Entrepôts et analyse en ligne de données complexes centrés utilisateur : un nouveau défi

Fadila Bentayeb

Université de Lyon - Lumière Lyon 2

Habilitation à Diriger des Recherches

SPÉCIALITÉ : INFORMATIQUE

présentée par

Fadila Bentayeb

Maître de conférences

Entrepôts et analyse en ligne de données complexes centrés utilisateur : un nouveau défi

Soutenue publiquement le 24 novembre 2011 devant le jury :

Mme Rokia Missaoui	Professeur, Université du Québec en Outaouais	Rapportrice
Mme Anne Doucet	Professeur, Université Pierre et Marie Curie	Rapportrice
M. Gilles Zurfluh	Professeur, Université Toulouse 1 Capitole	Rapporteur
Mme Corine Cauvet	Professeur, Université Aix-Marseille 3	Examinatrice
Mme Danielle Boulanger	Professeur, Université Jean Moulin Lyon 3	Examinatrice
M. Stefano Spaccapietra	Professeur, École Polytechnique Fédérale de Lausanne	Examinateur
M. Abdelkader Zighed	Professeur, Université Lumière Lyon 2	Examinateur

ERIC - Equipe de Recherche en Ingénierie des Connaissances - EA 3083
École Doctorale Informatique et Mathématiques

Préambule

L'objet de ce document est de retracer mon parcours d'enseignant-chercheur depuis mon recrutement en tant que maître de conférences à l'Université Lumière Lyon 2 en 2001, jusqu'à la fin de l'année universitaire 2010-2011. Durant toute cette période, j'ai bénéficié d'une grande liberté d'entreprendre au sein de l'Université Lyon 2 ainsi que d'un contexte de recherche très favorable au sein du laboratoire ERIC (Equipe de Recherche en Ingénierie de Connaissances). Comme l'annonce le titre de ce document : "Entrepôts et analyse en ligne de données complexes centrés utilisateur : un nouveau défi", mes travaux de recherche trouvent leurs fondements dans la conception et l'ingénierie des systèmes d'aide à la décision et visent à élaborer/valider des modèles d'entrepôts de données qui replacent l'utilisateur au cœur du système décisionnel. Ce mémoire décrit mes activités de recherche et d'animation de la recherche depuis ces dix dernières années. Les travaux décrits ici ont été menés au laboratoire ERIC (Université Lyon 2) dirigé par Monsieur Djamel Zighed au sein de l'axe BDD (Bases de Données Décisionnelles) devenu par la suite ENA-DC (ENtrepôts et Analyse en ligne de Données Complexes). Mes activités de recherche consistent depuis 2001 à étudier les différentes façons d'entreposer et d'analyser des données complexes. Les problèmes d'entreposage s'étendent à tous les types de données (structurés ou non) et dans tous les domaines d'application. Il s'agit en particulier d'intégration de données, de modélisation et d'analyse en ligne de ces données. L'originalité de mes travaux de recherche a consisté à montrer qu'il est pertinent d'intégrer la sémantique dans tout le processus d'entreposage, soit en invitant l'utilisateur à exprimer ses propres connaissances métier, soit en utilisant les méthodes de fouille de données pour extraire des connaissances cachées. Ces travaux ont été proposés pour répondre aux nouveaux défis liés aux entrepôts de données, notamment pour la prise en compte de l'utilisateur d'une part, et d'autre part pour la prise en compte de données complexes. Dans cet objectif, j'ai mené, dirigé, encadré et valorisé à travers des collaborations scientifiques et industrielles des travaux de recherche dont je rapporte ici une synthèse ainsi que les développements futurs. Dans ce mémoire, je présente mes principaux travaux dans un ordre thématique plutôt que chronologique.

Résumé

Les entrepôts de données répondent à un réel besoin en matière d'accès à l'information résumée. Cependant, en suivant le processus classique d'entreposage et d'analyse en ligne (OLAP) de données, les systèmes d'information décisionnels (SID) exploitent très peu le contenu informationnel des données. Alors même que les SID sont censés être centrés utilisateur, l'OLAP classique ne dispose pas d'outils permettant de guider l'utilisateur vers les faits les plus intéressants du cube. La prise en compte de l'utilisateur dans les SID est une problématique nouvelle, connue sous le nom de *personnalisation*, qui pose plusieurs enjeux peu ou pas étudiés.

Le travail présenté dans ce mémoire vise à proposer des solutions innovantes dans le domaine de la personnalisation dans les entrepôts de données complexes. L'originalité de nos travaux de recherche a consisté à montrer qu'il est pertinent d'intégrer la sémantique dans tout le processus d'entreposage, soit en invitant l'utilisateur à exprimer ses propres connaissances métier, soit en utilisant les méthodes de fouille de données pour extraire des connaissances cachées.

En s'appuyant sur l'intuition que des connaissances sur le métier, sur les données entreposées et leur usage (requêtes) peuvent contribuer à aider l'utilisateur dans son exploration et sa navigation dans les données, nous avons proposé une première approche de personnalisation basée sur les connaissances explicites des utilisateurs. En empruntant le concept d'évolution de schéma, nous avons relâché la contrainte du schéma fixe de l'entrepôt, pour permettre d'ajouter ou de supprimer un niveau de hiérarchie dans une dimension. Ces travaux ont été étendus pour recommander à l'utilisateur des hiérarchies de dimension nouvelles basées sur la découverte de nouvelles structures naturelles grâce aux principes d'une méthode de classification (K-means). Nous avons par ailleurs développé la fouille en ligne en s'appuyant uniquement sur les outils offerts par les systèmes de gestion de bases de données (SGBD). La fouille en ligne permet d'étendre les capacités analytiques des SGBD, support des entrepôts de données, de l'OLAP vers une analyse structurante, explicative et prédictive ; et venir en appui à la personnalisation.

Afin de prendre en compte à la fois l'évolution des données et celle des besoins tout en garantissant l'intégration structurelle et sémantique des données, nous avons proposé une approche d'analyse en ligne à la demande, qui s'appuie sur un système de médiation à base d'ontologies. Par ailleurs, nous avons proposé un modèle multidimensionnel d'objets complexes basé sur le paradigme objet qui permet de représenter les objets de l'univers de façon plus naturelle et de capter la sémantique qu'ils véhiculent. Un opérateur de projection cubique est alors proposé pour permettre à l'utilisateur de créer des cubes d'objets complexes personnalisés.

Toutes nos solutions ont été développées et testées dans le contexte des entrepôts de données relationnels et/ou XML.

Mots-clés : Données complexes, Entrepôt de données, Évolution de schéma, Fouille de données en ligne, Hiérarchie de dimension, Mise à jour, Objet complexe, Ontologie, OLAP, Performance, Personnalisation, Recommandation, Sémantique, Utilisateur.

Abstract

The main goal of data warehouses is to facilitate decision making. In order to satisfy the whole analysis needs of the majority of the users, a promising issue consists in integrating a personalization process for OLAP analysis by taking into account user's own knowledge, preferences, needs,... In other words, the objective is to provide a user-centric decision-making system. In this thesis, we aim at proposing novel solutions for user-centric data warehouses.

First, we have designed an original approach to achieve a user-driven model evolution that provides answers to personalized analysis needs. Our key idea consists in generating new anlysis axes based on users' knowledge by dynamically extending dimension hierachies or creating new ones. Moreover, to help users to find non expected and pertinent aggregates expressing deep relations within a data warehouse, we propose to combine data mining techniques with OLAP. We have more precisely defined a new roll-up operator based on the K-means clustering method.

In addition, we have proposed a framework for mining large databases without size limit in very acceptable processing times. For this end, we have integrated data mining techniques within database management systems (DBMSs) by exploiting only their features. This helps to facilitate the extension of the capabilities of OLAP towards explicative and predictive analysis.

To take into account both data sources changes and users requirements evolution, we have designed a user-centric approach for producing OLAP data cubes on the fly. This is based on a mediation system using ontologies. To generate the global merged ontology from local ontologies, we use the agglomerative hierarchical clustering method.

Int the other hand, to warehouse complex data, we have designed a complex object-based multidimensional model. This is defined at two layers : (1) the package diagram layer which describes complex objects and their complex relationships and (2) the class diagram layer which provides details about the structure of each complex object. From the complex object-based multidimensional model, personalized complex object cubes can be derived.

Eventually, for evaluationg our user-centric data warehouses solutions, we have implemented and carried out some experiments in both the contexts of relational and XML data warehouses.

Keywords : Complex object, Data warehouses, Dimension hierarchy, OLAP, On-line data mining, Ontologies, Performance, Personnalization, Recommendation, Schema evolution, Semantics, Updates, User-centric data warehouses.

À Elyas

Remerciements

J'exprime tout d'abord mes remerciements à Madame Rokia Missaoui, à Madame Anne Doucet et à Monsieur Gilles Zurfluh qui ont accepté d'être les rapporteurs de mon travail. Je suis honorée qu'ils aient consacré de leur temps précieux à cet effet. Je remercie également de tout cœur Madame Corine Cauvet, Madame Danielle Boulanger, Monsieur Stefano Spaccapietra et Monsieur Djamel Zighed, d'avoir accepté de faire partie de mon jury.

Je remercie Djamel pour m'avoir accueillie au laboratoire ERIC. J'ai pu bénéficier d'une grande liberté d'entreprendre et d'une grande latitude pour tracer mon propre sillon dans le monde de la recherche. Ce mémoire est le fruit de longues années de travail d'équipe et de collaborations scientifiques. À ce titre, je témoigne du mérite des doctorants et stagiaires que j'ai encadrés ou que j'encadre encore, nommément Cédric, Maguy, Cécile, Amandine, Adrien, Nora, Ony, Rym, Doulkifli, Rachid, François, Yannick et Sid-Ali. Je leur souhaite un avenir plein de succès.

Je remercie mes collègues du laboratoire ERIC qui m'ont beaucoup apportée, que ce soit par leur conseil, leur aide ou tout simplement par leur amitié. Merci à Omar pour notre longue et fructueuse collaboration pendant toutes ces années, pour sa lecture avisée de ce mémoire et pour ses précieux conseils. Je remercie également de tout cœur Ricco et Stéphane pour leur regard critique sur la partie "fouille de données" de mon travail. Nos échanges ont beaucoup aidé à améliorer ce mémoire. Merci à Rafik, à Jean-Hugues et à Nadia pour leurs encouragements.

Merci à mes collègues de l'axe ENA-DC, Cécile, Nouria, Omar, Jérôme, Sabine et à tous les doctorants, pour le dynamisme "scientifique" que nous avons pu créé autour de nos thématiques de recherche. Merci également à tous les autres collègues du laboratoire ERIC, enseignants-chercheurs, doctorants, stagiaires et personnel administratif. Je remercie en particulier Valérie et Julien pour leur amabilité et leur disponibilité.

Un grand merci à Djamal Benslimane pour ses encouragements et sa gentillesse.

Je voudrais remercier mon mari Salah pour son soutien sans lequel ce travail n'aurait pas pu aboutir... en supportant mes absences répétées les week-ends et mes rentrées (très) tardives le soir. Je remercie également toute ma famille et mes amis d'ici ou d'ailleurs. Enfin, merci à mon fils Elyas d'être simplement ce qu'il est, ma source de bonheur.

Dédicaces

Je dédie ce mémoire à Nicolas qui nous a quittés en 2007. C'est avec lui que j'ai eu le plaisir de co-encadrer ma première thèse. Je le remercie pour son humilité, sa gentillesse, sa disponibilité et ses encouragements de tous les jours.

Enfin, je dédie ce mémoire à mes parents vava (papa) et yemma (maman) qui nous ont quittés en 2009 et qui me manquent énormément... Qu'ils reposent en paix.

Table des matières

1 Introduction **1**

1.1 Contexte de nos travaux de recherche 1

1.2 Contributions 8

1.3 Organisation du mémoire 13

2 Philosophie de la personnalisation dans les entrepôts de données **15**

2.1 Motivation 15

2.2 Définitions préalables 16

2.3 Personnalisation dans les systèmes d'information 18

2.4 L'utilisateur dans l'entrepôt de données 20

2.5 Discussion 24

2.6 Publications 26

3 Evolution de schémas pour la personnalisation des analyses dans les entrepôts de données **27**

3.1 Motivation 28

3.2 Travaux sur l'évolution de schéma 28

3.3 Principe général de notre approche de personnalisation 30

3.4 WEDriK : une architecture d'entrepôt centré utilisateur 31

3.5 Modèle d'entrepôt à base de règles 32

3.6 Méta-modèle d'entrepôts de données évolutifs 38

3.7 Mise à jour de hiérarchies de dimension 39

3.8 Conclusion 40

3.9 Publications 42

4 Recommandation de niveaux de hiérarchies dans une dimension **45**

4.1 Motivation 45

4.2 Principe général 46

4.3 RoK : un opérateur d'agrégation basé sur les k-means 48

4.4 Formalisation 53
4.5 Implémentation et Expérimentation 57
4.6 Conclusion 59
4.7 Publications 61

5 Fouille de données en ligne 63
5.1 Motivation 64
5.2 Intégration des techniques de fouille dans les SGBD 65
5.3 Les arbres de décision 66
5.4 Fouille en ligne utilisant les vues relationnelles 69
5.5 Fouille en ligne utilisant la table de contingence 70
5.6 Fouille en ligne utilisant les index bitmap 75
5.7 Implémentation et performance 80
5.8 Fouille en ligne dans les cubes OLAP 82
5.9 Synthèse de notre approche de fouille de données en ligne 85
5.10 Conclusion 86
5.11 Publications 87

6 Intégration sémantique de données pour l'analyse en ligne à la demande 91
6.1 Motivation 92
6.2 Système de médiation et ontologies 95
6.3 Alignement et Fusion des ontologies 98
6.4 Classification ascendante hiérarchique pour la fusion des ontologies 101
6.5 Conclusion 110
6.6 Publications 112

7 Entrepôts d'objets complexes 115
7.1 Motivation 116
7.2 État de l'art 117
7.3 Entreposage et analyse en ligne de données complexes 118
7.4 Modèle multidimensionnel d'objets complexes 125
7.5 Conclusion 139
7.6 Publications 141

8 Conclusion générale 145
8.1 Bilan et contributions 145
8.2 Projet de recherche 147

Bibliographie 153

Table des figures

2.1 Principes de la personnalisation 17
2.2 Entrepôt LCL 21
2.3 Personnalisation selon le langage IRAH appliqué au cas de LCL 22

3.1 Personnalisation selon l'évolution de schéma dans les entrepôts 31
3.2 WEDriK : data Warehouse Evolution Driven by Knowledge 32
3.3 Méta-modèle d'entrepôts de données évolutifs 38

4.1 Schéma de l'entrepôt de données VENTE. 50
4.2 Création du niveau GROUPE_ VILLE à partir du niveau VILLE 51
4.3 Création du niveau TYPOLOGIE_ VILLE selon NOMBREHABITANTS et SUPERFICIE à partir du niveau VILLE 52
4.4 Entrepôt de données VENTE après ajout de deux niveaux d'analyse GROUPE_ VILLE et GROUPE_ PRODUIT 53
4.5 Résultats des k-means selon les deux scenarii. 58
4.6 Cubes de données générés avec les deux nouveaux niveaux d'analyse. 59

5.1 Exemple d'arbre de décision 67
5.2 Vues relationnelles associées à l'arbre de décision de l'exemple de la Figure 5.1 . . 69
5.3 Arbre de décision Titanic associé à la vue relationnelle de la Table 5.2 71
5.4 Temps de traitement en fonction de la taille de la base 71
5.5 Vue relationnelle associée à la table de contingence : Exemple du Titanic 72
5.6 Construction du nœud racine. 77
5.7 Arbre de décision obtenu après segmentation selon l'attribut Sexe. 79
5.8 Comparaison de la performance des implémentations d'ID3 82
5.9 Entrepôt de données Titanic 83
5.10 Requête décisionnelle pour la construction du Cube de données Titanic 84
5.11 Requête d'extraction du nœud racine 84
5.12 Nœud racine de l'arbre d'ID3 84

5.13 Requête d'extraction du nœud fils *Sexe* = 'Femme' 85
5.14 Processus de fouille de données en ligne 86

6.1 Différents modèles d'intégration 97
6.2 Architecture de médiateur à base d'ontologies selon GLAV 98
6.3 Méthode de fusion d'ontologies 104
6.4 Exemple de trois ontologies du même domaine 105
6.5 Comparaison de la moyenne de la *Précision*, du *Rappel* et du *Fallout* pour OMerSeC, FCA et COMA++ 110

7.1 Approche d'intégration de données complexes 120
7.2 Modèle UML générique pour la représentation de données complexes 121
7.3 La fouille pour l'aide à la modélisation multidimensionnelle de données complexes . 122
7.4 Architecture du système SMAIDoC 124
7.5 Diagramme de classes des ventes aux enchères (*auctions*) 128
7.6 Méta-modèle d'objet complexe 130
7.7 Exemple d'objet complexe représentant les items 131
7.8 Exemple de relations entre objets complexes 133
7.9 Exemple de hiérarchie d'attributs associée à Person_ ID 134
7.10 Exemple de hiérachie d'objets 135
7.11 Modèle multidimensionnel d'objets complexes des *auctions* 136
7.12 Représentation à trois niveaux d'un cube d'objets complexes 137
7.13 Exemple de cube d'objets complexe des *auctions* 138
7.14 Processu d'entreposage et d'analyse en ligne des données complexes *auctions* . . . 139

Liste des tableaux

4.1 Application des k-means sur le niveau VILLE (attribut NOMBREHABITANTS) de la dimension *Région* 52
4.2 Création du niveau d'analyse GROUPE_ VILLE dans la dimension *Région* 52
4.3 Instances de la table de fait VENTE. 54
4.4 Niveau d'analyse VILLE décrit par les mesures. 56
4.5 Niveau d'analyse VILLE décrit par ses propres descripteurs. 56

5.1 Extrait de la base d'apprentissage Titanic 68
5.2 Vue relationnelle associée à l'arbre de décision Titanic 70
5.3 Table de contingence correspondant à la base Titanic 72
5.4 Table relationnelle associée à la table de contingence de la base Titanic 73
5.5 Table Titanic et index bitmap construit sur l'attribut *Survivant*. 76
5.6 Bitmap (*Survivant*="Oui") AND Bitmap (*Sexe*="Homme"). 76
5.7 Base d'apprentissage : ensemble des index bitmap de la base Titanic. 77
5.8 Bitmaps caractérisant les hommes ayant et n'ayant pas survécu. 78
5.9 Vues CovType utilisées pour les tests 81
5.10 Cube OLAP TitanicCube associé à la requête décisionnelle de la Figure 5.10 . . . 89

6.1 Description des ontologies locales 108
6.2 Statistiques sur les données ontologiques 108

Chapitre 1

Introduction

1.1 Contexte de nos travaux de recherche

Face à la mondialisation et à la concurrence grandissante, la prise de décision est devenue cruciale pour les dirigeants d'entreprises (au sens large du terme, entreprises privées, publiques, institutions, organisations...). L'efficacité de cette prise de décision repose sur la mise à disposition d'informations pertinentes et d'outils d'analyse adaptés. L'objectif des entreprises est de pouvoir exploiter efficacement d'importants volumes d'informations, provenant soit de leurs systèmes opérationnels, soit de leur environnement extérieur, pour l'aide à la décision.

C'est dans ce contexte qu'est apparu le secteur de l'informatique décisionnelle. Aussi, les systèmes d'information décisionnels (SID) sont nés pour répondre au besoin exprimé par les entreprises. Il s'agit pour les entreprises d'avoir une vision transversale de leurs données permettant ainsi de les analyser selon plusieurs axes d'activité, ce que ne permettait pas d'accomplir les systèmes de bases de données traditionnels. De ce besoin sont apparus dans les années quatre-vingt-dix les entrepôts de données (*data warehouses, dans la terminologie anglosaxonne*), qui sont des bases de données orientées analyse pour l'aide à la décision. Ces dernières ont eu une répercussion importante aussi bien dans le monde industriel que dans la communauté de la recherche scientifique. Les entrepôts de données forment ainsi le socle des systèmes d'information décisionnels et sont le support de l'analyse multidimensionnelle en ligne (*On-Line Analytical Processing - OLAP*) [Cod93].

A la fin des années quatre-vingt-dix, les premières manifestations scientifiques dans le domaine des entrepôts de données et de l'OLAP sont apparues. Nous pouvons citer notamment l'atelier international DOLAP [1] (international workshop on Data warehousing and OLAP) créé en 1998, la conférence internationale DAWAK [2] (international conference

1. http ://www.cis.drexel.edu/faculty/song/dolap.html
2. http ://www.dexa.org/

on DAta WArehousing and Knoweldge Discovery) créée en 1999 et la revue internationale IJDWM [3] (International Journal of Data Warehousing and Mining) créée en 2005. Dans la même année, avec d'autres collègues, nous avons fondé la première conférence francophone EDA [4] (journées francophones sur les Entrepôts de Données et l'Analyse en ligne) que nous continuons à piloter jusqu'à ce jour. En 2011, nous avons organisé l'atelier internationl dans le domaine des entrepôts de données : WMCD [5] (international workshop on Warehousing and Mining Complex Data) adossé à la conférence internationale EDBT/ICDT [6].

En intégrant la notion d'entrepôt de données, le processus décisionnel apporte une première réponse au problème de la croissance continuelle des données et à leur exploitation. La première difficulté à laquelle personnellement nous nous sommes confrontée, en menant des recherches dans le domaine des systèmes d'information décisionnels, est le manque de consensus sur les définitions et les concepts relatifs aux entrepôts de données. Selon Widom [Wid99], le terme entrepôts de données signifie *un ensemble de vues matérialisées* alors que selon Inmon un entrepôt de données est *une collection de données orientées sujet, intégrées, non volatiles, historisées et organisées pour supporter un processus d'aide à la décision* [Inm02]. Cette deuxième définition, reconnue actuellement par la communauté scientifique, est la plus proche de la réalité pour l'élaboration des SID puisqu'elle regroupe à la fois les processus d'intégration, de modélisation et de stockage des données pour des fins d'analyse. Cependant, on peut regretter l'absence de l'utilisateur dans cette définition alors même que les SID sont censés être centrés utilisateur. En effet, la construction d'un entrepôt de données est basée à la fois sur les objectifs d'analyse globaux et sur les besoins individuels des utilisateurs. Ces derniers doivent être en interaction avec le SID qui doit répondre à leurs requêtes décisionnelles, via l'analyse OLAP, afin d'extraire des informations pertinentes pour l'aide à la décision. Tout en nous inscrivant personnellement dans le courant des entrepôts de données définis par Inmon, l'idée de remettre l'utilisateur au sein des SID a servi de ligne directrice dans nos travaux de recherche.

L'objectif principal des entrepôts de données est l'analyse en ligne (OLAP). Un entrepôt de données présente une modélisation dite dimensionnelle qui se compose classiquement d'une table des faits centrale et d'un ensemble de tables de dimensions. Cette modélisation conceptuelle a pour objectif d'observer les faits à travers des mesures (appelées aussi indicateurs), en fonction des dimensions qui représentent les axes d'analyse. Ce modèle est qualifié de modèle en étoile.

La technologie entrepôt-OLAP est apparue donc comme une technologie clef pour les entreprises désirant améliorer l'analyse de leurs données et leur système d'aide à la décision. Cette technologie développe des outils décisionnels qui permettent d'étudier,

3. http ://www.igi-global.com/bookstore/titledetails.aspx ?TitleId=1085
4. http ://eric.univ-lyon2.fr/ eda05/
5. http ://eric.univ-lyon2.fr/wmcd/
6. http ://edbticdt2011.it.uu.se/

par exemple, le comportement de consommateurs, de produits, de sociétés ; d'effectuer une veille concurrentielle ou technologique, etc. Pour cela, ils intègrent traditionnellement des données dites de production provenant de sources de données internes ou externes à l'entreprise dans une base de données centralisée à vocation analytique (l'entrepôt). Les données dans l'entrepôt sont alors agrégées, historisées et structurées de manière à permettre aux utilisateurs (décideurs) d'effectuer des analyses efficaces (navigation OLAP et reporting). Ainsi, en utilisant des applications décisionnelles, un décideur peut découvrir à partir d'un entrepôt de données les tendances générales significatives d'une activité ciblée de l'entreprise et par conséquent prendre les décisions adéquates. Par exemple, la direction d'une chaîne de magasins peut interroger l'entrepôt de données de l'entreprise pour connaître les dix meilleures ventes de produits par magasin ou pour comparer les ventes de deux années consécutives. A partir des informations obtenues, la direction de l'entreprise peut déterminer le plan d'action à mettre en place afin d'améliorer les ventes et augmenter les bénéfices.

Les entrepôts de données répondent donc à un réel besoin en matière d'accès à l'information résumée. Les offres commerciales actuelles proposent des logiciels d'extraction d'information et d'analyse de données ; ils permettent de collecter des informations provenant de sources différentes et d'exploiter ces données au travers d'outils d'analyse et d'interfaces utilisateurs. Néanmoins ces logiciels ne proposent pas une solution intégrée de conception et de développement de systèmes décisionnels.

En suivant le processus classique d'entreposage et d'analyse en ligne de données, les systèmes décisionnels exploitent très peu le contenu informationnel des données. Tout d'abord, le processus d'ETL (*Extract-Transform-Load*) classique est vu comme un processus mécanique pour extraire, transformer et charger les données dans l'entrepôt en utilisant les métadonnées nécessaires pour accomplir cette tâche. Ensuite, les modèles d'entrepôts obtenus permettent dans un premier temps de produire des cubes de données (*data cubes en anglais*) adaptés à l'analyse, puis dans un second temps, c'est à l'utilisateur de naviguer, explorer et analyser les données d'un cube pour en extraire des informations pertinentes pour la prise de décision. Dans ce cas, il est nécessaire de guider l'utilisateur dans sa phase d'exploration et de navigation dans les données pour obtenir les informations les plus pertinentes en fonction de ses propres besoins d'analyse.

Notons aussi que de plus en plus, les données exploitées dans le cadre des processus décisionnels sont diverses et hétérogènes. En effet, dans de nombreux domaines tels que la médecine, les sciences sociales, la gestion de la relation client,... il est question de données qui ne sont ni numériques ni symboliques. L'avènement du Web et la profusion de données multimédias ont en grande partie contribué à l'émergence de ces données, que nous qualifions de complexes.

Ainsi, face à la profusion de données complexes, à l'évolution croissante des données et

des besoins, à la prise en compte de l'utilisateur dans les systèmes décisionnels et enfin à la nécessité d'intégrer la sémantique dans le processus d'entreposage de données, les modèles d'entrepôts de données classiques ont montré certaines limites. L'environnement décisionnel doit donc s'élargir pour prendre en compte, en plus des données et des métadonnées habituelles nécéssaires à la gestion de l'entrepôt, la sémantique des données surtout lorsque celles-ci sont complexes, les connaissances du domaine, les connaissances métier ainsi que les connaissances sur l'utilisateur (ses besoins, son profil, etc.). Nous développons dans les paragraphes suivants certaines limites des entrepôts de données classiques que nous avons identifiées et qui nous ont permis de dégager plusieurs verrous scientifiques auxquels nous avons tenté d'apporter des solutions adaptées.

1.1.1 Entrepôts de données peu centrés utilisateur

La mise en œuvre d'un entrepôt de données nécessite un important travail d'étude de l'existant et de recueil de données à partir de sources opérationnelles pour bien pendre en compte les objectifs d'analyse. L'entrepôt de données est ainsi conçu pour répondre à un ensemble de besoins d'analyse recensés auprès des utilisateurs à un moment donné. Cependant, les utilisateurs peuvent avoir de nouveaux besoins divers et variés auxquels l'entrepôt n'est pas forcément en mesure de répondre, a fortiori dans une grande entreprise dans laquelle les utilisateurs exercent de nombreux métiers. La création de magasins de données (*data marts en anglais*) représente une première solution possible et tentent de se rapprocher des besoins utilisateurs en fonction de leurs métiers. En effet, un magasin de données, élaboré comme un extrait de l'entrepôt, regroupe les données utiles pour un sujet d'analyse. Les données sont alors organisées suivant un modèle facilitant leur analyse. Néanmoins, chaque utilisateur peut disposer de connaissances particulières du domaine et des besoins d'analyse qui lui sont propres. Il est donc difficile de recenser de façon exhaustive les besoins d'analyse des utilisateurs et il est quasiment impossible de prévoir tous les besoins d'analyse futurs.

Dans ce contexte, l'intégration de nouveaux besoins d'analyse dans le processus d'entreposage et d'analyse en ligne des données devient un enjeu majeur. Cela nécessite d'impliquer un peu plus l'utilisateur dans ce processus. Bien que les architectures décisionnelles soient considérées comme centrées utilisateurs, la prise en compte de ces derniers dans les systèmes décisionnels a finalement été peu étudiée. Des propositions ont été faites dans les méthodes de conception, notamment dans les méthodes descendantes (méthodes de conception des entrepôts de données dirigées par les besoins), pour prendre en compte les besoins utilisateurs [TLM03]. Néanmoins ces travaux ne se focalisent pas réellement sur l'utilisation individualisée d'un magasin de données ; le magasin est conçu à un instant donné pour un groupe d'utilisateurs. Toute adaptation du magasin de données nécessite de réitérer le processus de conception. En outre, la constitution d'un magasin de données

reste une tâche complexe qui met en jeu des processus d'extraction, de transformation et d'alimentation qui rendent plus diffficile voir impossible la constitution de magasins individualisés en fonction des spécificités et des usages particuliers de chaque décideur.

Guider l'utilisateur dans son exploration des données et l'aider à pouvoir intégrer ses connaissances dans l'entrepôt suppose une prise en compte par le système de ses nouveaux besoins, ses préférences, ses usages, etc. C'est ce qu'on appelle *personnalisation* qui constitue un champ de recherche qui reste largement à explorer dans le domaine des systèmes d'information décisionnels. Le but de la personnalisation est de faciliter l'expression des besoins de l'utilisateur et de lui permettre d'obtenir des informations pertinentes lors de ses accès à un système d'information.

En conclusion, il apparaît donc qu'il existe un besoin fort de remettre l'utilisateur au cœur du système décisionnel en l'impliquant davantage dans le processus d'entreposage de données. Pour cela, il faut intégrer ses connaissances et/ou ses besoins et/ou son profil,... dans le processus d'entreposage, que ce soit au niveau de la phase de conception de l'entrepôt, de création des cubes de données ou de leur exploitation.

Notre objectif est de pouvoir intégrer les connaissances utilisateurs dans le processus d'entreposage afin d'apporter au modèle d'entrepôt une flexibilité en terme d'évolution des contextes d'analyse. Cet objectif représente notre premier verrou scientifique baptisé *personnalisation des analyses dans les entrepôts de données.*

Par ailleurs, l'analyse en ligne (OLAP) classique ne dispose pas d'outils permettant de guider l'utilisateur vers les faits les plus intéressants du cube. C'est à l'utilisateur de manipuler au mieux le cube de données pour y découvrir des zones d'informations pertinentes. Dans un cube de données volumineux, la navigation suivant les différents axes d'analyse est encore moins aisée. Par conséquent, le recours à des opérateurs spécifiques pour détecter automatiquement des zones sensibles augmenteraient de façon significative le pouvoir analytique de l'OLAP.

Ainsi, pour aider l'utilisateur à découvrir des informations cachées et potentiellement pertinentes pour lui, nous nous sommes intéressée à *la recommandation de nouveaux axes d'analyse* qui constitue notre deuxième verrou scientifique. La personnalisation et la recommandation s'inscrivent dans le courant des systèmes adaptatifs et constituent un thème émergent dans les entrepôts de données.

1.1.2 Modèles d'entrepôts en étoile centralisés et figés

La construction d'un entrepôt de données consiste à copier physiquement via le processus d'ETL les données des sources dans une base de données centralisée selon un schéma en étoile défini à priori. Les deux contraintes de centralisation des données et du schéma fixe de l'entrepôt ne permettent pas de prendre en compte l'évolution des sources de don-

nées et des besoins. Aussi pouvons-nous dire que le concept d'entrepôt centralisé avec un schéma fixé à priori, par conséquent figé, n'est plus nécessairement pertinent, voire même inadapté, dans tous les cas de figure. En effet, la spécificité des sources des données, leur évolution éventuelle, ainsi que l'évolution des besoins des utilisateurs peuvent amener à envisager de nouvelles solutions décisionnelles.

Afin de prendre en compte l'évolution des données et des besoins dans les entrepôts de données, des travaux de recherche s'inspirant de l'évolution de schéma dans les bases de données y ont été adaptés selon deux approches différentes. La première approche propose une mise à jour du schéma [HMV99a], la seconde consiste à gérer différentes versions de l'entrepôt [MW03, MW04]. Ces deux approches apportent une réponse à l'émergence de nouveaux besoins d'analyse qui sont engendrés par l'évolution des données, mais pas lorsqu'ils sont engendrés par des connaissances du domaine dont disposent les utilisateurs. En effet, dans les modèles existants, seules les données sont utilisées pour atteindre les objectifs d'analyse.

Ainsi, pour pouvoir prendre en compte les nouveaux besoins d'analyse des utilisateurs, une des solutions que nous envisageons est de "relâcher" la contrainte du schéma fixe de l'entrepôt de données en nous inspirant des travaux sur l'évolution de schéma. En outre, un schéma d'entrepôt flexible permettrait aussi, dans le cadre des systèmes de recommandation, de suggérer à l'utilisateur de nouveaux axes d'analyse non attendus à priori.

1.1.3 Analyse OLAP limitée

L'objectif des entrepôts de données est d'être un support pour les applications d'analyse en ligne. Ce type d'applications est caractérisé par une vision multidimensionnelle des données de l'entrepôt. Il s'agit essentiellement d'analyser des indicateurs (mesures) d'une activité selon différents axes d'analyse (dimensions). Par exemple, la direction d'une entreprise souhaite observer l'évolution des ventes en terme de *chiffre d'affaire* en fonction des axes d'analyse *produit*, *magasin* et *année*. Des modèles particuliers, tels que le schéma en étoile ou le schéma en flocon de neige, ont été conçus afin de rendre les données d'un entrepôt prêtes à l'analyse. Ces modèles permettent de créer des vues multidimensionnelles des données appelées cubes de données dont la vocation est l'analyse en ligne (OLAP) [CD97]. Ainsi, la grande capacité de stockage, la structuration multidimensionnelle et les opérateurs OLAP font de l'entrepôt de données une plate-forme décisionnelle servant à l'exploration, à la navigation et à la visualisation dans les grandes bases de données à vocation analytique.

Avec l'avènement des entrepôts de données, l'analyse OLAP [Cod93] a constitué une première étape en matière d'intégration de l'analyse au sein des systèmes de gestion de

bases de données (SGBD) pour des fins exploratoire et navigationnelle dans les données. Grâce aux opérateurs OLAP s'appuyant sur des fonctions d'agrégat de type *somme*, *comptage*, *moyenne*, *écart type* etc, l'information dans l'entrepôt de données peut être résumée selon différents axes d'observation. La navigation dans les données de l'entrepôt permet alors de produire les premiers résultats d'analyse pertinents pour l'aide à la décision. Le SGBD devient alors à la fois un système de gestion et d'analyse en ligne de bases de données. Cependant, cette analyse est assez limitée puisqu'elle ne permet pas de découvrir de nouvelles structures, d'expliquer un phénomène ou de prédir de nouvelles tendances. C'est dans ce contexte que personnellement nous nous sommes intéressée à la façon d'étendre les capacités analytiques des SGBD de l'OLAP vers la fouille de données. Cela permet d'une part de bénéficier des capacités des SGBD en termes de chargement et de traitement des données sans limitation de taille ; ce qui est non négligeable lorsque les méthodes de fouille sont appliquées sur les entrepôts de données qui sont très volumineux. D'autre part, cela va dans le sens des travaux sur l'enrichissement des capacités analytiques des SGBD avec des opérateurs de structuration, d'explication et de prédiction. Ceci constitue notre troisième verrou scientifique que nous avons baptisé *fouille de données en ligne*.

1.1.4 Avènement de données complexes

A l'heure actuelle, la communauté scientifique s'accorde pour dire que les données ne sont pas seulement numériques ou symboliques, mais qu'elles peuvent être représentées dans des formats différents (texte, image, son, vidéo, base de données, etc.), provenir de sources diverses (données de production, scanners, satellites, enregistrements vidéos, compte-rendus médicaux, résultats d'analyse, web, etc.), avoir une sémantique différente (langues différentes, échelles différentes, évolution de la définition d'une donnée dans le temps, etc.). De telles données sont désignées par les termes de données complexes. L'exploration des données complexes implique de nombreux problèmes, notamment en ce qui concerne leur modélisation (leur structuration et leur stockage) d'une part, et leur analyse d'autre part. L'une des difficultés engendrées par le premier point est due à la diversité des formats des données complexes. La description de ces dernières nécessite une certaine précision et un espace de représentation adapté. Par ailleurs, la prise en compte de la sémantique des données complexes représente un enjeu majeur aussi bien dans la phase de modélisation que dans la phase de l'analyse. Si bien que l'intégration des données complexes exige une modélisation permettant de prendre en considération les différents types de complexité de ces données.

L'un des principes de base de l'entrepôt de données est de proposer un modèle de données unique et structuré. La plupart des entrepôts reposent sur une modélisation et une algèbre OLAP trop rigides et inadaptées à la représentation de données complexes. En effet, les architectures classiques d'entrepôts de données [Inm02, KR02] ont montré leur

utilité et leur efficacité lorsque les données sont “simples” (numériques ou symboliques). En revanche, elles doivent être complètement reconsidérées lorsque les données sont complexes. Lorsqu'il s'agit d'analyser par exemple une image, une vidéo ou tout autre objet de l'univers, il est alors plus efficace de considérer chacun de ces éléments comme une entité, à part entière, à observer.

Ainsi, ces dernières années le domaine des entrepôts de données et de l'OLAP a été marqué par la croissance des travaux traitant des données complexes. Ces travaux couvrent différents aspects de l'entreposage et de l'analyse en ligne. Nous trouvons les travaux sur l'intégration des données provenant du Web, l'entreposage de données non structurées et des données semi-structurées, représentées notamment en XML, l'entreposage des données médicales et des données spatiales, la prise en compte d'autres aspects de la complexité comme la temporalité et l'incertitude, etc.

Partant de ce constat, il apparaît évident qu'il est nécessaire de proposer un nouveau processus d'entreposage et d'analyse en ligne qui permettrait de concevoir des systèmes d'information décisionnels centrés utilisateur et adaptés aux données complexes. Parfois même, il faut repenser les modèles d'entrepôts et relâcher les contraintes que nous venons de citer. Dans ce contexte, nous avons identifié deux verrous scientifiques majeurs, à savoir l'*intégration sémantique de données* dans le processus d'entreposage et l'*élaboration de nouveaux modèles d'entrepôts de données complexes.*

1.2 Contributions

Nous avons présenté dans les sections précédentes les différents problèmes relatifs aux entrepôts de données auxquels nous nous sommes intéressée ces dernières années. Ces problèmes constituent les différents verrous scientifiques auxquels nous avons tenté d'apporter des solutions adaptées. Quelle est maintenant la place de nos travaux et de ce mémoire dans ce contexte ? Dans leur finalité, nos contributions et développements trouvent écho dans le domaine des entrepôts et de l'analyse en ligne de données complexes, mais aussi dans la fouille de données en ligne. Le fil conducteur de nos travaux est la prise en compte de l'utilisateur dans tout le processus d'entreposage de données complexes.

Les travaux exposés dans ce mémoire visent donc à proposer des solutions innovantes dans le domaine de la modélisation multidimensionnelle et de l'analyse en ligne des données complexes. Ils traitent plus particulièrement de la personnalisation dans les entrepôts de données complexes. L'originalité de nos travaux de recherche a consisté à montrer qu'il est pertinent d'intégrer la sémantique dans tout le processus d'entreposage de données. Par la suite, ces travaux ont été étendus pour répondre aux nouveaux défis liés aux entrepôts de données complexes, notamment pour la prise en compte de l'utilisateur. Pour aborder ces problèmes, nous avons, de façon plus personnelle, orienté nos travaux de recherche

en suivant plusieurs angles d'approche. (1) Une approche personnalisée pour l'analyse en ligne basée sur l'évolution de schéma, (2) une approche de recommandation de nouveaux axes d'analyse pertinents exploitant les techniques de fouille de données, et (3) une approche d'intégration des méthodes de fouille de données au sein des SGBD pour assurer d'une part, de bonnes performances de la fouille en ligne, et d'autre part pour étendre les capacités analytiques des SGBD, de l'exploration et la navigation dans les données vers la structuration, l'explication et la prédiction. (4) Une approche d'analyse en ligne à la demande fondée sur un système de médiation en utilisant les ontologies, et enfin (5) une nouvelle approche de modélisation multidimensionnelle d'objets complexes centrée utilisateur.

