol. 39, pp. 1-12, 2001.
27. A. Rivet, "Normes de qualité et systèmes d'information", *les journées réseaux*, Strasbourg, France, 20-23 Novembre2007.

28. "Glossary of Statistical terms, OECD: Organization for Economic Co-operation and Development", 2007. http://www.stats.oecd.org/glossary/
29. C. Toulemonde, "Des données de qualité : exploitez le capital de votre organisation", *livre blanc de JEMM Research Informatica*, pp. 1-26, Janvier 2008.
30. R. Devillers, A. Stein, Y. Bédard, N. Chrisman, P. Fisher, W. Shi, "Thirty Years of Research on Spatial Data Quality: Achievements, Failures, and Opportunities", *Transactions in GIS,* Vol. 14, No. 4, pp. 387-400, 2010.
31. X.H. Huynh, F. Guillet, H. Briand, "Une plateforme exploratoire pour la qualité des règles d'association: apports pour l'analyse implicative", *Actes du 3ème Rencontres Internationales Analyse Statistique Implicative,* Palermo, Italie, pp. 339-349, October 6-8 2005.
32. S. Lin, J. Gao, A. Koronios, V. Chanana, "Developing a data quality framework for asset management in engineering organizations", *International Journal of Information Quality (IJIQ)*, Vol. 1, No. 1, pp. 100–126., 2007.
33. E. Rahm, H. H. Do, "Data Cleaning: Problems and Current Approaches", *IEEE Data Engineering Bulletin*, University of Leipzig, Germany, Vol. 23, No. 4, pp. 3-13, 2000.
34. C. White, "Developing a Universal Approach to Cleansing Customer and Product Data", *white paper*, *BI Research, Business Objects SAP company,* 2010.
35. H. Galhardas, D. Florescu, D. Shasha, E. Simon, C. Saita, "Declarative Data Cleaning: Language, Model, and Algorithms", *Rapport de Recherche*, n° 4149, INRIA, March 2004.
36. H. Müller, J. C. Freytag, "Problems, Methods, and Challenges in Comprehensive Data Cleansing", *Technical Report*, No. HUB-IB-164, Humboldt-Universität zu Berlin, Germany, 2003.
37. C. C. Guitierrez Rodriguez, "Qualité des données capteurs", *Editions Universitaires Européennes*, 2011.
38. R.R. Nemani, R. Konda, "A Framework for Data Quality in Data Warehousing", *J. Yang et al. (eds): UNISCON'09, LNBIP*, Vol. 20, Part 2, Part 5, pp. 292-297, Springer-Verlag Heidelberg, 2011.
39. C. Batini, M. Scannapieco, "Data Quality: concepts, methodologies and techniques", 1st edition, Springer-Verlag Berlin Heidelberg, Gmbh & Co., 2010.
40. J. E. Olsen, "Data Quality: the accuracy dimension", Morgan Kaufmann Publishers, Elsevier, 2003.
41. Y. Hao, D. X. Chun, L. Kai-qi, "Research on Information Quality Driven Data Cleaning Framework", In *Proceedings of IEEE International Seminar on Future Information Technology and Management Engineering, FITME'08,* Leicestershire, United Kingdom, 20-22 Novembre 2008, pp. 537 – 539, 2008.
42. S. Madnick, R. Y. Wang, Y. W. Lee, H. W. Zhu, "Overview and Framework for Data and Information Quality Research", *ACM Journal of Data and Information Quality*, Vol. 1, No. 1, Article 2, 2009.
43. D.D. Fehrenbacher, M. Helfert, "An empirical research on the evaluation of data quality dimensions", In *Proceedings of the 13th International Conference on Information Quality (ICIQ),* MIT, Cambridge, USA, November 14-16 2008, pp. 230-245,2008.

44. T. T. P. Thi, M. Helfert, "Discovering Dynamic Integrity Rules with a Rules-Based Tool for Data Quality Analyzing", In *Proceedings of the 11th International Conference on Computer Systems and Technologies (CompSysTech'10) ,* Sofia, Bulgaria, June 17-18 2010, pp. 89-94, 2010.
45. A. Caro, C. Calero, M. Piattini, "A Portal Data Quality Model for Users and Developpers", In *Proceedings of the 12th International Conference on Information Quality (ICIQ*), MIT, Cambridge, USA, November 9-11 2007, pp. 462-476, 2007.
46. G. Shankaranarayanan, A. Even, "Measuring Data Quality in Context", In *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends*, (Ferraggine, V. E., Doorn, J. H. and Rivero, L. C. - Editors), IGI - Global, Hershey, Pennsylvania, 2009.
47. J. Akoka, L. Berti-Équille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer, V. Goasdoué, Z. Kedad, S. Nugier, V. Peralta, M. Quafafou, S. Sisaïd-Cherfi, "Évaluation de la qualité des systèmes multi sources Une approche par les patterns", *Actes du 8ème Journées Francophones sur l'Extraction et Gestion des Connaissances (EGC),* Sophia Antipolis, Nice France, 29 Janvier 2008.
48. S. Ben Hassine-Guetari, "Data quality evaluation in an e-business environment: A survey", In *Proceedings of 14th International Conference on Information Quality (ICIQ),*Hasso Plattner Institute, University of Potsdam, Germany, November 7-8 2009, pp. 189-201, 2009.
49. J. L. Kulikowski, "Data Quality Assessment", *Handbook of Research on Innovations in Database Tchnologies and Applications: Current and Future Trends,* V. E. Ferraggine, J. H. Doorn, L. C. Rivero, editors, Information Science Reference by IGI Global, pp. 378–384, 2009.
50. L. Berti-Équille, "Data quality awarness: a case study for cost optimal association rule mining", *Special Issue of Knowledge Information Systems,* London, UK, pp. 191 -215, 2007.
51. M. Helfert, C. Herrmann, "Proactive Data Quality Management for Data Warehouse Systems - A Metadata based Data Quality System -", *Journal od Data mining and Data warehouse (DMDW*), Vol. 2002 pp 97 -106, 2002.
52. K. Ali, M. A. Warraich, "A framework to implement Data Cleaning in Enterprise Data Warehouse for Robust Data Quality", In *Proceedings of 2010 IEEE International Conference on Information and Emerging Technologies (ICIET),* Karachi, Pakistan, June 14-16 2010, pp. 1-6, 2010.
53. G. Shankaranarayanan, "Towards implementing total data quality management in a data warehouse", *Journal of Information Technology Management*, Vol. XVI, No. 1, pp. 21-30, 2005.
54. A. Koronios, S. Lin, "Information Quality in Engineering Asset Management", Chapter X in book: "*Information Quality Management: Theory and Applications*", edited by L. El-Hakim, Idea Group PublishIng (IGP), 2007.
55. L. Berti-Equille, "Measuring and Modelling Data Quality for Quality-Awareness in Data Mining", *Studies in Computational Intelligence (SCI),* Vol. 43, pp. 101–126, 2007.

56. T. Dasu, L. Berti-Equile, "Data Quality Mining: New Research Directions", Tutorial presented at *IEEE International Conference on Data Mining (ICDM),* Miami, Florida, USA, 7 December 2009.
57. J. Akoka, L. Berti-Équille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer, V. Goasdoué-Thion, Z. Kedad, S. Nugier, V. Peralta, S. Sisaid-Cherfi, "A framework for quality evaluation in data integration systems", In *Proceeding of 9th International Conference on Enterprise Information Systems, Madeira (ICEIS*), Funchal, Madeira, Portugal, June 12-16 2007, pp. 170-175, 2007.
58. R. Choquet, S. Qouiyd, E. Pasche, C. Daniel, O. Boussaid, M.Christine jaulen, "Un modèle de connaissances pour mesurer la qualité d'une source d'information", *Actes des 21ème Journées francophones d'Ingénierie des Connaissances (IC),* Nîmes, France, June 8-11 2010.
59. L. Berti-Équille, "Évaluation de la qualité des données et informations: la prise en compte de l'utilisateur", Tutorial dans les *Actes du Journée sur La Prise en Compte de l'Utilisateur dans les Systèmes d'Information (PeCUSI)*, Toulouse, France, 26 Mai 2009.
60. J. Han, M. Kamber, J. Pei, "Data Mining: Concepts and Techniques", 3rd edition Morgan Kaufmann Publishers, 2011.
61. T. Dasu, T. Johnson, "Exploratory Data Mining and Data Cleaning", AT&T Labs, Research Division Florham Park, NJ, 2003.
62. P. Oliveira, F. Rodrigues, P. Henriques, H. Galhardas, "A taxonomy of data quality problems", In *Proceedings of 2nd International Workshop on Data and Information Quality (DIQ' 05),* Porto, Portugal, June 14 2005, pp. 219-233, 2005.
63. H. Ye, D. Wu, S. Chen, "An Open Data Cleaning Framework Based on Semantic Rules for Continuous Auditing", In *Proceedings of IEEE 2010 2nd International Conference on Computer Engineering and Technology (ICCET),*Chengdu, China, April 16-18 2010, Vol. 2, pp. 158-162, 2010.
64. F. Guillet, H. J. Hamilton, "Quality Measures in Data Mining", Springer-Verlag, Berlin Heidelberg, 2007.
65. K. G. Herbert, "Biological data cleaning: a case study", *International Journal of Information Quality (IJIQ)*, Vol. 1, No. 1, pp 60-82, 2007.
66. S. Watts, G. Shankaranarayanan, A. Even, "Data Quality Assessment in Context: A Cognitive Perspective", *Journal of Decision Support Systems*, Vol. 48, pp. 202-211, 2009.
67. L. Dubrovin, C. Brault, "Qualité des Données : Organisation, Confiance et Initiatives", *Business Application Research Center (BARC)*, Juin 2011.
68. J.LIE, Y. KOH, "Correlation-Based Methods for Biological Data Cleaning", A Thesis, University of Singapore, Malaysia, Mars 2007.
69. P. Ponniqh, "Data warehousing fundamentals for it professionals", 2nd Edition, John Wiley & Sons, INC., Publication, 2010.
70. H. Prade, N. Aussenac, J. L. Soubie, H. Farreny, M. P. Gleizes, P. Glize, "L'Intelligence Artificielle, mais enfin de quoi s'agit-il? ", *les livrets du service culture UPS*, No. 13, IRIT, 2001.

71. F. Fürst, M. Leclère, F. Trichet, "Operationalizing Domain Ontologies: A Method and a Tool", In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, (ECAI'04), including Prestigious Applicants of Intelligent Systems, (PAIS'04),* Valencia, Spain, August 22-27 2004, pp. 318-322, 2004.
72. C. C. Chan, Z. Su, "From Data to Knowledge: an Integrated Rule-Based Data Mining System", In *Proceedings of the 17th International Conference of Software Engineering and Knowledge Engineering (SEKE'05),* Taipei, Taiwan, July 14-16 2005, pp. 508-513, 2005.
73. C. Angeli, "Diagnostic Expert Systems: From Expert's Knowledge to Real-Time Systems", *TMRF e-Book, Advanced Knowledge Based Systems: Model, Applications & Research (Eds. Sajja & Akerkar)*, Vol. 1, pp. 50 – 73, 2010.
74. S. Lukichev, G. agner, "UML-Based Rule Modeling with Fujaba", In *Proceedings of the 4th International Fujaba days 2006,* University of Bayreuth, Germany, September 28-30 2006, pp. 31-35, 2006.
75. F. Guillet, G. Ritschard, D. A. Zighed, H. Briand (eds), "Advances in Knowledge Discovery and Management", *Studies in Computational Intelligence*, Vol. 292, Springer, 2010.
76. G. Dondossola, "Formal Methods in the development of safety critical knowledge-based components", In *Proceedings of the KR'98 European Workshop on Validation and Verification of Knowledge-Based Systems,* Trento, Italy, June 1 1998, pp. 232-237, 1998.
77. A. J.RHEM, "UML for developing knowledge management systems", *Auerbach Publications, Taylor é Francis Group, Boca Raton New York*, 2006, http://www.auerbach-publications.com
78. L. Brisson, "Intégration de connaissances expertes dans le processus de fouille de données pour l'extraction d'informations pertinentes", Thèse de Doctorat, Université de Nice – Sofia Antipolis- UFR Sciences, France, 2006.
79. A. Abraham, "Rule-based Expert Systems", In Handbook of Measuring System Design, John Wiley & Sons Ltd, pp. 909-919, 2005.
80. S. M. Marlar Soe, M.Paing Zaw, "Design and Implementation of Rule-based Expert System for Fault Management", *World Academy of Science, Engineering and Technology*, Vol. 48, pp. 34-39, 2008.
81. MC. Lai, HC Huang and W. K. Wang, "Designing a knowledge-based system for benchmarking: A DEA approach", *Journal of Knowledge-Based Systems*, Vol. 24, pp. 662–671, 2011.
82. F. Le Ber, J. Lieber, A. Napoli, "Les systèmes à base de connaissances", *Encyclopédie de l'informatique et des systèmes d'information*, pp. 1197-1208, 2006.
83. C. Riou, P. Le Beux, P. Lenoir, "La représentation des connaissances dans le système d'aide à la décision médicale", *ADM, Informatique et Santé*, Vol. 5, pp. 95-107, 1992.
84. D. Sheeren, S. Mustière, J. D. Zucker, "A data-mining approach for assessing consistency between multiple representations in spatial databases", *International Journal of Geographical Information Science*, Vol. 23, No. 8, pp. 961-992, 2009.
85. P. Rozière, "Logique mathématique : une introduction au calcul des prédicats du premier ordre", Université Paris 7, M63010, 2011.

86. S. Hedman, "A First Course in Logic: An introduction to model theory, proof theory, computability, and complexity", Department of Mathematics, Florida Southern College, OXFOED University Press, 2006.
87. D. van Dalen, "Logic and Structure", Fourth edition, Springer-Verlag Berlin Heidelberg, 2008.
88. M. Ribarić, D. Gašević, M. Milanović, "A Rule-based Approach to Modeling of Semantically-enriched Web Services", *Web4Web Workshop, International Workshop on Semantic Web Technologies, Belgrade, Serbia*, September 29-30 2008.
89. S. Antony, R. Santhanam, "Could the use of a knowledge-based system lead to implicit learning?", *Journal of Decision Support System*, Vol. 43, pp. 141–151, 2007.
90. M. S. Abdullah, C. Kimble, I. Benest, R. Paige, "Knowledge-based systems: a re-evaluation", *Journal Of Knowledge Management, Emerald Group Publishing* , Vol. 10, No. 3, pp. 127-142, 2006.
91. A. Paschke, A. Kozlenkov, "A Rule-based Middleware for Business Process Execution", In *Proceedings of Multikonferenz WirtschaftsInformatik (MKWI'08),* Munchen, Germany, February 26-28 2008, pp. 1409-1420, GITO-Verlag, Berlin, 2008.
92. D. Batra, N. A. Wishart, "Comparing a rule-based approach with a pattern-based approach at different levels of complexity of conceptual data modelling tasks", *International Journal of Human-Computer Studies*, Vol. 61, No. 4, pp. 397-419, 2004.
93. V. Legendre, G. P. Jean, T. Lepatre, "Gestion des règles « métier »", *Compendium, Revue Génie logiciel*, vol. 92, Mars 2010.
94. R.K. Kumar,R. M. Chadrasekaran, "Attribute correlation-Data cleaning using Association rule and clustering methods", *International Journal of Data Mining & Knowledge Management Process (IJDKP)*,Vol.1, No.2, pp. 22-32, March 2011.
95. "Validation de règles actives: Etat de l'art et prototypes d'outils", *Rapport final du projet 'Bases de Données Actives'*, Facultés Universitaires Notre-Dame de la Paix, NAMUR, Charleroi, Février 2001.
96. A. Andreescu, M. Mircea, "Managing Knowledge as Business Rules*", Journal of Informatica Economica*, Vol. 13, No. 04, pp. 63-74, 2009.
97. M. Nickles, D. Sottara , "Approaches to uncertain or imprecise rules: a survey", In: Governatori, G., Hall, J., Paschke, A., (eds), *Rule Interchange and Applications*. Springer, pp. 323-336, 2009.
98. A.Lige¸ J. Nalepa, "A study of methodological issues in design and development of rule-based systems: proposal of a new approach", *Reviews Data Mining and Knowledge Discovery, John Wiley & Sons,* Vol. 1, pp. 117-137, 2011.
99. G. J. Nalepa, A. Ligeza, "The HEKATE methodology hybrid engineering of intelligent systems", *International Journal of Applied Mathematics and Computer Science*, Vol. 20, No. 1, pp. 35–53, 2010.
100. C. Marinica, "Fouille interactive de connaissances à l'aide de mesures et de représentations sémantiques", Thèse de Doctorat, Ecole polytechnique de l'université de Nantes, France, 2009.

101. E. F. Hill, "Rule-Based Systems in Java: Jess in Action", Manning Publications Co (eds), USA, 2003.
102. M. Diouf, S. Maabout, K. Musumbu, "Merging Model Driven Architecture and Semantic Web for Business Rules Generation", In *Proceedings of the 1th International Conference on Web Reasoning and Rule Systems (RR'07),* Innsbruck, Austria, June 7-8 2007, LNCS, Vol. 4524, pp. 118-132, Springer, 2007.
103. F. Guillet, "Mesure de la qualité des connaissances en ECD", Tutoriel de la 4ème *Conférence d'Extraction et Gestion des Connaissances (EGC'4),* Clermont Ferrand, France, 20 Janvier 2004.
104. L. Bradji, M. Boufaida, "Nettoyage des Données et Maintien de la Cohérence Dans les Entrepôts des Données", *2nd Workshop sur la Cohérence des Données dans l'Univers Réparti (CDUR'O8)*, Lyon, France, 23 Juin 2008. (Poster)
105. L. Bradji, M. Boufaida, "Parallélisme et cohérence pour l'optimisation du processus d'entreposage et la qualité des données", In *Proceedings of* 3ème *Workshop sur la Cohérence des données* en *Univers Réparti* (*CDUR*'09), Toulouse, France, 11 Septembre 2009.
106. C. Tongchuay , P. Praneetpolgrang , "Knowledge Quality and Quality Metrics in Knowledge Management Systems", In *Proceedings of 5th International Conference on e-Learning for Knowledge-Based Society, Bangkok, Thailand*, December 11-12 2008, pp. 211 – 216, 2008.
107. T. R. Chung, M. Boucher, W. R. King, "Knowledge Quality, User Motivation, and Knowledge Use: A Theoretical Framework and Research Proposal", the *7th Annual Pre-ICIS Information Systems Cognitive Research Workshop, Montreal, Quebec*, December 9 2007.
108. Y. Le Bras, P. Lenca, S. Lallich, "Mining interesting rules without support requirement: A general universal existential upward closure property", *Annals of Information Systems*, Vol. 8, pp. 75–98, 2010.
109. J. Blanchard, F. Guillet, H. Briand, "Interactive visual exploration of association rules with rule-focusing methodology", *Journal of Knowledge and Information Systems*, Vol. 13, No. 1 pp. 43-75, 2007.
110. S. Lallich, O. Teytaud, E. Prudhomme, "Association rules interestingness: measure and validation", *Quality Measures in Data Mining (eds. Guillet F. and Hamilton H. J.), Studies in Computational Intelligence*, Vol. 43, pp. 251-275, Springer-Verlag, Berlin Heidelberg, 2007.
111. A. D. Preece, R. Shinghal, "Foundation and Application of Knowledge Base Verification" , *International Journal of Intelligent Systems*, Vol. 22, pp. 23-41, 1994.
112. Y. SHI, H. Lam "Integrated verification of constraints and event-and-action business rules", A Thesis, University of Florida, USA, 2001.
113. C. W. Soo, T. M. Devinney, D. F. Midgley, "The Role of Knowledge Quality in Firm Performance", in H. Tsoukas and N. Mylonopoulos (eds.) *Organisations as Knowledge Systems*, London, Palgrave, 2004.
114. L. Berti-Équille, "Modelling and Measuring Data Quality for Quality-Awareness in Association Rule Mining", *Chapter in the book: "Quality Measures in Data Mining", edited by F. Guillet and H. Hamilton (Eds.),* pp. 101-126, Springer, 2007..