Pour atteindre ces objectifs, nous nous sommes fixée un certain nombre d'hypothèses de travail qui nous ont permis d'une part, de cerner le champ des actions de recherche à mener, et d'autre part d'organiser ces actions. Une partie des propositions et résultats obtenus dans ce mémoire sont le fruit des travaux menés dans le cadre de la thèse de C. Favre soutenue en 2007 et de celle de N. Maïz soutenue en 2010 à l'Université Lumière Lyon 2. Les autres propositions et résultats sont le fruit de travaux développés dans le cadre des masters recherche ou de collaborations avec des collègues au sein du laboratoire ERIC.

1.2.1 Evolution de schéma pour la personnalisation des analyses

La personnalisation a pour objectif de faciliter l'expression des besoins des utilisateurs et de rendre l'information sélectionnée intelligible à l'usager et exploitable. Elle se définit, entre autres, par un ensemble de connaissances, de préférences individuelles, par des ordonnancements de critères ou par des règles sémantiques spécifiques à chaque utilisateur ou communauté d'utilisateurs. Ces modes de spécification servent à décrire le centre d'intérêt de l'utilisateur, le niveau de granularité des données qu'il désire ou des modalités de présentation de ces données.

L'encadrement de la thèse de C. Favre [Fav07] nous a permis d'approfondir le problème de la personnalisation des analyses dans les entrepôts de données en empruntant le concept d'évolution de schéma [FBB06b, FBB07b]. La recherche avait pour objectif de faciliter la tâche de l'utilisateur 'averti' d'un système décisionnel en lui permettant d'intégrer ses propres connaissances métier en termes d'axes d'analyse [BFB08]. Le travail de doctorat a permis de définir une architecture décisionnelle dans laquelle le schéma de l'entrepôt de données est évolutif. L'accent a été mis sur la spécification et l'intégration des connaissances explicites d'un utilisateur ou d'un groupe d'utilisateurs dans l'entrepôt de données. Cette connaissance utilisateur est traduite en termes de règles d'agrégation permettant de créer de nouveaux niveaux de granularité dans une hiérarchie de dimension prêts à être

exploités dans les analyses futures [FBB06a, FBB07a]. Cela pose de nombreux problèmes de cohérence des données et des analyses dans l'entrepôt de données pour lesquels des solutions originales ont été apportées.

1.2.2 Recommandation de requêtes OLAP en utilisant la fouille

La démarche classique de l'analyse multidimensionnelle commence par la sélection des niveaux d'analyse et des mesures qui sont susceptibles de répondre au besoin d'analyse de l'utilisateur. Une fois que le cube de données associé à ce besoin est construit, l'utilisateur va explorer ce cube pour tenter de déceler rapidement des similarités entre les faits selon les dimensions qu'il étudie. Ce sont les différents niveaux de granularité dans les hiérarchies de dimensions qui permettent de détecter ces similarités. A partir du plus haut niveau de la hiérarchie, le décideur (utilisateur) observe un niveau plus bas d'une hiérarchie en regardant les valeurs agrégées et en identifiant visuellement des valeurs intéressantes. Si une exploration ne donne pas de résultats intéressants, le décideur remonte au niveau le plus haut des hiérarchies des dimensions du cube et continue son analyse dans une autre direction, en allant observer d'autres dimensions. Le but de ces manipulations est de pouvoir découvrir des aspects insoupçonnables dans la masse de données de l'entrepôt permettant ainsi l'enrichissement de l'analyse exploratoire du décideur.

Cependant, il n'existe pas d'outils pour guider l'utilisateur vers de nouvelles explorations non prédéfinies par le modèle de l'entrepôt ni pour approfondir l'analyse vers la structuration, l'explication et la prédiction. En effet, dans un processus décisionnel, un utilisateur peut vouloir anticiper la réalisation d'évènements futurs. Dans ce cas, combiner les techniques de fouille de données avec la technologie OLAP permet d'assister l'utilisateur dans cette tâche pour l'extraction de nouvelles connaissances pouvant être exploitées dans ses analyses futures.

Pour aider l'analyste dans cette démarche, nous avons proposé un nouvel opérateur d'agrégation *RoK* (*Roll-up with K-means*) en utilisant la fouille de données [RB07]. L'opérateur *RoK* permet de créer un nouveau niveau de granularité dans une hiérarchie de dimension en se basant sur une méthode de classification automatique [Ben08, BF09].

L'opérateur *RoK* utilise la méthode des *k-means* qui permet de rechercher des structures naturelles dans les données. Il s'agit dans notre cas, de trouver un bon regroupement des instances d'un niveau d'analyse existant choisi par l'utilisateur, à partir duquel un nouveau niveau d'analyse peut être créé. Ainsi, notre approche enrichit considérablement l'analyse multidimensionnelle car elle offre de nouveaux angles de vues intéressants sur les faits pouvant être explorés par l'utilisateur.

1.2.3 Fouille de données en ligne

Notre objectif dans ce domaine de recherche est étroitement lié aux entrepôts de données et à l'analyse en ligne. Notre travail a consisté à étendre les possiblités d'analyse des SGBD, qui hébergent les entrepôts de données, de l'OLAP vers la fouille de données. En ce sens, faire des SGBD, en plus d'un outil de gestion de gros volumes de données qui est leur finalité première, une plate-forme logicielle dédiée à l'analyse exploratoire et navigationnelle mais aussi descriptive, explicative et prédictive tout en garantissant de bonnes performances.

Contrairement aux solutions présentes dans la littérature qui utilisent des extensions du langage SQL et des API ("*Application Programming Interface*") pour intégrer des méthodes de fouille dans les SGBD, nous avons souhaité utilisé dans nos travaux uniquement les outils offerts par ces derniers (tables, vues, index, procédures stockées...) afin de garantir des temps de traitement acceptables sur des bases d'apprentissage sans limitation de taille. Pour atteindre cet objectif, il est nécessaire d'adapter les algorithmes de fouille de données à l'environnement des SGBD.

Dans un premier temps, nous nous sommes intéressée à l'intégration de méthodes d'apprentissage supervisé au sein des SGBD, notamment les *arbres de décision*. Tout d'abord, nous avons utilisé les vues relationnelles pour modéliser les arbres de décision [BD02]. Par la suite, nous avons proposé une amélioration de cette approche en introduisant une phase de préparation des données, en remplaçant la table d'apprentissage initiale par la table de contingence correspondante [BDU04, UBDB04]. D'autre part, pour exploiter les outils d'optimisation des SGBD, nous avons proposé une approche originale de fouille de données intégrée qui consiste à utiliser les index bitmap pour construire des arbres de décision [FB05b, FB05a]. Pour terminer, nous avons étendu ces travaux vers la fouille en ligne dans les cubes OLAP [Mad04]. De même que pour la table de contingence, une adaptation des algorithmes de fouille était nécessaire afin de les appliquer sur des données agrégées.

Dans un second temps et dans le cadre de nos travaux sur la recommandation de requêtes décisionnelles, nous nous sommes intéressée à l'intégration de méthodes d'apprentissage non-supervisé au sein des SGBD, en particulier les *méthodes de classification* [RB07, Ben08].

1.2.4 Intégration sémantique des données pour l'analyse en ligne à la demande

L'encadrement de la thèse de N. Maiz [Mai10] nous a permis d'aborder le problème d'évolution à la fois des sources de données et des besoins d'analyse dans le contexte décisionnel. La solution que nous avons apportée à ce problème réside dans le choix de la stratégie de construction des cubes OLAP. Garder un entrepôt centralisé suppose une

dépendance vis-à-vis du processus d'ETL qui rend impossible l'accès en temps réel aux sources de données modifiées. L'autre inconvénient de l'entrepôt centralisé réside dans son modèle figé qui ne permet pas de prendre facilement en considération les nouveaux besoins d'analyse. Nous avons donc opté pour une approche d'intégration par médiation représentée par le médiateur qui joue le rôle d'intermédiaire entre l'utilisateur et les sources de données. Malheureusement dans cette approche nous perdons le bénéficie de l'historisation des données qui permet bien évidemment des analyses OLAP de type tendances. En revanche, cette approche peut être très bénéfique lorsque les données à exploiter dans les sources représentent des données redondantes à vocation décisionnelle (données de simulation par exemple) stockées de manière répartie. Le principe d'une telle approche serait de permettre le calcul de cubes de données à la demande, en fonction des besoins évolutifs des utilisateurs.

Notre approche de conception de cubes OLAP à la demande guidée par les utilisateurs (approche personnalisée) est fondée sur une architecture par médiation à base d'ontologies pour assurer à la fois l'intégration structurelle et sémantique des données [MBB06, MBB07]. Cette approche permet l'accès en temps réel aux sources de données, et créer des cubes personnalisés à la demande sur des données actualisées. Notre contribution dans ce domaine porte principalement sur la proposition d'un algorithme de fusion des ontologies et d'une mesure de similarité sémantique adaptée. L'originalité de notre approche de fusion des ontologies provient de l'utilisation de la méthode de classification ascendante hiérarchique et du mécanisme d'inférence OWL (*Web Ontology Language*) pour extraire les classes de concepts les plus similaires à partir de plusieurs ontologies pour construire l'ontologie globale [MFBB10].

1.2.5 Entrepôts d'objets complexes centrés utilisateur

Le survol des travaux sur la modélisation multidimensionnelle des données complexes permet de constater la diversité des modèles multidimensionnels proposés pour tenir compte des différents aspects de complexité des données. Il apparaît donc difficile d'avoir un modèle d'entrepôt unifié pour prendre en compte les différents types de complexité des données. Il en ressort néanmoins que la modélisation objet est la plus appropriée pour modéliser des données complexes bien qu'il n'existe pas de consensus sur la représentation des concepts multidimensionnels. Il y a presque autant de modèles que de types de données complexes.

Dans le cadre de nos travaux sur la modélisation multidimensionnelle des données complexes, nous ne ferons pas exception ; nous traitons un aspect particulier des données complexes. Tout d'abord, nous considérons une donnée complexe comme un agrégat de données hétérogènes qui, une fois réunies, forment une unité sémantique. Nous parlons

alors d'objet complexe. Nous nous sommes particulièrement intéressée aux liens sémantiques existants entre les éléments d'un même objet (liens intra-objets) et entre les objets complexes eux-mêmes (liens inter-objets). Le challenge est alors de pouvoir trouver de nouveaux modèles orientés analyse, pour représenter ces objets complexes. D'une part, les concepts multidimensionnels (fait, dimension, hiérarchie, niveau, attribut,...) sont à redéfinir dans le cadre de ces objets complexes. D'autre part, il est nécessaire de définir de nouvelles métriques sémantiques afin d'agréger et d'observer les objets complexes. En effet, les métriques quantitatives sont souvent insuffisantes ou inadaptées.

D'autre part, de par sa définition, un modèle d'entrepôt en étoile fixe les indicateurs et les axes d'analyse pendant la phase de conception. Par conséquent, les espaces d'analyse dépendent fortement de cette modélisation. En effet, la définition à priori des mesures et des dimensions fige l'analyse multidimensionnelle autour de ces mêmes mesures et dimensions. Cependant, dans le monde réel, un utilisateur peut vouloir créer des espaces d'analyse dans lesquels une même donnée peut représenter, selon ses objectifs d'analyse, un sujet ou un axe d'observation. Sachant que les modèles d'entrepôts actuels ne permettent pas cela, il apparaît donc nécessaire de proposer un nouveau modèle d'objets complexes centré utilisateur.

Dans ce contexte, nous avons proposé un modèle multidimensionnel d'objets complexes basé sur le paradigme objet qui permet de représenter les objets de l'univers de façon plus naturelle et de capter la sémantique qu'ils véhiculent [BBBL10]. Par ailleurs, nous avons défini un opérateur de projection cubique qui permet à l'utilisateur, de choisir ses objets, de leur affecter le rôle de sujet ou d'axe d'analyse, et enfin de créer le cube d'objets complexes souhaité [BBB10a, BBB10b].

1.3 Organisation du mémoire

Les travaux exposés dans ce mémoire ont été menés au laboratoire ERIC au sein de l'axe de recherche ENA-DC (ENtrepôts et Analyse de Données et Complexes) avec des contributions venant de stagiaires, doctorants et collègues, qui ont naturellement influencé notre perception des différents verrous scientifiques auxquels nous avons personnellement tenté d'apporter des solutions, et par conséquent les directions de recherche que nous avons suivies.

Le document suit la progression de nos différentes contributions. Hormis le chapitre introductif et le chapitre consacré à la conclusion, ce mémoire comporte six chapitres. Le chapitre 2 introduit et développe la thématique de la personnalisation dans les entrepôts de données. Le chapitre 3 présente notre approche de personnalisation des analyses en empruntant le concept d'évolution de schéma dans les bases de données. Le chapitre 4 porte sur la recommandation de requêtes décisionnelles en utilisant la fouille de données. Le

chapitre 5 est consacré à la fouille en ligne nécessaire pour le traitement de gros volumes de données tout en augmentant les capacités analytiques des SGBD. Le chapitre 6 développe notre approche d'analyse en ligne à la demande basée sur un système de médiation en utilisant les ontologies. Le chapitre 7 présente notre approche d'entreposage de données complexes en utilisant les techniques de fouille de données et les systèmes multi-agents, puis propose un modèle multidimensionnel d'objets complexes centré utilisateur pour la prise en compte des liens sémantiques intra- et inter-objets. Nous terminons par la conclusion qui résume nos contributions tout en présentant notre projet de recherche qui introduit les multiples perspectives de ce travail dans le domaine des entrepôts de données complexes.

Chapitre 2

Philosophie de la personnalisation dans les entrepôts de données

L'objectif de ce chapitre est de présenter les enjeux et les opportunités relevant de la prise en compte des utilisateurs au sein des entrepôts de données. Pour ce faire, nous présentons tout d'abord un panorama des travaux sur la personnalisation dans les domaines connexes des bases de données et de la recherche d'information. Nous présentons ensuite un état de l'art des quelques travaux qui émergent dans le domaine des entrepôts de données vis-à-vis de la prise en compte de l'utilisateur. Ceci nous permet de définir les nouveaux défis relevant des entrepôts de données centrés utilisateur.

2.1 Motivation

Les résultats d'une enquête menée en 2005 par le magazine CIO [1] (magazine destiné aux décideurs informatiques) auprès de 140 grandes entreprises [IDG05] ont révélé une volonté des entreprises de «disposer d'outils souples, plus près des objectifs métiers et des usages que peuvent en faire les opérationnels». En effet, parmi les facteurs clés de succès de la mise en place d'un projet décisionnel identifiés par les entreprises interrogées, l'adéquation aux objectifs métiers et l'adhésion des utilisateurs arrivent en tête. En outre, cette enquête a également révélé qu'un tiers des entreprises envisage une extension du parc d'utilisateurs. Face à ces constats concernant à la fois le nombre d'utilisateurs et la réponse à leurs besoins, la personnalisation des possibilités d'analyse trouve un grand intérêt.

La personnalisation est une approche qui vise à mieux répondre aux besoins des usagers par une meilleure connaissance de leurs caractéristiques [Sea03]. Les mécanismes mis en jeu pour la personnalisation d'un système peuvent se résumer en deux principales phases :

1. http ://www.decisio.info/Entreprises-et-decisionnel.html

une phase d'acquisition des caractéristiques pour améliorer la connaissance de l'usager par le système et une phase d'exploitation de ces caractéristiques pour améliorer l'adéquation du système aux besoins de l'usager.

Bien que les architectures décisionnelles soient considérées centrées utilisateurs, la prise en compte de ces derniers dans les systèmes décisionnels a finalement été peu étudiée. Des propositions ont été faites dans les méthodes de conception, notamment dans les méthodes descendantes, pour prendre en compte les besoins utilisateurs [LMT]. Néanmoins ces travaux ne se focalisent pas réellement sur l'utilisation individualisée d'un cube de données ; il est conçu à un instant donné pour un groupe d'utilisateurs. Toute adaptation du cube de données voire de l'entrepôt nécessite de réitérer le processus de conception, qui reste une tâche complexe et fastidieuse mettant en jeu des processus ETL complexes. En effet, la spécificité des sources de données, leur évolution éventuelle ainsi que l'émergence de nouveaux besoins des utilisateurs peuvent amener à envisager de nouvelles solutions décisionnelles.

Les systèmes OLAP mettent à disposition des utilisateurs des résumés d'information extraits à partir de l'entrepôt afin de les aider à prendre des décisions. Pour présenter à l'utilisateur des informations pertinentes, le système doit intégrer dans le processus d'entreposage les préférences et les besoins individuels des utilisateurs. De manière générale, l'accès à une information pertinente, adaptée aux besoins et au profil de l'utilisateur est un enjeu capital dans le contexte décisionnel. En effet, à partir d'énormes volumes de données collectées, l'utilisateur d'un entrepôt de données souhaite dégager les informations les plus intéressantes, selon tels ou tels critères, afin de mieux étayer ses prises de décision. Cela permet de toute évidence de réduire l'espace de recherche et de répondre de façon ciblée à l'utilisateur tant sur le contenu que sur la présentation des résultats.

La prise en compte par le système décisionnel des besoins, des préférences, des usages et des interactions du décideur constitue un champ de recherche, appelé *personnalisation*, qui reste à explorer dans le domaine des systèmes décisionnels [Riz07].

2.2 Définitions préalables

Y. Ioannidis et G. Koutrika définissent la personnalisation en ces termes "*...providing an overall customized, individualized user experience by taking into account the needs, preferences and characteristics of a user or group of users*" [KI05]. Généralement, la personnalisation d'un système consiste à définir, puis à exploiter un profil utilisateur [Kor96] qui regroupe un ensemble de caractéristiques servant à configurer ou à adapter le système à l'utilisateur, afin de fournir des réponses plus pertinentes à l'utilisateur [DJ07]. Ce profil peut s'apparenter à une modélisation de l'utilisateur (identification, antécédents, droits d'accès, préférences...). Nous proposons de caractériser un profil selon deux perspectives :

l'implication de l'utilisateur et les fonctions systèmes liées au profil [BBF+09].

- 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.
- Les fonctions systèmes liées au profil consistent dans un premier temps à définir le profil et dans un second temps à exploiter ce dernier pour une meilleure prise en compte de l'utilisateur.

La Figure 2.1 présente les quatre principes mis en jeu lors de la personnalisation. La définition d'un profil réalisé de façon explicite correspond au paramétrage d'un système tandis que la définition implicite s'apparente à l'apprentissage. L'exploitation du profil peut soit nécessiter l'intervention explicite de l'utilisateur qui réalise un choix par rapport à la recommandation du système, soit induire une transformation automatique du système. Les termes de configuration et d'adaptation sont caractérisés en fonction de ces quatre principes.

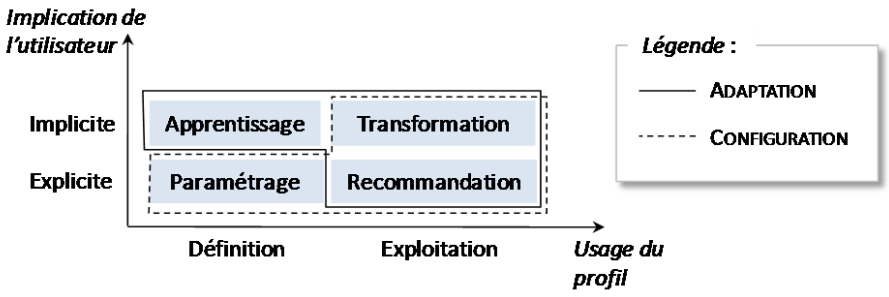

FIGURE 2.1 – Principes de la personnalisation

- La configuration ("*customisation*" ou "*user modeling*") consiste donc pour l'utilisateur à paramétrer explicitement son profil. Le profil ainsi déterminé est exploité au travers de mécanismes de transformation ou de recommandation. Par exemple, dans le logiciel Word, l'opération consistant à placer manuellement un bouton dans la barre d'outils est une tâche qui s'apparente à une configuration.
- L'adaptation ("*user profiling*") consiste pour le système à définir implicitement le profil de l'utilisateur, puis à l'exploiter selon les principes de transformation ou de recommandation. Par exemple, dans le logiciel Word, depuis sa version 97, les items des menus sont rendus automatiquement visibles en fonction de l'usage qui est fait du traitement de texte.

2.3 Personnalisation dans les systèmes d'information

La personnalisation de l'information a été abordée principalement dans la communauté Recherche d'Information (RI) et la communauté Bases de Données (BD) [KI04, KI05]. Nous reprenons les principes de la personnalisation évoqués précédemment pour présenter brièvement les travaux sur la configuration et l'adaptation dans les sytèmes d'information.

2.3.1 Configuration

Différents travaux se sont focalisés sur la définition du profil dans une logique de configuration par un paramétrage du profil. Dans le contexte de la RI, le profil utilisateur consiste souvent à regrouper un ensemble plus ou moins structuré de mots-clés, définis par l'utilisateur, avec éventuellement des poids qui leur sont associés [PG99]. Le profil utilisateur peut également correspondre à des fonctions d'utilité sur un domaine d'intérêt, exprimant l'importance relative des sujets de ce domaine, les uns par rapport aux autres [CGFZ03]. Le profil est ensuite exploité de diverses manières :

- Les systèmes de filtrage d'information ("*information filtering*") visent à distribuer des informations de façon personnalisée en comparant les caractéristiques de l'utilisateur fournies par ce dernier avec une collection d'informations [Aas97].
- Les systèmes de recommandation collaboratifs ("*collaborative filtering*") ont le même objectif de mise à disposition d'informations pertinentes en faisant reposer le processus de sélection d'information sur une approche collaborative (comparaison d'un utilisateur avec d'autres en fonction d'éléments fournis par ceux-ci) [GNOT92, AT08, CEP09].
- Les systèmes temps réel fondés sur les stratégies d'interactions sociales pour la recherche et l'accès à l'information [Cas08].

La notion de préférence a également été introduite dans le domaine plus structuré des BD [LL87]. Ces travaux ont essentiellement porté sur la personnalisation de requêtes, notamment à travers l'intégration des préférences utilisateurs [KI05, AW00, Cho03, Kie02, LL87]. Dans ce contexte, deux approches principales ont émergé : quantitative et qualitative. L'approche quantitative consiste à exprimer les préférences d'une façon indirecte par l'utilisation de fonctions de score qui associent un score numérique à chaque n-uplet du résultat d'une requête. Dans l'approche qualitative, les préférences sont spécifiées directement à l'aide de relations binaires. L'intégration consiste alors à un enrichissement des requêtes opérées sur la BD par de nouveaux prédicats.

2.3.2 Adaptation

La particularité des systèmes adaptatifs réside dans la détermination "automatique" des caractéristiques de l'utilisateur et dans une certaine mesure de leur évolutivité. La difficulté réside alors dans la modélisation de l'utilisateur (de ses caractéristiques, de ses préférences).

On peut distinguer les différents travaux concernant les systèmes adaptatifs selon le type d'apprentissage des caractéristiques, des préférences. En effet, l'apprentissage peut se faire soit à partir d'éléments fournis par l'utilisateur, soit à partir du comportement même de l'utilisateur, i.e. de l'interaction de ce dernier avec le système. Dans la première catégorie, on retrouve les travaux sur les systèmes éducatifs qui se basent sur un questionnaire préliminaire rempli par l'apprenant pour "apprendre" le niveau de celui-ci ou sur des tests au fur et à mesure de l'apprentissage ; c'est le cas des systèmes ACE [SO98] et ARTHUR [GH99]. On trouve également des travaux se basant sur la programmation par l'exemple où les utilisateurs sont invités à fournir des exemples pour la personnalisation d'applications Web [MP08]. Dans la deuxième catégorie, on retrouve également des travaux sur les systèmes éducatifs, mais ceux-ci vont exploiter l'interaction système-utilisateur. C'est le cas dans le système iMANIC [SW00], où les données concernant l'interaction avec l'étudiant sont analysées pour déterminer quelles ressources doivent lui être recommandées. Poursuivant cette idée d'exploiter le comportement de l'utilisateur, il a été proposé dans le contexte de la RI, un agent nommé Letizia, qui enregistre les URL parcourues par l'utilisateur, lit les pages et détermine au fur et à mesure le profil de l'utilisateur [Lie95]. Sur cette base, il effectue une recherche pour recommander d'autres pages susceptibles d'intéresser l'utilisateur. Bradley et al. proposent une personnalisation de contenu dans le projet CASPER en définissant les profils et en les mettant à jour par rapport au comportement des usagers à travers les statistiques d'interaction entre le système et l'usager (nombre de clics, etc.) [BRS00].

2.3.3 Bilan

Bien que très proches dans leur objectif final (accès efficace à l'information), les deux domaines de recherche RI et BD se distinguent généralement par la nature de l'information traitée (documents textuels pour la RI et tableaux structurés pour les BD) et le mode d'accès à cette information (accès par mots-clés plus ou moins complexes et pas à pas pour la RI, accès par expressions logiques et de façon globale pour les BD).

La personnalisation dans les bases de données a souvent été envisagée sous l'angle d'extension du langage de requêtes (SQL en général) par un ensemble de clauses ou de prédicats censés traduire les préférences de l'utilisateur, sans pour autant interagir avec ce dernier durant l'évaluation. Grâce aux nouvelles clauses, il est possible d'exprimer des préférences

hiérarchiques ("*prefering*" et "*cascading*") [Kie02], ou de définir un ordre entre les critères de sélection ("*domain*" et "*utility*") [CGFZ03]. L'utilisateur est contraint d'écrire à chaque fois la requête complète qui définit son besoin d'information, ce qui est un inconvénient non négligeable.

Aucune proposition de profil n'a encore fait l'objet d'un consensus. Cette absence de standard rend cette notion parfois ambiguë ; le concept même n'est pas toujours clairement formalisé dans les solutions proposées (certaines approches ne faisant pas appel explicitement à un profil, mais recueillant néanmoins les préférences d'un utilisateur). Bouzeghoub et Kostadinov ont alors apporté un modèle générique multidimensionnel de profil, convenant ainsi à une majorité de contexte [BK05]. De son côté, Kobsa a proposé un état de l'art sur la modélisation des utilisateurs en fonction des besoins des systèmes [Kob07].

2.4 L'utilisateur dans l'entrepôt de données

La conception de magasins de données a pour objectif de répondre aux objectifs métiers. Mais, comme il est souligné dans [RTZ07], compte tenu de la complexité de mise en œuvre des magasins de données (conception, alimentation, rafraîchissement, maintenance), il n'est pas envisageable de déployer un magasin de données pour chaque décideur. Par conséquent, il est encore plus difficile de concevoir, construire et alimenter un entrepôt de données qui prendrait en compte tous les besoins actuels et futurs des utilisateurs. En effet, il est difficile d'être exhaustif dans le recensement des besoins d'analyse des utilisateurs au moment de la conception du schéma de l'entrepôt. De plus, la prise en compte des besoins d'analyse lors de la phase de conception n'est pas évidente ; ceci est en partie dû à l'absence de standard pour la conception des entrepôts de données [RALT06]. En outre, il est difficile de prévoir des besoins d'analyse futurs. Or, de nouveaux besoins individuels peuvent émerger dans la mesure où les utilisateurs peuvent s'intéresser à de nouveaux objectifs d'analyse.

Bien que les travaux dans les domaines de la RI et des BD pour la personnalisation aient été beaucoup développés [KI05], nous nous intéressons ici aux travaux moins nombreux et plus récents dans le domaine des entrepôts de données [Riz07, BBF+09]. La personnalisation dans le cadre des entrepôts de données présente un réel intérêt dans un contexte où les analyses devant permettre la prise de décision sont réalisées par l'utilisateur lui-même. Nous présentons tout d'abord les différentes notions liées aux entrepôts de données à travers un exemple qui illustrera la présentation des différents travaux. Nous comparons ensuite les différentes approches dans le but de faire émerger les enjeux qu'il reste à explorer dans ce domaine.

2.4.1 Exemple illustratif

Pour illustrer les différents travaux, nous choisissons d'utiliser une étude de cas simplifiée définie avec l'établissement bancaire LCL-Le Crédit Lyonnais (LCL) [2]. Il s'agit d'analyser un fait correspondant aux performances de l'établissement bancaire. Cette analyse est effectuée à travers une mesure : le Produit Net Bancaire (PNB). Ce PNB représente ce que rapporte la gestion des comptes des clients à l'établissement bancaire. Cette mesure est analysée selon différentes dimensions : CLIENT, TEMPS et AGENCE. Certaines dimensions peuvent être hiérarchisées, comme c'est le cas de la dimension AGENCE, dont la hiérarchie représente la structure commerciale de LCL (H-StructCommerciale) ; les agences sont regroupées en UC (Unité Commerciale), elles-mêmes regroupées en DPP (Direction Particuliers Professionnels), les DPP étant regroupées en DE (Direction d'Exploitation). Notons que les hiérarchies sont déterminées grâce à des attributs que l'on nomme paramètres (UC), la sémantique de ces derniers étant complétée par des attributs dits faibles (NOM_UC). La Figure 2.2 décrit le schéma conceptuel de l'entrepôt LCL pouvant supporter une telle analyse.

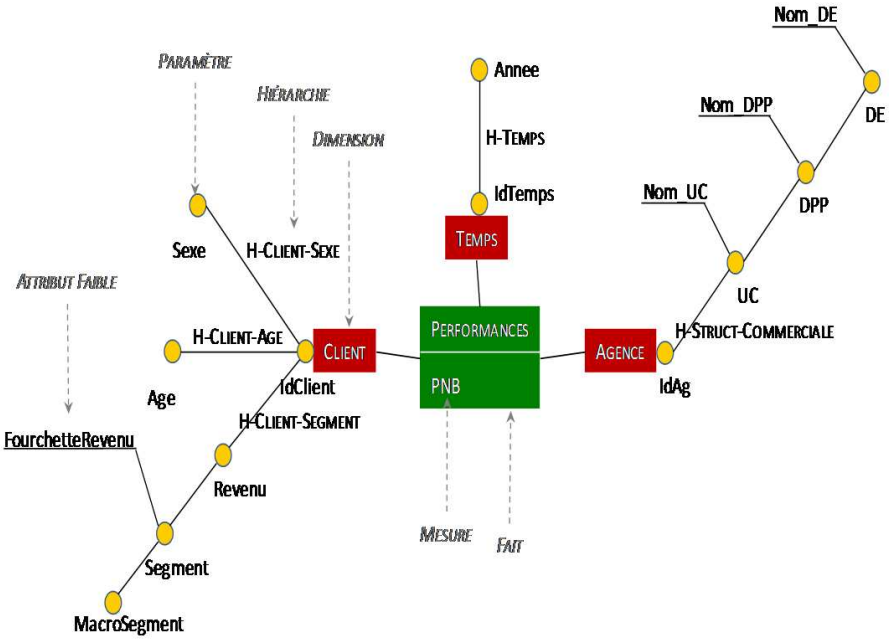

FIGURE 2.2 – Entrepôt LCL

2. Entreprise partenaire dans le cadre de la thèse CIFRE de C. Favre

2.4.2 Présentation et illustration des travaux

Initialement, les entrepôts étaient destinés à la consultation de données organisées en fonction d'une vision d'analyse. L'utilisateur ne pouvait interagir avec les valeurs et les structures mises à sa disposition autrement que par des requêtes d'interrogation. Face à la nécessité d'offrir davantage de flexibilité pour répondre au mieux aux besoins des utilisateurs, une première approche [EV01] a consisté à proposer le langage à base de règles nommé IRAH ("*Intensional Redefinition for Aggregation Hierarchies*") visant à transformer les appariements de valeurs entre les niveaux d'agrégation au sein des hiérarchies. Cette approche permet à l'utilisateur de construire ses propres chemins de navigation en réorganisant les instances mises à sa disposition. Considérons notre exemple illustratif en nous focalisant sur la hiérarchie H-ClientSegment (Figure 2.3 -a-). La hiérarchie H-ClientSegment signifie que chaque client a un certain revenu qui permet de déterminer le segment auquel il appartient ; les segments pouvant s'agréger dans un macrosegment (Figure 2.3 -b-). Un conseiller commercial peut alors réaffecter un client dans un autre segment, pour répondre à ses besoins, en exprimant une règle avec le langage IRAH, produisant ainsi une révision du chemin d'agrégation pour la dimension CLIENT (Figure 2.3 -c-).

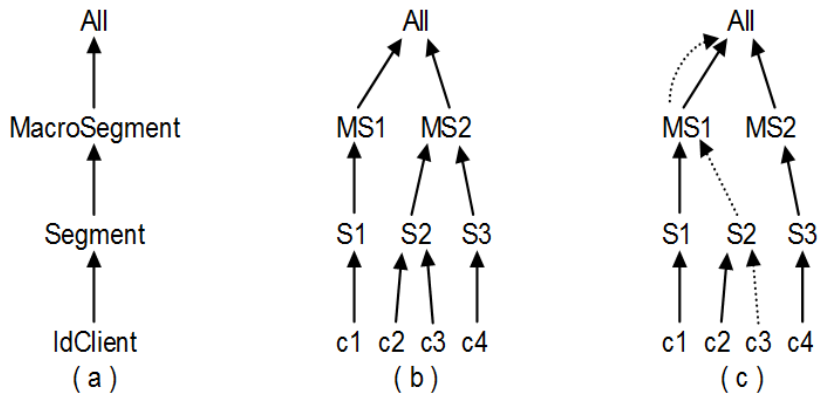

FIGURE 2.3 – Personnalisation selon le langage IRAH appliqué au cas de LCL

Thalhammer et al. présentent un système à base d'entrepôt actif au sein duquel l'utilisateur doit spécifier ses scénarios d'analyse au travers d'un mécanisme de type ECA (Evénement - Condition - Action) [TSM01]. L'objectif est, par une meilleure connaissance des analyses effectuées sur l'entrepôt, d'améliorer le prétraitement des données. Au-delà de l'amélioration des performances, les auteurs proposent d'exploiter les résultats obtenus lors des analyses pour induire des changements dans les données opérationnelles. Dans notre exemple, on peut imaginer une règle d'analyse qui permettrait de lancer une action

marketing pour améliorer les ventes, pour l'UC de Lyon, lorsque l'analyse du PNB faite à la fin de l'année est inférieure à 100 000 euros pour cette UC.

Concernant la personnalisation vis-à-vis des analyses multidimensionnelles, Bellatreche et al. se sont inspirés des techniques de filtrage d'information en fonction de profil utilisateur pour affiner des requêtes en y ajoutant des prédicats [BGM+05]. L'objectif de ces travaux est de pouvoir fournir à l'utilisateur un résultat focalisé sur son centre d'intérêt, tout en prenant en compte des contraintes de visualisation pour adapter le résultat à l'utilisateur. Par exemple, supposons qu'un utilisateur veuille connaître le PNB total détaillé par UC, année et segment de clients. Si le nombre de segments et/ou celui d'UC et/ou celui des années sont importants, la réponse complète ne peut être visualisée sur l'écran. Selon l'utilisateur, l'intérêt pour telles ou telles UC, tels ou tels segments peut varier. Si l'utilisateur est un responsable de la clientèle "haut de gamme", certains segments sont moins "intéressants" que d'autres. Concernant la segmentation, il s'agirait de créer un profil "responsable clientèle haut de gamme" dans lequel seraient classés les segments client par ordre de préférence pour la visualisation.

Pour faciliter la navigation de l'utilisateur dans les données, Ravat et al. proposent une approche de personnalisation des données multidimensionnelles manipulées [RT08]. Un poids fixé par l'utilisateur est associé aux données exprimant ainsi les préférences de ce dernier. Un système basé sur des règles ECA permet de générer des tables multidimensionnelles contenant uniquement les données identifiées comme pertinentes en fonction des poids. Cette solution quantitative permet de simplifier l'expression des requêtes d'analyse. Par exemple, un utilisateur peut exprimer ses préférences pour indiquer que, lors de l'analyse du PNB, les paramètres DE et DPP sont prioritaires, tandis que les paramètres UC et AGENCE ne le sont pas. Ainsi, si l'affichage de la mesure du fait PERFORMANCES est demandé en fonction de la dimension AGENCE, le système affiche automatiquement les attributs DE et DPP.

Pour aller au-delà de cette approche quantitative, une solution de personnalisation qualitative est introduite par Jerbi et al. [JRTZ08]. Il s'agit non plus d'exploiter des poids, mais plutôt des ordres (représentation qualitative des préférences), ce qui rend la tâche plus aisée pour l'utilisateur. En outre, ces ordres ne sont pas exprimés de façon absolue, mais par rapport à un contexte d'analyse donné. Ceci permet de prendre en compte le fait que les préférences peuvent varier d'un contexte d'analyse à l'autre. Par exemple, un utilisateur peut exprimer comme précédemment que les paramètres DE et DPP sont prioritaires par rapport à UC simplement par un ordre entre les paramètres des préférés aux moins préférés. En outre, dans un contexte d'analyse tel que l'analyse du PNB, cette préférence peut s'exprimer par l'affichage de la mesure du fait PERFORMANCES. Ainsi l'ordonnancement est pris en compte uniquement dans ce contexte, tandis que d'autres préférences peuvent être exprimées dans un autre contexte sur cette même dimension

AGENCE. Jerbi et al. poursuivent ces travaux en présentant un environnement OLAP intégrant des mécanismes de recommandation contextuelle des requêtes [JRTZ09]. Citons aussi la proposition de Giacometti et al. qui permet la recommandation de requêtes pour anticiper sur une séquence de requêtes d'un utilisateur grâce à l'analyse des historiques de navigations réalisées par les autres utilisateurs [GMN08, GMN09]. Par exemple, supposons qu'un utilisateur a réalisé une analyse du PNB par année et DE, puis avec un forage vers le bas par année et DPP, et enfin par année et UC (avec un second forage vers le bas). Si un nouvel utilisateur réalise une analyse du PNB par année et DE, puis une analyse du PNB par année et DPP, une analyse par année et UC lui sera recommandée, sa navigation étant similaire à une navigation réalisée précédemment. Pour terminer, nous pouvons citer, d'un point de vue un peu plus formel, la proposition de Golfarelli et Rizzi d'une algèbre permettant la prise en compte de préférences pour l'analyse en ligne (OLAP) [GR09]. Ces préférences peuvent porter sur les mesures, les attributs des dimensions ou les hiérarchies elles-mêmes.

2.5 Discussion

Les approches de personnalisation et de recommandation dans les entrepôts de données présentes dans la littérature s'intéressent particulièrement à l'intégration du profil de l'utilisateur dans le processus d'analyse. Ces approches se distinguent essentiellement par la manière de déterminer le profil de l'utilisateur. Quant à l'exploitation du profil, le choix entre la personnalisation ou la recommandation dépend essentiellement de l'objectif poursuivi. A notre connaissance, aucune approche de personnalisation présentée à la Section 2.4.2 n'utilise les nouveaux besoins réels des utilisateurs pour les intégrer de façon interactive dans l'entrepôt afin de les exploiter pour des fins d'analyse. En effet, tous ces travaux se sont focalisés sur l'exploitation du profil de l'utilisateur (préférences, contexte, contraintes de visualisation) pour l'enrichissement de requêtes décisionnelles. D'autre part, les systèmes de recommandation de requêtes développés ces dernières années dans le cadre des entrepôts de données, s'intéressent plus souvent à l'anticipation de requêtes décisionnelles. Ils sont également basés soit sur le profil utilisateur, soit sur l'historique de leurs requêtes.