115. M.C.M. Batista, A.C. Salgado, "Information Quality Measurement in Data Integration Schemas", In *Proceedings of the 5th Workshop on Quality in DataBases (QDB'07), at the VLDB 2007 Conference,* Vienna, Austria, September 23, 2007 pp. 61-72, 2007.
116. Y. Le Bras, P. Lenca, S. Lallich, "On optimal rules mining: a framework and a necessary and sufficient condition for optimality", In *Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD),* Bangkok, Thailand, April 27-30 2009, LNCS, Vol. 5476, pp. 705–712. Springer-Verlag Berlin Heidelberg, 2009.
117. Y. Le Bras, P. Meyer, P. Lenca, S. Lallich, "A robustness measure of association rules", In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD),* Barcelona, Spain, September 20-24 2010, LNCS, Vol. 6322, Part II, pp. 227–242, Springer-Verlag Berlin Heidelberg, 2010.
118. H. Galhardas, "Data Cleaning and Transformation Using the AJAX Framework", In *the Proceedings of Generative and Transformational Techniques in Software Engineering (GTTSE),* Praga, Portugal, July 4-8 2005, LNCS, Vol. 4143, pp. 327–343, Springer, 2006.
119. G. Helena, L. Antonia, S. Emanuel, "Support for User Involvement in Data Cleaning", In *Proceedings of 13th International Conference Data Warehousing and Knowledge Discovery, (DaWaK),* Toulouse, France, August 29-September02 2011, LNCS, Vol. 6862, pp. 136-151, Springer, 2011.
120. P. Oliveira, F. Rodrigues, P. Henriques, "An ontology-based approach for data cleaning", In *Proceedings of the 11th International Conference on Information Quality (ICIQ'07)*, MIT, Cambridge, MA, November 9-11 2007, USA, pp. 307-320, 2007.
121. A. Vavouras, "A Metadata-Driven Approach for Data Warehouse Refreshment", A Thesis, Der Universität Zürich, 2002.
122. K. Duncan, D. Wells, "Rule Based Data Cleansing for Data Warehousing", San Diego, 2000.
123. D. V. Kalashnikov, S. Mehrotra, S. Chen, "RelDC: a novel Framework for data cleaning", *ACM Transactions on Databases Systems*, Vol. V, No. N, pp 1-37, May 2004.
124. M. L. Lee, T. W. Ling, W. L. Low, "IntelliClean : A Knowledge Based Intelligent Data Cleaner", In *the Proceedings of the 6th ACM SIGKDD Conference on Knowledge Discovery and Data Mining,* Boston, MA, USA, August 20-23 2000, pp. 290-294, 2000.
125. V. Raman, J. M. Hellerstein, "An Interactive Framework for Data Cleaning", *Technical Report* No. UCB/CSD-0-1110 Computer Science Division (EECS), University of California Berkeley, September 2000.
126. V. Raman, J. M. Hellerstein, "Potter'sWheel: An Interactive Data Cleaning System", In *Proceedings of the 27th International Conference on Very Large DataBases (VLDB),* Roma, Italy, September 11-14 2001, pp. 381-390, Morgan Kaufmann, 2001.
127. L. Bradji, M. Boufaida, "A Rule Management System for Knowledge based Data Cleaning ", *Intelligent Information Management*, Vol. 03, No. 06, pp. 230-239, 2011. http://www.scirp.org/journal/iim/
128. L. Bradji, M. Boufaida, "Knowledge Based Data Cleaning for Data Warehouse Quality", In *Proceedings of International Conference (ICDIPC),* Ostrava, Republic of Czech, July 7-9

2011, CCIS, LNCS, Vol.189, Part II, pp. 373–384, Springer-Verlag Berlin Heidelberg, 2011. http://www.springerlink.com/content/n24278p472372263/

129. G. Helena, L. Antonia, S. Emanuel, "Explicitly Involving the User in a Data Cleaning Process", *Technical Report*, No. DI–FCUL–TR–2010–03, Department of Informatics, University of Lisbon, Portugal, February 14 2011.
130. K. Belhajjame, N.W. Paton, A.A.A. Fernands, C. Hedeler, S.M. Embury, "User Feedback as a First Class Citizen in Information Integration Systems", In *Online Proceedings of the 5th Biennial Conference on Innovative Data Systems Research,* Asilmor, CA, USA, January 9-12 2011.
131. X. Chai, B.Q. Vuong, A. Doan, J. F. Naughton, "Efficiently incorporating user feedback into information extraction and integration programs", In *Proceedings of the ACM SIGMOD International Conference on Management of Data,* Providence, Rhode Island, USA, June 29-July 2 2009, pp. 87-100, 2009.
132. A. Rashid, M. M. Sufyan Beg, "User Feedback Based Metasearching Using Rough Set Theory", In *Proceedings of 2008 International Conference on Information and Knowledge Engineering (IKE),* Las Vegas, Nevada, USA, July 14-17 2008, pp. 489-495, CSREA Press , 2008.
133. P. Christen, "Febrl -: An open source data cleaning, deduplication and record linkage system with a graphical user interface", In *Proceedings of the 14th ACM SIGKDD International on Knowledge Discovery and Data Mining (KDD),* Las Vegas, Nevada, USA, August 24-27 2008, pp. 1065-1068, 2008.
134. A. Arasu, R. Kaushik, "A Grammar-based Entity Representation Framework for Data Cleaning", In *Proceedings of the 35th ACM SIGMOD International Conference on Management Data (SIGMOD'09),* Providence, Rhode Island, USA, June 29– July 2 2009, pp. 233-244, 2009.
135. D. A. Zighed, R. Rakotomala, "Extraction de connaissances à partir de données (ECD) ", *Technique de l'ingénieur*, Référence H3 744, 2003.
136. D. Cram, "Techniques d'extraction de connaissances pour la facilitation des tâches à base de traces d'interaction", *Deuxième partie du livrable T3.1: " États de l'art sur les traces*", LIRIS, Février 2008.
137. D. A. Zighed, S. Tsumoto, Z. W. Ras, H. Hacid (Eds.), "Mining Complex Data", *Studies in Computational Intelligence*, Vol. 165, Springer 2009.
138. O. Couturier, " Contribution à la fouille de données : règles d'association et interactivité au sein d'un processus d'extraction de connaissances dans les données", Thèse de Doctorat, Université d'Artois, France, 2005.
139. A. Vautier, M. O. Cordier, R. Quiniou, "Towards Data Mining Without Information on Knowledge Structure", In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Warsaw, Poland, September 17-21 2007, LNCS, Vol. 4702, pp. 300-311, Springer, 2007.
140. O. Maimon, L. Rokach, "Data Mining and Knowledge Discovery", 2nd Edition, Springer, 2010.

141. A. Laurent, "Fuzzy and Complex Data Mining: Knowledge Discovery from Multidimensional Data", A Thesis, University of Montpellier 2, France, 2009.
142. C. Candillier, "Méthodes d'Extraction de Connaissances à partir de Données (ECD) appliquées aux Systèmes d'Information Géographiques (SIG)", Thèse de Doctorat, Université de Nantes, France, 2006.
143. A. Simitisis, P. Vassiliadis, S. Skiadopoulos, T. Sellis, "DataWarehouse Refreshment", chapter in the book of "*Data Warehouses and OLAP: Concepts, Architectures and Solutions*", pp. 111-135, IRM Press Robert Wrembel, 2007.
144. M. Refaat, "Data Preparation for Data Mining Using SAS"", Morgan Kaufmann Publishers, Elsevier, 2007.
145. G. Singh, F. Masseglia, C. Fiot, A. Marascu, P. Poncelet, "Data Mining for Intrusion Detection: From Outliers to True Intrusions", In *Proceedings of the 13th Pacific-Asia Conference Advances in Knowledge Discovery and Data Mining, (PAKDD'09)*, Bangkok, Thailand, April 27-30 2009, LNCS, pp. 891-898, Springer, 2009.
146. R. Lefébure, C. Venturi, "Data mining : Gestion de la relation client personnalisation de sites web", 2nd edition, Edition Eyrolles, 2001.
147. D. Francisci, "Techniques d'optimisation pour la fouille des données", Thèse de Doctorat, Université de Nice-Sophia Antipolis, France, 2004.
148. L. Bradji, M. Boufaida, "Rules-based Data Warehouse Quality Framework for Data Mining", In *Proceedings of the 23rd European Conference on Operational Research (EURO'09)*, Bonn, Germany, July 5-8 2009.
149. E. Ziyati, A. El Qadi, A. Driss, "Genetic optimization to data warehouse Design", *International Journal of Computational Science (IJCS*), Vol. 4, No. 1, pp. 70-79, 2010.
150. L. Bellatreche, "Optimization and Tuning in Data Warehouses*", Encyclopedia of Database Systems, Ling Liu and Tamer Özsu*, pp. 69-76, Springer, 2009.
151. P. Ponniah, "Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals", A Wiley-Interscience Publication John Wiley & Sons, INC. Copyright John Wiley & Sons, Inc. 2001.
152. W. H. Inmon, "Building the Data Warehouse", 4ème Edition, John Wiley Publishing, New York, Inc., 2005.
153. M. Mannino, S. N. Hong, "Efficiency evaluation of data warehouse operations", *Decision Support Systems*, Vol. 44, pp. 883–898, 2008.
154. F. Bentayeb, N. Maiz, H. Mahboubi, C. Favre, S. Loudcher, N. Harbi, O. Boussaid, J. Darmont, "Innovative Approaches for efficiently Warehousing Complex Data from the Web", Chapter in book of "*Business Intelligence Applications and the Web: Models, Systems and Technologies",* (Marta E. Zorrilla, Jose-Norberto Mazón, Óscar Ferrández, Irene Garrigós, Florian Daniel, Juan Trujillo (eds)), *Business Science Reference*, IGI Book, pp. 26-52, 2011.
155. B. Griesemer, "Oracle Warehouse Builder 11gR2: Getting Started", Packt Publishing, May 2011.

156. M.T. Serna-Encinas, M. Adiba, "Entrepôt de données pour l'aide à la décision médicale : Conception et expérimentation", *1ère journée francophone sur les Entrepôts de Données et l'Analyse en ligne (EDA 2005), Lyon,* France, Juin 10 2005, *Revue des Nouvelles Technologies de l'Information,* RNTI- B-1, pp. 122-140, Cépaduès, 2005.
157. F. Ravat, O. Teste, R. Tournier, G. Zurfluh, "Integrating Complex Data into a Data Warehouse", In *Proceedings of the 19th International Conference on Software Engineering Knowledge Engineering (SEKE'2007*),Boston, Massachusetts, USA, July 9-11 2007, pp. 483-486, Knowledge Systems Institute Graduate School, 2007.
158. E. Guérin, G. Marquet, A. Burgun, O. Loréal, L. Berti-Equille, U. Leser, F. Moussouni, "Integrating and Warehousing Liver Gene Expression Data and Related Biomedical Resources in GEDAW", , In *Proceedings of the 2nd International Workshop on Data Integration in the Life Sciences (DILS'05),* San Diego, CA, USA, July 20-22 2005, LNCS, Vol. 3615, pp. 158-174, Springer 2005.
159. M.D. Van Damme, "Entrepôts de données dans le domaine spatial pour l'inventaire forestier", Rapport technique, Conservatoire National des Arts et Métiers, France, 2010.
160. J. Adzic, F. Sisto, "Extraction, Transformation, and Loading Processes", chapter in the book of "*Data Warehouses and OLAP: Concepts, Architectures and Solutions*", pp. 88-110, IRM Press Robert Wrembel, 2007.
161. C. Gueydan, "XeuTL : un outil ETL pour l'intégration de données", Rapport technique, Conservatoire National des Arts et Métiers (CNAM), France, 30 juin 2010.
162. F. Francheteau, A. Giicquel, C. Jermann, "Etude des ETL Open Source", *Rapport de stage*, Université de Nantes, France, 2008.
163. L. Nemuraite, J. Tonkunaite, B. Paradauskas, "Model-Driven Development for Enabling the Feadback from Warehouses and OLAP to Operational systems", *Databases and Information Systems IV*, O. Vasilecas et al. (Eds), IOS Press, pp. 147-158, 2007.
164. C. Pierkot, "Gestion de la Mise à Jour de Données Géographiques Répliquées", Thèse de Doctorat, Université Toulouse III - Paul Sabatier, France, July 02 2008.
165. C. Salinesi, I. Gam, "How Specific should Requirements Engineering be in the Context of Decision Information Systems?", In *Proceedings of International Conference on Research Challenges in Information Science (RCIS*), Fès, Maroc, April 22-24 2009, pp. 247-254,2009.
166. E. Pacitti, "Réplication asynchrone des données dans trois contextes: entrepôts, grappes et systèmes pair-à-pair", HdR, Université de Nantes, France, July 8 2008.
167. D. Goldman, F. Gadi, I. Ankorian, "Attunity Oracle-CDC for SSIS, User Guide, Version 2.1", *Attunity Ltd.*, December, 2009.
168. J. Rodic, M. Baranovic, "Generating Data Quality Rules and Integration into ETL Process", In *Proceeding of ACM 12th International Workshop on Data Warehousing and OLAP (DOLAP'09) ,* Hong Kong, China, November 6 2009, pp. 65-72, 2009.
169. A. Adarsh, K.R. Rajendra, "Implementing a Data Quality Module in an ETL Process", *A Project Report*, Rochester Institute of Technology, Rochester, New York, USA, April 2011.

170. Y. Cui, J. Widom, "Lineage Tracing for General Data Warehouse Transformations", In *Proceedings of the 27th International Conference on Very Large DataBases (VLDB),* Roma, Italy, September 11-14 2001, pp. 471-480, 2001.
171. M. Zhang, X. Zhang, X. Zhang, S. Prabhakar, " General Data Warehouse Transformations Operators", In *Proceedings of the 33th International Conference on Very Large DataBases (VLDB* '07), Vienna, Australia, September23-27 2007, pp. 1116-1127, 2007.
172. B. Duval, A. Salleb, C. Vrain, "On the Discovery of Exception Rules: A Survey", *Studies in Computational Intelligence*, Vol. 43, pp. 77-98, Springer-Verlag, Berlin Heidelberg, 2007.
173. V. Basili, G. Caldiera, H. Rombach, "The Goal Question Metric Approach", John Wiley & Sons, Inc. 1994.
174. T. Baudin, H. Chattou, V. Sitthisack, D. Valin, "Méthodes de métriques du Logiciel", *Projet Industrial*, Valeo Climate Control, 2007.
175. H. Hinrichs, "CLIQ – Intelligent Data Quality Management", In *Proceedings of the 4th IEEE International Baltic Workshop on Databases and Information System,* In *Proceedings of the 7th Doctoral Consortium (CAiSE '00*), FU Berlin, Germany, pp. 25-36, 2000.
176. W. Li, L. Lei, "An Object-Oriented Framework for Data Quality Management of Enterprise Data Warehouse", In *Proceedings of the 9th Pacific Rim international conference on Artificial intelligence*, Guilin, China, August 7-11 2006, Vol. 4099, pp. 1125 – 1129, Springer Berlin, 2006.
177. C. Quix, "Repository Support for Data Warehouse Evolution", In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'99) Heidelberg, Germany*, June 14 – 15 1999.
178. M. Jarke, "Data Warehouse Quality and Agent Technology", *Cooperative Information System (CIA),* LNCS, Vol. 2182, pp. 56–75, 2001.
179. P. Vassiliadis, M. Bouzeghoub, C. Quix, "Towards Quality-Oriented Data Warehouse Usage and Evolution", *Journal of Information System,* Vol. 25, No. 2, pp. 89-115, 2000.
180. M. Helfert, C. Herrmann, "Introducing data-quality management in data warehouse" chapitre dans "Information Quality", *Advances in Management Information System*s, Vol. 1, pp. 135-150, ME Sharp, 2005.
181. L. Etcheverry, V. Peralta, M. Bouzeghoub, "Qbox-Foundation: a Metadata Platform for Quality Measurement", In *Proceedings of 4th Workshop on Data and Knowledge Quality (DKQ'2008*), Sophia-Antipolis, France, January 2008.
182. G. C. Moura Amaral, M. L. Machado Campos, "AQUAWARE: A Data Quality Support Environment for Data Warehousing", SBBD'04, *XIX Simpósio Brasileiro de Bancos de Dados*, 18-20 de Outubro, 2004,Brasília, Distrito Federal, Brasil, Anais, pp. 121 – 133 , 2004.
183. J. Depauw, "Qualité de l'information et vigilance collective sur le web Étude des stratégies d'évaluation des sources en ligne par les professionnels de la gestion de l'information dans les organisations", Thèse de Doctorat, Université Libre de Bruxelles, Belgique, 2009.
184. A. Y. Liu, Y. Shi, H. Lam, S. Y. W. Su, K. Pillamarri, M. Islamraja, "A Rule Warehouse System for Knowledge Sharing and Business Collaboration", In *Proceedings of Information and Knowledge Sharing (IKS*), St. Thomas, US Virgin Islands, 18-20 Novembre 2002.

185. A. Y. Liu, "A Rule Warehouse System for Knowledge Sharing and Business Collaboration", Ph. D. Dissertation,, Department of Computer and Information Science and Engineering, University of Florida, USA, August 2001.
186. H. R. Nemati, D. M. Steiger, L. S. Iyer, R. T. Herschel, "Knowledge Warehouse: An Architectural Integration of Knowledge Management, Decision Support, Data Mining and Data Warehousing", *Journal of Decision Support Systems*, Vol. 33, No. 2, pp. 143-161, June 2002.
187. S. Rizzi, "OLAP Preferences: a Research Agenda", In *Proceedings of of the ACM 10th International Workshop on Data warehousing and OLAP (DOLAP 07),* Lisbon, Portugal, November 9 2007, pp. 99–100, 2007.
188. F. Bentayeb, O. Boussaid, C. Favre, F. Ravat, O. Teste, "Personnalisation dans les entrepôts de données : bilan et perspectives", 5ème Journées francophones sur les Entrepôts de Données et l'Analyse en ligne : EDA'09, Revue des Nouvelles Technologies de l'Information, 2009.
189. A. Ligeza and G. J. Nalepa, "Knowledge Representation with Granular Attributive Logic for XTT-Based Expert Systems", In *Proceedings of the 20th Florida Artificial Intelligence Research Society Conference-FLAIRS,* Key West, Florida, USA, May 7-9 2007, pp. 530-535,AAAI Press, 2007.
190. J. H. Lin, P. J. Haug, "Exploiting missing clinical data in Bayesian network modeling for predicting medical problems", *Journal of Biomedical Informatics*, Vol. 41, No. 1, pp. 1–14, 2008.
191. L. Bradji, M. Boufaida, « Open User Involvement in Data Cleaning for Data Warehouse Quality", Online International Journal of Digital Information and Wireless Communications (IJDIWC), Vol. 1, No. 2, pp. 574-582, Février 2012 (à paraitre).
192. F. Tip, "A Survey of Program Slicing Techniques," *Journal of Programming Languages,* Vol. 3, pp. 121-189, 1995.
193. L. Bradji, M. Boufaida, "Users expectations feedback consistency as a first step for a better Data quality", *Journal of Applied and Theoretical Information Technology,* Vol. 33 No. 1, 2011, pp. 58-68, November 2011. http://www.jatit.org/volumes/Vol33No1/7Vol33No1.pdf
194. C. Clemmen, "Amélioration de la qualité des données dans les entrepôts de données et son impact dans les pratiques organisationnelles - Application aux bases de données administratives de l'ULB", *Rapport technique*, Université de Mons-Hainaut, Mons, Belgium, 2001.

Annexe

Annexe : Code JAVA de l'implémentation du processus ETCTC

Le listing suivant décrit le code java simplifié de l'implémentation du processus d'extraction, monotransformation, monochargement, transformation et chargement des données que nous avons proposé et développé dans les chapitres 3 et 4.

**********/connexion/**********

Cette première partie est le code java qui permet d'ajouter les connexions à tout système de gestion des bases de données via via divers connecteurs : ODBC, JDBC, SQL natif, Fichiers plats ou encore avec des connecteurs spéciaux.

**********Base des données**********

Cette partie du code java permet la création des bases des données, dans notre cas les zones de préparation des données et l'entrepôt des données.