Or, dans le cadre de notre collaboration avec LCL, les besoins des utilisateurs auxquels nous étions confrontée et qui étaient directement issus de la réalité de l'entreprise ont soulevé de nouveaux problèmes dans le contexte de la personnalisation. En effet, l'entreprise LCL était dans une phase de restructuration de ses agences et leur objectif était de disposer d'un système décisionnel qui peut les aider à analyser efficacement les demandes de marketing local, pour permettre une capitalisation des connaissances. Le système doit prendre en compte les évolutions éventuelles à venir dans le cadre des analyses futures.

Le premier constat auquel nous étions arrivée est qu'un modèle d'entrepôt de données classique ne pouvait pas répondre aux attentes de l'entreprise LCL. En effet, leurs produits apparaissent ou disparaissent, leurs structures organisationnelles se modifient, de nouveaux clients arrivent, certains partent et pour d'autres clients, leurs caractéristiques peuvent être modifiées (statut familial, pouvoir d'achat, etc.). De nouveaux besoins peuvent émerger, en réaction à l'évolution des données par exemple, ou tout simplement parce que les utilisateurs expriment de nouveaux besoins qui n'avaient pas été recensés lors de la conception de l'entrepôt et parce qu'il est difficile de prévoir tous les besoins à venir. LCL est un établissement regroupant des employés exerçant divers métiers et ayant donc des besoins d'analyses variés. La question qui s'est alors posée est comment adapter et faire évoluer le modèle d'entrepôt en étoile à la réalité de l'entreprise ?

Pour répondre à cette question, nous avons cherché dans la littérature des travaux, relatifs à l'évolution de schémas dans les entrepôts, capables de prendre en compte l'évolution des données et des besoins. Notre conclusion est que ces travaux répondent en partie à cette question en trouvant leur intérêt pour répondre au problème de modification dans les sources de données. Ce sont des solutions qui doivent être mises en œuvre par l'administrateur pour faire évoluer l'entrepôt de données. Cependant, elles n'impliquent pas directement les utilisateurs dans le processus d'évolution. En effet, une fois l'entrepôt créé, les utilisateurs peuvent uniquement réaliser les analyses prévues par le modèle. De ce fait, aucune solution n'est apportée à l'émergence de nouveaux besoins d'analyse exprimés par les utilisateurs, et par conséquent au problème de la personnalisation des analyses.

C'est dans ce contexte que nous avons proposé une approche de personnalisation, différente des approches existantes, qui permet de prendre en compte les nouveaux besoins réels des utilisateurs en les intégrant au cœur du système décisionnel. C'est une approche de personnalisation dirigée par les connaissances utilisateurs (cf. Chapitre 3). L'originalité de nos travaux consiste à "relâcher" la contrainte du schéma fixe de l'entrepôt pour permettre d'ajouter de nouveaux niveaux de hiérarchies dans les dimensions qui constitueront les nouveaux axes d'observation pour l'utilisateur. Nous pensons que l'évolution de schéma dans les entrepôts de données est une piste prometteuse dans le domaine de la personnalisation dans les entrepôts de données.

Nous avons ensuite étendu le concept d'évolution de schéma à la recommandation de nouveaux axes d'analyse permettant des requêtes décisionnelles non prévues initialement par l'entrepôt. Pour cela, nous avons utilisé la fouille de données pour extraire de nouvelles structures à partir des données de l'entrepôt pour les matérialiser sous forme de nouveaux niveaux de hiérarchies de dimensions. C'est une approche dirigée par les connaissances extraites à partir des données de l'entrepôt (cf. Chapitre 4).

Par ailleurs, pour rendre efficace l'application de la fouille de données sur les entrepôts ou les cubes de données, nous avons proposé une approche de fouille en ligne qui consiste

à intégrer les techniques de fouille au sein des SGBD (cf. Chapitre 5). L'objectif de la fouille en ligne est d'étendre les capacitiés analytiques des SGBD, de l'analyse OLAP qui est exploratoire et navigationnelle, vers une analyse structurante, explicative et prédictive.

2.6 Publications

La liste suivante présente nos publications concernant les travaux de synthèse que nous avons réalisés sur l'évolution de schéma et la personnalisation.

Chapitre d'ouvrage d'audience internationale

[1] C. Favre, **F. Bentayeb**, O. Boussaid, "A Survey of Data Warehouse Model Evolution", Handbook of Research on Innovations in Database Technologies and Applications, Vol. II, IGI Global, February 2009, 129-136.

Conférences francophones

[2] **F. Bentayeb**, O. Boussaïd, C. Favre, F. Ravat, O. Teste, "Personnalisation dans les entrepôts de données : bilan et perspectives", 5èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 09), Montpellier, Juin 2009 ; Revue des Nouvelles Technologies de l'Information, Vol. B-5, Cépaduès Editions, Toulouse, 7-22.

[3] C. Favre, **F. Bentayeb**, O. Boussaid, "Evolution de modèle dans les entrepôts de données : existant et perspectives", 3èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 07), Poitiers, Juin 2007 ; Revue des Nouvelles Technologies de l'Information, Vol. B-3, Cépaduès, 21-36.

Chapitre 3

Evolution de schémas pour la personnalisation des analyses dans les entrepôts de données

Dans ce chapitre, nous présentons notre approche de personnalisation des analyses dans les entrepôts de données. Contrairement aux travaux existants qui exploitent le profil de l'utilisateur, notre approche tente de prendre en compte les nouveaux besoins des utilisateurs exprimés sous forme de connaissances nouvelles. L'objectif est d'intégrer ces connaissances utilisateurs au sein de l'entrepôt. Le contenu de l'entrepôt est ainsi enrichi et permet par conséquent de nouvelles analyses pertinentes. Pour atteindre cet objectif, nous avons proposé une solution originale, qui repose sur le concept d'évolution de schéma, pour assister de façon automatique les utilisateurs des entrepôts dans leur tâche d'analyse.

Tout d'abord, nous présentons brièvement les différentes approches d'évolution de schéma présentées dans la littérature afin de mieux comprendre les défis associés à la personnalisation dans les entrepôts de données. Nous motivons et détaillons ensuite nos contributions dans ce domaine. Nous développons ainsi les quatre phases du processus de personnalisation que nous avons défini autour d'une architecture décisionnelle évolutive : acquisition des connaissances utilisateurs sous forme de règles, intégration de ces règles dans l'entrepôt, évolution du schéma de l'entrepôt en fonction des connaissances intégrées, et enfin application de l'analyse en ligne. Pour terminer, nous présentons notre plate-forme logicielle appliquée aux données bancaires.

Ce travail a fait l'objet de la thèse de doctorat de C. Favre que nous avons co-encadrée, préparée au sein du laboratoire ERIC dans le cadre d'une thèse CIFRE (Convention Industrielle de Formation par la Recherche) en collaboration avec l'entreprise bancaire LCL-Le Crédit Lyonnais et soutenue en 2007 à l'Université Lyon 2.

3.1 Motivation

Dans le cadre de notre collaboration avec la banque LCL-Le Crédit Lyonnais, l'objectif était de développer un système d'information décisionnel pour le marketing local qui consiste à mener des actions commerciales pour répondre à des besoins de vente spécifiques lors de la création d'une agence par exemple. Il s'agit alors, pour les responsables commerciaux, de faire des demandes de ciblage de clients et d'utiliser ces ciblages pour optimiser les ventes des conseillers. Il est alors nécessaire de pouvoir mesurer l'intérêt de ces demandes marketing. Dans ce contexte, nous avons été amenée à réfléchir à la façon de personnaliser les analyses des utilisateurs et à la mise en place d'une solution décisionnelle centrée utilisateur.

Si la personnalisation n'est pas une idée nouvelle dans les domaines des BD et de la RI, elle constitue un axe de recherche émergent dans le domaine des entrepôts de données, alors même que les caractéristiques de ces derniers lui sont favorables. En effet, le volume des données connu pour être important dans les entrepôts de données et le rôle central que joue l'utilisateur dans l'analyse en ligne des données, sont deux éléments qui justifient pleinement le recours à la personnalisation. D'autre part, lorsque les sources de données et/ou les besoins d'analyse évoluent dans une entreprise, il devient nécessaire de faire évoluer le modèle de l'entrepôt. Ainsi, l'enjeu réside dans le fait d'accorder à l'utilisateur une réelle place dans le processus décisionnel, au-delà de l'exploration des analyses possibles de l'entrepôt.

3.2 Travaux sur l'évolution de schéma

Les modèles multidimensionnels classiques [CT98, Kim96, Leh98] considèrent les faits comme la partie dynamique des entrepôts de données et les dimensions comme des entités statiques. L'historisation des données est assurée par la dimension *Temps*. Les autres dimensions sont considérées comme étant temporellement invariantes, compte tenu de l'hypothèse selon laquelle les dimensions sont supposées être orthogonales les unes par rapport aux autres et donc orthogonales par rapport à la dimension *Temps*. Cependant, en pratique, le schéma peut être amené à évoluer suite à l'évolution des sources de données ou des besoins d'analyse comme l'attestent de nombreux travaux sur l'évolution de schémas [NSF+05]. Dans la littérature, on peut distinguer deux approches pour remédier à ce problème : la mise à jour de schéma, et la modélisation temporelle.

La première approche consiste à migrer les données d'un ancien schéma vers le plus récent en proposant des opérateurs pour faire évoluer le schéma [BSH99, HMV99b]. Dans ce cas, un seul schéma est supporté, et les évolutions que le schéma subit ne sont donc pas conservées. Un autre courant s'inscrit dans cette approche de mise à jour, mais se

base sur l'hypothèse que, conceptuellement, un entrepôt de données est un ensemble de vues matérialisées construites à partir des sources de données. Dans ce cas, il s'agit de ramener le problème de l'évolution des sources de données à celui de la maintenance des vues [Bel02]. Dans cette approche, le fait de ne pas garder trace des évolutions peut induire des problèmes de cohérence du point de vue des analyses.

La deuxième approche consiste, elle, à garder justement la trace des évolutions, en utilisant des étiquettes de validité temporelle. Ces étiquettes sont apposées soit au niveau des instances, soit au niveau des liens d'agrégation, ou encore au niveau des versions du schéma. Le premier courant propose ainsi de gérer la temporalité des instances de dimensions [BSSJ98] grâce à un schéma en étoile temporel. Le principe est d'omettre la dimension temps qui permet habituellement l'historisation des données et d'ajouter une étiquette temporelle au niveau de chacune des instances des tables de faits et de dimensions de l'entrepôt. Le deuxième courant propose, quant à lui, de gérer la temporalité des liens d'agrégation [MV00]. Le chemin d'agrégation défini pour une instance le long d'une hiérarchie peut alors évoluer au cours du temps. Pour interroger ce modèle, un langage de requêtes nommé TOLAP (*Temporal OLAP*) est proposé [BSSJ98, VM00, MW03, MW04]. Le troisième courant consiste à gérer différentes versions du modèle de l'entrepôt, chaque version étant valide pendant une durée donnée. Le modèle proposé dans [EDE 01] définit des fonctions de mise en correspondance qui permettent la conversion entre des versions de structures. Dans [BMBT03], les auteurs proposent une approche qui permet à l'utilisateur d'obtenir des analyses en fonction de ses besoins. En effet, le modèle proposé permet de choisir dans quelle version analyser les données (en temps consistant, dans une version antérieure, ou dans une nouvelle version). Dans [RTZ06], un modèle multidimensionnel en temps consistant est proposé pour gérer des évolutions sur un modèle en constellation. Le versionnement permet également de répondre à des questions de type "*what-if analysis*", en créant des versions alternatives, en plus des versions temporelles, pour simuler des changements de la réalité [BEK$^+$04]. Différents travaux se sont par ailleurs focalisés sur la réalisation d'analyses prenant en compte différentes versions [GLRV06, MW04].

Dans le cadre de l'approche de la modélisation temporelle, les évolutions du schéma sont donc bien conservées et assurent la cohérence des analyses. Mais ce type de solutions nécessite une réimplémentation des outils de chargement de données et d'analyse avec la nécessité d'étendre les langages de requêtes afin de gérer les particularités de ces modèles. Il est donc nécessaire dans ce cas de prévoir la gestion des évolutions à venir au moment de la conception.

Nous trouvons également d'autres travaux qui se sont focalisés sur la gestion des hiérarchies. En effet, l'analyse dans les entrepôts de données est fortement liée aux hiérarchies de dimension qui sont définies. Lors de la conception des entrepôts de données, l'approche naïve consiste à faire émerger les hiérarchies en fonction des besoins d'analyse et des sources

de données qui sont mises à disposition. Pour rendre l'approche moins naïve, il a été proposé de définir des hiérarchies à un niveau conceptuel, puis logique, en les déterminant en fonction des relations de généralisation et d'agrégation de la modélisation UML (*Unified Modeling Language*) des besoins [ACWP01]. Les auteurs définissent leur approche en différentes étapes dont une doit inclure la confrontation avec les sources de données qui constitue un problème majeur dans la définition des hiérarchies. Les hiérarchies doivent non seulement être stucturées mais également alimentées. La précédente approche ne répond qu'au premier aspect. Dans [MT06], les auteurs proposent d'enrichir les hiérarchies de dimension à la fois pour la structure et les données, et ce de façon automatique. En partant du principe qu'une hiérarchie de dimension représente des relations sémantiques entre des valeurs, ils proposent d'exploiter les relations d'hypéronymie (*is-a-kind-of*) et de méronymie (*is-a-part-of*) de WordNet2 [1].

3.3 Principe général de notre approche de personnalisation

Les approches d'évolution de schémas présentées dans la littérature trouvent leur intérêt pour répondre au problème de changement de schéma suite à une modification dans les sources de données. Ce sont des solutions devant être mises en œuvre par l'administrateur pour faire évoluer l'entrepôt de données. Cependant, elles n'impliquent pas directement les utilisateurs dans le processus d'évolution. En effet, une fois l'entrepôt créé, les utilisateurs peuvent uniquement réaliser les analyses prévues par le modèle. De ce fait, aucune solution n'est apportée à l'émergence de nouveaux besoins d'analyse exprimés par les utilisateurs, et par conséquent au problème de la personnalisation des analyses. C'est pour répondre à ce problème que nous avons proposé une approche originale de personnalisation des analyses basée sur l'évolution de schéma [Fav07, BFB08]. Cette approche permet la création de nouveaux niveaux de granularité supplémentaires dans les hiérarchies de dimension existantes de l'entrepôt offrant ainsi à l'utilisateur de nouvelles possibiltés d'analyse.

Reprenons l'exemple de l'entrepôt LCL (Figure 2.2). Supposons qu'un utilisateur veuille analyser le produit net bancaire PNB, ce qui correspond à ce que rapporte un client à l'établissement bancaire, non pas en fonction de la structure commerciale de l'établissement, mais en fonction du type d'agence, information qui n'est pas présente dans l'entrepôt. L'utilisateur veut créer le niveau type d'gence (TypeAg) dans lequel il y aura : type *étudiant* pour les agences qui ne gèrent que les comptes des étudiants, type *non résident* lorsque les agences ne gèrent que des clients ne résidant pas en France et le type *classique* pour les agences ne présentant pas de particularité à partir du niveau AGENCE (Figure 3.1). L'utilisateur pourra ainsi réaliser des analyses du PNB en fonction de TypeAg.

1. http ://wordnet.princeton.edu/

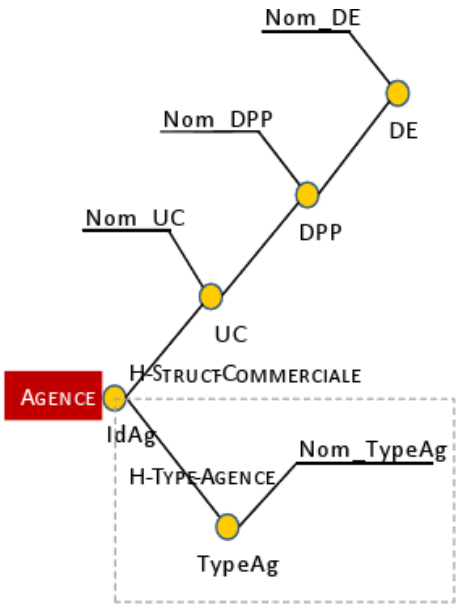

FIGURE 3.1 – Personnalisation selon l'évolution de schéma dans les entrepôts

3.4 WEDriK : une architecture d'entrepôt centré utilisateur

L'émergence des nouveaux besoins d'analyse des utilisateurs dans un contexte décisionnel impose une nouvelle approche des processus décisionnels. En effet, les architectures classiques d'entrepôts de données ont montré leur utilité et leur efficacité pour prendre en compte les besoins globaux lors de la phase de conception. En revanche, elles doivent être complètement repensées lorsque les besoins évoluent. L'architecture décisionnelle que nous présentons implique l'utilisateur dans le processus d'évolution de l'entrepôt.

C'est dans ce contexte que nous avons proposé un modèle d'entrepôt de données évolutif guidé par les utilisateurs. Nous avons conçu pour cela une architecture décisionnelle, dénommée WEDriK ("*data Warehouse Evolution Driven by Knowledge*"), qui permet aux utilisateurs d'intégrer dans l'entrepôt leurs propres connaissances métier sous la forme de règles d'agrégation de type "si-alors" (Figure 3.2). L'architecture WEDriK est composée de quatre modules. Le premier module *acquisition* des connaissances utilisateurs permet aux utilisateurs d'exprimer leurs connaissances métier sous la forme de règles de type «si-alors». Le deuxième module est l'*intégration* des règles dans l'entrepôt de données. Ensuite, le module d'*évolution* du schéma permet d'ajouter un nouveau niveau de granularité grâce aux règles, en étendant une hiérarchie de dimension existante, ou en créant une nouvelle. Enfin, le module d'*analyse* permet de réaliser des analyses en ligne, en se basant sur le nouveau schéma de l'entrepôt. Il s'agit là d'un processus d'évolution de schéma incrémental dans la mesure où les nouveaux besoins exprimés par les utilisateurs font évoluer au

fur et à mesure le schéma courant de l'entrepôt.

L'architecture décisionnelle WEDriK que nous proposons est centrée utilisateur, mais il est primordial que l'implication des utilisateurs ne compromette pas le schéma initial de l'entrepôt qui répond à des besoins d'analyse globaux communs à l'ensemble des utilisateurs. Ainsi, les nouveaux besoins d'analyse ne doivent modifier ni la table des faits, ni les niveaux de granularité directement liés à celle-ci (tables de dimension). Ceci justifie le fait que nous proposions un modèle d'entrepôt de données évolutif *R-DW* pour *Rule-based Data Warehouse* contenant une partie fixe (table de faits et dimensions) et une partie évolutive (hiérarchies de dimensions).

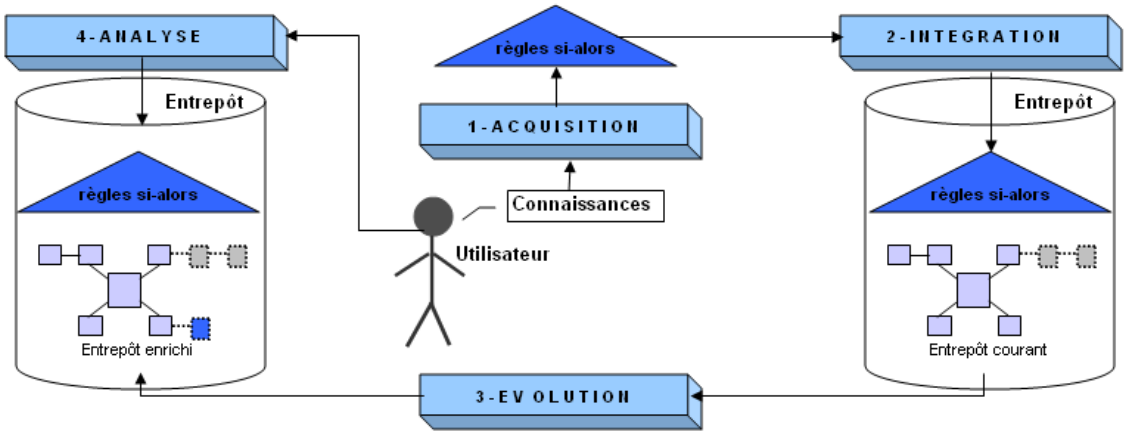

FIGURE 3.2 – WEDriK : data Warehouse Evolution Driven by Knowledge

3.5 Modèle d'entrepôt à base de règles

Notre choix du modèle *R-DW* à base de règles est motivé par le fait que les langages à base de règles permettent d'introduire une certaine flexibilité dans les systèmes qui les utilisent. Par exemple, Espil et al. ont défini un langage à base de règles pour la gestion des exceptions dans le processus d'agrégation qui permet de rendre l'analyse plus flexible en redéfinissant, au niveau des instances, des chemins d'agrégation dans les hiérarchies de dimension [EV01]. Dans notre cas, les règles que nous définissons sont des règles d'agrégation qui permettent de créer un nouveau niveau de hiérarchie à partir d'un niveau existant.

3.5.1 Principe

Notre modèle *R-DW* à base de règles est composé d'une partie fixe et d'une partie évolutive [FBB06a]. La partie fixe comprend la table des faits et les dimensions qui lui sont directement reliées. La partie évolutive comprend l'ensemble des hiérarchies du modèle.

Comme nous l'avons déjà annoncé, notre modèle *R-DW* ne permet pas l'ajout d'un niveau directement relié à la table des faits, qui constituerait ainsi une nouvelle dimension. Ce choix est motivé par deux aspects. Premièrement, il assure une cohérence des données stockées dans l'entrepôt. Si l'on envisageait la création de dimension par les utilisateurs, cela engendrerait une évolution dans le processus d'alimentation de l'entrepôt. En effet, une telle évolution nécessite une modification du processus d'ETL qui ne peut être réalisée par l'utilisateur. Deuxièmement, la partie fixe du modèle *R-DW* constitue une réponse à des besoins d'analyse initiaux, communs aux différents utilisateurs, définis lors de la conception de l'entrepôt. La modélisation initiale fournit ainsi le schéma global de l'entrepôt.

Le modèle *R-DW* est basé sur des règles qui vont permettre la création de nouvelles hiérarchies par ajout de niveau de granularité. L'ajout peut être réalisé au-dessus d'une table de dimension ou par extension des hiérarchies existantes par ajout d'un niveau de granularité en fin de hiérarchie ou au sein de celle-ci.

Les règles utilisées dans le modèle *R-DW* sont de type «si-alors». La clause "si" permet d'exprimer les conditions sur les attributs caractérisant le niveau de granularité inférieur, c'est-à-dire le niveau à partir duquel sera généré le nouveau niveau. Dans la clause "alors" figure la définition du niveau de granularité à créer, c'est-à-dire la définition des valeurs des attributs caractérisant ce nouveau niveau de granularité. Nous qualifions ces règles de *règles d'agrégation* puisqu'elles établissent un lien d'agrégation entre deux niveaux de granularité dans une hiérarchie de dimension.

Il existe différents types de liens d'agrégation au sein d'une hiérarchie de dimension [MZ04]. Nous considérons ici le cas classique, que l'on peut qualifier de hiérarchie symétrique stricte selon la typologie présentée dans [MZ04]. Ainsi on prend en considération le cas où toutes les instances d'un niveau donné ont une et une seule instance correspondante dans le niveau supérieur. Les règles exprimées par les utilisateurs doivent satisfaire deux contraintes.

La première contrainte est que les clauses "si" des règles d'agrégation définissent une partition des instances du niveau inférieur. Par définition, la partition d'un ensemble est un ensemble de parties non vides de cet ensemble, deux à deux disjointes et dont la réunion est égale à l'ensemble initial. La deuxième contrainte est liée au lien d'agrégation défini entre le niveau inférieur et le niveau créé. Chaque sous-ensemble d'instances de cette partition est associé à une et une seule instance du niveau créé. Les données concernant chaque instance du niveau inférieur pourront alors être agrégées en une instance du niveau créé. Ainsi la mise en correspondance entre les deux niveaux revient à l'application d'une fonction bijective entre les sous-ensembles de la partition du niveau inférieur et les instances du niveau créé. Nous conservons la notation "si-alors" dans le texte, mais nous emploierons son équivalent anglais «*if-then*» dans les formalisations ou les exemples.

3.5.2 Formalisation

Dans cette section, nous présentons formellement le modèle *R-DW*.

Définition 3.1 *Modèle d'entrepôt évolutif R-DW*
Le modèle d'entrepôt évolutif à base de règles *R-DW* est défini par le triplet suivant : $R\text{-}DW = (\mathcal{F}, \mathcal{E}, \mathcal{U})$ où $\mathcal{F}$ est la partie fixe, $\mathcal{E}$ la partie évolutive et $\mathcal{U}$ l'univers de *R-DW*.

Définition 3.2 *Univers de l'entrepôt*
Soit l'entrepôt $R\text{-}DW = (\mathcal{F}, \mathcal{E}, \mathcal{U})$. L'*univers* $\mathcal{U}$ de *R-DW* est un ensemble d'attributs tel que : $\mathcal{U} = \mathcal{U}_1 \cup \mathcal{U}_2$ où $\mathcal{U}_1 = \{B_\alpha, 1 \leq \alpha \leq z\}$ est l'ensemble des z attributs existants dans le schéma initial de l'entrepôt et $\mathcal{U}_2 = \{C_\beta, \beta \geq 1\}$ est l'ensemble des attributs générés, présents dans la partie évolutive $\mathcal{E}$ de *R-DW*.

Définition 3.3 *Partie fixe de R-DW*
La *partie fixe de* $R\text{-}DW = (\mathcal{F}, \mathcal{E}, \mathcal{U})$ est définie par $\mathcal{F} =< F, \mathcal{D} >$, où F est une table de faits, et $\mathcal{D} = \{D_s, 1 \leq s \leq t\}$ est l'ensemble des t dimensions de premier niveau qui ont un lien direct avec F. Nous supposons que ces dimensions sont indépendantes.

Exemple 3.1

Dans l'entrepôt simplifié de LCL (Figure 2.2) présenté dans le Chapitre 2, <PERFORMANCES, {AGENCE, TEMPS, CLIENT}> constitue la partie fixe de l'entrepôt *R-DW* pour l'analyse du PNB.

Définition 3.4 *Hiérarchie de dimension et niveau de granularité*
Soit $R\text{-}DW = (< F, \mathcal{D} >, \mathcal{E}, \mathcal{U})$ un entrepôt de données évolutif. On note $D_s.H_k, D_s \in \mathcal{D}, k \geq 1$ une *hiérarchie de la dimension* D_s. La hiérarchie de dimension $D_s.H_k$ est composée d'un ensemble de w niveaux de granularité ordonnés notés L_i : $D_s.H_k = \{L_0, L_1, \ldots, L_i, \ldots, L_w, w \geq 0\}$ avec $L_0 \prec L_1 \prec \cdots \prec L_i \prec \cdots \prec L_w$ où $\prec$ exprime l'ordre total sur les L_i.

Le *niveau de granularité* L_i de la hiérarchie H_k de la dimension D_s est noté $D_s.H_k.L_i$ ou plus simplement L_i^{sk}. L_0^{sk} correspond au premier niveau de la hiérarchie, il s'agit de la dimension elle-même.

Exemple 3.2

Selon le schéma de la Figure 3.1, on a :
$D_{\text{AGENCE}}.H_1$={AGENCE, UC, DPP, DE} et $D_{\text{AGENCE}}.H_2$={AGENCE, TypeAg}

Définition 3.5 *Partie évolutive de R-DW.*
La *partie évolutive de* $R\text{-}DW = (\mathcal{F}, \mathcal{E}, \mathcal{U})$ est l'ensemble des hiérarchies de dimension de

l'entrepôt, privé des niveaux correspondant aux dimensions elles-mêmes ; autrement dit c'est l'ensemble des différents niveaux de granularité composant ces hiérarchies à partir du niveau 1 : $\mathcal{E} = \{D_s.H_k\} - \{L_0^{sk}\} = \{L_i^{sk}\},\ avec\ 1 \leq s \leq t,\ k \geq 1,\ i > 0$

Notons qu'ici, i et k ne sont pas fixés a priori puisque le principe de notre modèle est justement que de nouveaux niveaux de granularité peuvent être créés dans les hiérarchies existantes et de nouvelles hiérarchies peuvent être créées également via la création de niveaux de granularité au-dessus d'une dimension.

Définition 3.6 *Attribut généré.*
Soient $R\text{-}DW = (< F, \mathcal{D} >, \mathcal{E}, \mathcal{U})$, L_i^{sk} le niveau de granularité L_i de la hiérarchie H_k de la dimension D_s et $\mathcal{U}_2$ l'ensemble des attributs générés dans la partie évolutive $\mathcal{E}$ de $R\text{-}DW$. Le niveau de granularité L_i^{sk} peut alors être créé et est alors défini par un ensemble de λ *attributs générés* qui décrivent ce niveau : $L_i^{sk}.A\ avec\ A = \{a_\delta, 1 \leq \delta \leq \lambda, a_\delta \in \mathcal{U}_2\}$

Exemple 3.3

Selon l'exemple de l'entrepôt LCL, on a : $L_2^{\text{AGENCE }2}.A = \{Nom\_TypeAg\}$

Définition 3.7 *Termes d'une règle d'agrégation.*
Une *règle d'agrégation* permet de définir le lien d'agrégation qui existe entre deux niveaux de granularité dans une hiérarchie de dimension. Elle est basée sur un ensemble $\mathcal{T}$ de n termes de règles, notés RT_p, tel que : $\mathcal{T} = \{RT_p, 1 \leq p \leq n\}$. Chaque règle RT_p relève de la forme générique suivante : $u\ op\ \{ens|val\}$ où u est un attribut de l'univers $\mathcal{U}$ de l'entrepôt ; op est un opérateur ($=, <, \leq, \geq, \neq, \in, ...$) ; ens est un ensemble de valeurs et val est une valeur finie.

Exemple 3.4

$RT_1 : IdAg \in \{$'01903','01905','02256'$\}$; $RT_2 : Age > 80$

Définition 3.8 *Règle d'agrégation.*
Une règle d'agrégation est une règle de type «si-alors». La conclusion de la règle (clause «alors») définit la valeur des attributs générés à l'aide de conjonctions d'égalité. La prémisse de la règle (clause «si») est basée sur des conjonctions de termes de règles :

r_{ij} : $if\ RT_1\ and\ ...\ and\ RT_p\ and\ ...\ and\ RT_n$
$then\ L_i^{sk}.a_1 = val_1^{ij}\ and\ ...\ and\ L_i^{sk}.a_\delta = val_\delta^{ij}\ and\ ...\ and\ L_i^{sk}.a_\lambda = val_\lambda^{ij}$
où $val_\delta^{ij} \in Dom_{L_i^{sk}.a_\delta}$ est le domaine de définition de l'attribut $L_i^{sk}.a_\delta$.

Exemple 3.5

Les règles suivantes déterminent les valeurs des attributs ClasseAge et ClasseAgeDescription qui caractérisent le niveau de granularité AGE construit au-dessus de la dimension CLIENT :

r_{11} : *if* $Age < 18$
then $AgeClassDescription =$ '*mineur*' *and* $ClasseAge =$ '< 18 ans'
r_{12} : *if* $Age \geq 18$
then $DescriptionClasseAge =$ '*majeur*' AND $ClasseAge =$ 'plus que 18 ans' ———

Le principe du lien d'agrégation est que les règles construisent une partition des instances du niveau inférieur. Pour satisfaire cette contrainte, nous définissons trois propriétés sur les règles. La propriété 1 a pour but d'exprimer le fait que les sous-ensembles d'instances du niveau inférieur définis par les clauses «si» des règles d'agrégation ne sont pas vides. La propriété 2 a pour but d'exprimer le fait que les sous-ensembles d'instances du niveau inférieur définis par les clauses «si» des règles d'agrégation sont deux à deux disjoints, autrement dit, l'intersection des ces sous-ensembles pris deux à deux doit être vide. La propriété 3 a pour but d'exprimer le fait que l'union des sous-ensembles d'instances du niveau inférieur définis par les clauses «si» des règles d'agrégation correspond à l'ensemble initial de toutes les instances de ce niveau.

Propriété 1 *Soit $L_i^{sk}.A$ l'ensemble des attributs générés qui caractérisent le niveau de granularité créé L_i de la hiérarchie H_k de la dimension D_s.*
Soit $\mathcal{R}_{i.}^{sk} = \{r_{ij}, 1 \leq j \leq v\}$ l'ensemble des v règles d'agrégation définissant les valeurs de l'ensemble des attributs générés $L_i^{sk}.A$. Chaque clause «si» des règles de $\mathcal{R}_{i.}^{sk}$ définit un ensemble d'instances I_{ij} dans le niveau inférieur L_{i-1}^{sk}. On a alors : $\forall i, \forall j, I_{ij} \neq \emptyset$

Propriété 2 *Soit $L_i^{sk}.A$ l'ensemble des attributs générés qui caractérisent le niveau de granularité L_i de la hiérarchie H_k de la dimension D_s.*
Soit $\mathcal{R}_{i.}^{sk} = \{r_{ij}, 1 \leq j \leq v\}$ l'ensemble des v règles d'agrégation définissant les valeurs de l'ensemble des attributs générés $L_i^{sk}.A$. Chaque clause «si» des règles de $\mathcal{R}_{i.}^{sk}$ définit un ensemble d'instances I_{ij} dans le niveau inférieur L_{i-1}^{sk}.
On a alors : $\forall i, \forall j, q$ tels que $j < q, j \in [1, v-1], q \in [2, v]$, $I_{ij} \bigcap I_{iq} = \emptyset$

Propriété 3 *Soit $L_i^{sk}.A$ l'ensemble des attributs générés qui caractérisent le niveau de granularité L_i de la hiérarchie H_k de la dimension D_s.*
Soit $\mathcal{R}_{i.}^{sk} = \{r_{ij}, 1 \leq i \leq w, 1 \leq j \leq v\}$ l'ensemble des v règles d'agrégation définissant les valeurs de l'ensemble des attributs générés $L_i^{sk}.A$. Chaque clause «si» des règles de $\mathcal{R}_{i.}^{sk}$ définit un ensemble d'instance I_{ij}.
Le niveau L_{i-1}^{sk} sur lequel est construit le niveau L_i^{sk} comprend un ensemble d'instances noté I_{i-1}^{ini}. On a alors : $\forall i, \bigcup_{j=1}^{v} I_{ij} = I_{i-1}^{ini}$

Chaque sous-ensemble d'instances du niveau inférieur est mis en correspondance avec une et une seule instance du niveau créé. Ainsi, nous définissons une fonction de correspondance qui détermine le lien d'agrégation entre le niveau créé et le niveau inférieur sur lequel il est basé.

Définition 3.9 *Fonction de correspondance.*
Soit $\mathcal{C}$ une fonction de correspondance de $\mathcal{I}^{inf}$ dans $\mathcal{I}^{cre}$ où $\mathcal{I}^{cre}$ désigne l'ensemble des instances du niveau créé et $\mathcal{I}^{inf}$ désigne l'ensemble des instances du niveau inférieur. La fonction $\mathcal{C}$ a une propriété de bijectivité au sens où nous la décrivons. On a alors : $\forall \iota \in \mathcal{I}^{cre}, \exists! \Theta \subset \mathcal{I}^{inf}, \mathcal{C}(\Theta) = \iota$

3.5.3 Version utilisateur

Dans notre approche, nous supposons que les utilisateurs disposent d'un entrepôt de donnée initial. Nous leur offrons la possibilité de créer de nouveaux axes d'analyse. Comme nous l'avons détaillé précédemment, ils peuvent exprimer de nouveaux besoins d'analyse en fonction de leurs propres connaissances : connaissance du domaine, objectif métier, etc.

Reprenons le cas de l'entreprise LCL. Il est possible que deux utilisateurs aient besoin d'analyser le PNB en fonction de l'âge des clients. Selon la fonction qu'ils occupent au sein de l'entreprise et les objectifs qu'ils se sont fixés, la définition des classes d'âge n'est pas la même pour les deux utilisateurs. L'un peut vouloir se baser sur deux catégories d'âge : les plus et les moins de 60 ans, parce qu'il travaille dans le service marketing dédié aux produits d'épargne de retraite. L'autre, qui travaille dans le service dédié aux offres étudiantes, doit par exemple distinguer les mineurs des majeurs dans ses analyses.

Ainsi nous devons faire face à des besoins d'analyse identique, en l'occurrence la définition de classes d'âge, mais avec des sémantiques différentes. Ainsi, nous avons proposé de gérer des versions de règles différentes. Cela correspond à gérer des versions de hiérarchies différentes [FBB06b, BFB08].

Pour cela, nous avons introduit la notion de version de niveau de granularité qui se greffe au schéma de façon parallèle. Dans le cas d'une analyse en fonction de ce niveau, il faudra donc choisir la version qui comprend la définition souhaitée. Notons qu'il s'agit ici de considérer des versions qui dépendent d'utilisateurs différents. Nous ne traitons pas ici de versions temporelles.

Définition 3.10 *Version de niveau de granularité*
Soit L_i^{sk} le niveau de granularité L_i de la hiérarchie H_k de la dimension D_s qui peut être défini par différents utilisateurs. Une version de ce niveau est noté : $L_i^{sk}(v_c),\ \ c \geq 1$ où v_c représente le numéro de la version. Si une seule version existe, nous adoptons la première notation : L_i^{sk}.

Exemple 3.6

Supposons que $L_2^{\text{CLIENT }2}$ correspond au niveau ClasseAge de la dimension CLIENT. Deux utilisateurs définissent différemment ce niveau, nous notons : $L_2^{\text{CLIENT }2}(v_1)$ et $L_2^{\text{CLIENT }2}(v_2)$.

3.6 Méta-modèle d'entrepôts de données évolutifs

Afin de pouvoir appliquer notre démarche sur n'importe quel entrepôt de données, nous avons conçu un méta-modèle qui permet de représenter le schéma logique de l'entrepôt de données évolutif (Figure 3.3).

Ce méta-modèle permet d'assurer la généricité de notre modèle d'exécution. En effet, une fois mis en œuvre, il permet de gérer plusieurs entrepôts de données évolutifs.

L'interprétation de ce méta-modèle est la suivante. Un entrepôt de données évolutif est décrit comme un ensemble de tables. Ces tables sont soit des tables de dimension, soit des tables de faits. Chaque table de dimension possède un ou plusieurs niveaux de granularité qui forment ainsi une hiérarchie de dimension. Chaque niveau possède un ensemble d'attributs et une clé primaire. Chaque niveau de granularité peut être généré par un ensemble de règles d'agrégation. Donc chaque niveau correspond soit à un ensemble d'attributs explicites, soit à des attributs générés avec des règles. Parallèlement, les tables de faits présentent une ou plusieurs mesures et une clé primaire qui est une composition des clés étrangères correspondant aux clés primaires des tables de dimension qui leurs sont directement reliées.

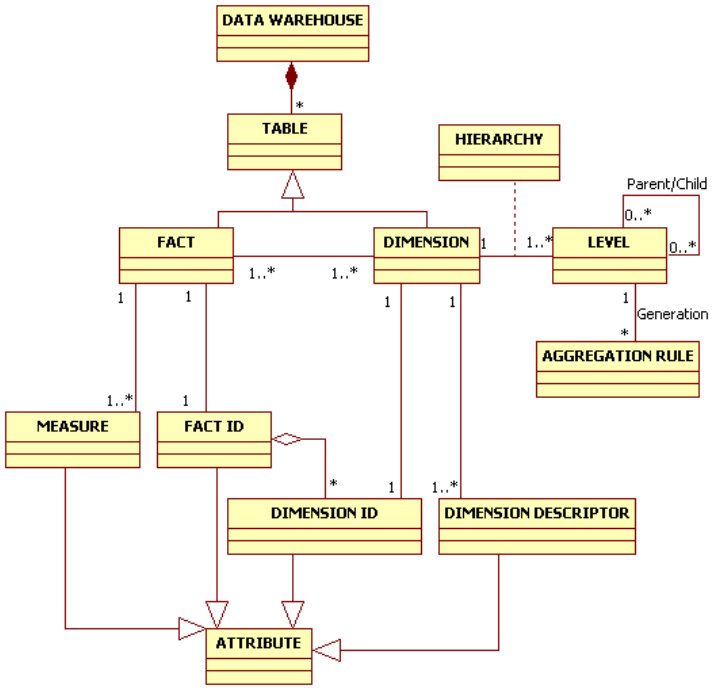

FIGURE 3.3 – Méta-modèle d'entrepôts de données évolutifs

3.7 Mise à jour de hiérarchies de dimension

Nous présentons dans cette section notre approche de mise à jour de hiérarchies de dimension pour la personnalisation des analyses dans les entrepôts de données de façon globale. Pour plus de détails, le lecteur intéressé pourra se reporter à la thèse de doctorat de C. Favre [Fav07].