****

```
package globalshemahouse;
import java.awt.*; import javax.swing.*; import java.awt.BorderLayout; import java.awt.Rectangle;
import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.sql.ResultSet;
import java.sql.Statement; import java.sql.SQLException; import java.sql.*;
import java.awt.event.MouseEvent; import java.awt.event.MouseAdapter; import java.awt.Font;
import javax.swing.BorderFactory; import java.awt.Color; import javax.swing.border.TitledBorder;
public class Connexion extends JFrame {
 Connection con;
   JPanel jPanel1 = new JPanel();
   JComboBox jComboBox1 = new JComboBox();
   JLabel jLabel1 = new JLabel();  JLabel jLabel2 = new JLabel();  JLabel jLabel3 = new JLabel();
   JComboBox jComboBox3 = new JComboBox();     JButton jButton1 = new JButton();
   JComboBox jComboBox2 = new JComboBox();          TitledBorder titledBorder1 = new
TitledBorder("");
   public Connexion() {
      try {
         jbInit();
      } catch (Exception exception) {
         exception.printStackTrace();   }   }
   private void jbInit() throws Exception {
      getContentPane().setLayout(null);
      jButton1.addActionListener(new Connexion_jButton1_actionAdapter(this));
      jComboBox1.addActionListener(new Connexion_jComboBox1_actionAdapter(this));
      jComboBox1.addMouseListener(new Connexion_jComboBox1_mouseAdapter(this));
```

```
    jComboBox1.setSelectedItem(this); jComboBox2.setBounds(new Rectangle(80, 107, 195,
22));
    jComboBox2.addActionListener(new Connexion_jComboBox2_actionAdapter(this));
    jComboBox3.addActionListener(new Connexion_jComboBox3_actionAdapter(this));
    jLabel1.setFont(new java.awt.Font("Verdana", Font.BOLD, 11));
    jLabel2.setFont(new java.awt.Font("Verdana", Font.BOLD, 11));
    jLabel3.setFont(new java.awt.Font("Verdana", Font.BOLD, 11));
    jButton1.setFont(new java.awt.Font("Verdana", Font.BOLD | Font.ITALIC, 11));
    jButton1.setBorder(titledBorder1);  jPanel1.setBackground(Color.lightGray);
    jPanel1.setBorder(BorderFactory.createLineBorder(Color.black));
    this.getContentPane().add(jPanel1, null);
    jLabel2.setText("Liste des tables");  jLabel2.setBounds(new Rectangle(71, 85, 184, 21));
    jLabel3.setText("Liste des champs");  jLabel3.setBounds(new Rectangle(70, 163, 229, 23));
    jComboBox3.setBounds(new Rectangle(70, 188, 367, 23));
    jButton1.setBounds(new Rectangle(227, 217, 150, 32));      jButton1.setText("Ajouter
Connexion");
    this.setTitle("Meta-données");          jPanel1.add(jLabel1);          jPanel1.add(jComboBox1);
jPanel1.add(jLabel2);
    jPanel1.add(jLabel3);                jPanel1.add(jComboBox3);                jPanel1.add(jButton1);
jPanel1.add(jComboBox2);
    jLabel1.setText("Liste des connexions disponibles"); jLabel1.setBounds(new Rectangle(70, 17,
264, 28));
    jPanel1.setLayout(null);  jComboBox1.setBounds(new Rectangle(71, 47, 200, 23));
    jPanel1.setBounds(new          Rectangle(10,          29,          444,          279));          try          {
Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection( "jdbc:mysql://localhost/metabase", "root", "");}
catch (SQLException s) {  jLabel1.setText("erreur Driver") ;  }
    //if (!jComboBox1.getSelectedItem().toString().equals(""))
     //    {  ResultSet rsAffichenomBdd; Statement stmtAffichenomBdd;
              try {
                 stmtAffichenomBdd = con.createStatement(ResultSet.
                       TYPE_SCROLL_SENSITIVE,
                   ResultSet.CONCUR_READ_ONLY);
                  rsAffichenomBdd = stmtAffichenomBdd.executeQuery(
                       "SELECT * FROM bdd   ");
                  while (rsAffichenomBdd.next()){
             jComboBox1.addItem(rsAffichenomBdd.getString("nomBdd")) ; }  }
              catch (SQLException n){  jLabel1.setText("erreur codeBdd diagnostic") ;  }
         // }    }
  public    void    jButton1_actionPerformed(ActionEvent    e)    {        DriverGateWay    f=new
DriverGateWay();
```

```
        f.setSize(500,500); f.setLocation(50,300); setVisible(true);  }
  public void jComboBox1_actionPerformed(ActionEvent e) {
    jComboBox2.removeAllItems() ;  ResultSet rsAffichenomtab;  Statement stmtAffichenomtab;
        try {
          stmtAffichenomtab = con.createStatement(ResultSet.  TYPE_SCROLL_SENSITIVE,
              ResultSet.CONCUR_READ_ONLY);          rsAffichenomtab =
stmtAffichenomtab.executeQuery(
              "SELECT * FROM bdd,tab WHERE bdd.codeBdd=tab.codeBdd AND
bdd.nomBdd='"+jComboBox1.getSelectedItem()+"'" );
          while (rsAffichenomtab.next()){
            jComboBox2.addItem(rsAffichenomtab.getString("nomtab") ) ;  } }
       catch (SQLException n){   jLabel1.setText("erreur tab") ;  }
  }
  public void jComboBox1_mouseClicked(MouseEvent e) {  }
  public void jComboBox3_actionPerformed(ActionEvent e) {   }
  public void jComboBox2_actionPerformed(ActionEvent e) {
    jComboBox3.removeAllItems()    ;    ResultSet    rsAffichenomchamp;    Statement
stmtAffichenomchamp;
            try {
              stmtAffichenomchamp = con.createStatement(ResultSet. TYPE_SCROLL_SENSITIVE,
                  ResultSet.CONCUR_READ_ONLY);
               rsAffichenomchamp = stmtAffichenomchamp.executeQuery(
                  "SELECT * FROM tab,champ,bdd,type WHERE tab.codetab=champ.codetab
AND bdd.codebdd=tab.codebdd AND champ.typechamp=type.typechamp AND
tab.nomtab='"+jComboBox2.getSelectedItem()+"'" );
               while (rsAffichenomchamp.next()){
 jComboBox3.addItem(rsAffichenomchamp.getString("nomchamp")+"
|"+rsAffichenomchamp.getString("Taille")+"   "+ rsAffichenomchamp.getString("nomtypechamp")
) ;
          }// AND champ.typechamp=type.typechamp  }
           catch (SQLException n){
        jLabel1.setText("erreur champ") ;  }  } }
class Connexion_jComboBox2_actionAdapter implements ActionListener {  private Connexion
adaptee;
  Connexion_jComboBox2_actionAdapter(Connexion adaptee) {  this.adaptee = adaptee; }
  public void actionPerformed(ActionEvent e) { adaptee.jComboBox2_actionPerformed(e); } }
class Connexion_jComboBox3_actionAdapter implements ActionListener {
  private Connexion adaptee; Connexion_jComboBox3_actionAdapter(Connexion adaptee)
   {this.adaptee = adaptee;  }
  public void actionPerformed(ActionEvent e) {
    adaptee.jComboBox3_actionPerformed(e); } }
```

```
class Connexion_jComboBox1_actionAdapter implements ActionListener {
   private Connexion adaptee; Connexion_jComboBox1_actionAdapter(Connexion adaptee)
      { this.adaptee = adaptee; }
   public void actionPerformed(ActionEvent e) {  adaptee.jComboBox1_actionPerformed(e);  } }
class Connexion_jComboBox1_mouseAdapter extends MouseAdapter {
   private Connexion adaptee;
   Connexion_jComboBox1_mouseAdapter(Connexion adaptee) { this.adaptee = adaptee; }
   public void mouseClicked(MouseEvent e) { adaptee.jComboBox1_mouseClicked(e); } }
class Connexion_jButton1_actionAdapter implements ActionListener {
   private Connexion adaptee; Connexion_jButton1_actionAdapter(Connexion adaptee)
     {  this.adaptee = adaptee; }
   public void actionPerformed(ActionEvent e) {  adaptee.jButton1_actionPerformed(e); } }
**********/connexions/**********
```