Dans les bases de données, l'évolution de schéma peut être vue comme la mise en œuvre d'opérations élémentaires d'ajout, de suppression et de modification. Dans le modèle relationnel par exemple, ces opérations sont appliquées au niveau des tables ou des attributs. Ainsi les concepts subissant les opérations ont tous le même rôle ; ce qui n'est pas le cas dans les entrepôts de données. En effet, la sémantique portée par le schéma d'un entrepôt de données induit des considérations différentes sur ses concepts et sur leur évolution a fortiori. De ce fait, lorsque l'on parle d'évolution de schéma dans un entrepôt de données, on ne peut pas se baser sur les distinctions adoptées dans les bases de données. Il nous faut distinguer la mise à jour des tables de faits, de celle des tables de dimension, ou encore de celle des hiérarchies de dimension.

Nous avons proposé un algorithme de création et de suppression de niveaux de granularité dans une hiérarchie de dimension. Nous avons également proposé un algorithme qui permet de gérer la propagation des mises à jour dans une hiérarchie. En effet, lorsqu'un niveau est créé ou supprimé en milieu de hiérarchie, il est nécessaire d'effectuer une propagation de la mise à jour sur toutes les hiérarchies concernées afin d'assurer la cohérence du schéma et des instances.

Lorsque l'utilisateur veut créer un nouveau niveau de granularité dans une hiérarchie de dimension, il définit tout d'abord une méta-règle. Celle-ci décrit la structure du lien d'agrégation entre deux niveaux. Puis, il définit les règles d'agrégation qui décrivent le lien entre les instances du niveau existant et le nouveau niveau à créer.

Pour déployer notre approche, nous avons proposé un modèle d'exécution indépendant de toute configuration logicielle [FBB07a]. Dans le contexte de notre collaboration avec l'entreprise LCL, nous avons utilisé l'environnement relationnel en suivant l'approche ROLAP (*Relational OLAP*). Ce modèle d'exécution permet de mettre en œuvre les différents modules de notre architecture globale. Nous avons implémenté le méta-modèle d'entrepôt qui permet de décrire différents schémas d'entrepôt dans le SGBD Oracle 10g [2]. Ensuite, nous avons développé un prototype portant le même nom que l'architecture décisionnelle WEDriK (cf. Section 3.4), selon une configuration client/serveur [FBB07a]. Pour mettre en œuvre la personnalisation des analyses, l'utilisateur interagit avec le système via une interface web.

2. http ://www.oracle.com

3.8 Conclusion

Nous avons présenté dans ce chapitre une approche tendant à personnaliser les analyses utilisateurs dans le contexte des systèmes d'information décisionnels. Notre contribution dans ce domaine porte principalement sur l'intégration interactive des connaissances utilisateurs au sein de l'entrepôt en empruntant le concept d'évolution de schéma. Cette approche nous a permis de remettre l'utilisateur au cœur du système décisionnel en lui permettant d'enrichir les possibilités d'analyse de l'entrepôt au fur et à mesure que ses besoins évoluent.

Nous avons conçu pour cela une architecture décisionnelle évolutive. Pour soutenir cette architecture, nous avons défini un modèle d'entrepôt de données à base de règles d'agrégation R-DW. Ce modèle est composé d'une partie «fixe» et d'une partie «évolutive». La partie fixe est constituée de la table des faits et des tables de dimension qui lui sont directement reliées. Elle constitue une réponse à des besoins d'analyse initiaux, définis lors de la conception de l'entrepôt. La partie évolutive est composée de l'ensemble des hiérarchies de dimension pouvant être mises à jour par les utilisateurs. Les algorithmes de mise à jour que nous avons proposés assurent la cohérence des données et celle des analyses.

Nous avons par ailleurs conçu notre approche de manière générique. Pour cela, nous avons proposé un méta-modèle d'entrepôts de données évolutifs qui nous permet d'appliquer notre démarche sur tout entrepôt de données. La démarche que nous avons proposée n'est en effet liée à aucun système en particulier. Elle peut être appliquée sur n'importe quel SGBD hôte. Nous avons déployé notre approche dans un contexte relationnel en proposant un modèle d'exécution qui a pour but de gérer l'ensemble des processus liés à l'architecture, de l'acquisition des règles à l'évolution du schéma.

L'entreprise LCL a non seulement suscité notre problématique de personnalisation, mais a constitué par la suite un terrain d'application pour la mise en œuvre de nos propositions. Ainsi, l'ensemble de nos propositions a fait l'objet d'une implémentation au travers de la plate-forme WEDriK sur l'entrepôt de données test LCL-DW construit à partir de données réelles de LCL.

L'aboutissement de notre approche de personnalisation des analyses dans les entrepôts de données permet de prendre réellement en compte les nouveaux besoins des utilisateurs. Elle s'inscrit dans une perspective différente des travaux émergents dans le domaine. En effet, les approches existantes se basent sur l'expression de préférences pour personnaliser le processus d'analyse en filtrant les réponses aux requêtes [BGMM06, BGM$^+$05] ou en optimisant le nombre d'opérations à réaliser lors de la navigation dans les données [RTZ07]. Notre approche tente alors d'étendre le domaine de la personnalisation dans les entrepôts pour permettre d'intégrer au sein de l'entrepôt des connaissances nouvelles émanant des

utilisateurs afin d'enrichir les possibilités d'analyse.

Notre approche se rapproche des travaux de Blaschka et al. [BSH99], qui proposent un ensemble d'opérateurs élémentaires (ajouter un niveau, ajouter un attribut, connecter un attribut à un niveau, etc.) dans une hiérarchie de dimension. En combinant ces opérateurs, il est possible de réaliser l'ensemble des mises à jour du schéma. Néanmoins, ce travail consiste à représenter ces opérateurs au niveau structurel alors que dans notre approche nous exploitons les connaissances des utilisateurs pour l'évolution de schéma. Nous pouvons également rapprocher nos travaux de ceux de Mazon et al. [MT06] en termes d'objectif commun : enrichir les possibilités d'analyse des entrepôts de données en étendant les hiérarchies de dimension. Les deux approches diffèrent néanmoins sur la méthode utilisée. En effet, alors que notre approche intègre les connaissances des utilisateurs, l'approche de Mazon et al. exploite certaines relations (hypéronymie et méronymie) de WordNet. Par ailleurs, notre approche permet de créer de nouveaux chemins d'agrégation, allant au-delà de la proposition faite dans [EV01], dans laquelle les utilisateurs peuvent seulement modifier les chemins d'agrégation existants, en exprimant des exceptions dans le processus d'agrégation au niveau des instances. C'est précisément cette limite que nous dépassons grâce à notre approche. Enfin, notre approche de personnalisation basée sur l'évolution de schéma permet de rendre les nouvelles possibilités d'analyse pérennes et partageables avec d'autres utilisateurs, ce qui n'est pas le cas avec les propositions faites par certains éditeurs de logiciels décisionnels.

A la faveur des discussions que nous avons pu mener sur notre travail tout au long de ce chapitre, nous avons pu faire émerger plusieurs perspectives, directement liées à notre travail sur la personnalisation, ou de façon plus large.

Tout d'abord, un des points cruciaux qu'il nous reste à explorer dans le cadre de notre proposition est la gestion de l'évolution des règles. Si ce problème peut être abordé sous l'angle de l'évolution de schéma de façon générale avec les approches que l'on connaît de mise à jour ou de versionnement, il n'en demeure pas moins que les particularités de notre approche doivent être prises en compte. L'une des particularités les plus notables est l'implication de l'utilisateur dans le processus de mise à jour des hiérarchies de dimension. Il s'agit alors de connaître quels sont les besoins réels au niveau de l'historisation des dimensions.

D'autre part, lorsque le nombre d'instances à identifier lors du regroupement est trop important, la tâche qui incombe à l'utilisateur devient fastidieuse. Dans ce cas, l'utilisation d'une méthode d'apprentissage permettant un regroupement automatique des instances paraît pertinente. Cette méthode permettrait également de découvrir des regroupements intéressants pour l'analyse auxquels l'utilisateur n'aurait pas pensé. Nous avons d'ores et déjà mené des travaux dans ce sens que nous présentons dans le Chapitre 4 et que nous comptons poursuivre dans le futur.

Par ailleurs, lorsqu'un utilisateur crée de nouveaux axes d'analyse, ces derniers peuvent intéresser d'autres utilisateurs. Il est donc crucial qu'un utilisateur, qui réalise une analyse en fonction d'un niveau créé par un autre utilisateur, connaisse exactement la sémantique de ce niveau. Pour ce faire, nous pensons que le recours à un processus d'annotations, comme il a pu être proposé dans [CCRT07], peut être pertinent. En effet, dans ce travail les auteurs traitent du concept de mémoire d'expertises décisionnelles. Un des objectifs de cette mémoire est d'éviter la perte de connaissances lors du départ d'un collaborateur et de faciliter le transfert de ces connaissances entre les collaborateurs. Deux aspects ont retenu plus particulièrement notre attention dans cette proposition. Il s'agit d'une part de l'idée de préciser la sémantique au niveau des concepts dans le schéma. D'autre part, il s'agit de l'idée d'usage collectif, de partage d'expertises. Ainsi, notre approche présente ces deux idées : il est en effet crucial de pouvoir préciser la sémantique du niveau créé dans le schéma afin de pouvoir partager cette possibilité d'analyse supplémentaire avec d'autres utilisateurs. Le créateur du nouveau niveau de hiérarchie pourrait annoter celui-ci afin de lui donner une bonne description, pour que la compréhension soit facilitée pour les autres utilisateurs. C'est ce qui permettra un réel partage des nouvelles possibilités d'analyse en assurant la bonne interprétation de ces analyses. Rappelons que cette nécessité est accentuée dans le cas de versions utilisateurs différentes qui consistent à représenter un même niveau avec des règles de construction différentes.

Pour terminer, comme notre modèle d'entrepôt est indépendent de toute implémentation logicielle, il est intéressant d'exploiter notre modèle dans le cadre de données complexes. Nous avons d'ores et déjà proposé une adaptation de notre modèle dans le cadre des entrepôts XML [BMM+11]. Dans ce cas, l'évolution de schéma concerne la mise à jour de documents XML.

3.9 Publications

La liste suivante présente nos publications relatives à la personnalisation des analyses par évolution de schéma.

Revue internationale

[1] **F. Bentayeb**, C. Favre, O. Boussaid, "A User-driven Data Warehouse Evolution Approach for Concurrent Personalized Analysis Needs", Integrated Computer-Aided Engineering (ICAE), Vol.15, No 1, 2008, 21-36.

Revue nationale

[2] C. Favre, M. Rougié, **F. Bentayeb**, O. Boussaid, "Gestion et analyse personnalisées des demandes marketing : cas de LCL-Le Crédit Lyonnais" ; Revue Ingénierie des Systèmes d'Information (RSTI série ISI), Numéro spécial Usage et Conception des SI : Prise en Compte de l'Utilisateur, Vl. 14, No. 3, 2009, 119-139.

Chapitre d'ouvrage d'audience internationale

[3] **F. Bentayeb**, C. Favre, O. Boussaid, "Dynamic Workload for Schema Evolution in Data Warehouses : a Performance Issue" ; Complex Data Warehousing and Knowledge Discovery for Advanced Retrieval Development : Innovative Methods and Applications, Series Advances in Data Warehousing and Mining, chapter 02, 28-46, IGI Publishing, 2010.

Conférences internationales

[4] C. Favre, **F. Bentayeb**, O. Boussaid, "Evolution of Data Warehouses' Optimization : a Workload Perspective", 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 07), Regensburg, Germany, September 2007 ; LNCS, Vol. 4654, Springer, 13-22.

[5] C. Favre, **F. Bentayeb**, O. Boussaid, "A Rule-based Data Warehouse Model", 23rd British National Conference on Databases (BNCOD 06), Belfast, Northern Ireland, July 2006 ; LNCS, Vol. 4042, Springer, 274-277.

[6] C. Favre, **F. Bentayeb**, O. Boussaid, " A Knowledge-driven Data Warehouse Model for Analysis Evolution", 13th ISPE International Conference on Concurrent Engineering : Research and Applications (CE 06), Antibes, France, September 2006 ; Frontiers in Artificial Intelligence and Applications, Vol. 143, IOS Press, 271-278.

Conférences nationales

[7] C. Favre, **F. Bentayeb**, O. Boussaid, "Evolution de modèle dans les entrepôts de données : existant et perspectives", 3èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 07), Poitiers, Juin 2007 ; Revue des Nouvelles Technologies de l'Information, Vol. B-3, Cépaduès, 21-36.

[8] C. Favre, **F. Bentayeb**, O. Boussaid, "Intégration des connaissances utilisateurs pour des analyses personnalisées dans les entrepôts de données évolutifs", 7èmes journées francophones d'Extraction et de Gestion des Connaissances (EGC 07), Namur, Belgique, Janvier 2007 ; Revue des Nouvelles Technologies de l'Information, Vol. E-9, Cépaduès, 217-222.

[9] C. Favre, **F. Bentayeb**, O. Boussaid, "Evolution de schémas dans les entrepôts de données : modèle à base de règles", 2ème journée francophone sur les Entrepôts de Données et l'Analyse en ligne (EDA 06), Versailles, Juin 2006 ; Revue des Nouvelles Technologies de l'Information, Vol. B-2, Cépaduès, Toulouse, 175-176.

Workshop national

[10] C. Favre, **F. Bentayeb**, O. Boussaid, "Modèle d'entrepôt de données à base de règles", 3ème atelier Fouille de Données Complexes dans un processus d'extraction des connaissances, EGC 06, Lille, 2006, 39-50.

Chapitre 4

Recommandation de niveaux de hiérarchies dans une dimension

L'objectif de ce chapitre est tout d'abord de motiver la nécessité de recommander à l'utilisateur des requêtes d'analyse dans un contexte décisionnel. Nous présentons ensuite notre approche de recommandation de nouveaux niveaux de hiérarchies dans une dimension qui s'inscrit dans la continuité de nos travaux sur la personnalisation des analyses. Même si la recommandation et la personnalisation des analyses diffèrent dans leur objectif, leur point commun dans le cadre de nos travaux est le recours à l'évolution de schéma dans les entrepôts de données. En outre, la particularité de nos travaux sur la recommandation de requêtes réside dans l'utilisation de la fouille de données afin d'extraire des connaissances cachées pouvant servir à construire de nouveaux axes d'analyse sémantiquement plus riches.

Ce travail a en partie fait l'objet du stage de master recherche de O. Rakotoarivelo que nous avons personnellement encadré. Ce stage a été réalisé au sein du laboratoire ERIC et soutenu en 2006 [RB07, Ben08, BF09].

4.1 Motivation

Les systèmes d'aide à la décision synthétisent l'information et permettent aux utilisateurs d'explorer leurs données pour en extraire des informations pertinentes. L'exploration de données est un processus de recherche d'informations destiné à détecter des corrélations cachées ou des informations nouvelles. Or, les utilisateurs doivent faire face à un volume de plus en plus important d'informations en raison de l'accroissement des capacités de stockage et de calcul. Par conséquent, il est de plus en plus difficile de savoir exactement quelles informations rechercher et où les chercher. Dans ce contexte, le recours à des

techniques informatiques plus sophistiquées que la seule analyse OLAP s'avèrent nécessaires pour aider l'utilisateur dans sa tâche d'exploration des données afin de lui faciliter la recherche d'informations pertinentes. L'une de ces techniques est la recommandation d'informations et, plus particulièrement la recommmandation de requêtes décisionnelles.

Dans la littérature, une requête recommandée est une requête existante (ou calculée) issue d'un ensemble de requêtes posées sur l'entrepôt (fichier log de requêtes par exemple). Notre avis sur cette définition est qu'elle est très restrictive puisqu'elle s'appuie uniquement sur les requêtes utilisateurs déjà posées et exclut de ce fait le contenu de l'entrepôt lui-même. Aussi, le regard que nous portons personnellement sur la question de la recommandation dans les entrepôts est plus large que la définition proposée dans la littérature. Nous pensons alors qu'un système de recommandation doit pouvoir proposer à un utilisateur des points de vue et des angles d'analyse nouveaux basés sur les connaissances pouvant être extraitee à partir des données de l'entrepôt. Pour cela, notre idée porte pricipalement sur l'application des techniques de fouille sur les données de l'entrepôt afin de découvrir de nouveaux axes d'analyse permettant de réaliser de nouvelles requêtes décisionnelles potentiellement pertinentes pour l'utilisateur. Si bien que nous étendons la notion de recommandation de requêtes à la notion de recommandation d'axes d'analyse.

4.2 Principe général

La définition et la construction d'un cube de données cible un contexte d'analyse bien précis. Le choix des dimensions et des mesures dépend des besoins de l'analyste. D'une manière générale, une dimension est organisée sur plusieurs hiérérchies traduisant différents niveaux de garnularité. Chaque niveau d'une hiérarchie comporte un ensemble de modalités (valeurs), et chaque modalité d'un niveau d'une hiérarchie agrège des modalités du niveau de la hiérarchie immédiatement inférieur selon un ordre d'appartenance logique. Par exemple, une dimension temporelle peut être structurée selon trois niveaux hiérarchiques : *jour*, *mois* et *année*, et une dimension *article* peut être structurée selon deux niveaux *produit* et *catégorie produit*.

La navigation [SAM98] est un terme utilisé pour caractériser le fait qu'un utilisateur explore de manière interactive un cube de données pour obtenir des résultats intéressants pour l'aide à la décision. La démarche exploratoire de l'OLAP suppose une expertise suffisante de l'utilisateur qui lui permettra de découvrir des informations pertinentes au regard de ses besoins d'analyse. D'après [Sar00], une analyse pilotée par la découverte (*Discovery driven analysis*) démarre typiquement au niveau le plus haut des hiérarchies de dimensions du cube. Puis, la navigation dans le cube se fait en appliquant une séquence d'opérations de *Drill-down* (forage vers le bas), *Roll-up* (agrégation), *Slice* (projection), et *Dice* (sélection).

L'utilisateur soumet une première requête à l'entrepôt en selectionnant des mesures, un ensemble de dimensions avec le niveau de granularité souhaité avec des contraintes de filtrage. Puis il affine sa requête intéractivement en ajoutant ou en supprimant des niveaux de hiérarchie (*roll-up* et *drill-down*), en ajoutant ou modifiant des conditions de filtrage (*slice*) ou enfin en déplaçant des colonnes d'un axe à l'autre (*dice*). A partir du plus haut niveau de la hiérarchie, le décideur observe un niveau plus bas d'une hiérarchie en regardant les valeurs agrégées et en identifiant visuellement des valeurs intéressantes. Si une exploration ne donne pas de résultats intéressants, le décideur remonte au niveau le plus haut des hiérarchies des dimensions du cube et continue son analyse dans une autre direction, en allant observer d'autres dimensions. Le but de ces manipulations est de pouvoir découvrir des aspects insoupçonnables dans la grande masse de données de l'entrepôt permettant ainsi l'affinement de l'analyse exploratoire du décideur.

Toutefois, dans un processus décisionnel, un utilisateur peut vouloir anticiper la réalisation d'évènements futurs. Malheureusement, la technologie OLAP offre des possibilités pour visualiser des faits décrits par des indicateurs et des axes d'analyse, mais ne permet pas de décrire l'ordre d'importance ou les relations possibles entre ces faits. L'OLAP ne permet pas non plus de classifier ou de regrouper les faits selon un ordre de proximité sémantique et ne dispose pas non plus de moyens pour expliquer les associations ou les implications entre ces faits. De ce constat, sont nés des travaux combinant la fouille de données avec l'OLAP afin de renforcer le processus d'aide à la décision, notamment en vue d'étendre les capacités de l'OLAP vers l'explication ou la prédiction [SAM98, CRST06, Mes06]. Cependant, ces travaux ne se sont peu ou pas du tout intéressés à impliquer l'utilisateur dans ce processus.

Par conséquent, pour une meilleure prise en compte de l'utilisateur dans le système décisionnel, nous soulignons l'intérêt de guider l'utilisateur vers les faits les plus intéressants du cube ou de l'aider à découvrir de nouveaux axes d'analyse non prévus initialement par le schéma de l'entrepôt. Nous avons pensé alors que le couplage de la fouille de données avec la technologie OLAP pouvait permettre d'assister l'utilisateur dans l'extraction de nouvelles connaissances à partir de l'entrepôt en lui proposant des pistes d'analyse non explorées. Il s'agit par exemple d'appliquer une méthode de fouille sur une partie de l'entrepôt sélectionnée par l'utilisateur, en extraire des connaissances pour enfin les réinjecter dans l'entrepôt sous une forme facilement exploitable par l'utilisateur. Ce dernier pourra ainsi appliquer des opérateurs OLAP sur l'entrepôt enrichi.

Pour aider l'utilisateur dans son processus d'exploration et de découverte, nous lui proposons un cadre de structuration-OLAP fondé à la fois sur la philosophie OLAP et sur la fouille de données dans le but de lui recommander de nouveaux axes d'analyse. Ces derniers lui ouvrent de nouvelles perspectives d'analyse en termes d'exploration et de navigation par la biais de requêtes OLAP. Pour atteindre cet objectif, notre approche s'appuie sur

le concept d'évolution de schéma. Plus précisément, notre approche se positionne dans le courant des travaux qui proposent des opérateurs permettant de faire évoluer la structure hiérarchique d'une dimension. Toutefois, notre originalité réside dans l'utilisation de la fouille de données pour réaliser cette évolution. En effet, en ayant recours à la méthode d'apprentissage non supervisé des k-means qui permet de découvrir de nouvelles structures "naturelles" ; ces dernières peuvent être exploitées pour la génération d'un nouveau niveau d'analyse dans une hiérarchie de dimension.

4.3 RoK : un opérateur d'agrégation basé sur les k-means

4.3.1 Idée générale

La démarche classique de l'OLAP commence par la sélection des mesures et des axes d'analyse qui sont susceptibles de répondre au besoin d'analyse de l'utilisateur. Une fois que le cube de données associé à ce besoin est construit, l'utilisateur va explorer ce cube pour tenter de déceler rapidement des similarités entre les faits selon les axes qu'il étudie. Ce sont les différents niveaux de granularité dans les hiérarchies de dimensions qui permettent de détecter et d'apprécier ces similarités. De son côté, la fouille de données propose des méthodes de classification automatique pour regrouper les individus (au sein d'une population) possédant des caractéristiques similaires (deux individus sont considérés comme similaires lorsque les valeurs des variables qui les décrivent sont proches). De ce point de vue, notre idée est alors de combiner l'analyse en ligne avec la fouille de données pour créer un nouvel opérateur d'agrégation capable de créer un nouveau niveau de hiérarchie, structurellement au sens de l'OLAP mais sémantiquement plus riche.

Nous considérons dans ce travail les liens d'agrégation, au sein d'une hiérarchie d'une dimension, représentés par les hiérarchies classiques, qualifiées de hiérarchies symétriques strictes. Dans ce cas, notre opérateur doit pouvoir regrouper les instances du niveau inférieur niv_{inf} en un ensemble de classes formant une partition. Chaque classe obtenue sera ensuite considérée comme une modalité (valeur) du nouveau niveau de hiérarchie à créer niv_{sup}. Ainsi, l'association entre chaque individu de niv_{inf} et leur classe d'affectation constituera le lien d'agrégation entre le niveau d'analyse source niv_{inf} et le niveau d'analyse cible niv_{sup}. Par conséquent, nous avons choisi d'utiliser la méthode des k-means qui, en plus de satisfaire la contrainte structurelle des hiérarchies strictes de l'OLAP, permet de regrouper les instances selon un lien sémantique défini par l'utilisateur. Par exemple, la hiérarchie (*ville* $\rightarrow$ *département*) est conforme au découpage départemental en France. Cette hiérarchie peut être utilisée dans tous les entrepôts souhaitant faire des agrégations de *ville* vers le *département* au sens du découpage départemental. Cependant, lors des analyses OLAP, d'autres hiérarchies possibles de *ville* peuvent intéresser un utilisa-

teur pour l'aider dans ses prises de décision. Il peut par exemple vouloir regrouper les instances de *ville* par *superficie*, *nombre d'habitants* et/ou par le *nombre de naissances*, etc. Bien évidemment, le choix d'une hiérarchie de *ville* à construire dépend des objectifs d'analyse de l'utilisateur. A titre d'exemple, les villes *Paris*, *Lyon* et *Marseille* qui appartiennent à des départements différents peuvent se retrouver dans la même classe (*Grande ville*) et donc, former un même agrégat selon un regroupement des villes par *superficie*. Le nouveau niveau de hiérarchie serait "*Groupe de villes*" avec les instances *Grande ville*, *Moyenne ville* et *Petite ville*. Si l'utilisateur accepte ce nouvel axe d'analyse, ce dernier peut être physiquement créé permettant ainsi à l'utilisateur de faire des analyses OLAP plus élaborées sur la hiérarchie *ville* $\rightarrow$ *Groupe de villes*. Pour cela, nous avons défini un nouvel opérateur d'agrégation *RoK* (*Roll-up with K-means*) qui permet de faire évoluer la structure hiérarchique d'une dimension en utilisant la méthode des k-means.

L'approche de recommandation de requêtes que nous proposons présente plusieurs avantages. Elle est centrée utilisateur, dynamique, et permet de créer plusieurs niveaux de hiérarchies dans une même dimension, sémantiquement plus riche que les hiérarchies existantes.

4.3.2 La méthode des k-means

La méthode des k-means est un algorithme de classification automatique qui procède par réallocation dynamique [For65, Mac67]. On l'appelle aussi *la méthode des centres mobiles*. En effet, il s'agit d'un algorithme itératif qui partitionne une population X en k classes les plus homogènes possibles où chaque classe est modélisée par son barycentre (c'est-à-dire la moyenne arithmétique de tous les individus affectés à la classe).

Pour répartir la population X dans une partition à k classes, l'algoritme des k-means peut être résumé comme suit :

1) *Prendre aléatoirement k individus comme centres initiaux ;*
2) *Affecter chaque individu x_i au centre C_j qui lui est le plus proche (au sens de la distance euclidienne) ;*
3) *Recalculer les coordonnées des k centres ;*
4) *Réitérer (2) et (3) tant que les centres bougent ;*

La qualité de la classification peut être évaluée par la dispersion totale des individus à l'intérieur des classes obtenues. Cette dispersion est faible lorsque les individus d'une classe sont très proches de leur centre. On parle alors d'inertie intra-classes. Par conséquent, la meilleure partition de X en k classes est la partition qui minimise cette dispersion.

Nous avons choisi la méthode des k-means parmi toutes les méthodes de classification car nous pensons que c'est la méthode la mieux adaptée aux exigences majeures de l'analyse

en ligne. D'abord pour sa complexité algorithmique qui est faible et linéaire, ensuite pour le type des classes fournies par la méthode (une partition de la population à classifier). Par ailleurs, la méthode des k-means présente d'autres avantages qui peuvent être exploités dans le cadre de nos travaux. C'est une méthode incrémentale facilitant ainsi la mise à jour des classes obtenues lors du rafraichissement de l'entrepôt. De plus, il n'est pas nécessaire de charger toutes les données en mémoire pour le calcul ; ce qui est non négligeable pour le passage à l'échelle. C'est donc une méthode qui est très adaptée pour le traitement des grandes bases de données et à fortiori lorsqu'il s'agit des entrepôts de données.

4.3.3 Exemple illustratif

Nous utilisons tout au long de ce chapitre l'entrepôt de données VENTE (Figure 4.1) qui comporte deux mesures : le revenu des ventes (REVENUDESVENTES) et la quantité vendue (QTEVENDUE). Ces mesures peuvent être étudiées selon trois dimensions : *Temps*, *Produit* et *Région*. La hiérarchie de la dimension *Temps* possède quatre niveaux SEMAINE, MOIS, TRIMESTRE et ANNEE. La hiérarchie de la dimension *Région* possède trois niveaux : MAGASIN, VILLE et PAYS. De même, la dimension *Produit* est hiérarchisée selon trois niveaux : PRODUIT, CATEGORIE (*catégorie de produits*) et FAMILLE (*famille de produits*).

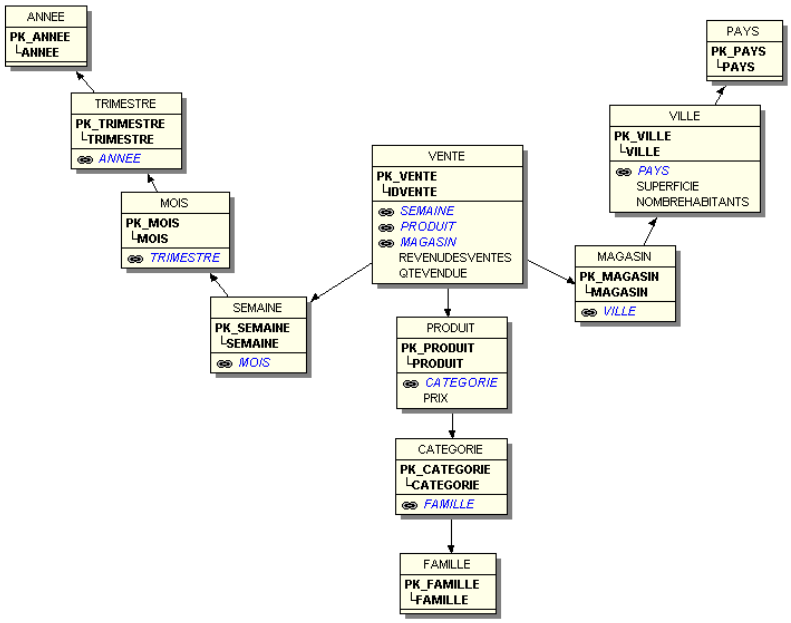

FIGURE 4.1 – Schéma de l'entrepôt de données VENTE.

Lorsqu'un utilisateur applique une requête OLAP sur l'entrepôt de données, il sélectionne les niveaux d'analyse qui sont susceptibles d'expliquer les faits qu'il veut observer. Cependant, il n'existe pas d'outils pour guider l'utilisateur vers de nouvelles explorations

non prédéfinies par le modèle de l'entrepôt ni pour approfondir l'analyse vers la structuration, l'explication et la prédiction. Nous nous intéressons plus particulièrement ici à la structuration de données en vue de créer de nouveaux axes d'analyse.

Par conséquent, chaque nouveau niveau d'analyse que l'opérateur *RoK* crée, doit répondre à un besoin d'analyse précis de l'utilisateur. C'est pour cette raison que nous proposons deux solutions pour le choix des descripteurs sur lesquels l'opérateur *Rok* va appliquer la méthode des k-means en vue de l'évolution de schéma de l'entrepôt.

4.3.4 Scenarii d'analyse

1. *Utilisation des descripteurs de dimension.* Supposons que l'objectif d'analyse de l'utilisateur soit de savoir s'il faut ouvrir (ou fermer) des points de ventes dans certaines zones. Pour trouver une réponse à cette question, il va essayer d'étudier les *revenus des ventes* à travers la dimension *Région* dont la hiérarchie actuelle est organisée comme suit :

$$\text{MAGASIN} \rightarrow \text{VILLE} \rightarrow \text{PAYS} \rightarrow all$$

Pour une analyse plus ciblée, l'utilisateur peut alors ressentir le besoin d'ajouter un nouveau niveau d'analyse GROUPE_VILLE qui doit regrouper les villes selon la densité de leur population (Figure 4.2).

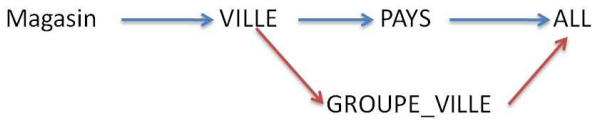

FIGURE 4.2 – Création du niveau GROUPE_VILLE à partir du niveau VILLE

Dans ce cas, il serait intéressant de regrouper les instances du niveau VILLE selon l'attribut *nombre d'habitants* (NOMBREHABITANTS), pour une classification en *petite*, *moyenne* et *grande ville* par exemple (Table 4.1). A la fin de la classification, l'opérateur *RoK*, basé sur la méthode des k-means, va créer le niveau de hiérarchie *groupe de villes* (GROUPE_VILLE) au dessus du niveau VILLE (Table 4.2).

Dans le cas où l'utilisateur peut vouloir regrouper les villes en fonction à la fois de leur nombre d'habitants et de leur superficie, la méthode des k-means est alors appliquée sur les données du niveau de hiérarchie VILLE en choisissant les variables (attributs) NOMBREHABITANTS et SUPERFICIE. Dans ce cas, un nouveau niveau de hiérarchie TYPOLOGIE_VILLE peut alors être créé au dessus du niveau VILLE (Figure 4.3).

MAGASIN	VILLE	NOMBREHABITANTS	CLASSE
M1	Grenole	400 000	1
M2	Lyon	1 200 000	2
M3	Amiens	160 000	3
M4	Nantes	500 000	1
M5	Lille	1 000 000	2
M6	Saint-Pierre	60 000	3
M7	Chambéry	100 000	3
M8	Strasbourg	390 000	1
M9	Marseille	1 400 000	2
M10	Calais	105 000	3
M11	Rouen	380 000	1

TABLE 4.1 – Application des k-means sur le niveau VILLE (attribut NOMBREHABITANTS) de la dimension *Région*

CLASSE	CENTRE	INTERVALLE	GROUPE_VILLE
1	417 000	[380 000, 500 000]	moyenne
2	1 200 000	[1000 000, 1 400 000]	grande
3	106 000	[60 000, 160 000]	petite

TABLE 4.2 – Création du niveau d'analyse GROUPE_VILLE dans la dimension *Région*

2. *Utilisation des mesures de la table des faits.* Cette deuxième proposition permet de répondre à un besoin d'analyse de tendances. Supposons par exemple que, pour trouver la politique commerciale la plus adaptée à chaque *produit*, l'utilisateur veuille connaître le comportement d'achat des clients. Pour cela, il désire créer un nouveau niveau d'analyse qui regroupe les produits en fonction du *chiffre d'affaire* qu'ils rapportent. Pour satisfaire ce besoin de l'utilisateur, l'opérateur *RoK* va agréger la mesure REVENUDESVENTES sur le niveau d'analyse PRODUIT. La méthode des k-means sera ensuite exécutée sur le résultat obtenu. A l'issue de cette classification, l'opérateur va créer le niveau d'analyse GROUPE_PRODUIT au dessus du niveau d'analyse PRODUIT (Figure 4.4).

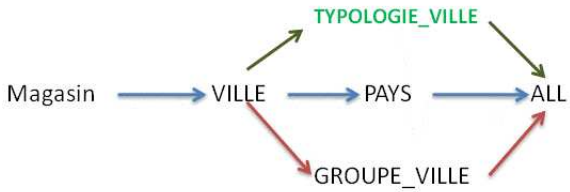

FIGURE 4.3 – Création du niveau TYPOLOGIE_VILLE selon NOMBREHABITANTS et SUPERFICIE à partir du niveau VILLE

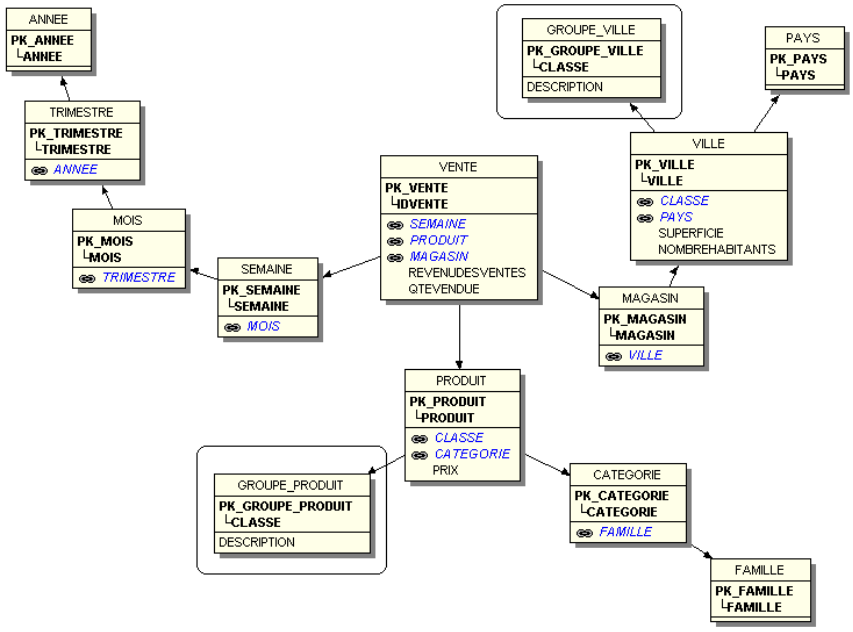

FIGURE 4.4 – Entrepôt de données VENTE après ajout de deux niveaux d'analyse GROUPE_VILLE et GROUPE_PRODUIT

4.4 Formalisation

4.4.1 Rappels

Nous rappelons ici brièvement les principaux concepts des entrepôts de données (nous utilisons ici les notations qui ont été proposées par Hurtado et al. [HMV99a]).

Définition 4.1 *Entrepôt de données*
Un entrepôt de données est un couple $\mu = (\delta, \varphi)$ où δ est un ensemble de *dimensions* et φ est un ensemble de *faits*.

Définition 4.2 *Dimension*
Le *schéma d'une dimension* est un couple $D = (L, \preceq)$ où L est un ensemble fini de niveaux de hiérarchie et $\preceq$ est une relation binaire transitive et reflexive sur L traduisant le lien hiérarchique entre les éléments de L. Cette relation possède au moins deux niveaux spécifiques :

- l_{bottom} : représente l'unique niveau le plus bas de la relation $\preceq$
- all : représente le niveau le plus haut de la relation $\preceq$.

$$L = \{l_{bottom}, ..., l, ..., all \mid \forall\ l,\ l_{bottom} \preceq l \preceq all\}$$

Chaque niveau $l \in L$ possède un ensemble d'instances qui prend ses valeurs dans un domaine $dom(l)$; $dom(all) = \{all\}$). Pour toute paire de niveaux $(l, l') \in L$ telle que $l \preceq l'$, il existe une fonction de correspondance f qui associe chaque instance du niveau l à une instance du niveau l' : $f_l^{l'} : dom(l) \longrightarrow dom(l')$.

Exemple 4.1

Considérons la dimension *Region* de la Figure 4.1. Nous avons :

$L_{Région} = \{\text{MAGASIN}, \text{VILLE}, \text{PAYS}, all \mid \text{MAGASIN} \preceq \text{VILLE} \preceq \text{PAYS} \preceq all\}$

Pour la paire de niveaux (VILLE, PAYS), nous avons $dom(\text{VILLE}) = \{Paris, Lyon, Berlin\}$, $dom(\text{PAYS}) = \{France, Allemagne\}$ ainsi que la fonction $f_{\text{VILLE}}^{\text{PAYS}}$ définie par :

$f_{\text{VILLE}}^{\text{PAYS}} = \{(Paris; France), (Lyon; France), (Berlin; Allemagne)\}$

Définition 4.3 *Fait*

Le schéma d'un fait est un couple $F = (L_{group}, M)$ où $L_{group} = l_{D_1} \cup .. \cup l_{D_q}$ est la réunion de q niveaux d'analyse appartenant respectivement à q dimensions différentes et M est un niveau spécifique qu'on appelle *mesure*. Le domaine $dom(M)$ est un ensemble sur lequel, des opérations d'agrégation sont possibles (somme, moyenne, ...). Une instance x du fait F est donc une mesure $m \in dom(M)$ décri sur q niveaux :

$$\begin{array}{rcl} F : dom(l_{D_1})X...Xdom(l_{D_q}) & \longrightarrow & dom(M) \\ x(l_{D_1}, ..., l_{D_q}) & \longmapsto & m \end{array}$$

Exemple 4.2

Pour la table de fait VENTE de la Figure 4.1, nous avons

$$\text{VENTE} = ((\text{SEMAINE} \cup \text{PRODUIT} \cup \text{MAGASIN}), \text{QTEVENDUE})$$

Le tableau 4.3 présente cinq instances du fait VENTE.

SEMAINE	PRODUIT	MAGASIN	QTEVENDUE
1	p3	m1	10
1	p4	m2	2
2	p1	m2	5
2	p2	m3	7
3	p2	m2	4

TABLE 4.3 – Instances de la table de fait VENTE.

Définition 4.4 *Cube de données*

L'algèbre mutidimensionnelle fournit un opérateur *Cube* qui peut être défini comme suit :

Soient une table de fait $F = (L_{group} = \{l_1 \in D_1 \cup ... \cup l_p \in D_p\}, M)$ et un ensemble de niveaux d'analyse $GL = \left\{l'_1 \in D_1, ..., l'_p \in D_p \mid l_i \preceq\ l'_i\ \forall i = 1..p\right\}$. L'opérateur d'agrégation $CUBE(F, GL)$ fournit une nouvelle table de faits $F' = (GL, M')$ où M' est le résultat de l'agrégation de la mesure M du groupe de niveau L_{group} vers le groupe de niveau GL.

4.4.2 Cadre formel de l'approche

4.4.2.1 Ajout d'un nouveau niveau d'analyse

Pour générer un nouveau niveau de hiérarchie dans une dimension, nous utilisons l'opérateur *Generalize* [HMV99a] dont la définition formelle est donnée ci-après.

Définition 4.5 *Opérateur Generalize*
Considérons une dimension $D = (L = \{l_{bottom}, ..., l, ..., all\}, \preceq)$, deux niveaux hiérarchiques $l \in L$ et $l_{new} \notin L$, et une fonction $f_l^{l_{new}}$ qui associe chaque instance de l à une instance de l_{new}. $Generalize(D, l, l_{new}, f_l^{l_{new}}) = D' = (L', \preceq')$ où $L' = L \cup \{l_{new}\}$ et $\preceq' = \preceq \cup \{(l \rightarrow l_{new}), (l_{new} \rightarrow All)\}$ conformément à la fonction de correspondance $f_l^{l_{new}}$.

Exemple 4.3

Considérons la dimension *Région* de la Figure 4.1 et la fonction suivante :
$f_{\text{PAYS}}^{\text{CONTINENT}} = \{(France, Europe), (Espagne, Europe), ..., (Canada, Amerique), ..., (Chine, Asie), ...\}$.
$Generalize(Région, \text{PAYS}, \text{CONTINENT}, f_{\text{PAYS}}^{\text{CONTINENT}})$ ajoute un nouveau niveau CONTINENT dans la hiérarchie de la dimension *Région* et fournit la nouvelle structure hiérarchique suivante : MAGASIN → VILLE → PAYS → CONTINENT

L'originalité de notre approche réside dans la construction de la fonction de correspondance $f_l^{l_{new}}$ en utilisant l'opérateur *RoK*.

Définition 4.6 *L'opérateur RoK*
Soient k un nombre entier strictement positif, $X = \{x_1, x_2, ..., x_n\}$ une population de n individus et $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_k\}$ un ensemble de k classes. L'opérateur $RoK(X, k)$ calcule (en utilisant l'algorithme des k-means décrit dans la section 4.3.2) l'ensemble $C = \{c_1, ..., c_k \mid \forall i = 1..k, c_i = barycentre(\mathcal{C}_i)\}$ et retourne la fonction de correspondance f_x^c telle que : $f_x^c = \{(x_i \rightarrow \mathcal{C}_j) \mid \forall i = 1..n\ et\ \forall m = 1..k, distance(x_i, c_j) \leq distance(x_i, c_m)\}$

Exemple 4.4

- $X = \{x_1 = 2, x_2 = 4, x_3 = 6, x_4 = 20, x_5 = 26\}$
- $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$

RoK(X, 2) retourne l'ensemble $C = \{c_1 = 4, c_2 = 23\}$ ainsi que l'application :
$f_x^c = \{(x_1 \to \mathcal{C}_1), (x_2 \to \mathcal{C}_1), (x_3 \to \mathcal{C}_1), (x_4 \to \mathcal{C}_2), (x_5 \to \mathcal{C}_2)\}$

4.4.3 Algorithme de génération de niveau de hiérarchie

Paramètres en entrée :
- une dimension $D = (L, \preceq)$,
- un niveau d'analyse $l \in L$,
- un nouveau niveau d'analyse $l_{new} \notin L$,
- un nombre entier $k \geq 2$ qui va être le nombre de modalités de l_{new},
- une variable *dataSource* qui peut prendre deux valeurs : 'F' (pour *fait*) ou 'D' (pour *dimension*).

1) Etape 1 : Construction de la population d'apprentissage X_l

   Cette première étape a pour objectif de constituer une population X_l à partir des instances du niveau d'analyse l. La population X_l sera décrite directement par les attributs de l si la valeur du paramètre *dataSource* est égale à 'D'. Dans le cas contraire (*dataSource* est égale à 'F'), X_l sera construite en exécutant l'opération $CUBE(F, l)$.

   Exemple. Supposons que l'on désire créer un nouveau niveau GROUPE_VILLE au dessus du niveau d'analyse VILLE. Si le paramètre *dataSource* est égal à 'F', l'algorithme exécute l'opération $CUBE$(VENTE, VILLE). Nous obtenons ainsi la population décrite par le tableau 4.4. Dans le cas contraire, les villes seront décrites par leurs descripteurs dans l'entrepôt (Table 4.5).

VILLE	REVENUDESVENTES	QTEVENDUE
Paris	10000	400
Chambéry	240	20
Lyon	120000	300
Saint-Etienne	1200	50

TABLE 4.4 – Niveau d'analyse VILLE décrit par les mesures.

VILLE	SUPERFICIE	NOMBREHABITANTS
Paris	105	12 000
Chambéry	6	100
Lyon	60	6 000
Saint-Etienne	8	180

TABLE 4.5 – Niveau d'analyse VILLE décrit par ses propres descripteurs.

2) Etape 2 : Classification

Durant cette étape, l'algorithme applique l'opérateur *RoK* sur la population d'apprentissage X_l qui a été créée durant l'étape précédente. Si, à titre d'exemple, le paramètre k est égale à 2, l'exécution de l'opérateur *RoK* sur le tableau 4.5 nous donne l'ensemble $\mathcal{C} = \{\mathcal{C}_1(82.5, 9000), \mathcal{C}_2(7, 140)\}$ ainsi que la fonction de correspondance : $f_{\text{VILLE}}^{\text{GROUPE\_VILLE}} = \{(Paris, \mathcal{C}_1), (Chambery, \mathcal{C}_2), (Lyon, \mathcal{C}_1), (SaintEtienne, \mathcal{C}_2)\}$

3) Etape 3 : Création du nouveau niveau d'analyse

Cette étape consiste à matérialiser le nouveau niveau d'analyse l_{new} au cœur du schéma de l'entrepôt de données. Pour ce faire, notre algorithme utilise l'opérateur *Generalize* sur la dimension D, à partir du niveau l et en utilisant la fonction de correspondance $f_l^{l_{new}}$ qui a été générée durant l'étape précédente de l'algorithme. En reprenant les exemples que l'on a pris dans les étapes 1 et 2, la création du niveau d'analyse *groupe de villes* consistera à exécuter l'opération :

$$Generalize(Région, \text{VILLE}, \text{GROUPE\_VILLE}, f_{\text{VILLE}}^{\text{GROUPE\_VILLE}})$$

4.5 Implémentation et Expérimentation

4.5.1 Environnement technique

L'algorithme que nous venons de présenter a été intégré au sein du SGBD Oracle 10g [RB07]. Ainsi, nous avons programmé l'algorithme des *k-prototypes* avec le langage PL/SQL du SGBD Oracle 10g. La méthode des *k-prototypes* est une variante des k-means permettant de traiter simultanément des descripteurs numériques et catégoriels [Hua97]. Le choix d'intégrer la méthode des k-means à l'intérieur du SGBD Oracle est motivé par les performances de la fouille en ligne appliquée sur les données de l'entrepôt qui sont volumineuses (cf. Chapitre 5).

4.5.2 Scenarii de test

Nos expériences et tests ont été effectués en utilisant l'entrepôt de données *Emode* qui sert de base de démonstration de l'outil Business Object. Nous avons normalisé le schéma de cet entrepôt pour qu'il soit identique au schéma de la Figure 4.1. Sa table des faits VENTE contient 89200 enregistrements et la table de dimension PRODUIT contient 663 enregistrements. Le niveau de granularité le plus fin de la dimension PRODUIT contient 211 articles regroupés sur 34 catégories de produits et sur 12 lignes de produits. Notre objectif principal a été d'apprécier la pertinence des résultats de l'opérateur *RoK* sur des données réelles. Ainsi, nous avons prévu les deux scenarii de tests suivants :

1) Créer un axe d'analyse *fourchette de prix* qui classifie les articles selon leur prix.

2) Créer un axe d'analyse *groupe d'articles* qui regroupe les articles selon la mesure *revenu des ventes* (REVENUDESVENTES) de la table de fait .

La Figure 4.5 illustre les résultats de ces deux scenarii de test.

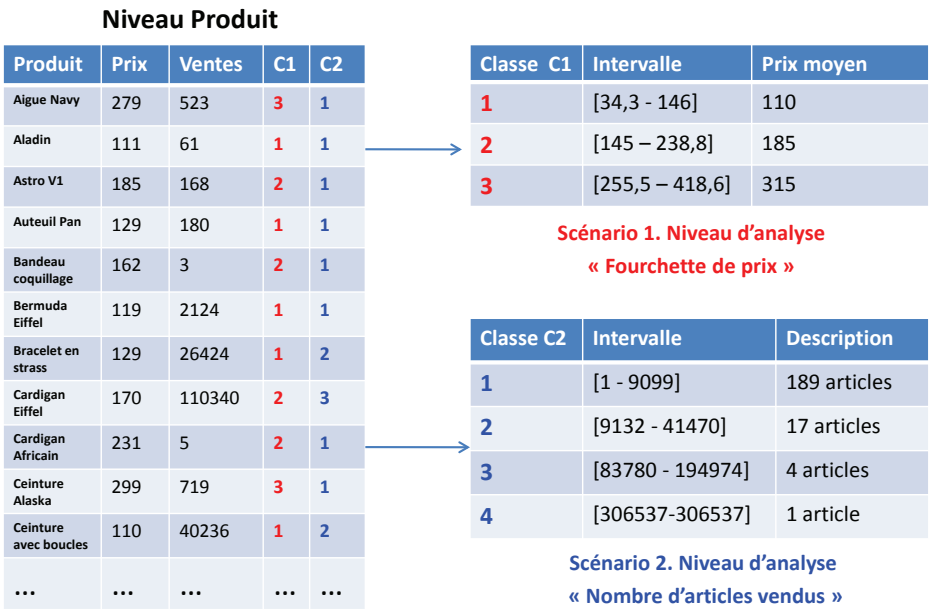

FIGURE 4.5 – Résultats des k-means selon les deux scenarii.

4.5.3 Discussion

Les résultats que nous avons obtenus mettent en évidence les points suivants :

- L'axe d'analyse créé avec le scénario numéro 1 permet d'étudier efficacement l'influence des prix sur les ventes. L'utilisation de ce nouvel axe dans l'analyse montre une corrélation assez forte entre le niveau des ventes et le prix des produits (Figure 4.6).
- L'axe d'analyse créé avec le scénario numéro 2 permet de voir les articles de produits qui se vendent bien et ceux qui se vendent moins bien (Figure 4.6).
- Nous avons remarqué aussi que la valeur des mesures pour les individus qui ont été classifiés différemment par les deux scenarii sont assez atypiques. De ce point de vue, nous pouvons dire que notre approche permet d'identifier les individus à comportement atypique. Dans ce cas, il est intéressant d'approfondir l'analyse pour

tenter d'expliquer les individus atypiques.

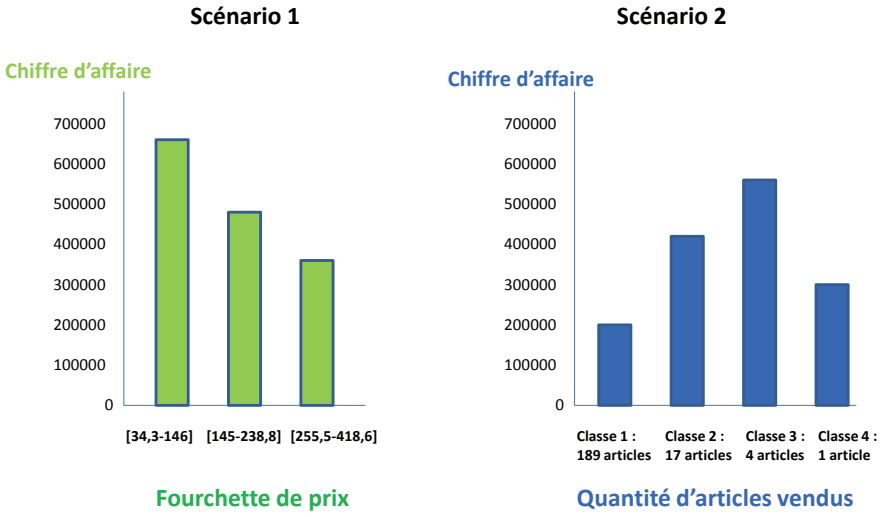

FIGURE 4.6 – Cubes de données générés avec les deux nouveaux niveaux d'analyse.

4.6 Conclusion

La technologie entrepôts-OLAP est apparue comme une technologie clef pour les entreprises désirant améliorer l'analyse de leurs données et leur système d'aide à la décision. En utilisant les applications décisionnelles, un décideur peut découvrir à partir de l'entrepôt, par le biais de l'analyse OLAP, les tendances générales significatives d'une activité ciblée de l'entreprise. Toutefois, l'analyse OLAP est très limitée puisqu'elle ne permet pas de découvrir de nouvelles structures, d'expliquer un phénomène ou de prédire de nouvelles tendances. C'est dans ce contexte que nous nous sommes intéresée à renforcer l'analyse en ligne en la combinant avec les techniques de fouille de données. Nous pensons alors que l'extraction de connaissances à partir des données entreposées peut aider l'utilisateur à découvrir de nouveaux regroupements inattendus dans la masse de données de l'entrepôt. En suivant notre intuition concernant l'intérêt de combiner la fouille de données avec l'analyse OLAP, nous avons développé des travaux originaux pour assister l'utilisateur dans son processus d'exploration et de navigation. Ces travaux consistent en la recommandation de nouveaux niveaux de hiérachies dans une dimension permettant à l'utilisateur d'écrire de

nouvelles requêtes décisionnelles.

Classiquement, pour anticiper l'écriture de requêtes décisionnelles selon l'usage des utilisateurs, la technique d'exploitation des requêtes déjà posées par ces derniers (fichier log de requêtes) est très adaptée. En revanche, pour recommander de nouvelles requêtes décisionnelles non prévues par l'entrepôt, l'élaboration de systèmes de recommandation basés sur les connaissances extraites à partir du contenu de l'entrepôt serait plus approprié. Dans ce contexte, nous soulignons l'intérêt de l'utilisation de la fouille de données qui permet d'extraire, à partir de l'entrepôt, de nouveaux espaces d'analyse insoupsonnés. Une telle approche apporte une sémantique plus riche aux nouveaux axes d'analyse découverts, pouvant être créés et ajoutés dans l'entrepôt, permettant ainsi de générer des requêtes décisionnelles pertinentes.

Nous avons présenté dans ce chapitre une approche visant à recommander à l'utilisateur de nouveaux axes d'analyse basés sur la découverte de nouvelles structures naturelles grâce aux principes d'une méthode de classification (K-means). Pour cela, nous avons défini un nouvel opérateur d'agrégation RoK (*Roll-up with K-means*) qui permet d'extraire à partir des données de l'entrepôt de nouvelles structures sémantiquement plus riches. En effet, en appliquant la méthode des k-means sur un niveau de hiérarchie d'une dimension niv_{inf}, un nouveau regroupement des instances de niv_{inf} selon un ordre de proximité sémantique peut être proposé à l'utilisateur. Le choix des variables de regroupement est basé soit sur les propres descripteurs du niveau de hiérarchie choisi, soit sur les mesures de l'entrepôt de données. Les classes obtenues assurent une vue concise et structurée des données et des regroupements inattendus apparaissent. Le résultat de la classification sert à créer un nouveau niveau de hiérarchie niv_{sup} qui agrège les instances de niv_{inf}, porteur d'une nouvelle sémantique et pouvant être pertinent pour l'utilisateur. En effet, le nouveau niveau de hiérarchie offre de nouveaux angles de vues sur les faits non prévus initialement par l'entrepôt. Par conséquent, l'utilisateur peut poser de nouvelles requêtes décisionnelles sur l'entrepôt enrichi.

Nous avons par ailleurs conçu et implémenté notre approche de manière intégrée dans le SGBD Oracle pour les raisons suivantes. Tout d'abord, comme nous allons le voir et le démontrer dans le Chapitre 5, la fouille de données en ligne permet de traiter de grandes bases et/ou entrepôts de données sans limitation de taille avec des temps de traitement très intéressants. Ensuite, l'opérateur *Rok* basé sur les k-means constitue au sein du SGBD un nouvel opérateur d'agrégation sémantique. Les expériences et les tests que nous avons menés attestent de l'intérêt de notre approche en fournissant des axes d'analyse pertinents. Nous avons montré dans ce chapitre l'intérêt de combiner la fouille de données avec l'analyse multidimensionnelle pour la création de nouveaux axes d'analyse sémantiquement plus riches.

Les perspectives concernant ce travail sont nombreuses. La principale évolution pos-

sible pour notre travail réside dans l'amélioration de l'automaticité de notre proposition. En effet, lorsque l'entrepôt est mis à jour, il faut pouvoir étiqueter les nouvelles instances avec les classes déjà créées. Dans ce cas, une piste intéressante serait d'étendre l'approche que nous avons proposée à l'apprentissage supervisé (cf. Chapitre 5). On peut par exemple construire des modèles de prédiction à partir des résultats de la classification. Ces modèles peuvent être exploités pour identifier des règles d'analyse pouvant prédire la valeur des nouvelles données et ainsi les classer. On peut également envisager d'étendre l'opérateur *Rok* pour permettre de créer des axes d'analyse de tendance qui tiennent compte de l'évolution des données dans le temps. Pour ce faire, nous proposons un "découpage horizontal" de la table des faits sur une unité de temps choisi par l'utilisateur. On applique alors la classification sur chaque sous-population de la table de faits et l'on fusionne les résultats au sein d'un axe d'analyse unique.

Pour terminer, nous soulignons l'intérêt d'évaluer le système de recommandation que nous proposons. Pour cela, il faut étudier la qualité des axes d'analyse obtenus par notre système de recomandation. Cela revient à évaluer la qualité des classes obtenues par la méthode des k-means. Autrement dit, est-ce que le nouveau regroupement des instances de niv_{inf} est un "bon" regroupement ? Dans le cadre de nos travaux sur la recommandation de nouveaux niveaux de hiérarchie, un bon regroupement d'instances doit certainement dépendre de l'utilisateur.

Nous citons deux approches de validation possibles. Dans la première approche, nous supposons que l'utilisateur veuille regrouper par exemple les villes en 3 classes par la méthode des k-means en choisissant les variables (attributs) *superficie* et *nombre d'habitants*. L'utilisateur peut dans ce cas comparer le regroupement obtenu avec une classification dont il dispose déjà. Dans la deuxième approche, l'utilisateur impose des contraintes. Par exemple, deux instances doivent appartenir à la même classe (*must-link*) ou au contraire deux instances ne doivent pas appartenir à une même classe (*cannot-link*). Dans ce cas, nous pouvons utiliser le clustering contraint (*Constrained clustering with k-means*) afin de prendre en compte les contraintes de l'utilisateur. Ces approches de validation peuvent aider l'utilisateur à confirmer ou infirmer la création d'un nouveau niveau de hiérarchie dans l'entrepôt.

4.7 Publications

La liste suivante présente nos publications relatives aux travaux que nous avons menés sur la recommandation de nouveaux axes d'analyse pertinents en utilisant la méthode d'apprentissage non supervisé, *les k-means*.

Conférences internationales

[1] **F. Bentayeb**, C. Favre, "RoK : Roll-Up with the K-means Clustering Method for Recommending OLAP Queries", 20th International Conference on Database and Expert Systems Applications (DEXA 2009), 2009, Linz, Austria, Lecture Notes in Computer Science, Vol. 5690, Springer, 501-515.

[2] **F. Bentayeb**, "K-means-based approach For OLAP dimension updates", 10th International Conference on Enterprise Information Systems (ICEIS 08), Barcelona, Spain, 12-16 June 2008, 531-534.

Conférence nationale

[3] O. Rakotoarivelo, **F. Bentayeb**, "Evolution de schéma par classification automatique pour les entrepôts de données", 3èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 07), Poitiers, Juin 2007 ; Revue des Nouvelles Technologies de l'Information, Vol. B-3, Cépaduès, 99-112.

Workshop national

[4] O. Rakotoarivelo, **F. Bentayeb**, "Evolution de schéma par classification automatique pour les entrepôts de données", 4ème atelier Fouille de Données Complexes dans un Processus d'Extraction des Connaissances (FDC-EGC 07), Namur, Belgique, Janvier 2007.

Chapitre 5

Fouille de données en ligne

L'évolution des bases de données aux entrepôts de données a permis le passage de la gestion à l'analyse des données grâce à une première intégration de l'OLAP au sein des SGBD. Pour étendre le volet analytique des SGBD, nous pensons que l'intégration des techniques de fouille au sein de ces derniers apporterait une dimension analytique sémantiquement plus riche, nécessaire à l'exploitation des données entreposées.

L'objectif de ce chapitre est tout d'abord de présenter et de discuter l'existant en matière d'intégation de tehniques de fouille de données au sein des SGBD. Nous motivons et présentons ensuite notre démarche de fouille en ligne qui est conjointement liée à l'intégration de l'analyse en ligne OLAP au sein des SGBD. Le challenge était alors d'étendre les capacités analytiques des SGBD de l'OLAP vers une analyse structurante, explicative et prédictive. Autrement dit, construire des opérateurs de fouille en ligne intégrés au sein des SGBD. C'est dans ce contexte que nous avons initié les premiers travaux sur la fouille en ligne puis développé une approche globale pour bénéficier des capacités des SGBD en termes de préparation, chargement et de traitement des données sans limitation de taille ; ce qui est non négligeable lorsque les méthodes de fouille sont appliquées sur les entrepôts de données. L'utilisateur aura donc a sa disposition un système décisionnel capable de lui fournir aussi bien des analyses exploratoire et navigationnelle de l'OLAP, que des analyses structurante, explicative et prédictive grâce aux techniques de fouille. De plus, l'application des techniques de fouille sur les cubes peut produire des résultats inattendus, potentiellement intéressants pour l'utilisateur.

Chacune des approches de fouille de données en ligne présentée dans ce chapitre a fait l'objet d'un travail de master, à l'exception de la première approche basée sur les vues relationnelles.

5.1 Motivation

Les entrepôts de données sont des bases de données dédiées à l'analyse pour l'aide à la prise de décision. Les modèles en étoile structurent les données entreposées de manière multidimensionnelle et permettent dans un premier temps de produire des cubes de données adaptés à l'analyse. Dans un second temps, c'est à l'utilisateur de naviguer, explorer et analyser les données d'un cube afin d'en extraire des informations pertinentes. Ceci permet à l'utilisateur d'anticiper, intuitivement, la réalisation d'événements futurs.

Or, comme nous l'avons annoncé dans l'introduction générale (cf. Chapitre 1), l'analyse OLAP qui est une analyse exploratoire et navigationnelle est insuffisante pour produire des analyses plus élaborées telles que des analyses descriptives et/ou explicatives et/ou prédictives. Si bien que plusieurs travaux se sont intéressés à la combinaison des techniques de fouille de données avec l'analyse en ligne afin d'étendre les capacités analytiques de l'OLAP [SAM98, CRST06, MRBB04, Mes06, MRMB07]. La majorité de ces travaux se situent dans l'approche multidimensionnelle des entrepôts de données et de l'OLAP (MOLAP - *Multidimensional OLAP*) dans laquelle les opérateurs OLAP et les techniques de fouille appliqués opèrent en mémoire. Cela pose donc le problème de la capacité des systèmes proposés selon l'approche MOLAP à gérer de gros volumes de données puis à les traiter avec des temps acceptables pour l'utilisateur. Par ailleurs, dans la réalité les entrepôts de données sont stockés sous forme de bases de données volumineuses, souvent au sein de SGBD relationnels selon l'approche ROLAP (*Relational OLAP*).

Notre objectif dans ce chapitre est de proposer un environnement décisionnel dans lequel peuvent co-exister à la fois l'OLAP et la fouille en ligne pour assister et aider l'utilisateur dans ses choix de scénarios d'analyse. Il s'agit surtout de proposer à l'utilisateur de nouveaux scénarios d'analyse plus élaborés et sémantiquement plus riches en appliquant directement les techniques de fouille sur les données entreposées ou en combinant la fouille avec l'OLAP. Les méthodes d'apprentissage non supervisé peuvent aider par exemple à rechercher des structures naturelles dans les données (cf. Chapitre 4) alors que les arbres de décision peuvent être utilisés pour la prédiction. Par ailleurs, l'utilisateur qui interagit de façon interactive avec le système, a besoin d'un temps de réponse quasi-instantané. La solution que nous proposons consiste alors à intégrer les techniques de fouille au sein des SGBD. Notre choix s'est donc porté naturellement vers l'approche ROLAP (*Relational OLAP*) du fait de la capacité des SGBD de traiter des bases de données sans limitation de taille avec des temps de traitement acceptables.

5.2 Intégration des techniques de fouille dans les SGBD

La fouille de données (*data mining*) est une discipline qui a largement fait ses preuves depuis le début des années 90. Elle emploie des méthodes d'apprentissage afin d'induire des modèles de connaissances exprimés dans des formalismes valides et compréhensibles. Aujourd'hui, on peut considérer la fouille de données comme une nécessité imposée par le besoin des entreprises de valoriser leurs données contenues dans les bases de données gérées par les SGBD. Pendant longtemps, le processus de fouille de données était dissocié des SGBD. Le SGBD n'était considéré dans ce cas que comme un système de stockage auquel la fouille de donnée accédait via des API ("*Application Programming Interface*"). Or, avec l'avènement des entrepôts de données et de la technologie OLAP, les SGBD se sont dotés d'outils exploratoires et navigationnels pour l'analyse en ligne. De la même manière que l'analyse OLAP fut intégrée au sein des SGBD, les éditeurs de logiciels ont tenté d'intégrer les méthodes de fouille au sein de leur SGBD. Dans le même temps, des travaux de recherche dans le domaine ont vu également le jour. L'intérêt porté à l'intégration des méthodes de fouille au sein des SGBD peut s'expliquer pour les raisons suivantes.

Les algorithmes traditionnels de fouille de données s'appliquent sur des tableaux attributs/valeurs [ZR00]. De ce fait, lorsque la volumétrie des bases traitées est importante, les algorithmes classiques de fouille se heurtent au problème de la limitation de la taille de la mémoire centrale dans laquelle les données sont traitées. La "scalabilité" ou le passage à l'échelle (capacité de maintenir des performances malgré un accroissement du volume de données), peut alors être assurée en optimisant soit les algorithmes [AMSea96, MAR96, GRG98, GRG00], soit l'accès aux données [RMZ02, DS99]. Une autre issue au problème consiste à réduire la volumétrie des données à traiter. Pour cela, une phase de prétraitement est généralement appliquée sur les données : l'échantillonnage [Toi96, CR00] ou la sélection d'attributs [LM98].

Dans ce contexte, l'intégration des méthodes de fouille au sein des SGBD constitue une solution évidente pour pallier le problème de limitation de la taille de la mémoire. Il s'agit d'intégrer les méthodes de fouille de données au cœur des SGBD [Cha98]. Ainsi, le volume de données traitées n'est plus limité par la taille de la mémoire. Plusieurs travaux ont étudié ce problème. Ils portent sur des extensions du langage SQL, pour la création de nouveaux opérateurs [MPC96, STA98, GS02] et le développement de nouveaux langages [HFW+96, IV99, WZL03]. Dans le même temps, quelques éditeurs de logiciels ont intégré certaines méthodes de fouille de données au sein de leur SGBD [IBM01, Ora01, STY01], grâce à des extensions du langage SQL et à l'utilisation d'API.

Contrairement aux solutions citées ci-dessus et dans le but de bénéficier des avantages certains des SGBD, nous avons voulu orienté nos travaux vers une intégration totale des méthodes de type "arbres de décision" dans les SGBD en utilisant uniquement les outils

offerts par ces derniers (tables, vues, ...). Pour cela, il était nécessaire d'adapter les algorithmes de fouille de données à l'environnement des SGBD. Une fois les méthodes de fouille intégrées au sein des SGBD, la fouille dans de gros volumes de données peut être appliquée. Nous appelons cela la *fouille en ligne*. Notre motivation de rapprocher la fouille de données et les SGBD peut s'expliquer pour les raisons suivantes.

- Les méthodes de fouille de données nécessitent des données consolidées, nettoyées et préparées dans un format approprié pour l'analyse. Cela conïcide exactement avec les différentes étapes nécessaires à la construction d'une base de données.
- Les algorithmes de fouille opèrent en mémoire vive, ce qui limite la taille des bases à traiter. Les SGBD quant à eux sont conçus pour supporter de gros volumes de données sans limitation de taille.
- Plusieurs algorithmes de fouille, tels que les arbres de décison, calculent des fréquences pour construire des modèles d'apprentissage (arbres). Or, les SGBD et le langage SQL en particulier fournissent les commandes d'agrégation telles que *Count* et *Group by* qui facilitent le calcul de ces fréquences. De plus, l'utilisation des structures d'accès telles que les index améliore l'accès aux données de façon significative.
- De même que l'analyse en ligne est intégrée au sein des SGBD, il apparaissait utile d'étendre les possibilités d'analyse des SGBD vers la fouille en ligne. Les SGBD deviennent alors, en plus de leur vocation de gestion, des outils d'analyse et de fouille en ligne.

Ainsi, contrairement aux approches existantes, nous avons proposé une approche de fouille totalement intégrée au sein des SGBD pour laquelle nous n'avons eu recours à aucune extension du langage SQL ni utilisé des API spécifiques. Nous avons tout d'abord montré qu'en utilisant le principe de vues relationnelles, les arbres de décision étaient facilement implémentables au sein des SGBD [BD02]. Ensuite, nous avons étendu ces travaux afin d'améliorer les temps de traitement en utilisant une table de contingence [UBDB04, BDU04], puis les index bitmap [FB05b, FB05a]. Une synthèse de ces travaux a été publiée dans une revue internationale [BDFU07]. A partir des premiers résultats encourageants concernant la fouille en ligne dans les grandes bases de données sans limitation de taille et avec des temps de traitement acceptables, nous avons orienté ces travaux vers la fouille dans les entrepôts de donnnées et en particuliers dans les cubes OLAP [Mad04].

5.3 Les arbres de décision

5.3.1 Principe

Les arbres de décision sont des méthodes de fouille de données qui relèvent de l'apprentissage supervisé et produisent des règles du type "si-alors" [ZR00]. Le processus d'ap-

prentissage consiste ensuite à déterminer la classe d'un objet quelconque d'après la valeur de ses variables. Les arbres de décision utilisent en entrée un ensemble d'objets (n-uplets) décrits par des variables (attributs). Chaque objet appartient à une classe, les classes étant mutuellement exclusives. Pour construire un arbre de décision, il est nécessaire de disposer d'une population d'apprentissage (table ou vue) constituée d'objets dont la classe est connue.

Les méthodes de construction d'arbres de décision segmentent la population d'apprentissage afin d'obtenir des groupes au sein desquels la proportion d'une classe est maximisée. Cette segmentation est ensuite réappliquée de façon récursive sur les partitions obtenues. La recherche de la meilleure partition lors de la segmentation d'un nœud revient à rechercher la variable la plus discriminante pour les classes. C'est ainsi que l'arbre (ou plus généralement le graphe) est constitué.

Finalement, les règles de décision sont obtenues en suivant les chemins partant de la racine de l'arbre (la population entière) jusqu'à ses feuilles. La Figure 5.1 montre un exemple d'arbre de décision ainsi que les règles associées. p(Classe i) représente la probabilité d'un objet d'appartenir à la Classe numéro i.

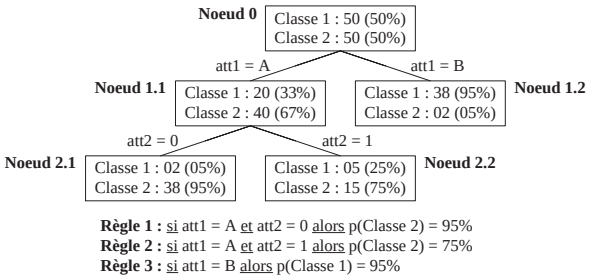

FIGURE 5.1 – Exemple d'arbre de décision

5.3.2 Entropie et gain d'information

Dans l'algorithme ID3 (*Induction Decision Tree*) [Qui86], le pouvoir discriminant d'une variable pour la segmentation d'un nœud est exprimé par une variation d'entropie. L'entropie h_s d'un nœud s_k (plus précisément, son entropie de Shannon) est :

$$h_s(s_k) = -\sum_{i=1}^{c} \frac{n_{ik}}{n_k} \log_2 \frac{n_{ik}}{n_k} \tag{5.3.1}$$

où n_k est l'effectif de s_k, n_{ik} le nombre d'objets de s_k qui appartiennent à la classe C_i et c la cardinalité de la classe. L'information portée par une partition S_K de K nœuds est alors la moyenne pondérée des entropies :

$$E(S_K) = \sum_{k=1}^{K} \frac{n_k}{n_j} h_s(s_k) \tag{5.3.2}$$

où n_j est l'effectif du nœud s_j qui est segmenté. Finalement, le gain informationnel associé à S_K est :

$$G(S_K) = h_s(s_j) - E(S_K) \tag{5.3.3}$$

Comme $G(S_K)$ est toujours positif ou nul, le processus de construction d'arbre de décision revient à une heuristique de maximisation de $G(S_K)$ à chaque itération et à la sélection de la variable correspondante pour segmenter un nœud donné. L'algorithme s'arrête lorsque $G(S_K)$ devient inférieur à un seuil (gain minimum) défini par l'utilisateur.

5.3.3 Exemple de Titanic

Pour illustrer notre approche de fouille de données en ligne, nous nous référons tout au long de ce chapitre à la base d'apprentissage Titanic (Table 5.1). Nous l'utilisons ici à titre d'exemple comme une table relationnelle. Il s'agit d'une table qui comporte trois attributs prédictifs *Classe* (1ère, 2ième, 3ième, Equipage), *Age* (Adulte, Enfant) et *Sexe* (Homme, Femme) ainsi qu'une classe à prédire *Survivant* (Oui, Non) pour une population totale de 2201 individus.

C'est une base d'apprentissage qui permet de déterminer si un individu aurait survécu ou non au naufrage du Titanic, en fonction de sa classe dans le navire, de son âge et de son sexe. L'attribut à prédire est donc *Survivant* et les trois attributs prédictifs sont : *Age*, *Sexe* et *Classe*.

Classe	*Age*	*Sexe*	*Survivant*
1ère	Adulte	Femme	Oui
3ème	Adulte	Homme	Oui
2ème	Enfant	Homme	Oui
3ème	Adulte	Homme	Oui
1ère	Adulte	Femme	Oui
2ème	Adulte	Homme	Non
1ère	Adulte	Homme	Oui
Equipage	Adulte	Femme	Non
...	...	...	...

TABLE 5.1 – Extrait de la base d'apprentissage Titanic

5.4 Fouille en ligne utilisant les vues relationnelles

Dans l'approche basée sur les vues relationnelles [BD02], nous représentons la racine d'un arbre de décision par une vue relationnelle qui correspond à la population d'apprentissage entière. Comme chaque nœud de l'arbre de décision représente une sous-population de son nœud parent, nous associons à chaque nœud une vue construite à partir de sa vue parente. Ces vues sont ensuite utilisées pour dénombrer les effectifs de chaque classe dans le nœud en utilisant de simples requêtes de regroupement et de comptage. Ces comptages servent finalement à déterminer le critère de partitionnement des nœuds en sous-partitions ou à conclure qu'un nœud est une feuille. La Figure 5.2 présente à titre d'exemple les commandes SQL permettant de créer les vues associées à l'arbre de décision de la Figure 5.1. Nous avons ensuite implémenté la méthode ID3 sous forme d'une procédure stockée *Buildtree* au sein du SGBD Oracle 9i.

```
nœud 0 :   CREATE VIEW v0 AS SELECT att1, att2, class FROM training_set
nœud 1.1 : CREATE VIEW v11 AS SELECT att2, class FROM v0 WHERE att1='A'
nœud 1.2 : CREATE VIEW v12 AS SELECT att2, class FROM v0 WHERE att1='B'
nœud 2.1 : CREATE VIEW v21 AS SELECT class FROM v11 WHERE att2=0
nœud 2.2 : CREATE VIEW v22 AS SELECT class FROM v11 WHERE att2=1
```

FIGURE 5.2 – Vues relationnelles associées à l'arbre de décision de l'exemple de la Figure 5.1

5.4.1 Construction de l'arbre de décision

En appliquant notre procédure stockée sur l'exemple Titanic présenté dans la Table 5.1, nous obtenons la vue relationnelle présentée dans la Table 5.2 qui permet par la suite d'obtenir l'abre de décision correspondant (Figure 5.3).

5.4.2 Discussion

L'avantage d'intégrer des méthodes de fouille de données au sein d'un SGBD est de bénéficier de sa puissance au niveau de l'accès aux données persistantes. En effet, les logiciels de fouille classiques nécessitent de charger la base de données en mémoire pour la traiter. Ils sont donc limités au niveau de la quantité de données analysables. Cet état de fait est illustré par la Figure 5.4, qui représente le temps de construction d'un arbre de décision sur la base d'apprentissage Titanic dont la taille augmente, avec d'une part des logiciels de fouille classiques (en l'occurrence SIPINA [ZR96] configuré pour utiliser la méthode ID3) et d'autre part, notre implémentation *Buildtree* d'ID3 sous Oracle.

Les différentes tailles de la base Titanic utilisées dans les tests sont obtenues par duplication de la base. Ces tests ont été effectués sur un ordinateur PC disposant de 128 Mo

Niveau	nœud	nœud parent	Règle	Survivant Non	P(NonSurvivant)	Survivant Oui	P(Survivant)
1	0			1490	68%	711	32%
2	1	0	Sexe = Femme	126	27%	344	73%
3	13	1	Classe = equipage	3	13%	20	87%
3	14	1	Classe = 1ière	4	3%	141	97%
4	21	14	Age = Enfant	0	0%	1	100%
4	22	14	Age = Adulte	4	3%	140	97%
3	15	1	Classe = 2ième	13	12%	93	88%
4	19	15	Age = Enfant	0	0%	13	100%
4	20	15	Age = Adulte	13	14%	80	86%
3	16	1	Classe = 3ième	106	54%	90	46%
4	17	16	Age = Enfant	17	55%	14	45%
4	18	16	Age = Adulte	89	54%	76	46%
2	2	0	Sexe = Homme	1364	79%	367	21%
3	3	2	Classe = Equipage	670	78%	192	22%
3	4	2	Classe = 1ière	118	66%	62	34%
4	11	4	Age = Enfant	0	0%	5	100%
4	12	4	Age = Adulte	118	67%	57	33%
3	5	2	Classe = 2ième	154	86%	25	14%
4	9	5	Age = Enfant	0	0%	11	100%
4	10	5	Age = Adulte	154	92%	14	8%
3	6	2	Classe = 3ième	422	83%	88	17%
4	7	6	Age = Enfant	35	73%	13	27%
4	8	6	Age = Adulte	387	84%	75	16%

TABLE 5.2 – Vue relationnelle associée à l'arbre de décision Titanic

de mémoire vive. Or, dans cette configuration SIPINA ne peut pas traiter des bases dont la taille dépasse 50 Mo.