Cette deuxième partie du code java permet de fixer la valeur des paramètres et de lancer les connexions et d'afficher des messages d'erreurs lorsqu'un évenement ou une érreur se produit.

```
**************************************************************************
package globalshemahouse; import java.awt.*; import java.awt.BorderLayout; import
java.awt.Dimension;
import java.sql.Connection; import java.sql.DriverManager; import java.sql.SQLException;
import java.sql.Statement; import javax.swing.JFrame; import javax.swing.JPanel;
import javax.swing.JComboBox; import java.awt.CardLayout; import
com.borland.jbcl.layout.XYLayout;
import com.borland.jbcl.layout.*; import java.awt.FlowLayout; import java.awt.Rectangle;
import javax.swing.JToggleButton; import javax.swing.JLabel; import javax.swing.JOptionPane;
import javax.swing.BorderFactory; import javax.swing.JTextField; import
javax.swing.JPasswordField;
import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.awt.Color;
import javax.swing.JButton; import java.awt.*; import java.awt.event.*; import javax.swing.*;
import java.sql.*; import javax.swing.*; import java.awt.Rectangle; import
java.sql.DriverManager;
import java.sql.SQLException;
public class Connexions extends JFrame {
   Connection con; //connection a metabase
   JPanel jPanel1 = new JPanel();     java.awt.GridLayout gridLayout1 = new GridLayout();
   JComboBox jComboBox1 = new JComboBox();     JButton jButton1 = new JButton();
   JLabel jLabel1 = new JLabel();     JComboBox jComboBox2 = new JComboBox();
   JLabel jLabel2 = new JLabel();
   public Connexions() {
```

```
    try {  jbInit();  } catch (Exception exception) { exception.printStackTrace(); }  }
  private void jbInit() throws Exception { getContentPane().setLayout(null);
    this.getContentPane().setBackground(UIManager.getColor("InternalFrame.borderColor"));
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setForeground(UIManager.getColor("ProgressBar.foreground"));
    this.setSize(new Dimension(416, 357)); this.setTitle("Connexions");  this.setVisible(true);
    try { Class.forName("com.mysql.jdbc.Driver"); con = DriverManager.getConnection(
          "jdbc:mysql://localhost/metabase", "root", ""); }
 catch (SQLException s) { }
    Statement stmtAfficheCleBdd; ResultSet rsAfficheCleBdd;
    stmtAfficheCleBdd = con.createStatement(ResultSet. TYPE_SCROLL_SENSITIVE,
                        ResultSet.CONCUR_READ_ONLY);
    rsAfficheCleBdd = stmtAfficheCleBdd.executeQuery( "SELECT * FROM bdd");
    //rsAfficheCleBdd.last();
    while (rsAfficheCleBdd.next()) {
jComboBox1.addItem(rsAfficheCleBdd.getString("nomBdd"));}
    jPanel1.setOpaque(false);  jPanel1.setBounds(new Rectangle(89, 275, 203, 25));
    jPanel1.setLayout(gridLayout1);  jLabel1.setForeground(new Color(215, 98, 138));
    jLabel1.setText("Liste des connexions Disponibles");
    jLabel1.setBounds(new Rectangle(38, 21, 348, 32));
    jComboBox1.addActionListener(new Connexions_jComboBox1_actionAdapter(this));
    jComboBox2.setBackground(Color.orange);  jComboBox2.setBounds(new  Rectangle(44,  176,
291, 25));
    jLabel2.setText("Liste des Tables");  jLabel2.setBounds(new Rectangle(46, 148, 198, 26));
    jButton1.addActionListener(new Connexions_jButton1_actionAdapter(this));
    this.getContentPane().add(jPanel1);  jButton1.setBounds(new Rectangle(205, 263, 126, 40));
    jButton1.setText("Ajouter connexion"); jComboBox1.setForeground(Color.blue);
    jComboBox1.setBorder(BorderFactory.createLineBorder(Color.black));
    jComboBox1.setBounds(new Rectangle(0, 0, 265, 26));
    this.getContentPane().add(jLabel1); this.getContentPane().add(jComboBox1);
    this.getContentPane().add(jComboBox2); this.getContentPane().add(jLabel2);
    this.getContentPane().add(jButton1);  }
  public  void  jButton1_actionPerformed(ActionEvent  e)  {  DriverGateWay  f  =  new
DriverGateWay();
    f.setSize(300, 400);  f.setLocation(100, 100); f.setVisible(true); }
  public void jComboBox1_actionPerformed(ActionEvent e) {
    jLabel1.setText(jComboBox1.getSelectedItem().toString());  ResultSet rsAfficherCleBd;
    Statement stmtAfficheCleBdd;
    try { stmtAfficheCleBdd = con.createStatement(ResultSet.
          TYPE_SCROLL_SENSITIVE,
          ResultSet.CONCUR_READ_ONLY);
```

```
        rsAfficherCleBd = stmtAfficheCleBdd.executeQuery(
            "SELECT tab.nomTab FROM bdd,tab where tab.codeBdd=bdd.CodeBdd and nomBdd='"
+             jComboBox1.getSelectedItem().toString() + "'");
        while (rsAfficherCleBd.next()) {
jComboBox2.addItem(rsAfficherCleBd.getString("nomtab")); } }
 catch (SQLException n) { jLabel1.setText("erreur codeBdd diagnostic"); } }
   class Connexions_jComboBox1_actionAdapter implements ActionListener {
     private Connexions adaptee; Connexions_jComboBox1_actionAdapter(Connexions adaptee)
     { this.adaptee = adaptee; }
     public void actionPerformed(ActionEvent e) { adaptee.jComboBox1_actionPerformed(e); } }
   public void actionPerformed(ActionEvent e) { } }
class Connexions_jButton1_actionAdapter implements ActionListener { private Connexions
adaptee;
   Connexions_jButton1_actionAdapter(Connexions adaptee) { this.adaptee = adaptee; }
   public void actionPerformed(ActionEvent e) { adaptee.jButton1_actionPerformed(e); } }

package globalshemahouse; import java.awt.BorderLayout; import java.awt.Dimension;
import java.sql.Connection; import java.sql.DriverManager; import java.sql.SQLException;
import java.sql.Statement; import javax.swing.JFrame; import javax.swing.JPanel;
import javax.swing.JComboBox; import java.awt.CardLayout; import
com.borland.jbcl.layout.XYLayout;
import com.borland.jbcl.layout.*; import java.awt.FlowLayout; import java.awt.Rectangle;
import javax.swing.JToggleButton; import javax.swing.JLabel; import javax.swing.JOptionPane;
import javax.swing.BorderFactory; import javax.swing.JTextField; import
javax.swing.JPasswordField;
import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.awt.Color;
import javax.swing.JButton; import java.awt.*; import java.awt.event.*; import javax.swing.*;
import java.sql.*; import java.awt.Font;
/**
 * <p>Title: </p> *
 * <p>Description: </p>  *
 * <p>Copyright: Copyright (c) 2009</p> *
 * <p>Company: </p>  *
 * @author not attributable  * @version 1.0  */
public class DriverGateWay extends JFrame { Connection con;//connection de bdd sélectionnée
   Connection conMySql;//connection a metabase
   ResultSet rsAfficheCleBdd; //
   private Statement statement;  JPanel contentPane;  JComboBox jComboBox1 = new
JComboBox();
   JLabel jLabel1 = new JLabel(); JPanel jPanel1 = new JPanel();  JTextField jTextField1 = new
JTextField();
```

```
  JTextField jTextField2 = new JTextField();  JPasswordField jPasswordField1 = new
JPasswordField();
  JLabel jLabel2 = new JLabel();  JLabel jLabel3 = new JLabel();  JLabel jLabel4 = new JLabel();
  JLabel jLabel6 = new JLabel();  JButton jButton1 = new JButton();  JLabel jLabel7 = new JLabel();
  JComboBox jComboBox2 = new JComboBox();  JComboBox jComboBox3 = new JComboBox();
  JScrollPane jScrollPane1 = new JScrollPane();  JTextArea resulta = new JTextArea();
  public DriverGateWay() {
    try {  setDefaultCloseOperation(EXIT_ON_CLOSE);  jbInit(); }
    catch (Exception exception) {  exception.printStackTrace(); }  }
  /**
   * Component initialization.  *
   * @throws java.lang.Exception */
  private void jbInit() throws Exception { contentPane = (JPanel) getContentPane();
    this.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE); setSize(new Dimension(400, 413));
    setTitle("DriverGateWay"); contentPane.setLayout(null);
    jComboBox1.setBounds(new Rectangle(6, 68, 377, 22));
    jComboBox1.addActionListener(new DriverGateWay_jComboBox1_actionAdapter(this));
    jLabel1.setText("Liste de Drivers"); jLabel1.setBounds(new Rectangle(32, 43, 212, 14));
    jPanel1.setBackground(Color.lightGray);
jPanel1.setBorder(BorderFactory.createRaisedBevelBorder());
    jPanel1.setBounds(new Rectangle(6, 101, 377, 134));  jPanel1.setLayout(null);
    jTextField1.setBounds(new Rectangle(95, 26, 271, 22));
    jTextField2.setBounds(new Rectangle(95, 51, 152, 21));
    jPasswordField1.setBounds(new Rectangle(95, 75, 152, 22));  jLabel2.setText("URL:");
    jLabel2.setBounds(new Rectangle(11, 29, 34, 14)); jLabel3.setText("Nom Utilsateur:");
    jLabel3.setBounds(new Rectangle(11, 55, 104, 14));  jLabel4.setText("Mot de passe:");
    jLabel4.setBounds(new Rectangle(11, 81, 73, 14));  jLabel6.setText("Errors Results:");
    jLabel6.setBounds(new Rectangle(49, 271, 183, 14));
    jButton1.setBounds(new Rectangle(243, 241, 140, 41));
    jButton1.setFont(new java.awt.Font("Verdana", Font.BOLD | Font.ITALIC, 11));
    jButton1.setText("Connect");
    jButton1.addActionListener(new DriverGateWay_jButton1_actionAdapter(this));
    jLabel7.setForeground(new Color(210, 0, 0));  jLabel7.setBounds(new Rectangle(50, 315, 284,
14));
    jComboBox2.setBounds(new Rectangle(8, 20, 384, 22));
    jComboBox3.setBounds(new Rectangle(55, 239, 121, 19));
    contentPane.setBackground(UIManager.getColor("InternalFrame.activeTitleGradient"));
    jScrollPane1.setBounds(new Rectangle(26, 304, 118, 80));
    resulta.setFont(new java.awt.Font("Dialog", Font.BOLD, 9));
    resulta.setText("Bienvenue");  resulta.setWrapStyleWord(true);
```

```
      jPanel1.add(jTextField2);          jPanel1.add(jPasswordField1);          jPanel1.add(jTextField1);
jPanel1.add(jLabel2);
      jPanel1.add(jLabel3);                    jPanel1.add(jLabel4);          contentPane.add(jLabel1);
contentPane.add(jPanel1);
      contentPane.add(jComboBox1, null); contentPane.add(jButton1); contentPane.add(jLabel6);
      contentPane.add(jLabel7); contentPane.add(jComboBox2); contentPane.add(jComboBox3);
      contentPane.add(jScrollPane1); jScrollPane1.getViewport().add(resulta);
      jComboBox1.addItem("com.mysql.jdbc.Driver") ;
      jComboBox1.addItem("sun.jdbc.odbc.JdbcOdbcDriver") ;
      jComboBox1.addItem("com.borland.datastore.jdbc.DataStoreDriver") ;
      jComboBox1.addItem("interbase.interclient.Driver") ;
      jComboBox1.addItem("oracle.jdbc.driver.OracleDriver") ;
      try{  Class.forName("com.mysql.jdbc.Driver");
conMySql=DriverManager.getConnection("jdbc:mysql://localhost/metabase","root",""); }
   catch (SQLException s){  jLabel6.setText("mysql error"); }  }
   public void jComboBox1_actionPerformed(ActionEvent e) {
try{  Class.forName(jComboBox1.getSelectedItem().toString()); }
catch (ClassNotFoundException s){ }
//Connection a la base de données
switch                              (jComboBox1.getSelectedIndex()){                              case
0:jTextField1.setText("jdbc:mysql://hostname/MyBase");
    jTextField2.setText("root") ;  jPasswordField1.setText("") ;  break;
case 1:jTextField1.setText("jdbc:odbc:odbcDataSource");
    jTextField2.setText("sysdba") ;   jPasswordField1.setText("masterkey") ;  break;
     case 2:jTextField1.setText("jdbc:borland:dslocal:directoryAndFile.jds");break;
      case 3:jTextField1.setText("jdbc:interbase://hostname/directoryAndFile.gdb");break;
      case 4:jTextField1.setText("jdbc:oracle:thin:@hostname:1521:MyBase");break;
      default : jTextField1.setText("jdbc:mysql://hostname/MyBase"); }
String DBurl=jTextField1.getText(); String User=jTextField2.getText();
String Password=jPasswordField1.getText() ;    }
   public void jButton1_actionPerformed(ActionEvent e) {
//connection a la base
      try{ try{
con=DriverManager.getConnection(jTextField1.getText(),jTextField2.getText(),jPasswordField1.ge
tText());} catch(SQLException f)    {   }
      try{  Statement statement; Statement  stmt=null; stmt=conMySql.createStatement() ;
   stmt.executeUpdate("INSERT INTO bdd VALUES ("+0+",'"+jTextField1.getText()+"')"); }
catch (SQLException m)
{ jLabel7.setText("erreur lors de l'insertion dans la tabe Bdd ") ; }
// définition dune cle pour tab dans la variable stepTab
// capture de la clé de bdd
```

```
   Statement stmtAfficheCleBdd;
   try{ stmtAfficheCleBdd = conMySql.createStatement(ResultSet.
                                TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);
   rsAfficheCleBdd = stmtAfficheCleBdd.executeQuery(
        "SELECT codeBdd FROM bdd   ");  rsAfficheCleBdd.last();
   //jLabel7.setText(rsAfficheCleBdd.getString("codeBdd"));
   } catch (SQLException n){  jLabel7.setText("erreur codeBdd diagnostic") ; }
    DatabaseMetaData dmd=con.getMetaData() ; Statement stmtAfficheCletab; ResultSet
rsAfficheCletab ;
stmtAfficheCletab=conMySql.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY) ; ResultSet tables=dmd.getTables(con.getCatalog()
,null,"%",null);
       Statement stmtTabInsert;  stmtTabInsert=conMySql.createStatement() ;
          while (tables.next() ){  stmtTabInsert.executeUpdate("INSERT INTO tab VALUES (" +0+ ",'" +
            tables.getString("TABLE_NAME") + "','" +   tables.getString("TABLE_TYPE")  + "'," +
                        rsAfficheCleBdd.getString("codeBdd") +")");
stmtAfficheCletab=conMySql.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_READ_ONLY) ;
                        rsAfficheCletab=stmtAfficheCletab.executeQuery("SELECT * FROM tab") ;
                       rsAfficheCletab.last(); DatabaseMetaData dmdcol = con.getMetaData();
//récupération des informations
      String nomDeLaTable = rsAfficheCletab.getString("nomTab");
      ResultSet resultatcol = dmdcol.getColumns(con.getCatalog(),null,nomDeLaTable, "%");
//affichage des informations
      ResultSetMetaData rsmdcol = resultatcol.getMetaData();  Statement stmtcolInsert;
      stmtcolInsert=conMySql.createStatement();
// trouver les cles primaires
      //DatabaseMetaData metadata = con.getMetaData();
              //  ResultSet clefs = dmd.getPrimaryKeys(con.getCatalog(),null,"Patient");
          //  while(clefs.next()){
          //  String nomColonne = clefs.getString("COLUMN_NAME");
          //  jComboBox2.addItem(nomColonne) ; //}
      while(resultatcol.next()){
            stmtcolInsert.executeUpdate("INSERT INTO champ VALUES
(0,'"+resultatcol.getString("COLUMN_NAME") +"','"+resultatcol.getString("DATA_TYPE")
+"','"+2+"','"+
      3+"',"+ 1+","+ rsAfficheCletab.getString("codetab")+","
+resultatcol.getString("COLUMN_SIZE") +")");
   } } }
      catch(SQLException s){ }
        this.setVisible(false); } }
```

```
class DriverGateWay_jButton1_actionAdapter implements ActionListener { private
DriverGateWay adaptee;
   DriverGateWay_jButton1_actionAdapter(DriverGateWay adaptee) { this.adaptee = adaptee; }
   public void actionPerformed(ActionEvent e) { adaptee.jButton1_actionPerformed(e); } }
class DriverGateWay_jComboBox1_actionAdapter implements ActionListener {
   private DriverGateWay adaptee; DriverGateWay_jComboBox1_actionAdapter(DriverGateWay
adaptee)
        {  this.adaptee = adaptee; }
   public void actionPerformed(ActionEvent e) {  adaptee.jComboBox1_actionPerformed(e); } }

/* * @(#)EventDispatchThread.java       1.54 05/03/03
 * * Copyright 2005 Sun Microsystems, Inc. All rights reserved.
 * SUN PROPRIETARY/CONFIDENTIAL. Use is subject to license terms. */
package java.awt; import java.awt.event.InputEvent; import java.awt.event.MouseEvent;
import java.awt.event.ActionEvent; import java.awt.event.WindowEvent; import
java.lang.reflect.Method;
import java.security.AccessController; import sun.security.action.GetPropertyAction;
import sun.awt.DebugHelper; import sun.awt.AWTAutoShutdown; import sun.awt.SunToolkit;
import sun.awt.dnd.SunDragSourceContextPeer;
/**
 * EventDispatchThread is a package-private AWT class which takes
 * events off the EventQueue and dispatches them to the appropriate
 * AWT components.
 *
 * The Thread starts a "permanent" event pump with a call to
 * pumpEvents(Conditional) in its run() method. Event handlers can choose to
 * block this event pump at any time, but should start a new pump (<b>not</b>
 * a new EventDispatchThread) by again calling pumpEvents(Conditional). This
 * secondary event pump will exit automatically as soon as the Condtional
 * evaluate()s to false and an additional Event is pumped and dispatched.
 *  * @author Tom Ball
 * @author Amy Fowler
 * @author Fred Ecks
 * @author David Mendenhall
 *   * @version 1.54, 03/03/05
 * @since 1.1  */
class EventDispatchThread extends Thread {
   private static final DebugHelper dbg = DebugHelper.create(EventDispatchThread.class);
   private EventQueue theQueue;     private boolean doDispatch = true;
   private static final int ANY_EVENT = -1;
   EventDispatchThread(ThreadGroup group, String name, EventQueue queue)
```

```
{  super(group, name);  theQueue = queue;  }
   void stopDispatchingImpl(boolean wait) {
      // Note: We stop dispatching via a flag rather than using
      // Thread.interrupt() because we can't guarantee that the wait()
      // we interrupt will be EventQueue.getNextEvent()'s.  -fredx 8-11-98
      StopDispatchEvent stopEvent = new StopDispatchEvent();
      // wait for the dispatcher to complete
      if (Thread.currentThread() != this) {
         // fix 4122683, 4128923
         // Post an empty event to ensure getNextEvent is unblocked  //
         // We have to use postEventPrivate instead of postEvent because
         // EventQueue.pop calls EventDispatchThread.stopDispatching.
         // Calling SunToolkit.flushPendingEvents in this case could
         // lead to deadlock.
         theQueue.postEventPrivate(stopEvent);
         if (wait) { try { join();
            } catch(InterruptedException e) { }  }  } else {  stopEvent.dispatch();   }
      synchronized (theQueue) {
         if (theQueue.getDispatchThread() == this) { theQueue.detachDispatchThread(); } } }
   public void stopDispatching() { stopDispatchingImpl(true);  }
   public void stopDispatchingLater() { stopDispatchingImpl(false);  }
   class StopDispatchEvent extends AWTEvent implements ActiveEvent {
      public StopDispatchEvent() {  super(EventDispatchThread.this,0); }
      public void dispatch() { doDispatch = false;  } }
   public void run() {
          try {  pumpEvents(new Conditional() { public boolean evaluate() { return true; }   });}
 finally {   /*
               * This synchronized block is to secure that the event dispatch
               * thread won't die in the middle of posting a new event to the
               * associated event queue. It is important because we notify
               * that the event dispatch thread is busy after posting a new event
               * to its queue, so the EventQueue.dispatchThread reference must
               * be valid at that point.  */
              synchronized (theQueue) {
             if (theQueue.getDispatchThread() == this) {  theQueue.detachDispatchThread();  }
             /** Event dispatch thread dies in case of an uncaught exception.
              * A new event dispatch thread for this queue will be started
              * only if a new event is posted to it. In case if no more
              * events are posted after this thread died all events that
              * currently are in the queue will never be dispatched. */
             /* * Fix for 4648733. Check both the associated java event
```

```
            * queue and the PostEventQueue.
            */
           if (theQueue.peekEvent() != null || !SunToolkit.isPostEventQueueEmpty())
{  theQueue.initDispatchThread(); }
                    AWTAutoShutdown.getInstance().notifyThreadFree(this); } } }
   void pumpEvents(Conditional cond) { pumpEvents(ANY_EVENT, cond);  }
   void pumpEventsForHierarchy(Conditional cond, Component modalComponent) {
      pumpEventsForHierarchy(ANY_EVENT, cond, modalComponent); }
   void pumpEvents(int id, Conditional cond) { pumpEventsForHierarchy(id, cond, null);  }
   void pumpEventsForHierarchy(int id, Conditional cond, Component modalComponent)
   {   while (doDispatch && cond.evaluate()) {
         if (isInterrupted() || !pumpOneEventForHierarchy(id, modalComponent)) { doDispatch =
false; } } }
   boolean checkMouseEventForModalJInternalFrame(MouseEvent me, Component modalComp)
{
      // Check if the MouseEvent is targeted to the HW parent of the
      // LW component, if so, then return true. The job of distinguishing
      // between the LW components is done by the LW dispatcher.
      if (modalComp instanceof javax.swing.JInternalFrame) {
         Container c; synchronized (modalComp.getTreeLock()) {
            c = ((Container)modalComp).getHeavyweightContainer(); }
         if (me.getSource() == c) return true; } return false; }
   boolean pumpOneEventForHierarchy(int id, Component modalComponent) {
      try {
         AWTEvent event; boolean eventOK;
         do {
               event = (id == ANY_EVENT)
                     ? theQueue.getNextEvent() : theQueue.getNextEvent(id);
            eventOK = true;
            if (modalComponent != null) {
               /* * filter out MouseEvent and ActionEvent that's outside
                * the modalComponent hierarchy.
                * KeyEvent is handled by using enqueueKeyEvent
                * in Dialog.show  */
               int eventID = event.getID();
               if (((eventID >= MouseEvent.MOUSE_FIRST && eventID <= MouseEvent.MOUSE_LAST)
&&
                     !(checkMouseEventForModalJInternalFrame((MouseEvent)
                        event, modalComponent))) ||
                     (eventID >= ActionEvent.ACTION_FIRST &&
```

```
                eventID <= ActionEvent.ACTION_LAST) || eventID ==
WindowEvent.WINDOW_CLOSING) {
            Object o = event.getSource();
            if (o instanceof sun.awt.ModalExclude) {
              // Exclude this object from modality and
              // continue to pump it's events.  } else if (o instanceof Component) {
              Component c = (Component) o; boolean modalExcluded = false;
              if (modalComponent instanceof Container) {
                while (c != modalComponent && c != null) {
                  if ((c instanceof Window) &&
                    (sun.awt.SunToolkit.isModalExcluded((Window)c))) {
                      // Exclude this window and all its children from
                      //  modality and continue to pump it's events.
                    modalExcluded = true;  break;  }
                  c = c.getParent();  }   }
              if (!modalExcluded && (c != modalComponent)) { eventOK = false; } } } }
          eventOK = eventOK && SunDragSourceContextPeer.checkEvent(event);
          if (!eventOK) { event.consume(); } }
         while (eventOK == false);
           if ( dbg.on ) dbg.println("Dispatching: "+event); theQueue.dispatchEvent(event); return
true;   }
 catch (ThreadDeath death) {  return false;   } catch (InterruptedException interruptedException) {
        return false; // AppContext.dispose() interrupts all
                // Threads in the AppContext
           // Can get and throw only unchecked exceptions
     } catch (RuntimeException e) {  processException(e, modalComponent != null); } catch (Error
e)
          {  processException(e, modalComponent != null);  } return true;  }
   private void processException(Throwable e, boolean isModal) {
     if (!handleException(e)) {
       // See bug ID 4499199.
       // If we are in a modal dialog, we cannot throw
       // an exception for the ThreadGroup to handle (as added
       // in RFE 4063022).  If we did, the message pump of
       // the modal dialog would be interrupted.
       // We instead choose to handle the exception ourselves.
       // It may be useful to add either a runtime flag or API
       // later if someone would like to instead dispose the
       // dialog and allow the thread group to handle it.
       if (isModal) {
         System.err.println(
```

```
                "Exception occurred during event dispatching:");  e.printStackTrace(); }
               else if (e instanceof RuntimeException) { throw (RuntimeException)e; }
else if (e instanceof Error) { throw (Error)e; } } }
  private static final String handlerPropName = "sun.awt.exception.handler";
  private static String handlerClassName = null;
  private static String NO_HANDLER = new String();
  /**
   * Handles an exception thrown in the event-dispatch thread.
   *
   * <p> If the system property "sun.awt.exception.handler" is defined, then
   * when this method is invoked it will attempt to do the following:
   *
   * <ol>
   * <li> Load the class named by the value of that property, using the
   *      current thread's context class loader,
   * <li> Instantiate that class using its zero-argument constructor,
   * <li> Find the resulting handler object's <tt>public void handle</tt>
   *      method, which should take a single argument of type
   *      <tt>Throwable</tt>, and
   * <li> Invoke the handler's <tt>handle</tt> method, passing it the
   *      <tt>thrown</tt> argument that was passed to this method.
   * </ol>
   *
   * If any of the first three steps fail then this method will return
   * <tt>false</tt> and all following invocations of this method will return
   * <tt>false</tt> immediately.  An exception thrown by the handler object's
   * <tt>handle</tt> will be caught, and will cause this method to return
   * <tt>false</tt>.  If the handler's <tt>handle</tt> method is successfully
   * invoked, then this method will return <tt>true</tt>.  This method will
   * never throw any sort of exception.
   *
   * <p> <i>Note:</i> This method is a temporary hack to work around the
   * absence of a real API that provides the ability to replace the
   * event-dispatch thread.  The magic "sun.awt.exception.handler" property
   * <i>will be removed</i> in a future release.
   *
   * @param  thrown  The Throwable that was thrown in the event-dispatch
   *                 thread
   *
   * @return  <tt>false</tt> if any of the above steps failed, otherwise
   *          <tt>true</tt>
```

```
   */
   private boolean handleException(Throwable thrown) {
      try { if (handlerClassName == NO_HANDLER) { return false;   /* Already tried, and failed */ }
         /* Look up the class name */
         if (handlerClassName == null) { handlerClassName = ((String) AccessController.doPrivileged(
               new GetPropertyAction(handlerPropName)));
            if (handlerClassName == null) {  handlerClassName = NO_HANDLER; /* Do not try this
again */
               return false; } }
         /* Load the class, instantiate it, and find its handle method */
         Method m; Object h;
         try {
            ClassLoader cl = Thread.currentThread().getContextClassLoader();
            Class c = Class.forName(handlerClassName, true, cl);
 m = c.getMethod("handle", new Class[] { Throwable.class }); h = c.newInstance(); }
 catch (Throwable x) {  handlerClassName = NO_HANDLER; /* Do not try this again */
            return false; }
         /* Finally, invoke the handler */
         m.invoke(h, new Object[] { thrown });  } catch (Throwable x) { return false; } return true;  }
   boolean isDispatching(EventQueue eq) { return theQueue.equals(eq);  }
   EventQueue getEventQueue() { return theQueue; } }
**********/globahouse/**********
package globalshemahouse; import java.awt.Toolkit; import javax.swing.SwingUtilities;
import javax.swing.UIManager; import java.awt.Dimension;
/**
 * <p>Title: </p>  *
 * <p>Description: </p>  *
 * <p>Copyright: Copyright (c) 2009</p>  *
 * <p>Company: </p>  *
 * @author not attributable
 * @version 1.0  */
public class GlobaHouse { boolean packFrame = false;
   /**
    * Construct and show the application.  */
   public GlobaHouse() {
      Connexion frame = new Connexion();
      // Validate frames that have preset sizes
      // Pack frames that have useful preferred size info, e.g. from their layout
      if (packFrame) { frame.pack(); } else { frame.validate(); }
      // Center the window
      Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
```

```
      Dimension frameSize = frame.getSize();
      if (frameSize.height > screenSize.height) { frameSize.height = screenSize.height;  }
      if (frameSize.width > screenSize.width) { frameSize.width = screenSize.width;   }
      frame.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height -
frameSize.height) / 2);
      frame.setVisible(true);  }
   /**
    * Application entry point  *
    * @param args String[]  */
   public static void main(String[] args) { SwingUtilities.invokeLater(new Runnable()
 { public void run() {
            try { UIManager.setLookAndFeel(UIManager.  getSystemLookAndFeelClassName());}
            catch (Exception exception) { exception.printStackTrace(); } new GlobaHouse(); } }); } }
**********************************/tables/**********************************
```