Ce résultat montre que nos travaux permettent de continuer à traiter des bases de données de grande taille là où les logiciels travaillant en mémoire ne peuvent plus opérer. Cependant si nos résultats en terme de taille de bases à traiter sont prometteurs, les temps de traitements demeurent très longs. Bien que le temps de calcul ne soit pas généralement considéré comme un point critique dans un processus de fouille de données, il est néanmoins nécessaire de le réduire au maximum. Par ailleurs, l'expérience de la Figure 5.4 met en œuvre une base de données dont seul le nombre de n-uplets augmente. Or, en schématisant, la complexité des algorithmes de construction d'arbres de décision est linéaire selon le nombre d'objets (n-uplets), mais exponentielle selon le nombre de variables (attributs). Il est donc primordial d'optimiser le temps de traitement des algorithmes de fouille fonctionnant au sein d'un SGBD afin d'obtenir des temps de réponse acceptables.

5.5 Fouille en ligne utilisant la table de contingence

5.5.1 Construction de la table de contingence

Pour améliorer les temps de traitement, nous avons eu recours à un pré-traitement de la table d'apprentissage initiale afin de réduire sa taille. Ce pré-traitement consiste à

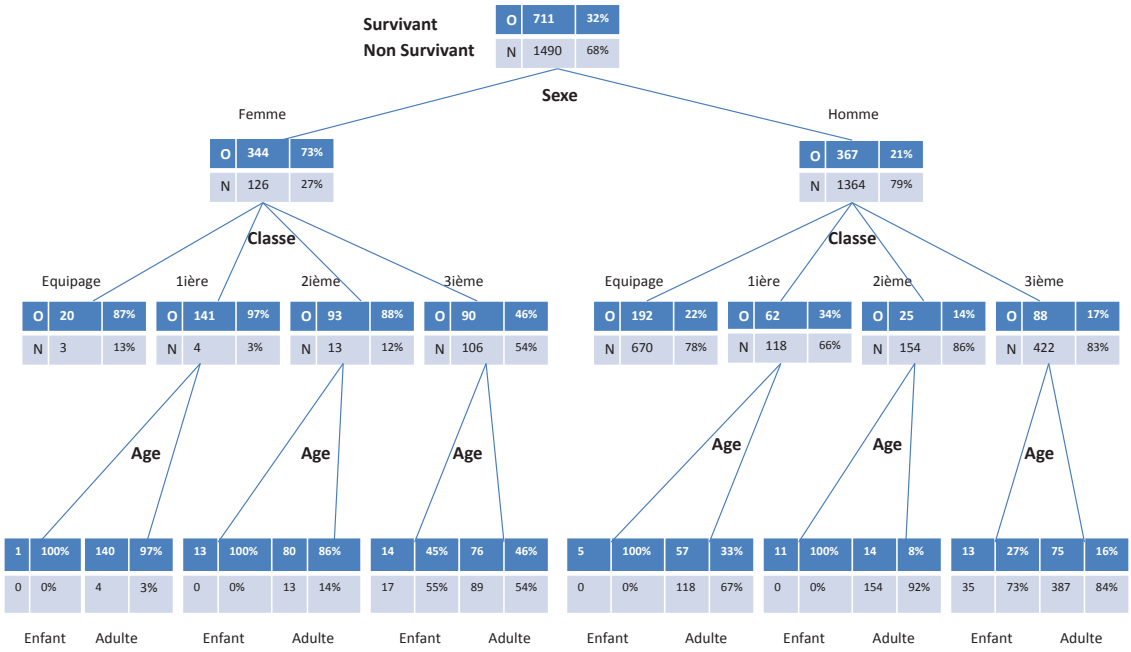

FIGURE 5.3 – Arbre de décision Titanic associé à la vue relationnelle de la Table 5.2

construire la table de contingence de Titanic (Table 5.3). Dans ce cas, au lieu de s'appliquer sur 2201 individus, la méthode ID3 va s'appliquer sur un nombre d'individus beaucoup plus réduit.

La table relationnelle (Table 5.4) équivalente à la table de contingence est obtenue avec

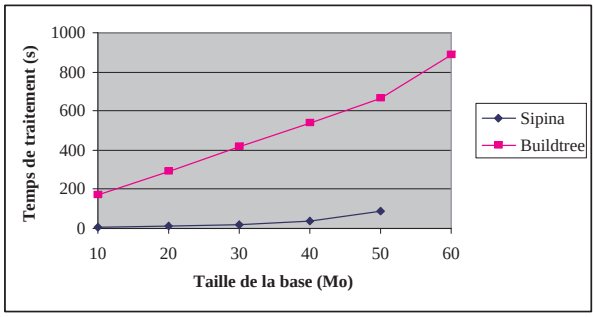

FIGURE 5.4 – Temps de traitement en fonction de la taille de la base

		Adulte		Enfant	
		Homme	Femme	Homme	Femme
1ère	Oui	57	140	5	1
	Non	118	4	0	0
2ième	Oui	14	80	11	13
	Non	154	13	0	0
3ième	Oui	75	76	13	14
	Non	387	89	35	17
Equipage	Oui	192	20	0	0
	Non	670	3	0	0

TABLE 5.3 – Table de contingence correspondant à la base Titanic

une simple requête SQL (Figure 5.5). Elle ne contient que 24 n-uplets.

```
CREATE VIEW Contingence AS SELECT Classe, Sexe, Age, Survivant, COUNT(*)
AS Effectif FROM Titanic GROUP BY Classe, Sexe, Age, Survivant
```

FIGURE 5.5 – Vue relationnelle associée à la table de contingence : Exemple du Titanic

5.5.2 Calcul du gain d'information et de l'entropie

Pour démontrer la pertinence et l'efficacité de notre approche, nous avons implémenté à nouveau la méthode ID3 en tenant compte de la phase de préparation des données. La véritable différence se situe lors du calcul du gain d'information pour chaque attribut prédictif et par conséquent lors du calcul de l'entropie.

Les implémentations de l'algorithme standard sont contraintes, pour calculer le gain d'information d'un attribut prédictif, de lire tous les n-uplets de la partie de la base de départ correspondant au nœud courant de l'arbre d'induction afin de déterminer la répartition des n-uplets en fonction de chaque valeur de l'attribut prédictif et de chaque valeur de la classe.

Dans notre approche, pour connaître l'effectif de la population d'un nœud obtenu à partir d'un ensemble de critères E_r (*Age*=Enfant et *Sexe*=Femme, par exemple), il suffit d'effectuer la somme des valeurs de l'attribut *Effectif* de la table de contingence (Table 5.4) pour les n-uplets satisfaisant E_r. Cette technique est donc beaucoup plus rapide car elle s'applique sur un nombre bien plus restreint de n-uplets.

Enfin, en modifiant la manière de le calculer, il devient possible de n'effectuer qu'une seule lecture pour connaître le gain d'information d'un attribut prédictif. En effet, comme nous l'avons vu dans la section 5.3.2, le calcul normal du gain pour un attribut ayant K

Classe	*Age*	*Sexe*	*Survivant*	*Effectif*
1ère	Adulte	Homme	Oui	57
1ère	Adulte	Homme	Non	118
1ère	Adulte	Femme	Oui	140
1ère	Adulte	Femme	Non	4
1ère	Enfant	Homme	Oui	5
1ère	Enfant	Femme	Oui	1
2ième	Adulte	Homme	Oui	14
2ième	Adulte	Homme	Non	154
2ième	Adulte	Femme	Oui	80
2ième	Adulte	Femme	Non	13
2ième	Enfant	Homme	Oui	11
2ième	Enfant	Femme	Oui	13
3ième	Adulte	Homme	Oui	75
3ième	Adulte	Homme	Non	387
3ième	Adulte	Femme	Oui	76
3ième	Adulte	Femme	Non	89
3ième	Enfant	Homme	Oui	13
3ième	Enfant	Homme	Non	35
3ième	Enfant	Femme	Oui	14
3ième	Enfant	Femme	Non	17
Equipage	Adulte	Homme	Oui	192
Equipage	Adulte	Homme	Non	670
Equipage	Adulte	Femme	Oui	20
Equipage	Adulte	Femme	Non	3

TABLE 5.4 – Table relationnelle associée à la table de contingence de la base Titanic

valeurs possibles et avec une classe ayant c valeurs possibles est de :

$$gain = (entropiedunoeud) - \sum_{k=1}^{K} \left(\frac{n_k}{n_j} \times \left(- \sum_{i=1}^{c} \frac{n_{ik}}{n_k} \times log_2 \left(\frac{n_{ik}}{n_k} \right) \right) \right) \tag{5.5.1}$$

où n_k est l'effectif du nœud ayant la valeur V_k pour l'attribut prédictif, n_j est l'effectif de la population du nœud, n_{ik} est l'effectif du nœud ayant la valeur V_k pour l'attribut prédictif et la valeur C_i pour la classe. Or, en développant (5.5.1), on obtient :

$$gain = (entropie\ du\ noeud) + \frac{1}{n_j} \times \sum_{k=1}^{K} \left(n_k \times \frac{1}{n_k} \times \sum_{i=1}^{c} n_{ik} \times log_2 \left(\frac{n_{ik}}{n_k} \right) \right) \tag{5.5.2}$$

De plus, $log_2 \frac{a}{b}$ étant égal à $log_2 a - log_2 b$ on obtient d'après (5.5.2) :

$$gain = (entropie\ du\ noeud) + \frac{1}{n_j} \times \sum_{k=1}^{K} \left(\sum_{i=1}^{c} n_{ik} \times (log_2 n_{ik} - log_2 n_k) \right) \tag{5.5.3}$$

En développant (5.5.3), on obtient :

$$gain = (entropie\ du\ noeud) + \frac{1}{n_j} \times \left(\sum_{k=1}^{K} \sum_{i=1}^{c} n_{ik} \times log_2 n_{ik} - \sum_{k=1}^{K} \sum_{i=1}^{c} n_{ik} \times log_2 n_k \right) \quad (5.5.4)$$

Or

$$\sum_{i=1}^{c} n_{ik} \times log_2 n_k = n_k \times log_2 n_k$$

D'où, d'après (5.5.4) :

$$gain = (entropie\ du\ noeud) + \frac{1}{n_j} \times \left(\sum_{k=1}^{K} \sum_{i=1}^{c} n_{ik} \times log_2 n_{ik} - \sum_{k=1}^{K} n_k \times log_2 n_k \right) \quad (5.5.5)$$

En appliquant la Formule 5.5.5 sur la table de contingence que nous ne lisons qu'une seule et unique fois, nous obtenons facilement le gain. En effet, dans cette formule il n'est pas nécessaire de connaître au même moment les différents effectifs (n_j, n_k, n_{ik}) et on obtient n_k par somme sur les n_{ik} et n_j par somme sur les n_k.

5.5.3 Etude de complexité

L'étude de complexité suivante nous permet d'appuyer les résultats expérimentaux obtenus. Soient N le nombre de n-uplets de la base de départ, K le nombre d'attributs prédictifs et T la taille (nombre de n-uplets) de la table de contigence. Notre objectif est de comparer la complexité entre l'algorithme avec vues relationnelles (*Buildtree*) et celui utilisant la table de contingence ("TC_ID3"). Nous considérons que les deux algorithmes sont optimisés dans leur implémentation de telle sorte que seuls les n-uplets nécessaires sont lus.

Dans cette étude, nous nous intéressons au temps passé à la lecture et à l'écriture des données car ce sont les opérations les plus coûteuses. Nous considérons qu'un n-uplet est lu ou écrit en une unité de temps. Enfin, nous considérons que l'arbre de décision obtenu est équilibré et complet, c'est-à-dire qu'à chaque niveau de l'arbre, l'union des populations des différents nœuds du niveau équivaut à la base toute entière.

Pour l'algorithme avec vues relationnelles, pour un niveau i quelconque de l'arbre, pour aboutir au niveau $i+1$, il faut lire chaque nœud autant de fois qu'il existe d'attributs prédictifs à ce niveau, c'est-à-dire $(K - i)$. Comme la somme des populations des nœuds du niveau correspond à la population de la base de départ, il est donc nécessaire de lire N n-uplets $(K - i)$ fois, autrement dit le nombre de n-uplets $\times$ le nombre d'attributs candidats. Le temps total de lecture pour le niveau i est donc de $N(K - i)$. Or, pour

obtenir ce niveau, il a fallu écrire les n-uplets correspondants. Le temps d'écriture est donc de N.

En rappelant que $\sum_{i=1}^{K} i = K(K+1)/2$, nous obtenons alors une complexité finale de la racine jusqu'aux feuilles (niveau K) égale à :

- en lecture : $N(K^2/2 - K/2)$ unités de temps, donc en NK^2 ;
- en écriture : NK unités de temps.

Pour notre approche, il convient d'abord de créer la table de contingence donc un temps d'écriture de T. Pour l'obtenir, il convient de lire intégralement la base de départ une première fois, soit un temps de lecture de N. A chaque niveau i, pour aboutir au niveau $i+1$, on lit l'intégralité des T n-uplets $(K-i)$ fois, soit un temps de $T(K-i)$ pour chaque niveau.

Avec création de la table de contingence, les résultats sont donc :

- en lecture : $T(K^2/2-K/2)+N$ unités de temps, donc en TK^2 ou en N si $N > TK^2$;
- en écriture : T unités de temps.

Ainsi, en temps de traitement, notre approche apporte une amélioration en N/T ou en K^2 (si $N > TK^2$) pour la lecture et en NK/T pour l'écriture. Comme N est normalement très supérieur à T, cette amélioration est importante et, de plus, elle augmente avec le nombre d'attributs.

5.6 Fouille en ligne utilisant les index bitmap

5.6.1 Index bitmap

Un index bitmap est une structure de données définie dans un SGBD, utilisée pour optimiser l'accès aux données dans les bases de données. C'est un type d'indexation qui est particulièrement intéressant et performant dans le cadre des requêtes de sélection. L'index bitmap d'un attribut est codé sur des bits, d'où son faible coût en terme d'espace occupé. Toutes les valeurs possibles de l'attribut sont considérées, que la valeur soit présente ou non dans la table. A chacune de ces valeurs correspond un tableau de bits, appelé bitmap, qui contient autant de bits que de n-uplets présents dans la table. Ainsi, ce type d'index est très efficace lorsque les attributs ont un faible nombre de valeurs distinctes. Chaque bit représente donc la valeur d'un attribut, pour un n-uplet donné. Pour chacun des bits, il y a un codage de présence/absence (1/0), ce qui traduit le fait qu'un n-uplet présente ou non la valeur caractérisée par le bitmap.

Les index bitmap possèdent une propriété très intéressante qui consiste à répondre à certains types de requêtes sans retourner aux données elles-mêmes, optimisant ainsi les temps de réponse. Cela est possible grâce aux opérations de comptage ($COUNT$) et aux

opérateurs logiques (*AND*, *OR*, etc) qui agissent "bit à bit" sur les bitmap.

Pour illustrer nos propos, nous nous référerons à un échantillon de la base d'apprentissage Titanic (Table 5.1) (ne sont représentés ici que les huit premiers n-uplets de la table). Ainsi, l'index bitmap construit sur l'attribut *Survivant* de cette table se présente de la façon suivante (Table 5.5).

Classe	Age	Sexe	Survivant
1ère	Adulte	Femme	Oui
3ème	Adulte	Homme	Oui
2ème	Enfant	Homme	Oui
3ème	Adulte	Homme	Oui
1ère	Adulte	Femme	Oui
2ème	Adulte	Homme	Non
1ère	Adulte	Homme	Oui
Equipage	Adulte	Femme	Non
...	...	...	...

ID n-uplet	...	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*
"*Non*"	...	1	0	1	0	0	0	0	0
"*Oui*"	...	0	1	0	1	1	1	1	1

TABLE 5.5 – Table Titanic et index bitmap construit sur l'attribut *Survivant*.

Considérons, par exemple, la question : "Combien y a-t-il eu d'hommes qui ont survécu au naufrage du navire ?" Celle-ci peut être formulée par la requête SQL suivante : 'SELECT COUNT(*) FROM Titanic WHERE *Survivant* = "*Oui*" AND *Sexe* = "*Homme*"'. La réponse à cette requête est donnée en se basant uniquement sur les index bitmap des attributs *Sexe* et *Survivant*, sans parcourir la table Titanic.

Cela consiste à effectuer une opération AND entre le bitmap associé à la valeur "*Oui*" pour l'attribut *Survivant* et le bitmap associé à la valeur "*Homme*" pour l'attribut *Sexe* (Table 5.6). Un comptage des "*1*" dans le bitmap résultat permet de déterminer l'effectif.

ID n-uplet	...	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*
Survivant="*Oui*"	...	0	1	0	1	1	1	1	1
Sexe="*Homme*"	...	0	1	1	0	1	1	1	0
AND	...	0	**1**	0	0	**1**	**1**	**1**	0

TABLE 5.6 – Bitmap (*Survivant*="Oui") AND Bitmap (*Sexe*="Homme").

5.6.2 Construction de la base d'apprentissage

Etant donnée une base d'apprentissage initiale, nous construisons les index bitmap sur tous les attributs (attributs prédictifs et attribut à prédire) de celle-ci. L'ensemble de ces index bitmap constitue alors la nouvelle base d'apprentissage.

A partir de la table d'apprentissage Titanic, nous construisons la base d'apprentissage équivalente en utilisant les index bitmap (Table 5.7). Cela revient à considérer la représentation disjonctive complète de la population. Il s'agit en effet de coder de manière binaire chacune des valeurs distinctes de l'ensemble des attributs. Le fait de représenter la base

d'apprentissage par l'ensemble de ses index bitmap permet de réduire à la fois la taille de la base d'apprentissage et les temps de traitement. Nous montrons, dans ce qui suit, comment les seules informations contenues dans les index bitmap permettent de construire les arbres de décision.

	ID n-uplet	...	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*
Classe	*Equipage*	...	1	0	0	0	0	0	0	0
	1ère	...	0	1	0	1	0	0	0	1
	2ème	...	0	0	1	0	0	1	0	0
	3ème	...	0	0	0	0	1	0	1	0
Age	*Enfant*	...	0	0	0	0	0	1	0	0
	Adulte	...	1	1	1	1	1	0	1	1
Sexe	*Femme*	...	1	0	0	1	0	0	0	1
	Homme	...	0	1	1	0	1	1	1	0
Survivant	*Non*	...	1	0	1	0	0	0	0	0
	"Oui"	...	0	1	0	1	1	1	1	1

TABLE 5.7 – Base d'apprentissage : ensemble des index bitmap de la base Titanic.

5.6.3 Construction de l'arbre de décision

Construction du nœud racine. Le nœud racine est caractérisé par les effectifs de l'attribut à prédire, sans tenir compte des valeurs des différents attributs prédictifs. Pour construire le nœud racine et, par conséquent, pour obtenir les différents effectifs de l'attribut à prédire, il faut effectuer un simple comptage des "*1*" sur chacun des bitmaps de l'index de l'attribut à prédire. Ainsi, dans le cas de la base Titanic (5.7), le nœud racine est déterminé selon la Figure 5.6.

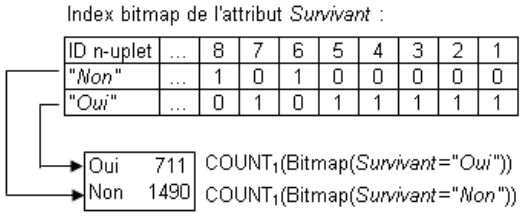

FIGURE 5.6 – Construction du nœud racine.

Construction d'un nœud quelconque. La détermination de la distribution des effectifs de l'attribut à prédire pour un nœud quelconque se fait en trois étapes.

1) *Caractériser la population du nœud courant par un unique bitmap*
   Chacun des nœuds descendant directement de la racine est caractérisé par le bitmap correspondant à la valeur de l'attribut de segmentation qui génère ce nœud courant. Pour les autres nœuds, ce bitmap est le résultat d'une ou de plusieurs opérations *AND*

qui traduisent la règle de construction du nœud, en considérant les valeurs prises par les attributs de segmentation successifs, de la racine jusqu'au nœud courant.

2) *Caractériser la population du nœud courant pour chaque valeur de l'attribut à prédire par un bitmap résultat*
   Appliquer l'opérateur *AND* entre le bitmap caractéristique de la population du nœud courant et chacun des bitmaps de l'index de l'attribut à prédire.

3) *Déterminer les effectifs de l'attribut à prédire pour le nœud courant*
   Cela revient à faire un comptage des "*1*" dans chacun des bitmaps résultats. Ces bitmaps résultats représentent les différentes populations du nœud courant, en tenant compte de la valeur de l'attribut à prédire.

Avant la création de chaque nouvelle partition, un attribut doit être sélectionné pour effectuer la segmentation. Il correspond à celui qui dispose d'un gain informationnel maximal. Le gain informationnel correspondant à une variation d'entropie, il peut être directement calculé à partir des index bitmap. En effet, le calcul d'entropie nécessite de déterminer différents effectifs et donc d'effectuer des comptages. Comme nous avons pu le montrer précédemment, ces comptages peuvent être faits de manière efficace en utilisant les index bitmap. Ainsi, la partition optimale est donc construite en utilisant les index bitmap, non seulement lors du choix de l'attribut de segmentation à travers les calculs d'entropie mais également pour générer la partition lors de la construction des différents nœuds.

Dans notre exemple, le calcul du gain informationnel indique que l'attribut à considérer pour la segmentation du nœud racine est *Sexe*. Suivant la méthode proposée, il s'agit alors de caractériser les populations par des bitmaps, pour ensuite calculer les effectifs correspondants (Table 5.8).

ID n-uplet	...	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*
Sexe="Homme"	...	0	1	1	0	1	1	1	0
Survivant="Oui"	...	0	1	0	1	1	1	1	1
AND	...	0	1	0	0	1	1	1	0

ID n-uplet	...	*8*	*7*	*6*	*5*	*4*	*3*	*2*	*1*
Sexe="Homme"	...	0	1	1	0	1	1	1	0
Survivant="Non"	...	1	0	1	0	0	0	0	0
AND	...	0	0	1	0	0	0	0	0

TABLE 5.8 – Bitmaps caractérisant les hommes ayant et n'ayant pas survécu.

En agissant de même pour la valeur *Femme* de l'attribut *Sexe*, on obtient alors la segmentation présentée dans la Figure 5.7. On procède alors de la même façon pour les autres nœuds, jusqu'à obtenir l'arbre de décision final.

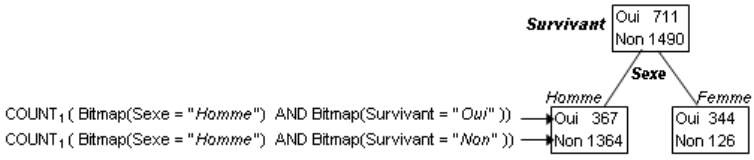

FIGURE 5.7 – Arbre de décision obtenu après segmentation selon l'attribut Sexe.

5.6.4 Etude de complexité

L'étude de complexité suivante nous permet de confirmer les résultats expérimentaux obtenus. Cette étude est menée d'un point de vue théorique. Nous raisonnons dans le "pire des cas", en l'occurence, nous supposons que les index ne tiennent pas en mémoire.

Soient N le nombre de n-uplets de la base initiale, K le nombre d'attributs, L la longueur moyenne en bits de chaque attribut. Soit A le nombre moyen de valeurs par attribut.

Nous nous intéressons d'abord à la taille des bases d'apprentissage. Etant données les notations adoptées, la taille de la base initiale est de $N * L * K$ bits. Pour notre approche, l'étape préalable à la construction de l'arbre est de créer la population d'apprentissage constituée de l'ensemble des index bitmaps construits sur chacun des attributs. K index bitmap sont donc créés avec en moyenne A bitmaps par index. Chaque bitmap a une taille de N bits. La taille de l'ensemble des index bitmap est donc de $N * A * K$ bits. En terme de taille de base d'apprentissage, et donc de temps de chargement, l'approche par index bitmap est plus avantageuse dès lors que $A < L$, ce qui correspond à une majorité des cas.

Nous nous intéressons à présent au temps passé à la lecture des données. Nous considérons qu'un bit est lu en une unité de temps.

Le nombre total de nœuds au niveau de profondeur i peut être approximé par A^{i-1}, puisque A est le nombre moyen de valeurs des attributs. En effet, l'hypothèse posée est que l'arbre de décision est équilibré et complet.

Le nombre d'attributs prédictifs restant à considérer pour le niveau i est de $(K-i)$. La base d'apprentissage doit être lue une fois par attribut restant dans les deux approches, soit $(K-i)$ fois.

Dans le cadre de l'approche "classique", la taille de la base d'apprentissage à lire est approximée par $N * L * K$. Ainsi, au niveau i, le temps de lecture s'exprime de la façon suivante (en unités de temps) : $(K-i) * N * L * K * A^{i-1}$. On obtient alors pour la construction de l'ensemble de l'arbre un temps de : $\sum_{i=1}^{K}(K-i) * N * L * K * A^{i-1}$

Dans le cadre de l'approche utilisant les bitmaps, la taille d'un index à lire est approximée par $N * A$ bits. Au niveau de profondeur i de l'arbre, pour un attribut prédictif donné,

le nombre d'index à lire pour pouvoir déterminer le niveau suivant est $(i+1)$. Ainsi, pour le niveau i, le temps de lecture dans l'approche par index bitmap, s'exprime de la façon suivante : $(i+1)(K-i)N.A^i$; ce qui donne pour la construction de l'ensemble de l'arbre : $\sum_{i=1}^{K}(i+1)(K-i)N * A^i$

Pour évaluer le gain, on construit le ratio :

$R = \frac{temps\ approche\ classique}{temps\ approche\ index\ bitmap} = \frac{\frac{KL}{A}\sum_{i=1}^{K}(K-i)*A^i}{\sum_{i=1}^{K}(K-i)(i+1)*A^i}$.

Après développement, on a : $R = \frac{\frac{KL}{A}\sum_{i=1}^{K}(K-i)*A^i}{\sum_{i=1}^{K}(K-i)*A^i+\sum_{i=1}^{K}i(K-i)*A^i}$

$R^{-1} = \frac{A}{KL}(1 + \frac{\sum_{i=1}^{K}i(K-i)*A^i}{\sum_{i=1}^{K}(K-i)*A^i}) = \frac{A}{KL}(1+G)$

En considérant les polynômes de plus haut degré, G est de complexité K. R^{-1} est donc de complexité $\frac{A}{L}$. En effet $R^{-1} = \frac{A}{KL}(1+K) = \frac{A}{L}(1+\frac{1}{K})$ et $\frac{1}{K}$ est négligeable. Notre approche par index bitmap est intéressante dans le cas où le ratio R^{-1} est inférieur à 1, donc si $A < L$. Ce qui correspond à une majorité des cas.

5.7 Implémentation et performance

Pour procéder à des tests de performance, nous avons implémenté la méthode ID3 selon les trois approches de fouille en ligne en utilisant les procédures stockées en PL/SQL, compatibles sous oracle 10g. L'adaptation de l'algorithme ID3 (pour le calcul de l'entropie et du gain d'information) fut donc nécessaire afin qu'il puisse s'appliquer sur la table de contingence.

Afin de valider nos différentes approches et de comparer leurs performances entre elles et vis-à-vis aussi des méthodes de fouille classiques, nous avons effectué des tests sur la base CovType [1].

La base de données CovType contient 581,012 n-uplets définis par 54 attributs prédictifs, dont les dix premiers sont discrets, et un attribut à prédire (avec 7 valeurs distinctes). A partir de la base CovType, nous avons construit 5 vues de telle sorte que la taille de la vue i est égale à i * taille (vue 1). Chaque vue créée est définie par 3 attributs prédictifs ayant chacun 5 valeurs distinctes et l'attribut à prédire de CovType. Les caractéristiques de chacune des vues créée sont données dans la Table 5.9. Ces tests ont été réalisés sur un ordinateur PC avec 1.50GHz et 512 MB de mémoire vive sous le SGBD Oracle 10g.

La Figure 5.8 montre les résultats des tests effectués sur les différentes implementations d'ID3. La méthode classique utilisant le logiciel Sipina [ZR96], la méthode intégrée basée sur les vues, la méthode intégrée basée sur la table de contingnece et celle basée sur les index bitmap sont baptisées $Sipina\_ID3$, $View\_ID3$, CT_ID3 et $Bitmap\_ID3$

1. http ://ftp.ics.uci.edu/pub/machine-learning-databases/covtype/

Vue	Attributs prédictifs utilisés	Taille de la vue (en n-uplets)	taille de la vue (en MB)
vue 1	1,2,3	116202	454
vue 2	4,5,6	232404	908
vue 3	7,8,9	348607	1362
vue 4	1,4,10	464810	1816
vue 5	2,5,8	581012	2270

TABLE 5.9 – Vues CovType utilisées pour les tests

respectivement. Pour les méthodes intégrées, nous avons ajouté au temps de traitement le temps nécessaire à la construction des index bitmap et la table de contingence. En ce qui concerne le temps de traitement de la méthode classique, il inclut le temps de chargement des données en mémoire.

Le premier résultat important auquel nous sommes arrivée est le suivant. Lorsque les bases de données utilisées pour la fouille de données atteignent une taille supérieure à 2270 MB, la méthode *Sipina_ID3* en mémoire n'est pas capable de construire l'arbre de décision alors que nos approches intégrées le sont. Cela paraît évident comme résultat puisque *Sipina_ID3* opérant en mémoire est limitée par la taille de celle-ci.

Nous pouvons également souligner le gain induit par nos méthodes intégrées comparées à l'approche classique. En effet, le temps de traitement pour $Sipina\_ID3$ croît de 16 à 80 secondes lorsque la taille de la base est multiplié par 5. Ainsi, le temps de traitement pour $Sipina\_ID3$ est multiplié par 5 alors qu'il est multiplié par 3 pour les méthodes basées sur les vues et les index bitmap. Quant à la méthode basée sur la table de contingence, le temps de traitement reste à peu près stable.

Maintenant si on compare nos différentes méthodes de fouille intégrées, le temps de traitement croît de façon plus lente. Cette croissance est presque identique pour $View\_ID3$ et $Bitmap\_ID3$ (de 9 à 22 secondes pour $View\_ID3$, et de 5 à 16 secondes pour $Bitmap\_ID3$). Le temps de traitement pour la méthode CT_ID3 est presque constant (de 2 à 3 secondes).

Les résultats expérimentaux montrent clairement que la méthode basée sur la table de contingence est la meilleure. Pour CT_ID3, le gain induit dépend principalement de la taille de la table de contingence qui est généralement beaucoup plus petite que la taille de la base d'apprentissage initiale. Néanmoins, dans des cas extrêmes, il peut arriver que la taille de la table de contingence soit si proche de celle de la base d'apprentissage de départ que le gain en devient infime. Ces cas restant toutefois en pratique très rares, l'utilisation de la table de contingence améliore considérablement les temps de traitement.

La méthode $View\_ID3$ est la plus lente car il y a de multiples accès aux données puisque nous n'utilisons aucun outil d'optimisation. La méthode basée sur les index bitmap

est environ 30% plus rapide que $View\_ID3$ en moyenne. Ce résultat était attendu car les index bitmap évitent les accès aux données de la base.

En conclusion, nous pouvons dire que nos méthodes de fouille intégrées sont très intéressantes pour de grandes bases de données. Cependant, Sipina demeure une méthode très rapide pour le traitement, mais est très lente pour le chargement de données en mémoire alors que nos méthodes intégrées se comportent de manière opposée. De plus, le temps de chargement augmente plus rapidement que le temps de traitement lorsque la base croît considérablement. Enfin, l'utilisation de la table de contingence comme un outil d'optimisation des tables d'apprentissage améliore de façon importante les temps de traitement.

Nous avons regroupé nos trois procédures stockées relatives à $View\_ID3$, CT_ID3 et $Bitmap\_ID3$ dans un même prototype logiciel nommé *Decision_Tree* accessible en ligne [2].

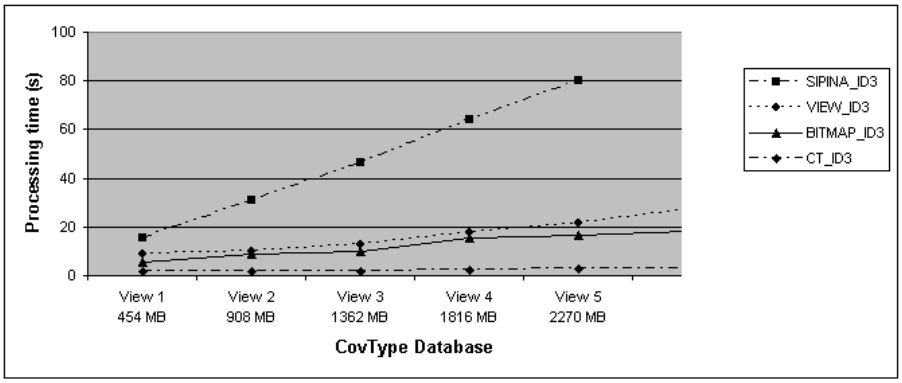

FIGURE 5.8 – Comparaison de la performance des implémentations d'ID3

5.8 Fouille en ligne dans les cubes OLAP

Les résultats prometteurs obtenus avec la fouille en ligne en utilisant la table de contingence nous ont conduit à orienter nos travaux vers la fouille dans les entrepôts de donnnées et en particulier dans les cubes OLAP [Mad04]. L'algorithme de fouille est ainsi appliqué sur les données agrégées du cube et évitent par conséquent le retour aux données de l'entrepôt. Néanmoins, les cubes construits doivent être basés sur la fonction d'agrégat *Count* dans le cas par exemple des méthodes basées sur les arbres de décision. En effet, c'est exactement cette fonction d'agrégat qui est utilisée pour calculer les effectifs des différents nœuds de l'arbre de décision.

2. http ://eric.univ-lyon2.fr/ bentayeb/logiciels.html

Le cube de données obtenu peut alors constituer notre base d'apprentissage et les effectifs des différentes sous populations de l'arbre de décision peuvent être obtenus par de simples requêtes SQL. L'avantage majeur de travailler avec les cubes est d'éviter l'accès aux données sources et de présenter une solution technique pour faire de la fouille dans les cubes. Par conséquent, ceci permet la diminution des accès récurrents à la base originelle et la réduction des temps induits par les traitements. Pour valider notre approche, nous avons implémenté la méthode ID3 opérant sur un cube de données au sein du SGBD Oracle 10g.

5.8.1 Entrepôt de données Titanic

Pour illustrer notre approche de fouille dans les cubes, nous avons construit un entrepôt de données à partir de la base d'apprentissage source des passagers du Titanic (Figure 5.9). Le fait à analyser est le "nombre de passagers" survivants ou non survivants selon les dimensions *Classes*, *Ages*, et *Sexes*.

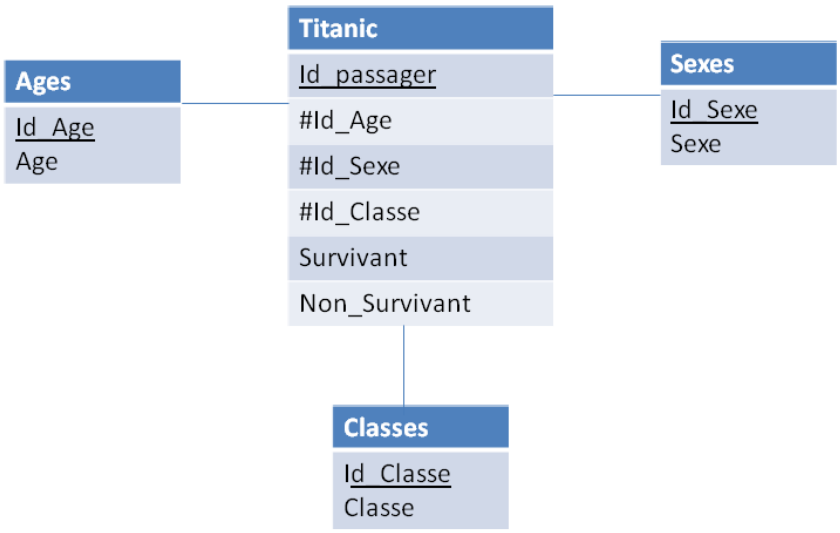

FIGURE 5.9 – Entrepôt de données Titanic

5.8.2 Construction du cube Titanic

En appliquant la requête décisionnelle présentée dans la Figure 5.10 sur l'entrepôt de données Titanic, nous obtenons le cube OLAP présenté dans la Table 5.10 qui constituera notre base d'apprentissage. Dans notre exemple, les attributs *Survivant* et *Non_Survivant* sont les attributs à prédire et correspondent aux mesures qui se trouvent dans la table de

faits de l'entrepôt Titanic. Les attributs prédictifs, *Age*, *Sexe* et *Classe* correspondent aux dimensions du cube.

```
CREATE VIEW TitanicCube AS
SELECT Age, Sexe, Classe, Count(Survivant), Count(NonSurvivant)
FROM Titanic, Ages, Sexes, Classes
WHERE Titanic.IdAge=Ages.IdAge
And Titanic.IdSexe=Sexes.IdSexe
And Titanic.IdClasse=Classes.IdClasse
GROUP BY CUBE Age, Sexe, Classe
```

FIGURE 5.10 – Requête décisionnelle pour la construction du Cube de données Titanic

5.8.3 Construction de l'arbre de décision d'ID3

La première étape consiste à créer le nœud racine. Il est caractérisé par les différents effectifs des sous-populations qui sont définies selon les modalités de la classe à prédire, sans tenir compte des valeurs des différents attributs prédictifs. Pour le nœud racine, nous cherchons donc à obtenir l'effectif de la population pour laquelle on a *Survivant* = 'oui' d'une part et celui pour laquelle *Survivant* = 'non' d'autre part. Pour cela, nous avons besoin de chercher dans le cube OLAP TitanicCube uniquement les effectifs des attributs mesures *Survivant* et *Non_Survivant* pour lesquels nous avons la valeur *ALL*, *ALL*, *ALL* sur tous les autres attributs prédictifs. Autrement dit, il suffit d'exécuter la requête SQL présentée dans la Figure 5.11. Le résultat obtenu constitue les effectifs attendus pour le nœud racine de l'abre de décision d'ID3 (Figure 5.12).

```
SELECT Survivant, Non_Survivant
FROM TitanicCube
WHERE Ages.Age = ALL
And Sexes.Sexe = ALL
And Classes.Classe = ALL
```

FIGURE 5.11 – Requête d'extraction du nœud racine

De manière générale, pour connaître l'effectif de la population d'un nœud obtenu à partir d'un ensemble de critères E, il suffit de sélectionner dans le cube les deux valeurs des attributs *Survivant* et *Non_Survivant* pour lesquels E est satisfait.

Survivant	*Non_Survivant*
711	1490

FIGURE 5.12 – Nœud racine de l'arbre d'ID3

Supposons que l'attribut prédictif qui segmente le nœud racine soit l'attribut Sexe. Il s'avère que cet attribut possède deux modalités : Homme et Femme. Les nœuds fils de la racine de l'arbre sont donc caractérisés par les règles Sexe = "Femme" d'une part et Sexe = "Homme" d'autre part. Etudions le nœud issu de la règle sexe = "Femme". Le principe est le même pour le nœud associé à la règle Sexe = "Homme". Deux effectifs lui sont rattachés : celui des survivants (*Survivant*) et celui des non survivants (*Non_Survivant*). Pour obtenir ces deux effectifs, nous exécutons la requête présentée dans la Figure 5.13.

```
SELECT Survivant, Non_Survivant
FROM TitanicCube
WHERE Sexes.Sexe = 'Femme'
And Ages.Age = ALL
And Classes.Classe = ALL
```

FIGURE 5.13 – Requête d'extraction du nœud fils *Sexe* = 'Femme'

5.8.4 Bilan

Grâce à notre approche de fouille de données dans les cubes, nous avons la possibilité d'extraire des connaissances à partir de données agrégées sans accéder aux données sources. L'application de la méthode ID3 sur les cubes OLAP nécessite une adaptation de son algorithme. La véritable différence se situe lors du calcul du gain informationnel pour chaque attribut prédictif et par conséquent lors du calcul de l'entropie. En effet, pour calculer le gain informationnel d'un attribut prédictif, les implémentations de l'algorithme classique sont contraintes de lire tous les n-uplets correspondants au nœud courant de l'arbre d'induction. Dans notre approche, pour connaître l'effectif de la population d'un nœud, une seule lecture suffit.