Cette partie du code java permet la création des tables dans les bases des données.

```
**************************************************************************
package globalshemahouse; import java.awt.*; import javax.swing.*; import java.awt.Rectangle;
import java.sql.Connection; import java.sql.DriverManager; import java.sql.ResultSet;
import java.sql.Statement; import java.sql.SQLException; public class Tables extends JFrame {
   Connection con; //connection a metabase
   public Tables() {
      try {  jbInit(); } catch (Exception exception) { exception.printStackTrace(); }   }
   private void jbInit() throws Exception {  getContentPane().setLayout(null);
      jPanel1.setBounds(new Rectangle(72, 59, 313, 205));  jPanel1.setLayout(null);
      jComboBox1.setBounds(new Rectangle(7, 5, 279, 22)); jLabel1.setText("jLabel1");
      jLabel1.setBounds(new Rectangle(71, 134, 240, 22));  this.getContentPane().add(jPanel1);
      this.getContentPane().add(jLabel1); jPanel1.add(jComboBox1);
      try { Class.forName("com.mysql.jdbc.Driver");
               con = DriverManager.getConnection("jdbc:mysql://localhost/metabase", "root", "");}
  catch (SQLException s) { }
            Statement stmtAfficheCleBdd; ResultSet rsAfficheCleBdd;
            stmtAfficheCleBdd = con.createStatement(ResultSet.
                                    TYPE_SCROLL_SENSITIVE,  ResultSet.CONCUR_READ_ONLY);
         rsAfficheCleBdd = stmtAfficheCleBdd.executeQuery( "SELECT * FROM tab ");
         //rsAfficheCleBdd.last();
      while  (rsAfficheCleBdd.next()){ jComboBox1.addItem(rsAfficheCleBdd.getString("nomtab") );
} }
   JPanel jPanel1 = new JPanel(); JComboBox jComboBox1 = new JComboBox(); JLabel jLabel1 =
new JLabel();
}
```

Résumé

L'extraction des connaissances à partir des données (ECD) et L'entreposage des données (ED) sont les technologies le plus utilisable dans le contexte de l'aide à la décision. La qualité des données et des connaissances est la clé de voûte de la réussite de ces technologies. L'ECD et l'ED sont traités comme étant deux processus indépendants malgré leur synergie et leur complémentarité. De ce fait, ce travail propose une solution pour la gestion totale et continue de la qualité des données et des connaissances en adaptant le processus ECD et l'ED. Nous dotons le processus l'ECD par un système d'entreposage des règles. Ce système est basé sur un nouveau formalisme de représentation des règles. La puissance de ce formalisme est de permettre la représentation et la gestion de tout type de règle et de leur qualité. Nous avons intégré un système d'élicitation des connaissances spécifique à l'ECD afin d'impliquer l'utilisateur directement dans l'amélioration de la qualité. Un autre avantage de ce travail est qu'il permet le nettoyage des données dans les sources des données originales. Pour valider nos différentes contributions, nous avons réalisé une expérimentation au sein d'un établissement sanitaire.

Mots clés: Qualité, Donnée, Connaissance, Règle, Entrepôt, ECD, Elicitation

Abstract

Knowledge Discovery from Data (KDD) and Data Warehouse (DW) are fundamentals tools to Decision Support System (DSS). The data and knowledge quality is essential to the success of these tools. However, research works deals with these process without take into account their synergy and complementarily. Then, this work proposes a system for the total and continuous management of the quality of data and knowledge by adapting the KDD and DW process. This system considers the DW as the first point of the KDD. For this, we have equipped the KDD by a system of rules warehousing and after we have adapt the DW which based in the Extract, Transformation and Load (ETL) processes. The rule warehousing is based in a strong and unified rule form inspired from Logic. This rule form allows the management of each type of rule and its quality. In this work, we proposed a process that allows the elicitation of knowledge in order to involve the user in the management of the quality of data. We have proposed different processes in order to improve the quality of knowledge and DW and also the quality of operational data sources. To validate our contributions we have realized experimentation in the health sector.

Keywords: Quality, Data, Knowledge, Rule, Warehouse, KDD, Elicitation

ملخص

إن إستخراج المعارف من المعطيات و تجميع المعطيات من أهم الأدوات التكنولوجيا في ميدان أنظمة القرار. و نوعية المعطيات و المعارف و دقتها هي التي تحدد صحة و دقة القرار. غير أن الأعمال العلمية الحالية لا تأخذ بعين الإعتبار إرتباط و تكامل هذه التكنولوجيات. من هنا نقترح في هذه الرسالة عملا علميا لتحسين نوعية المعطيات و المعارف و بالتالي التحصل على قرارات صائبة و دقيقة.