5.9 Synthèse de notre approche de fouille de données en ligne

La Figure 5.14 résume notre approche de fouille de données en ligne qui améliore considérablement les performances de l'analyse lorsque les bases sont très volumineuses. C'est pourquoi, nous avons étendu la fouille dans les bases de données à la fouille dans les entrepôts de données (cubes OLAP).

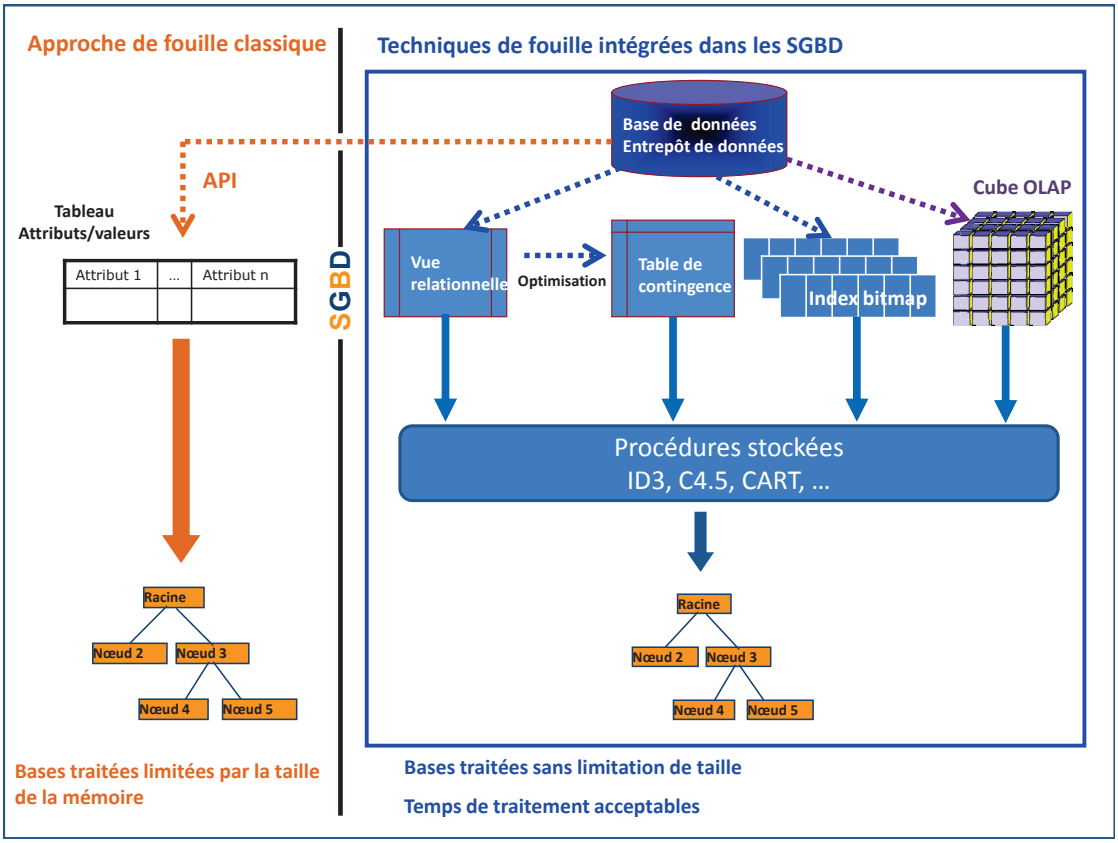

FIGURE 5.14 – Processus de fouille de données en ligne

5.10 Conclusion

L'objectif que nous avons poursuivi dans ce chapitre est d'étendre les capacités analytiques des SGBD, support des entrepôts de données, de l'OLAP vers la fouille de données afin de fournir à l'utilisateur des analyses plus élaborées que les simples analyses exploratoires et navigationnelles proposées par l'analyse en ligne. Pour atteindre cet objectif, nous avons présenté une approche de fouille de données en ligne intégrée au sein du SGBD Oracle. Notre contribution dans ce domaine porte essentiellement sur l'exploitation des outils offerts par les SGBD pour assurer des temps de chargement et de traitement acceptables pour des bases d'apprentissage sans limitation de taille. A notre connaissance, aucune approche de fouille intégrée existante n'a procédé de la sorte. En effet, tous les algorithmes de fouille en ligne disponibles dans les SGBD sont implémentés sous forme de "boites noires" difficiles à maîtriser et les propositions de recherche décrites dans la littérature font appel à des extensions de SQL ou à des API. Par ailleurs, nos travaux s'inscrivent dans la continuité de l'intégration de l'analyse OLAP dans les SGBD. En ef-

fet, l'intégration de la fouille au sein des SGBD étend les possibilités d'analyse des données entreposées vers la structuration, l'explication et la prédiction.

Nous avons aussi systématiquement cherché à démontrer l'efficacité de nos propositions de fouille en ligne en mettant en œuvre un processus d'expérimentation qui a impliqué leur implémentation sur un système existant, en l'occurence Oracle. Les résultats expérimentaux mettent en évidence deux résultats importants. D'une part, les méthodes de fouille opérant en mémoire sont limitées par la taille de celle-ci alors que les méthodes de fouille intégrées ne le sont pas. D'autre part, l'utilisation de la table de contingence améliore considérablement les temps de traitement. De plus, afin d'appuyer et de confirmer nos résultats expérimentaux, nous avons réalisé une étude de complexité théorique pour chacune de nos propositions.

Si l'approche d'intégration des méthodes de fouille de données au sein des SGBD peut paraître simple, elle n'en demeure pas moins intéressante. Comme nous l'avons souligné auparavant, l'intérêt de la fouille en ligne est double : d'abord un intérêt de l'ordre de l'optimisation de performance, et ensuite un intérêt de l'ordre de l'analyse descriptive, explicative et prédictive au sein des SGBD. C'est dans ce contexte que nous pouvons inscrire plusieurs pistes de recherche.

L'une des principales évolutions possibles de ce travail est l'intégration d'autres méthodes de fouille au sein des SGBD. Nous avons déjà adapté, dans le cadre des méthodes de fouille de type "arbres de décision", la méthode Cart. Par ailleurs, l'intégration de méthodes de fouille non supervisées apporteraient certainement une nouvelle dimension analytique dans le contexte des entrepôts de données. Par exemple, dans le cadre des travaux sur la combinaison de l'OLAP avec la fouille de données, notre démarche de fouille en ligne peut s'avérer très pertinente. Nos premiers résultats dans ce contexte ont consisté à intégrer la méthode de classification (*k-means*) au sein du SGBD Oracle et l'utiliser par la suite pour recommander de nouveaux axes d'analyse à l'utilisateur (Chapitre 4). En effet, nous considérons que l'exploitation de la fouille en ligne est une piste intéressante dans le cadre de la personnalisation et de la recommandation dans les entrepôts de données.

5.11 Publications

la liste suivante présente nos publications concernant les travaux que nous avons menés sur la fouille de données en ligne.

Revue internationale

[1] **F. Bentayeb**, J. Darmont, C. Favre, C. Udréa, "Efficient On-Line Mining of Large Databases", International Journal of Business Information Systems, Vol. 2, No. 3,

2007, 328-350.

Conférences internationales

[2] C. Favre, **F. Bentayeb**, "Bitmap Index-based Decision Trees", 15th International Symposium on Methodologies for Intelligent Systems (ISMIS 05), New York, USA, May 2005 ; LNAI, Vol. 3488, Springer, 65-73.

[3] **F. Bentayeb**, J. Darmont, C. Udréa, "Efficient integration of data mining techniques in DBMSs", 8th International Database Engineering and Applications Symposium (IDEAS 04), Coimbra, Portugal, July 2004 ; Institute of Electrical and Electronics Engineers (IEEE) proceedings, 59-67.

[4] **F. Bentayeb**, J. Darmont, "Decision tree modeling with relational views", 13 th International Symposium on Methodologies for Intelligent Systems (ISMIS 2002), Lyon, France ; Lecture Notes of Artificial Intelligence (LNAI), Vol. 2366, 423-431.

Conférences nationales

[5] C. Favre, **F. Bentayeb**, "Intégration efficace des arbres de décision dans les SGBD : utilisation des index bitmap", 5èmes journées francophones d'Extraction et de Gestion des Connaissances (EGC 05), Paris, Janvier 2005 ; Revue des Nouvelles Technologies de l'Information, Vol. E-3, Cépaduès, 319-330.

[6] C. Udréa, **F. Bentayeb**, "Fouille de Données Relationnelles dans les SGBD", 5èmes Journées d'Extraction et de Gestion des Connaissances (EGC 05), Paris, Janvier 2005 ; Revue des Nouvelles Technologies de l'Information, 356-356.

Age	Sexe	Classe	Survivant	Non_Survivant
Adulte	Femme	1ière	140	4
Adulte	Femme	2ième	80	13
Adulte	Femme	3ième	76	89
Adulte	Femme	Equipage	20	3
Adulte	Femme	ALL	316	109
Adulte	Homme	1ière	57	118
Adulte	Homme	2ième	14	154
Adulte	Homme	3ième	75	387
Adulte	Homme	Equipage	192	670
Adulte	Homme	ALL	338	1329
Adulte	ALL	1ière	97	122
Adulte	ALL	2ième	94	167
Adulte	ALL	3ième	151	476
Adulte	ALL	Equipage	212	673
Adulte	ALL	ALL	654	1438
Enfant	Femme	1ière	1	0
Enfant	Femme	2ième	13	0
Enfant	Femme	3ième	14	17
Enfant	Femme	ALL	28	17
Enfant	Homme	1ière	5	0
Enfant	Homme	2ième	11	0
Enfant	Homme	3ième	13	35
Enfant	Homme	ALL	29	35
Enfant	ALL	1ière	6	0
Enfant	ALL	2ième	24	0
Enfant	ALL	3ième	27	52
Enfant	ALL	ALL	57	52
ALL	Femme	1ière	141	4
ALL	Femme	2ième	93	13
ALL	Femme	3ième	90	106
ALL	Femme	Equipage	20	3
ALL	Femme	ALL	344	126
ALL	Homme	1ière	62	118
ALL	Homme	2ième	25	154
ALL	Homme	3ième	88	422
ALL	Homme	Equipage	192	670
ALL	Homme	ALL	367	1364
ALL	ALL	1ière	203	122
ALL	ALL	2ième	118	167
ALL	ALL	3ième	178	528
ALL	ALL	Equipage	212	673
ALL	ALL	ALL	711	1490

TABLE 5.10 – Cube OLAP TitanicCube associé à la requête décisionnelle de la Figure 5.10

Chapitre 6

Intégration sémantique de données pour l'analyse en ligne à la demande

La problématique traitée dans ce chapitre s'inscrit dans la continuité de nos travaux sur la conception et la construction de systèmes d'information décisionnels centrés utilisateur. Il s'agit ici de rendre possible l'analyse en ligne à la demande en développant un système d'information décisionnel capable de prendre en compte l'évolution des données (schémas et instances) ainsi que l'évolution des besoins des utilisateurs tout en garantissant, en plus de l'intégration structurelle, l'intégration sémantique des données. Se posent alors plusieurs questions : Quel est le modèle d'intégration de données le plus adapté ? (2) comment représenter puis intégrer les données ? et enfin (3) comment réaliser l'analyse en ligne à la demande ?

Notre contribution dans ce domaine réside dans notre choix d'un modèle d'intégration par médiation permettant d'assurer une intégration sémantique des données en utilisant des ontologies au niveau local (sources de données) et au niveau global (médiateur) selon le modèle d'intégration GLAV (*Generalized Local as View*). Pour traiter l'hétérogénéité structurelle et sémantique des données, nous avons proposé un algorithme de fusion des ontologies locales en une ontologie globale en utilisant la fouille de données. Par ailleurs, nous avons défini une mesure de similarité sémantique adaptée. Dans la suite de ce chapitre, nous motivons le choix de notre modèle d'intégration de données pour l'analyse en ligne à la demande et présentons ensuite notre algorithme de fusion d'ontologies locales en une ontologie globale.

Ce travail a fait l'objet de la thèse de doctorat de N. Maïz que nous avons co-encadrée, préparée au sein du laboratoire ERIC et soutenue en 2010 [Mai10].

6.1 Motivation

Mener la réflexion sur la place de l'utilisateur dans les systèmes d'information décisionnels nous a permis d'identifier des élements clés qui peuvent contribuer à remettre l'utilisateur au centre du système en tant que partie prenante. Dans un contexte où les sources de données évoluent, où de nouveaux besoins émergent au fur et à mesure du temps, où les usages individuels se distinguent d'un utilisateur à l'autre, et les données à entreposer sont complexes, il apparaît évident que de nouveaux modèles supportant des processus d'aide à la décision doivent voir le jour pour prendre en compte tous les élements nécessaires à la mise en place d'un système d'information décisionnel centré utilisateur.

L'idée principale derrière la conception d'un système d'information décisionnel centré utilisateur est de créer une réelle interactivité entre l'utilisateur et le système durant tout son cycle de vie. Cela suppose de suivre une démarche de conception du système qui prenne en compte tout d'abord l'évolution des sources tout en proposant des solutions pour la résolution de l'hétérogénéité structurelle et sémantique des données. Ensuite, il s'agit de prendre en compte l'évolution des besoins individuels des utilisateurs qui implique une mise en place d'une stratégie d'analyse en ligne à la demande. Dans ce contexte, la pertinence des résultats obtenus dépendra fortement de la qualité des données prises en compte pour l'analyse.

Intéressons-nous tout d'abord au processus d'intégration de données et aux modèles qui en découlent. L'intégration de données a pour objectif de combiner des sources de données autonomes distribuées et hétérogènes afin d'obtenir une vue homogène et uniforme des données intégrées. Pour y parvenir, toutes les données doivent être représentées selon un même schéma global et selon une sémantique unifiée. Deux approches principales pour la conception des systèmes d'intégration ont été définies dans la littérature :

1) l'intégration virtuelle de données où la vue unifiée est virtuelle et les données restent stockées dans les sources d'origine. L'architecture type pour l'intégration virtuelle de données est l'architecture *médiateur* [Wie92] ;
2) l'intégration matérialisée de données où la vue unifiée de données est matérialisée et les données sont rapatriées des sources d'origine et stockées dans un *entrepôt de données* [Inm96, KRRT00].

En réalité, quelle que soit l'architecture utilisée pour l'intégration de données, nous sommes confrontée aux deux problèmes importants suivants. Le premier problème est celui de la réconciliation des données avec un schéma (schéma global du médiateur ou schéma de l'entrepôt). Ce problème consiste à déterminer quel élément du schéma est représenté par quel élément de la source de données. Le deuxième problème se pose lorsqu'on s'intéresse à l'intégration des sources conformément à un schéma ou lorsque les sources sont hétérogènes.

Pour traiter l'hétérogénéité des schémas et des données, la majorité des travaux menés se contentent de l'exploitation des informations présentes dans les schémas et dans les données sous forme de métadonnées. Se pose alors la question suivante : Est-ce que les métadonnées suffisent pour régler l'hétérogénéité des données ? Malheureusement, la réponse est non car il suffit de prendre quelques exemples pour se rendre compte que l'hétérogénéité des données dans les sources pose des problèmes d'ordre sémantique que les seules métadonnées ne peuvent régler. En effet, la seule présence des homonymes dans les sources suffit pour dire que les métadonnées ne suffisent pas à régler l'hétérogénéité des données. Par exemple, dans l'une des sources la donnée "montant" représente le montant global des achats alors que dans l'autre source la donnée "montant" représente le montant global des ventes. Les mêmes données (même nom) peuvent ne pas représenter la même information et n'ont donc pas la même sémantique.

Concernant la prise en compte de l'évolution des sources et des besoins dans les entrepôts de données, des solutions de modélisation existantes ont pu répondre au problème de l'évolution des données en empruntant le concept d'évolution de schéma. Deux courants de modèles avec évolution de schéma existent dans la littérature : les modèles temporels dans lesquels nous trouvons les modèles avec plusieurs versions du modèle d'entrepôt et les modèles mis à jour qui ne gardent que la dernière version du modèle. Cependant, aucune des approches proposées ne répondait au problème de l'évolution des besoins. C'est pourquoi, nous avons dans un premier temps, orienté nos travaux afin de prendre en compte les nouveaux besoins des utilisateurs en proposant un modèle d'entrepôt évolutif pour la personnalisation des analyses en suivant l'approche mise à jour du schéma (cf. Chapitre 3). Enfin, lorsque nous observons toutes les propositions de modélisation à base d'évolution de schéma, y compris nos travaux, nous sommes contraints de dire qu'aucun de ces travaux ne prend en compte à la fois l'évolution des données et celle des besoins. Alors que les modèles avec plusieurs versions permettent de gérer l'évolution des sources de données mais échouent face à l'évolution des besoins, les modèles mis à jour prennent en compte l'évolution des besoins mais s'avèrent inadaptés à l'évolution des sources de données.

Partant du constat que les modèles d'intégration de données selon l'approche matérialisée (entrepôt) ne répondent pas à notre objectif d'élaboration de systèmes d'information décisionnels pour la prise en compte à la fois de l'évolution des données et celle des besoins, il apparaît indispensable d'élaborer de nouvelles architectures décisionnelles. Une première piste de recherche possible consiste à combiner dans un même modèle d'entrepôt les deux approches d'évolution de schéma dans les entrepôts (temporelle et mise à jour) qui permettrait de résoudre le problème. Cependant, cette solution présente un inconvénient majeur puisqu'il faut gérer en plus des versions des entrepôts, l'évolution de schéma dans chaque version de l'entrepôt. Une deuxième piste possible, dans laquelle peuvent s'inscrire nos travaux, est le recours à l'approche d'intégration virtuelle de données (médiateur) qui

a fait l'objet de nombreux travaux dans des domaines tels que la recherche d'information ou les bases de données, mais constitue un axe de recherche assez récent dans le domaine de l'analyse en ligne.

En suivant l'approche d'intégration par médiation, notre objectif est alors de construire des cubes de données à la demande tout en garantissant l'intégation sémantique des données. Pour intégrer les données, en plus des métadonnées, des techniques de comparaison syntaxique telle que les mesures de similarité ont été également utilisées, cependant cela n'a pas permis de régler le problème de l'hétérogénéité sémantique des données. Une idée possible pour rendre explicite la sémantique des sources (données et schémas) est de la définir de manière exhaustive pour toutes les données et tous les schémas. Seulement, il est impossible d'envisager une telle démarche pour décrire la sémantique des sources de données.

Le premier verrou scientifique auquel nous nous sommes confrontée est comment réaliser une intégration sémantique des données dans un système d'information décisionnel basé sur la médiation ? Les ontologies qui sont définies comme "une spécification explicite et formelle d'une conceptualisation commune" [Gru95] peuvent contribuer à la résolution du problème de l'hétérogénéité sémantique. En effet, elles offrent une description formelle des concepts et de leurs relations. Le deuxième verrou scientifique que nous avons soulevé consiste à réaliser l'analyse en ligne à la demande selon les besoins évolutifs des utilisateurs.

En conclusion, nous pouvons dire que dans un contexte où les données de l'entreprise changent rapidement et les besoins décisionnels peuvent évoluer eux aussi, l'intégration de données par médiation en utilisant les ontologies paraît l'approche d'intégration la mieux adaptée pour suivre ces différentes évolutions afin de réaliser des analyses en ligne à la demande.

Dans ce chapitre, nous proposons donc un système d'information décisionnel permettant de construire des cubes de données à la demande fondé sur un modèle d'intégration par médiation dans lequel chaque source de données est décrite par une ontologie locale. Ce modèle d'intégration permet à l'utilisateur d'être au cœur du processus décisionnel. En effet, contrairement aux entrepôts de donnnées qui limite le rôle de l'utilisateur à mener des analyses en naviguant dans les données, avec un système de médiation, il a la possibilité de fixer le schéma de son cube de données au moment même de réaliser l'analyse. Plusieurs questions scientifiques sont alors à considérer. Sachant que dans un système de médiation, le schéma global est l'intégration des schémas locaux, le problème qui se pose alors est comment fusionner les ontologies locales qui décrivent les schémas locaux en une ontologie globale décrivant le schéma global ? Quelle mesure de similarité utiliser pour aider à résoudre l'hétérogénéité sémantique des données ?

Nous avons choisi de développer dans ce chapitre le volet concernant la fusion des ontologies locales pour la création de l'ontologie globale qui décrit le médiateur. Notre

contribution dans ce travail consiste en la définition d'un algorithme de fusion des ontologies locales en une ontologie globale en utilisant la technique de classification ascendante hiérarchique (CAH) [LW67]. Pour comparer les concepts des ontologies entre eux, nous avons défini une mesure de similarité sémantique adaptée. L'approche de création de cubes de données à la demande est largement développée dans la thèse de N. Maïz [Mai10].

6.2 Système de médiation et ontologies

Un système de médiation est défini par trois couches : celle des sources de données, celle du médiateur et enfin celle des correspondances entre les sources et le médiateur. Dans la première couche, dite aussi "couche des schémas locaux", une définition des schémas des sources est imposée. La couche médiateur ou la couche du schéma global, donne une vision globale sur les sources de données. La troisième couche définit l'ensemble des correspondances entre le schéma global et les schémas locaux afin d'assurer le lien entre les deux types de schémas pour faciliter l'accès aux données locales et réconcilier les conflits sémantiques entre les systèmes locaux. Le problème qui se pose alors est l'identification puis la résolution des conflits entre les entités dans les différentes sources qui sont sémantiquement liées. Les conflits sémantiques se produisent lorsque deux contextes n'emploient pas la même interprétation d'informations. En effet, trois causes principales pour l'hétérogénéité sémantique sont identifiées [CMS03] : (1) les conflits de confusion qui se produisent quand les concepts semblent avoir la même signification, mais diffèrent en réalité ; (2) les conflits de graduation qui se produisent lorsque différents systèmes de référence sont employés pour mesurer une valeur et, (3) les conflits de nom qui se produisent lors de l'attribution des noms dans des schémas qui diffèrent de manière significative. Un phénomène fréquent est la présence des homonymes et des synonymes.

Par ailleurs, l'ontologie, dont l'avènement au sein du domaine de l'ingénierie des connaissances a eu lieu au cours des années 90, est une conceptualisation des objets du domaine selon un certain point de vue imposé par l'application. Elle est conçue comme un ensemble de concepts organisés à l'aide de relations structurantes, dont la principale avec laquelle est construite l'ossature taxonomique de l'ontologie, est la relation "$is-a$". Cette conceptualisation est écrite dans un langage de représentation des connaissances qui propose des "services inférentiels" (classification de concepts, capacité de construire des concepts définis à partir de concepts primitifs, etc.). Une ontologie peut prendre différentes formes. Cependant, elle inclut nécessairement un vocabulaire de termes et une spécification de leur signification [BAGC04]. L'utilisation des ontologies dans le processus d'intégration de données n'est pas nouvelle. Elle permet une intégration structurelle et sémantique des données et contribue à résoudre les conflits entre les concepts des sources de données. Plusieurs architectures d'ontologies ont été adoptées dans un système de médiation : avec une

seule ontologie, avec plusieurs ontologies ou de façon hybride [WVV+01]. Le choix d'une architecture dépend de la nature évolutive du système ainsi que de la difficulté de définir les correspondances entre les différentes ontologies. En effet, l'architecture hybride consiste à définir une ontologie locale pour chaque source de données, ainsi qu'une ontologie globale pour le médiateur. L'ontologie globale est le résultat de la fusion des différentes ontologies locales. Il s'avère indispensable de résoudre les conflits sémantiques entre les différentes ontologies locales et de définir les correspondances entre elles d'une part, et avec l'ontologie globale d'autre part.

Lors de la consultation des données au niveau de leurs sources, l'utilisateur doit formuler sa requête dans le langage du médiateur, c'est-à-dire celui du schéma global. Le médiateur prend par la suite la requête utilisateur et la décompose en un ensemble de sous-requêtes et procède à l'identification des sources qui peuvent apporter une réponse à chaque sous-requête en utilisant les correspondances définies auparavant. Une fois les sous-requêtes exécutées, le médiateur recompose les résultats afin d'avoir une réponse globale. En fait, la stratégie de décomposition des requêtes et de recomposition des sous-requêtes est appelée réécriture de requêtes. Dans un système de médiation, la stratégie de réécriture des requêtes dépend directement du modèle d'intégration utilisé pour la conception du médiateur. En réalité, il existe deux modèles d'intégration : le modèle GaV (*Global as View*) et le modèle LaV (*Local as View*) [HR04]. La combinaison des deux modèles précédents donne le modèle GLAV (*Generalized Local As View*) [Len01].

GaV a été la première approche proposée pour intégrer des données [HR04]. Elle consiste à définir à la main (ou de façon semi-automatique) le schéma global en fonction des schémas des sources de données à intégrer puis à le connecter aux différentes sources. Pour cela, les prédicats du schéma global, appelés aussi relations globales, sont définis comme des vues sur les prédicats des schémas des sources à intégrer. Parmi les systèmes utilisant l'approche GaV, on peut citer TSIMMIS [CGMH+94] et MOMIS [BBC+00, BBGV01].

L'approche LaV suppose l'existence d'un schéma global et consiste à définir les schémas des sources de données à intégrer comme des vues du schéma global. Les principaux systèmes développés selon cette approche sont : Infomaster [GK97], PICSEL [RBF+02, GLR00] et Information Manifold [LRO96].

Le modèle GLAV suppose l'utilisation des vues au niveau local et global. Le traitement de requêtes dans GLAV nécessite une réécriture et un dépliement et n'est pas toujours réalisable. Cependant, le traitement de requêtes dans le cadre d'une architecture avec plusieurs ontologies modélisées selon GLAV est possible si la requête est exprimée dans un langage qui prend en charge le niveau global et local [Len01].

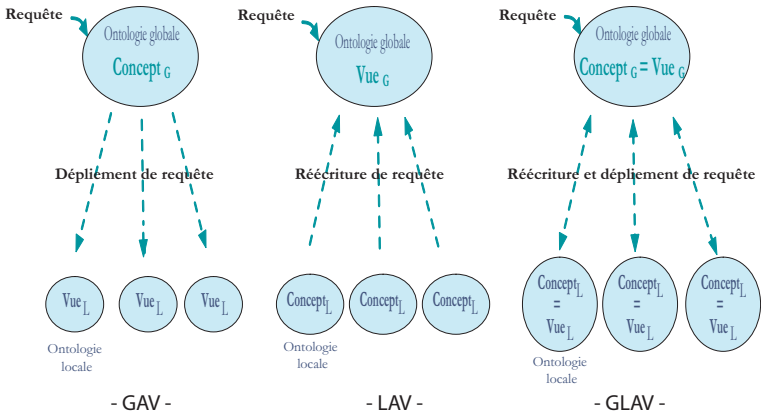

FIGURE 6.1 – Différents modèles d'intégration

Plusieurs modèles de médiation à base d'ontologies existent dans la littérature. En ce qui concerne nos travaux qui s'inscrivent dans un contexte où les sources de données évoluent, nous pensons que l'architecture de médiation à bases d'ontologies multiples est la mieux adaptée à nos attentes en termes d'analyse en ligne à la demande. Nous présentons dans la Figure 6.2 les différents couches qui composent l'architecture décisionnelle pour l'analyse en ligne à la demande. En effet, à chaque source de données est associée une ontologie locale. L'ontologie globale est obtenue par fusion des ontologies locales. L'utilisateur doit composer sa requête sous forme de conjonction de concepts et de propriétés du vocabulaire de l'ontologie globale et de celui des ontologies locales. Ensuite, un mécanisme de réécriture de requêtes doit assurer la décomposition et la recomposition des résultats élémentaires de la requête décomposée.

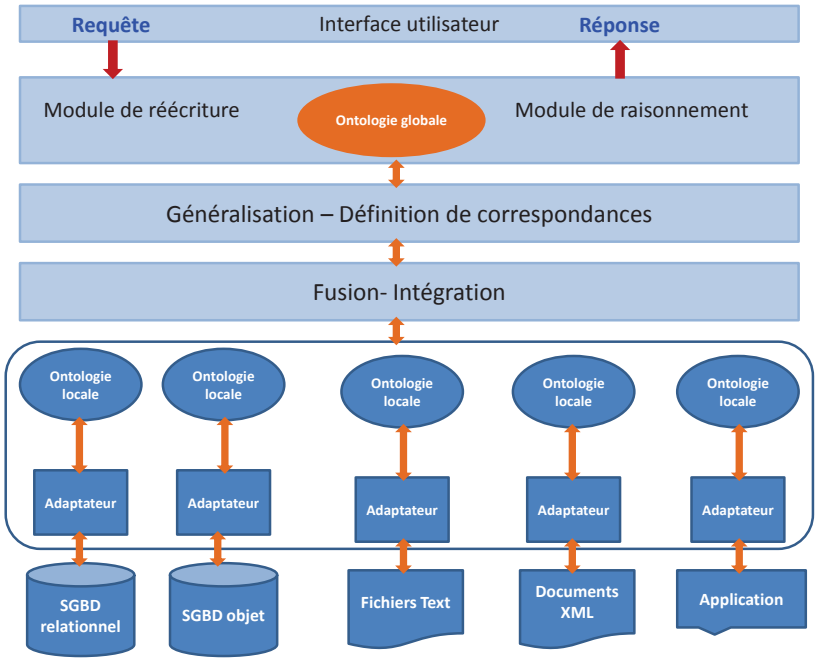

FIGURE 6.2 – Architecture de médiateur à base d'ontologies selon GLAV

6.3 Alignement et Fusion des ontologies

Nous trouvons dans la littérature plusieurs méthodes de fusion des ontologies, fondées sur des algorithmes d'alignement eux-mêmes basés sur diverses mesures de similarité. Nous présentons tout d'abord quelques unes de ces méthodes et mesures puis nous présentons notre approche de fusion. L'originalité de notre approche de fusion est l'utilisation de la technique de classification ascendante hiérarchique pour permettre la découverte de connaissances implicites dans les concepts (entités) des ontologies et la construction des classes de concepts (propriétés, instances) équivalents (synonymes). La construction des classes s'appuie sur une mesure de similarité que nous avons définie et qui prend en compte la terminologie, la structure et la sémantique des concepts. Les classes de concepts obtenues seront utilisées par la suite pour aider à résoudre les conflits sémantiques pouvant exister entre les concepts lors de la construction de l'ontologie globale.

6.3.1 Méthodes d'alignement et de réconciliation de schémas ou d'ontologies

Plusieurs travaux sur l'alignement des ontologies ont été publiés. Ils traitent une étape du processus de fusion qui est la découverte des correspondances entre les entités des on-

tologies à aligner. Nous présentons ici un ensemble d'approches permettant d'aligner des schémas ou des ontologies. Autrement dit, il s'agit de trouver des similarités ou des correspondances entre des éléments de deux schémas ou ontologies qui sont sémantiquement liées. De nombreux travaux se sont penchés sur la question dans le but d'automatiser le processus d'alignement [RB01, SE05]. Les correspondances trouvées permettent d'effectuer différentes tâches relatives à la manipulation de données hétérogènes exprimées dans des schémas ou des ontologies distincts. L'objectif étant d'intégrer ces données, soit grâce à un adaptateur dans le cadre d'un système d'intégration de type médiateur, soit grâce à un extracteur dans le cadre de la construction d'un entrepôt de données. Dans le contexte de nos travaux, ces correspondances peuvent permettre de fusionner des ontologies.

L'alignement d'ontologies peut être décrit comme suit [EGV05] : étant données deux ontologies dont chacune représente un ensemble d'entités (classes, propriétés ou instances), l'alignement d'ontologies consiste à définir l'ensemble des relations, telles que l'équivalence ou la subsomption qui existent entre ces deux ensembles d'attributs. Il s'agit donc de construire des *ponts* entre les différentes entités appartenant aux différentes ontologies. Le problème de fusion des ontologies consiste à utiliser les résultats du processus d'alignement (les relations entre les différentes entités) pour construire une ontologie globale contenant toutes les connaissances existantes dans les différentes ontologies. Le problème principal dans le processus d'alignement est le calcul des similarités entre les entités. D'autre part, le passage à l'échelle, constitue également une difficulté. Les mesures de similarité utilisées par les approches de réconciliation de schémas diffèrent en fonction :

(a) des connaissances qu'elles peuvent prendre en compte. Il s'agit par exemple de connaissances terminologiques (connaissances textuelles), structurelles ou sémantiques. Souvent, la richesse de ces connaissances dépend fortement du modèle de représentation utilisé pour le schéma ou l'ontologie. Il peut s'agir dans certaines applications d'une simple taxonomie alors que nous pouvons trouver une ontologie décrite avec le langage OWL-DL (*Ontology Web Language-Description Logics*) dans d'autres applications ;

(b) des types de relations de mise en correspondance qu'elles peuvent trouver telles que les relations d'équivalence entre des couples d'entités appartenant à des ontologies distinctes comme par exemple la subsomption, le recouvrement ou même la disjonction.

Plusieurs mesures de similarité existent dans la littérature. Nous trouvons les mesures basées sur la structure interne (les éléments sont pris de façon isolée), les mesures basées sur la structure externe (prise en compte des éléments et de leurs voisins), les méthodes terminologiques, les méthodes basées sur les ensembles d'instances (distance de Hamming [Ham50], indice de Jaccard [1]), et enfin les mesures de similarité sémantiques.

1. L'indice de Jaccard (ou coefficient de Jaccard) est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles.

6.3.2 Fusion des ontologies

Dans le domaine des ontologies, différentes méthodes de fusion ont été proposées dans la littérature. Ces dernières sont basées sur le côté syntaxique, lexical ou sémantique du vocabulaire de l'ontologie. Nous citons par exemple Anchor-Prompt [NM01], FCA-Merge [SM01], HCONE [KVS06], S-Match [GSY04]. Dans [NM01], les auteurs proposent une méthode de comparaison des ontologies de domaine par la définition des correspondances entre leurs concepts. Ils choisissent deux paires de concepts équivalents comme référence. Chaque paire appartient à une ontologie. Ensuite, ils sélectionnent tous les concepts intermédiaires deux à deux qui occupent les mêmes positions dans deux chemins de même longueur, reliant les deux concepts de la même paire. Cela permet aux auteurs de juger si ces paires de concepts sont équivalents ou pas. Autrement dit, deux concepts se trouvant dans la même position entre deux concepts équivalents sont également équivalents. Anchors-Prompt suppose que les deux ontologies sont construites de la même façon. Ce n'est pas le cas dans la réalité ! Dans FCA-Merge, les auteurs définissent une méthode formelle et ascendante de fusion des ontologies en se basant sur un ensemble de documents. Ils appliquent des techniques de traitement du langage naturel et d'analyse formelle de concepts pour dériver le treillis des concepts. Ce dernier est exploré et transformé en une ontologie par l'intervention de l'être humain. Fernandez-Breis et Martinez Béjar ont proposé une plate-forme coopérative pour l'intégration des ontologies [FBMB02]. L'idée principale de leur travail est de prendre en entrée un ensemble d'ontologies pouvant être distribuées mais représentant le même domaine. Ils appliquent un algorithme sur ces ontologies pour définir une ontologie intégrée basé sur des caractéristiques taxonomiques ainsi que sur la détection de concepts synonymes. L'algorithme prend également en considération les attributs des concepts. Par exemple, si les noms des concepts sont égaux (en se basant sur des critères d'égalité des noms), les concepts doivent avoir aussi les mêmes attributs.

6.3.3 Discussion

Les travaux existants concernant la fusion et l'alignement des ontologies sont très nombreux et variés. Nous avons cependant identifié quelques points principaux qui sont susceptibles d'être améliorés. Les approches de fusion et d'alignement des ontologies utilisent un seuil de stabilisation pour arrêter le processus d'alignement ; ce qui limite la propagation de la similarité et par conséquent réduit la précision de la méthode. De plus, ces méthodes sont utilisées pour aligner deux ontologies. En réalité, il peut exister plusieurs ontologies qui décrivent le même domaine et qui nécessitent d'être alignées pour devenir réutilisables. Il est donc nécessaire de proposer une approche de fusion capable de traiter plusieurs ontologies à la fois.

6.4 Classification ascendante hiérarchique pour la fusion des ontologies

6.4.1 Motivation

Notre objectif dans ce travaill est de proposer une approche de fusion des ontologies capable de prendre en compte les points soulevés dans la Section 6.3.3. En premier lieu, afin d'assurer le passage à l'échelle (plusieurs ontologies), il est nécessaire de mettre en œuvre une démarche automatique qui facilite le calcul des similarités entre les entités. Pour cela, nous avons combiné la technique de classification ascendante hiérarchique, avec le mécanisme d'inférence offert par le langage des ontologies OWL-DL. Notre idée est de construire des classes d'entités similaires et de propager ensuite cette similarité sur les axiomes pour en détecter de nouveaux définissant les nouvelles relations entre les entités. Pour cela, nous avons défini une mesure de similarité sémantique adaptée.

6.4.2 Principe général

Notre approche de fusion automatique des ontologies se base non seulement sur les classes de concepts obtenues par l'algorithme de CAH [DLST03] mais aussi sur un mécanisme d'inférence. La combinaison de la CAH avec le mécanisme d'inférence permet de trouver les classes d'entités (concepts, propriétés, instances) provenant des différentes ontologies ainsi que leurs relations d'équivalence ou de subsomption permettant de construire une nouvelle ontologie consistante. L'algorithme de classification considère au départ chaque entité comme une classe à part entière. Ensuite, il essaye de fusionner les classes qui sont équivalentes. A chaque fusion de deux classes, une nouvelle relation d'équivalence entre les entités est créée. Nous devons donc propager cette équivalence vers toutes les entités qui sont liées aux éléments de la nouvelle classe. L'algorithme continue à fusionner les classes jusqu'à ce qu'il n'y ait plus de classes à fusionner. Le résultat de notre algorithme est un ensemble de classes d'entités équivalentes ainsi qu'un ensemble d'axiomes définissant les relations entre les différentes classes.

6.4.3 Mesure de similarité sémantique

Nous définissons un concept C^i par le vecteur suivant :

$$C^i = (T^i, Att^i_1, ..., Att^i_{m(i)}, P^i_1, ..., P^i_{n(i)})$$

où T^i est le terme qui décrit le concept C^i, $Att^i_1, ..., Att^i_{m(i)}$ représentent les attributs du concept C^i et les $P^i_1, ..., P^i_{n(i)}$ représentent les propriétés du concept C^i ou ses relations avec

ses concepts voisins. Ces différents éléments vont être utilisés pour comparer la sémantique des différents concepts.

Le calcul de la similarité entre deux concepts permet de déterminer leur proximité. Cette mesure est basée sur la terminologie du concept, de ses attributs, de ses propriétés et de ses relations avec ses voisins. En effet, pour deux concepts qui possèdent la même terminologie, les mêmes attributs et ont des relations identiques avec des voisins similaires, il y a une forte chance qu'ils soient identiques. Pour déterminer la similarité entre deux concepts C^i et C^j, on doit d'abord calculer la similarité entre leurs termes T^i et T^j, puis entre les différentes paires d'attributs (Att^i_x $(x := 1, ..m(i))$, Att^j_y $(y := 1, ..m'(j))$).

Définition 6.1 *Similarité terminologique entre deux termes*
La similarité des termes notée Sim_T est une similarité lexicale et syntaxique. Elle est obtenue en utilisant une distance entre chaînes de caractères qui est l'indice de $Jaccard$ [2] et en faisant référence à Wordnet [3] [Mil95].

Définition 6.2 *Similarité terminologique entre deux attributs*
Soient C^i et C^j deux concepts. La similarité entre deux attributs Att^i_x et Att^j_y de C^i et C^j respectivement, est une similarité terminologique et est notée $Sim_T(Att^i_x, Att^j_y)$.

Définition 6.3 *Similarité d'attributs entre deux concepts*
Soient $C^i = (T^i, Att^i_1, ..., Att^i_{m(i)}, P^i_1, ..., P^i_{n(i)})$ et $C^j = (T^j, Att^j_1, ..., Att^j_{m'(j)}, P^j_1, ..., P^j_{n'(j)})$ deux concepts. La similarité d'attributs entre C^i et C^j est définie de la manière suivante :

$$Sim_A(C^i, C^j) = \sum_{x:=1}^{m(i)} \max_{y:=1..m'(j)}(Sim_T(Att^i_x, Att^j_y))$$

Définition 6.4 *Similarité terminologique entre deux propriétés*
Soient C^i et C^j deux concepts. La similarité entre deux propriétés P^i_x et P^j_y de C^i et C^j respectivement, est une similarité terminologique et est notée $Sim_P(P^i_x, P^j_y)$.

Définition 6.5 *Similarité de propriétés entre deux concepts*
Soient $C^i = (T^i, Att^i_1, ..., Att^i_{m(i)}, P^i_1, ..., P^i_{n(i)})$ et $C^j = (T^j, Att^j_1, ..., Att^j_{m'(j)}, P^j_1, ..., P^j_{n'(j)})$ deux concepts. La similarité de propriétés entre C^i et C^j est définie de la manière suivante :

$$Sim_P(C^i, C^j) = \sum_{x:=1}^{n(i)} \max_{y:=1..n'(j)}(Sim_T(P^i_x, P^j_y))$$

Définition 6.6 *Similarité locale entre deux concepts*
Soient C^i et C^j deux concepts. La similarité locale notée Sim_L entre C^i et C^j est définie de la manière suivante :

2. http ://www.limsi.fr/Individu/rosset/similarite2/node6.html
3. http ://wordnet.princeton.edu/.

$$Sim_L(C^i, C^j) = Sim_T(C^i, C^j) + Sim_A(C^i, C^j) \tag{6.4.1}$$

Définition 6.7 *Similarité globale entre deux concepts*
La similarité globale Sim_G entre deux concepts C^i et C^j est égale à leur similarité locale Sim_L, plus la similarité de leurs propriétés Sim_P, plus la similarité de leurs voisins Sim_V.

$$Sim_G(C^i, C^j) = Sim_L(C^i, C^j) + Sim_P(C^i, C^j) + Sim_V(C^i, C^j) \tag{6.4.2}$$

Définition 6.8 *Similarité des voisins*
Soient C^i et C^j deux concepts. Pour calculer la similarité des voisins de C^i et de C^j notée $Sim_V(C^i, C^j)$, il faut d'abord fixer le rayon de voisinage r (longueur du chemin) à considérer. Une fois r fixé, nous comparons tous les concepts intermédiaires deux à deux qui occupent les mêmes positions dans deux chemins de même longueur. La similarité entre les paires de concepts voisins est calculée uniquement sur la base de leur similarité locale Sim_L ainsi que celle de leurs propriétés Sim_P.

Remarque. Dans nos expérimentations (cf. Section 6.4.5), nous avons trouvé que les meilleurs résultats d'alignement correspondent à un chemin de longueur égale à 2.

6.4.4 Algorithme de fusion des ontologies

Pour automatiser le processus de fusion, nous avons proposé une nouvelle approche de fusion des ontologies locales en utilisant la technique de classification hiérarchique ascendante et le mécanisme d'inférence du langage OWL. Cela permet de trouver les classes d'entités ontologiques (concepts, attributs, propriétés) ainsi que leurs relations d'équivalence ou de subsomption. Pour cela, nous avons proposé une mesure de similarité qui prend en considération la terminologie, la structure ainsi que le voisinage de l'entité ontologique. Nous présentons à présent notre méthode de fusion des ontologies en utilisant l'algorithme de classification ascendante hiérarchique (Figure 6.3).

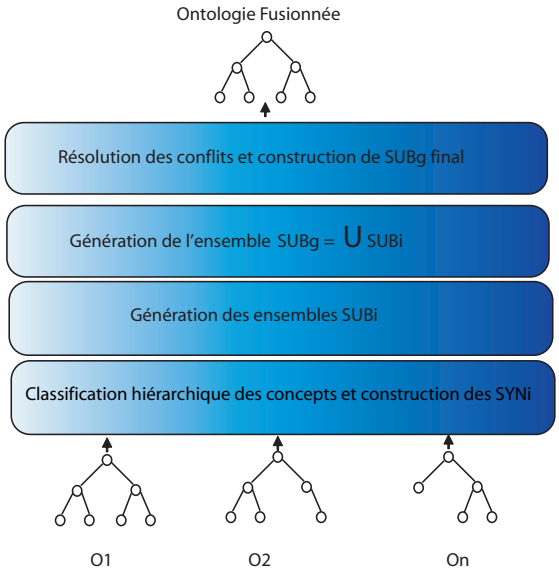

FIGURE 6.3 – Méthode de fusion d'ontologies

6.4.4.1 Classification hiérarchique de concepts

Dans cette section, nous allons détailler le processus de classification hiérarchique des concepts appartenant aux différentes ontologies à fusionner. La tâche de classification a comme objectif de construire un ensemble de classes de concepts dont chacune ne contiendra que des entités équivalentes appartenant aux différentes ontologies. Pour cela, nous appliquons l'algorithme de CAH basé sur notre mesure de similarité.

L'algorithme de classification utilise une matrice de similarité dont la première ligne et la première colonne de la matrice représentent tous les concepts des différentes ontologies locales. Les autres cases de la matrice représentent les similarités entre les différentes paires de concepts. Ensuite, en se basant sur la matrice de similarité, l'algorithme sélectionne d'abord la paire de concepts dont la similarité est maximale et construit une première classe qui va contenir ces deux concepts sémantiquement équivalents. Dans l'itération suivante, l'algorithme va considérer la classe construite dans l'itération précédente comme étant un seul individu et va calculer de nouveau sa similarité avec les autres concepts.

Pour mesurer la proximité ou l'écart entre une classe de synonymes SYN_α contenant les concepts $\{C^1, \ldots, C^t\}$ et un concept C^j, il existe plusieurs façons de procéder. On calcule par exemple la quantité : $sim(SYN_\alpha, C^j) = Min(d(C^1, C^j), ..., d(C^t, C^j))$ où d représente la distance entre deux concepts.

L'algorithme continue à tourner jusqu'à ce qu'il obtienne une classe qui contient tous les concepts. Ensuite, nous faisons la meilleure coupe ou bien nous fixons un seuil de similarité

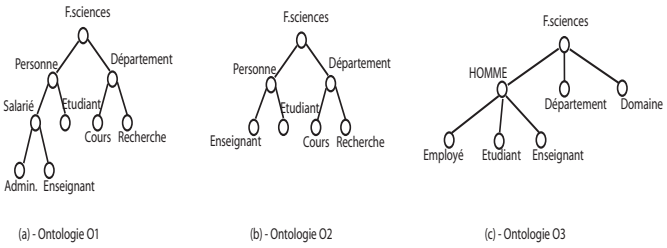

FIGURE 6.4 – Exemple de trois ontologies du même domaine

entre classes pour que l'algorithme s'arrête. Le résultat de cette étape est un ensemble de classes SYN_α dont chacune contient tous les concepts équivalents appartenant aux différentes ontologies locales. Nous détaillons l'algorithme de fusion des ontologies locales en se basant sur l'exemple suivant.

Considérons les trois ontologies présentées dans la Figure 6.4. Ces ontologies représentent un même domaine sauf qu'elles sont définies de manières différentes. Pour fusionner ces ontologies nous devons d'abord extraire l'ensemble de tous les concepts C.

$C =$ $\{F.sciences, Personne, Département, Salarié, Etudiant, Cours, Recherche, Admin, Enseignant, F.sciences, Personne, Département, Etudiant, Cours, Recherche, Enseignant, F.sciences, HOMME, Département, Employé, Etudiant, Cours, Recherche, Domaine, Enseignant\}$

Après application de l'algorithme de classification basée sur notre mesure de similarité sur C, nous obtenons sept classes $SYN_\alpha, \alpha := 1, \ldots, 7$ de concepts équivalents.

L'ensemble $SYN = \bigcup SYN_\alpha$ est présenté ci-après.

$SYN =$ $\{\{F.Sciences, F.sciences, F.sciences\},$
$\{Personne, Personne, HOMME\},$
$\{Département, Département, Département\},$
$\{Etudiant, Etudiant, Etudiant\},$
$\{Enseignant, Enseignant, Enseignant\},$
$\{Employé, Salarié\}, \{Cours, Cours\},$
$\{Recherche, Recherche\}, \{Admin\}, \{Domaine\}\}$

Ensuite à chaque classe SYN_α de concepts synonymes obtenue, un nom de concept représentatif lui est attribué. À chaque fois que nous affectons un nom de concept à une classe, nous vérifions qu'il n'est pas déjà attribué précédemment à une autre classe. Ainsi, en plus de la résolution du problème des conflits sémantiques en terme de synonymie, nous résolvons le problème des conflits sémantiques en terme d'antonymie et nous définissons des tables de correspondance qui permettent de garder les liens entre le nouveau nom et les autres concepts de la classe. Cette phase est nécessaire pour des tâches antérieures comme la réécriture de requêtes par exemple. De plus, les attributs d'un nouveau concept C^g, qui généralise un ensemble SYN_α, est l'union des attributs de tous les concepts C^i appartenant à SYN_α. Dans notre exemple, l'ensemble SYN est égal à :

$SYN = \{F.Sciences, Personne, Département, Etudiant, Enseignant, Salarié, Cours, Recherche, Admin, Domaine\}$

6.4.4.2 Construction de l'ensemble SUB_g

Après la construction des classes de synonymes SYN_α, nous entamons la troisième étape de l'algorithme de fusion des ontologies qui consiste à générer, à partir des hiérarchies des ontologies à fusionner, l'ensemble des paires de concepts (père, fils) de chaque ontologie locale $O_l, (l = 1..k)$ (k étant le nombre d'ontologies). La construction de cet ensemble et l'utilisation des ensembles des synonymes construits précédemment vont nous donner la structure de l'ontologie globale. Pour construire l'ensemble SUB, nous procédons en deux temps.

1. Génération des ensembles SUB_l. La première phase consiste à définir pour chaque ontologie O_l, l'ensemble $SUB_l = \{(père, fils)\}$. La détermination des ensembles SUB_l se fait par un simple parcours des hiérarchies des différentes ontologies O_l. Les ensembles SUB_l contiennent des conflits sémantiques qui seront résolus en utilisant les ensembles SYN_α.

Dans notre exemple, les trois ensembles SUB_1, SUB_2 et SUB_3 correspondant aux trois ontologies sont définis de la manière suivante.

$$\begin{aligned}
SUB_1 = \quad & \{(F.sciences, Personne),\ (F.sciences, Departement),\\
& (Personne, Salarie),\ (Personne, Etudiant),\\
& (Departement,\ Cours), (Departement, Recherche),\\
& (Salarie, Admin.),\ (Salarie, Enseignant)\}\\
SUB_2 = \quad & \{(F.sciences, Personne),\ (F.sciences, Departement),\\
& (Personne, Enseignant), (Personne, Etudiant),\\
& (Departement, Cours), (Departement, Recherche)\}\\
SUB_3 = \quad & \{(F.sciences, Homme), (F.sciences, Departement),\\
& (F.sciences, Domaine), (Homme, Employe),\\
& (Homme, Etudiant), (Homme, Enseignant)\}
\end{aligned}$$

2. Fusion des ensembles SUB_l.

$$SUB_g = \bigcup_{l=1,\ldots,k} SUB_l$$

Dans notre exemple, $SUB_g = SUB_1 \cup SUB_2 \cup SUB_3$ et est présenté ci-après.

$SUB_g =$ {(*F.sciences*, *Personne*), (*F.sciences*, *Departement*),
(*Personne*, *Salarie*), (*Personne*, *Etudiant*),
(*Departement*, *Cours*), (*Departement*, *Recherche*),
(*Salarie*, *Admin.*), (*Salarie*, *Enseignant*),
(*F.sciences*, *Personne*), (*F.sciences*, *Departement*),
(*Personne*, *Enseignant*), (*Personne*, *Etudiant*),
(*Departement*, *Cours*), (*Departement*, *Recherche*),
(*F.sciences*, *HOMME*), (*F.sciences*, *Departement*),
(*F.sciences*, *Domaine*), (*Homme*, *Employe*),
(*HOMME*, *Etudiant*), (*HOMME*, *Enseignant*)}.

6.4.4.3 Utilisation des classes SYN_α pour générer l'ensemble SUB_g

L'ensemble SUB_g contient des paires de concepts redondantes à éliminer. Pour pouvoir supprimer cette redondance, nous utilisons les connaissances extraites à partir des concepts des différentes ontologies. Nous procédons de la manière suivante.

Remplacer les concepts dans SUB_g par leur représentant. Dans l'ensemble SUB_g, nous parcourons les paires de concepts, paire par paire, et pour chaque concept de la paire en cours, nous cherchons la classe à laquelle ce concept appartient. Une fois la classe correspondante trouvée, nous remplaçons le concept de la paire par le nom de la classe. On réitère ce processus jusqu'à ce nous ayons parcouru toutes les paires de l'ensemble SUB_g. L'ensemble SUB_g de notre exemple devient alors :

$SUB_g =$ {(*F.sciences*, *Personne*), (*F.sciences*, *Departement*),
(*Personne*, *Salarie*), (*Personne*, *Etudiant*),
(*Departement*, *Cours*), (*Departement*, *Recherche*),
(*Salarie*, *Admin.*), (*Salarie*, *Enseignant*),
(*F.sciences*, *Personne*), (*F.sciences*, *Departement*),
(*Personne*, *Enseignant*), (*Personne*, *Etudiant*),
(*Departement*, *Cours*), (*Departement*, *Recherche*),
(*F.sciences*, *Personne*), (*F.sciences*, *Departement*),
(*F.sciences*, *Domaine*), (*Personne*, *Salarie*),
(*Personne*, *Etudiant*), (*Personne*, *Enseignant*)}

Supprimer les redondances. Cette phase consiste à parcourir l'ensemble SUB_g et à comparer les paires de concepts deux à deux. Les paires similaires redondantes vont être supprimées afin de n'en garder qu'une seule dans l'ensemble SUB_g.

TABLE 6.1 – Description des ontologies locales

Ontologies	Caractéristiques
1	L'ontologie de base
2	La hiérarchie des concepts a été réduite
3	Entités remplacées par leurs synonymes
4	La hiérarchie des concepts a été réduite

TABLE 6.2 – Statistiques sur les données ontologiques

	NbClass.	NbProp.	NbInst.	NbAxiom.
Min	24	17	15	12
Max	39	28	110	14
Average	31	22	43	13

$$\begin{aligned} SUB_g = \quad & \{(F.sciences, Personne), (F.sciences, Departement), \\ & (Personne, Salarie),\ (Personne, Etudiant), \\ & (Departement, Cours), (Departement, Recherche), \\ & (Salarie, Admin.),\ (Salarie, Enseignant), \\ & (Personne, Enseignant),\ (F.sciences, Domaine)\} \end{aligned}$$

6.4.4.4 Construction de l'ontologie globale

L'ensemble SUB_g contient la structure finale de l'ontologie globale. La racine de l'ontologie globale est définie par le concept qui n'a pas de père (qui n'est *fils* dans aucune paire de concepts).

6.4.5 Implémentation

Pour évaluer notre approche de fusion des ontologies, nous avons utilisé des variations différentes de l'ontologie géographique construite au sein du projet "Fouille de Données Multi-STratégie" (FoDoMust) [4] pour extraire et qualifier la végétation urbaine à partir de bases de données images. Ces variations d'ontologies sont présentées dans le tableau suivant (Table 6.1).

L'ontologie de base des objets géographiques est composée de 39 concepts, 110 instances, 28 propriétés et 14 axiomes. Le tableau suivant résume les statistiques sur les données ontologiques (Table 6.2).

4. http ://lsiit-old.u-strasbg.fr/afd/sites/fodomust/fr-accueil.php

Nos expérimentations sont effectuées en utilisant la plateforme *Eclipse* avec le raisonneur libre du langage OWL-DL Pellet [5] et la bibliothèque Jena [6]. Afin de calculer les similarités entre les concepts, nous avons utilisé notre mesure de similarité sémantique qui prend en considération la similarité des termes, des attributs, des propriétés ainsi que celle du voisinage. Le calcul de similarité de voisinage est récursif. Pour limiter le nombre de voisins pouvant être impliqués dans le calcul de similarité pour un concept donné, nous avons effectué des tests afin de mesurer la valeur optimale du rayon de voisinage (la longueur du chemin entre le concept et ses voisins). Dans notre cas, nous avons trouvé que la valeur optimale qui définit la meilleure similarité est de 2, c'est à dire le deuxième voisin du concept.

6.4.6 Métriques d'évaluation

Les mesures de *Précision*, *Rappel* et *Fallout* sont des métriques largement exploitées pour évaluer la qualité d'alignement d'ontologies. L'objectif principal de ces mesures est l'automatisation du processus de comparaison des méthodes d'alignement ainsi que l'évaluation de la qualité des alignements produits. La première phase dans le processus d'évaluation de la qualité de l'alignement consiste à résoudre le problème manuellement. Le résultat obtenu manuellement est considéré comme l'alignement de référence. La comparaison du résultat de l'alignement de référence avec celui de l'appariement obtenu par la méthode d'alignement produit trois ensembles : $AFound$, $AExpected$ et $ACorrect$. Le premier ensemble $AFound$ représente les paires d'entités alignées par l'approche d'alignement utilisée. Le deuxième ensemble $AExpected$ représente les paires d'entités alignées dans l'alignement de référence. Et enfin, le troisième ensemble $ACorrect$ représente l'intersection des deux ensembles $AFound$ et $AExpected$. En se basant sur ces trois ensembles de paires d'entités, les trois mesures de qualité *précision*, *rappel* et *fallout* peuvent alors être calculées. La *précision* est définie comme le rapport entre $ACorrect$ et $AFound$, le *rappel* est le rapport entre $ACorrect$ et $AExpected$ et le *fallout* est le rapport entre la différence entre $AFound$ et $ACorrect$, et $AFound$.

Pour procéder à l'évaluation expérimentale de notre approche, nous avons suivi les deux étapes suivantes. D'abord, nous avons aligné, avec l'aide de l'expert du domaine, les différentes ontologies manuellement. Les correspondances trouvées dans cette étape sont considérées comme un alignement de référence. La comparaison de l'alignement de référence avec l'alignement proposé par notre approche produit les trois ensembles : $AFound$, $AExpected$ et $ACorrect$. Nous avons utilisé les ontologies candidates pour mesurer la qualité de notre alignement nommé *OMerSec* [MFBB10]. Ensuite, nous avons comparé les résultats obtenus avec deux autres approches d'alignement qui sont COMA++ [MER06]

5. http ://www.mindswap.org/2003/pellet/
6. http ://jena.sourceforge.net/

et FCA-Merge [SM01]. Les résultats montrent que notre approche de fusion présente une meilleure précision (Figure 6.5).

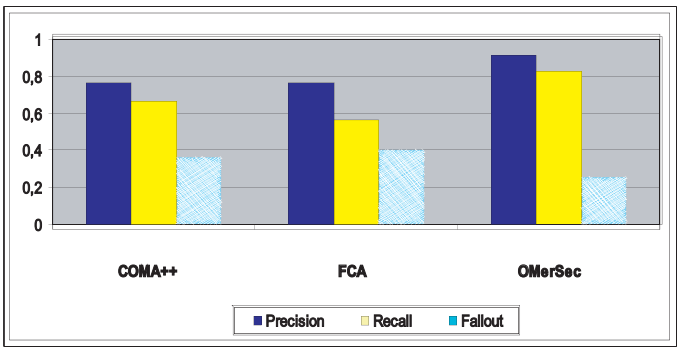

FIGURE 6.5 – Comparaison de la moyenne de la *Précision*, du *Rappel* et du *Fallout* pour OMerSeC, FCA et COMA++

6.5 Conclusion

Nous avons présenté dans ce chapitre une vision d'analyse en ligne à la demande qui remet l'utilisateur au cœur du système décisionnel. Notre choix s'est porté sur une architecture décisionnelle fondée sur un système de médiation en utilisant les ontologies pour décrire à la fois les sources de données et le médiateur. Cette architecture permet de prendre en compte l'évolution des sources de données et des besoins d'analyse. Pour cela, nous avons proposé un dispositif de médiation permettant, à partir d'une requête utilisateur, de déployer le processus d'intégration allant de la sélection des données dans les sources de données originelles jusqu'à la construction des cubes de données à la demande. Notre approche de conception de cubes de données à la demande permet d'accéder en temps réel aux données afin de créer des cubes personnalisés en fonction des besoins actuels des utilisateurs. Notre approche, même si elle ne permet pas de construire un entrepôt de données historisées, elle peut être très adaptée dans des applications où les données sources sont stockées selon un axe temporel. Les données de simulation en sont un bon exemple.

Dans ce travail, nous avons concentré nos efforts sur l'intégration sémantique des données. Notre contribution dans ce domaine porte principalement sur la proposition d'un algorithme de fusion des ontologies. L'originalité de notre approche de fusion des ontologies provient de l'utilisation de la méthode de CAH et du mécanisme d'inférence OWL pour extraire les classes de concepts les plus similaires à partir de plusieurs ontologies, puis de trouver leur subsumant afin de construire l'ontologie globale alors que la majorité des approches de fusion s'appliquent sur deux ontologies seulement. Nous avons également défini une mesure de similarité sémantique de manière à considérer le voisinage d'un concept lors de la comparaison entre les différents concepts des différentes ontologies.

Nous avons par ailleurs démontré l'efficacité de notre approche de fusion en mettant en œuvre un processus d'expérimentation qui a impliqué son implémentation et son application sur un cas réel. De plus, nous avons mesuré la précision de notre approche de fusion en la comparant à d'autres approches existantes. En effet, la combinaison de la CAH et du mécanisme d'inférence OWL ont permis d'augmenter la précision de notre approche de fusion. Notre approche de fusion des ontologies par la CAH est prometteuse puisque d'autres travaux récents combinent l'inférence logique et la classification bayésienne pour l'alignement des ontologies [TPRT10]. Cette méthode permet d'estimer la probabilité des mappings entre classes à partir des métadonnées des instances déclarées ou inférées dans ces classes.

En ce qui concerne nos autres contributions dans le domaine de l'analyse en ligne à la demande, le lecteur intéressé peut se referer à la thèse de N. Maïz [Mai10]. Nous avons en effet obtenu quelques résultats prometteurs. Une proposition a été faite dans ce sens en s'appuyant sur une ontologie métier dans laquelle sont définis les concepts multidimensionnels (cube, fait, dimension, mesure, niveau, attribut) et une ontologie de correspondances qui lie l'ontologie métier à l'ontologie globale créée à partir des ontologies locales décrivant les sources de données (ontologie fusionnée) pour assurer la création des cubes de données OLAP. En effet, pour pouvoir transformer les entités de l'ontologie globale en éléments multidimentionnels représentés par les entités de l'ontologie métier, nous devons expliciter les correspondances entre les entités des deux précédentes ontologies. L'ontologie de correspondances représente alors l'ensemble des liens existants entre l'ontologie métier et l'ontologie globale. Ces correspondances vont déterminer le rôle que peut jouer chaque entité de l'ontologie globale dans l'ontologie métier. Par exemple, définir un concept donné comme une dimension ou un attribut comme une mesure. Une fois les correspondances entre les deux ontologies établies, elles seront stockées sous forme d'axiomes OWL.

Enfin, l'utilisateur peut définir des requêtes décisionnelles pour construire des cubes à la demande représentant ses contextes d'analyse. Dans ce cas, un algorithme de réécriture de requêtes peut être utilisé permettant à l'utilisateur d'exprimer sa requête dans les termes du vocabulaire de l'ontologie globale (schéma du médiateur) et des ontologies locales (schémas locaux). Cette caractéristique permet d'assurer la possibilité au médiateur de traiter correctement une requête dans le modèle GLaV.

Plusieurs perspectives de recherche restent encore à explorer dans le domaine de l'analyse en ligne à la demande fondé sur un système de médiation à base d'ontologies. Créer par exemple un entrepôt de cubes de données où chaque cube représente un contexte d'analyse correspondant à un utilisateur donné à un instant donné. Cela peut s'apparenter au concept de versionnement (versions de cubes). Cette solution peut paraître lourde à mettre en place mais permet de répondre à des questions sur des données anciennes. Dans ce contexte, plusieurs problèmes restent ouverts. Comment gérer les versions des cubes,

comment composer un nouveau cube à partir de cubes existants, etc. ? Ou alors, ne rien stocker et rester dans une approche totalement virtuelle. Dans ce cas, l'utilisateur n'aura à sa disposition que les données disponibles dans les sources. Selon la nature des données et l'application décisionnelle à développer, l'une ou l'autre des approches peut alors être utilisée.

6.6 Publications

Nous présentons dans cette section nos publications concernant les travaux que nous avons menés sur l'intégration de données par médiation en utilisant les ontologies.

Conférences internationales

[1] N. Maiz, M. Fahad, O. Boussaid, **F. Bentayeb**, "Automatic Ontology Merging by Hierarchical Clustering", 10th International Conference on Knowledge Management and Knowledge Technologies (I-KNOW 2010), Graz, Austria., Special Issue of Journal of Universal Computer Science (J.UCS), 1-3 September 2010, 81-93.

[2] N. Maiz, **F. Bentayeb**, O. Boussaid, "Hybrid Architecture of OWL-Ontology for Relational Data Sources Integration", 18th Information Ressource Management Association International Conference (IRMA 07), Vancouver, Canada, May 2007, 857-860.

[3] N. Maiz, O. Boussaid, **F. Bentayeb**, "Ontology-Based Mediation System", 13th ISPE International Conference on Concurrent Engineering : Research and Applications (CE 06), Antibes, France, September 2006 ; Frontiers in Artificial Intelligence and Applications, Vol. 143, IOS Press, 181-189.

Conférences nationales

[4] N. Maiz, O. Boussaïd, **F. Bentayeb**, Fusion d'ontologies par classification hiérarchique pour la conception d'un entrepôt de données, 2ème Journées Francophones sur les Ontologies (JFO 08), Décembre 2008, Lyon, France.

[5] N. Maiz, O. Boussaïd, F. Bentayeb, Ontology Merging by Clustering For Data Warehouse Building on-the-fly, joint meeting of the Société Francophone de Classification and the Classification and Data Analysis Group of the Italian Society of Statistics (SFC-CLADAG), Caserta, Italy,(2008).

Workshops nationaux

[6] N. Maiz, O. Boussaid, **F. Bentayeb**, "Un système de médiation basé sur les ontologies", 3ème atelier Fouille de Données Complexes dans un processus d'extraction des connaissances, EGC 06, Lille, Janvier 2006, 27-38.

[7] N. Maiz, **F. Bentayeb**, O. Boussaid, "Un système de médiation basé sur les ontologies pour l'entreposage des données", Atelier Systèmes Décisionnels (ASD 06), 9th Maghrebian Conference on Information Technologies (MCSEAI 06), Agadir, Maroc, Décembre 2006.

Chapitre 7

Entrepôts d'objets complexes

Les architectures classiques d'entrepôts de données ont montré leur utilité et leur efficacité lorsque les données sont «simples» (numériques ou symboliques). En revanche, elles sont inadaptées dans le cas des données complexes (formats différents, sources diverses, sémantique différente, etc.). En effet, la spécificité des données complexes implique d'envisager de nouvelles solutions décisionnelles. Plusieurs questions se posent alors : quel modèle multidimensionnel pour les données complexes ? quel processus ETL adopter ? quelle est la place de l'utilisateur dans un tel système décisionnel ? L'objectif de ce chapitre est de tenter d'apporter des réponses à ces questions.

Dans un premier temps, nous motivons la nécessité d'entreposer et d'analyser les données complexes. Puis, nous présentons brièvement les travaux décrits dans la littérature consacrés à la modélisation multidimensionnelle de ces données.

Dans un deuxième temps, nous présentons deux approches de modélisation multidimensionnelle de données complexes. Dans la première approche, nous considérons la donnée complexe comme un ensemble de descripteurs de bas niveau et de descripteurs sémantiques. Dans ce cas, la modélisation multidimensionnelle des données complexes revient à la modélisation de leurs descripteurs. Dans la deuxième approche, nous considérons la donnée complexe comme un agrégat de données hétérogènes qui, une fois réunies, forment une entité sémantique que nous qualifions d'objet complexe. Dans ce cas, la modélisation multidimensionnelle de données complexes revient à la modélisation d'objets complexes.

La première partie de ce travail a fait l'objet des stages de master de A. Tanasescu, A. Duffoux et F. Clerc que nous avons co-encadrés. La deuxième partie concerne la thèse de doctorat de D. Boukraâ avec qui nous collaborons depuis deux ans, notamment durant son long séjour scientifique passé au laboratoire ERIC. Il prépare sa thèse au sein de l'Université de Jijel en Algérie ; sa soutenance est prévue fin 2011.

7.1 Motivation

Les données exploitées dans le cadre des processus décisionnels sont de plus en plus diverses, hétérogènes et porteuses de sémantique différente. En effet, le besoin d'exploiter des données qui ne sont ni numériques ni symboliques se fait jour dans de nombreux domaines : médecine, marketing, écologie, ... L'avènement du Web et la profusion de données non structurées (multimédias, texte, blogs, réseaux sociaux, etc.) ont en grande partie contribué à l'emergence de ces données, que nous qualifions de complexes. Si bien qu'un domaine de recherche centré sur l'entreposage et la fouille de données complexes a émergé depuis quelques années.

On estime dans la littérature que seulement 20% des données existantes dans des bases de données ou sur le web sont numériques et exploitables par les systèmes OLAP [TC06]. Cependant les 80% de données restantes, considérées souvent comme des données complexes, demeurent hors de portée de ces systèmes faute d'outils ou de méthodes appropriées. L'un des points clés de l'entreposage des données réside dans la conception du modèle de l'entrepôt ; en effet les possibilités d'analyse sont conditionnées par ce dernier. Cette démarche, si elle est valable dans le contexte des entrepôts de données classiques, elle l'est encore plus dans le contexte des entrepôts de données complexes. En effet, le concept du modèle en étoile dans lequel les faits et les dimensions sont fixés à priori, avec de simples liens d'agrégation entre les données, n'est plus nécessairement pertinent dans le cas des données complexes. Il apparaît donc évident que l'élaboration de nouveaux modèles multidimensionnels de données complexes devient une nécessité dans le domaine des SID. D'autre part, si la place de l'utilisateur est importante dans les entrepôts de données classiques, elle le demeure plus que jamais dans les entrepôts de données complexes. En effet, en plus des objectifs d'analyse globaux, les besoins personnalisés des utilisateurs doivent également être pris en compte.

En conclusion, dans le cadre des données complexes, il faut revisiter les principes de la modélisation multidimensionnelle classique. Dans ce contexte, nous avons suivi deux approches de modélisation différentes.

La première approche consiste à garder le modèle en étoile et repenser le processus d'ETL (Section 7.3). Il s'agit alors de définir une démarche globale d'entreposage de données complexes pour définir le schéma multidimensionnel en étoile en y intégrant la sémantique des données. Il s'agit de prendre en compte, en plus des descripteurs de bas niveau, les descripteurs sémantiques qui peuvent aider à la construction de contextes d'analyse pertinents. L'extraction des descripteurs sémantiques à partir des données complexes se fait grâce aux techniques de fouille de données et est motivée par les besoins des utilisateurs. Dans cette approche, la modélisation multidimensionnelle est dirigée par les descripteurs des données.

La deuxième approche consiste quant à elle, à définir un nouveau modèle multidimensionnel adapté aux exigences des données complexes (Section 7.4). Cependant, il apparaît difficile de définir un modèle multidimensionnel unifié qui peut prendre en compte tout type de complexité des données. Il s'agit de considérer une donnée complexe comme un objet de l'univers, lui même composé d'autres objets. Un objet est un agrégat de données, le tout formant une entité sémantique, appelée objet complexe. Nous considérons également les liens existants entre les composants d'un objet complex ou entre les objets complexes eux-mêmes. Par ailleurs, pour rendre le modèle multidimensionnel centré utilisateur, il faut s'assurer du traitement symétrique entre les faits et les dimensions. Si bien qu'un seul concept unique doit représenter une donnée complexe. L'affectation de rôle de fait ou de dimension à un objet complexe est effectuée par l'utilisateur au moment de la création du cube d'objets complexes. Dans cette approche, la modélisation multidimensionnelle est dirigée par les objets.

7.2 État de l'art

Ces dernières années, le domaine des entrepôts de données et de l'OLAP a été marqué par la croissance des travaux traitant des données complexes. Dans ce contexte, plusieurs travaux sur la modélisation multidimensionnelle de données complexes sont apparus soit par extension des modèles existants, soit en proposant de nouveaux concepts. Ces modèles multidimensionnels traitent des trois niveaux de modélisation : conceptuel, logique et physique. Ces travaux couvrent différents aspects de l'entreposage et de l'analyse en ligne de données complexes. Nous trouvons les travaux sur l'intégration des données provenant du Web [SSB03, Xyl01, BBDR03, BDBR08], l'entreposage de données non structurées [IT07, KKL05] et des données semi-structurées, représentées notamment en XML [GRV01, VBR03, PHS05], l'entreposage des données médicales [WHK+01] et des données spatio-temporelles [BTM06, GKMV09]. D'autres aspects de la complexité des données ont été abordés dans la littérature comme la temporalité [Tes00, PJ99, KT07] et l'incertitude [PJ99].

À la lumière des travaux portant sur la modélisation multidimensionnelle de données complexes, nous pouvons dire qu'il y a autant de modèles multidimensionnels que de types de données complexes. Il apparaît donc difficile d'avoir un modèle multidimensionnel unifié pour tout type de données complexes.

Par ailleurs, l'utilisation du paradigme objet pour la modélisation multidimensionnelle des données a également motivé certains travaux [ASS00]. Néanmoins, les modèles orientés-objet proposés diffèrent dans la manière de définir les concepts multidimensionnels même si la majorité de ces travaux utilisent les diagrammes UML pour définir les faits et les dimensions [JMP01, LM02, LA05, Tru99]. Les hiérarchies sont modélisées en utilisant le

concept d'agrégation [JMP01] ou d'association [LM02] d'UML. D'autres travaux utilisent les packages pour améliorer la lisibilité du modèle. Par exemple, les packages sont utilisés pour définir le modèle en constellation au premier niveau, les faits et les dimensions au deuxième niveau ; et enfin au troisième niveau, les diagrammes de classes sont utilisés pour modéliser les différents éléments contenus dans les pacakges du deuxième niveau [LMTS02]. Un travail similaire a été proposé dans lequel le modèle en étoile est représenté par un package de faits et les dimensions par des packages de dimensions [NRDR04].

Ce survol des travaux permet de constater la diversité des modèles multidimensionnels proposés dans la littérature. Dans le contexte des données complexes, la modélisation objet paraît la plus appropriée bien qu'il n'existe pas de consensus sur la représentation des concepts multidimensionnels. On peut constater aussi que le langage XML a beaucoup été utilisé pour décrire les données complexes au niveau logique. En conclusion, il en ressort la nécessité de définir de nouveaux modèles multidimensionnels permettant de prendre en compte les données complexes tant au niveau de leur contenu, leur structure que de leur sémantique.

7.3 Entreposage et analyse en ligne de données complexes

Lorsque nous observons les processus d'entreposage des données et celui de l'extraction des connaissances à partir des données (ECD), nous constatons qu'ils présentent des similitudes. Le processus ETL dans les entrepôts correspond au processus de préparation des données dans l'ECD. La phase de modélisation mulitidimensionnelle pour les entrepôts de données correspond à la phase de structuration en tableau "individus/valeurs" pour l'ECD. Enfin, la phase d'analyse OLAP dans les entrepôts correspond à la phase d'extraction des connaissances. C'est de ce constat que nous avons été amenée à utiliser la fouille de données non seulement comme un outil final d'analyse (cf. Chapitres 4 et 5) mais également pour l'aide à la modélisation multidimensionnelle de données, notamment lorsque celles-ci sont complexes.

La problématique d'intégration, de modélisation, de structuration et d'extraction de connaissances à partir de données complexes devient cruciale pour différents domaines (médical, bio-informatique, linguistique, téléphonie. . .). De ce constat découle l'idée de définir une méthodologie et des outils génériques pour l'entreposage et l'extraction automatique de connaissances à partir de données complexes. Pour enclencher un processus d'analyse en ligne et/ou d'extraction de connaissances à partir des données complexes, il faut intégrer puis représenter les données complexes sous une forme adaptée aux techniques de l'OLAP et/ou de fouille de données.

Dans ce domaine, nous avons entrepris plusieurs travaux de recherche comme l'attestent nos nombreuses publications sur le sujet (Section 7.6). Nous présentons brièvement dans

cette section notre approche de modélisation, d'intégration et d'analyse en ligne de données complexes [BBDR03].

7.3.1 Utilisation de la fouille pour l'aide à la modélisation multidimensionnelle de données complexes

Dans le cadre des entrepôts de données, plusieurs travaux ont proposé des méthodes de conception pour l'élaboration des schémas multidimensionnels qui sont habitullement classées en 3 catégories : ascendante, descendante et mixte. La méthode ascendante est une démarche de conception dirigée par les sources de données [GMR98], la méthode descendante [TLMS03, PACW06] est dirigée par les besoins des décideurs, et enfin la méthode mixte combine les deux précédentes méthodes [RA10]. Lorsque les données sont complexes, le processus de conception de schéma multidimensionnel devient plus difficile quelle que soit l'approche suivie. L'une des difficultés engendrées par les données complexes est due à la diversité de leurs formats (image, son, texte, etc.). Ainsi, la phase de modélisation doit prendre en compte la spécificité des données complexes qui nécessite non seulement la description de la structure des données mais également de leur contenu. En effet, les données complexes sont décrites par de nombreux descripteurs (données simples) et véhiculent aussi beaucoup d'informations (métadonnées) et de sémantique (connaissances). L'intégration des métadonnées et des connaissances dans le modèle d'entrepôt est un enjeu majeur aussi bien dans la phase de modélisation que dans la phase d'analyse en ligne des données complexes.

Dans ce contexte, les premiers travaux auxquels nous nous sommes intéressée dans le cadre de l'entreposage de données complexes portent sur leur intégration et leur modélisation. Notre objectif était de trouver le meilleur moyen possible pour regrouper dans un même espace de stockage unifié les données complexes à des fins d'analyse.

Nous avons conçu alors un ODS (*Operating Data Storage*) qui est un système d'entreposage de données complexes qui utilise XML comme langage pivot (Figure 7.1). Notre système repose sur un méta-modèle UML décrivant les données complexes sous forme de classes d'objets complexes selon le paradigme objet [DBB02]. Un objet complexe est composé de différents éléments de base selon le type de données auquel ils appartiennent (texte, image, son, vidéo, vue relationnelle matérialisée). Après instanciation du méta-modèle, le modèle UML obtenu peut être traduit directement en schéma logique XML, qu'il soit exprimé à l'aide d'une DTD ou à l'aide du langage XML-Schema. Pour terminer, le modèle logique XML est traduit en modèle physique sous forme de documents XML. A partir du modèle logique et des données complexes, les documents XML valides sont générés. Ces derniers peuvent être finalement stockés soit dans une base de données native XML, soit dans une base de données relationnelle via un processus de mapping.

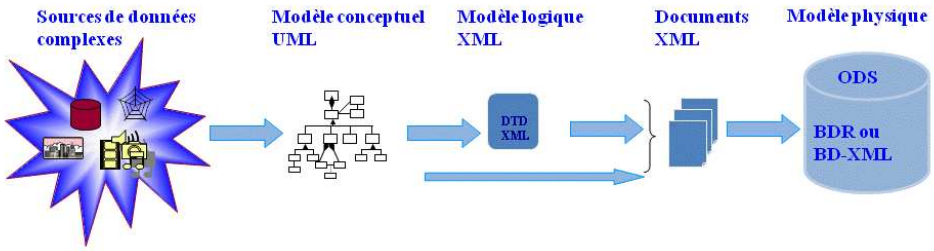

FIGURE 7.1 – Approche d'intégration de données complexes

Le modèle relationnel obtenu à l'issue de ce processus n'est en fait que l'image du mapping des documents XML dans des tables relationnelles. Il ne définit pas de liens sémantiques entre les différents documents. Seuls les descripteurs de bas niveau y sont stockés. C'est ainsi que nous avons étendu le méta-modèle initial en y incluant une méta-classe *specific* (Figure 7.2) qui permet d'enrichir la description des données complexes par des informations d'ordre sémantique [TBB05, BTBD07].

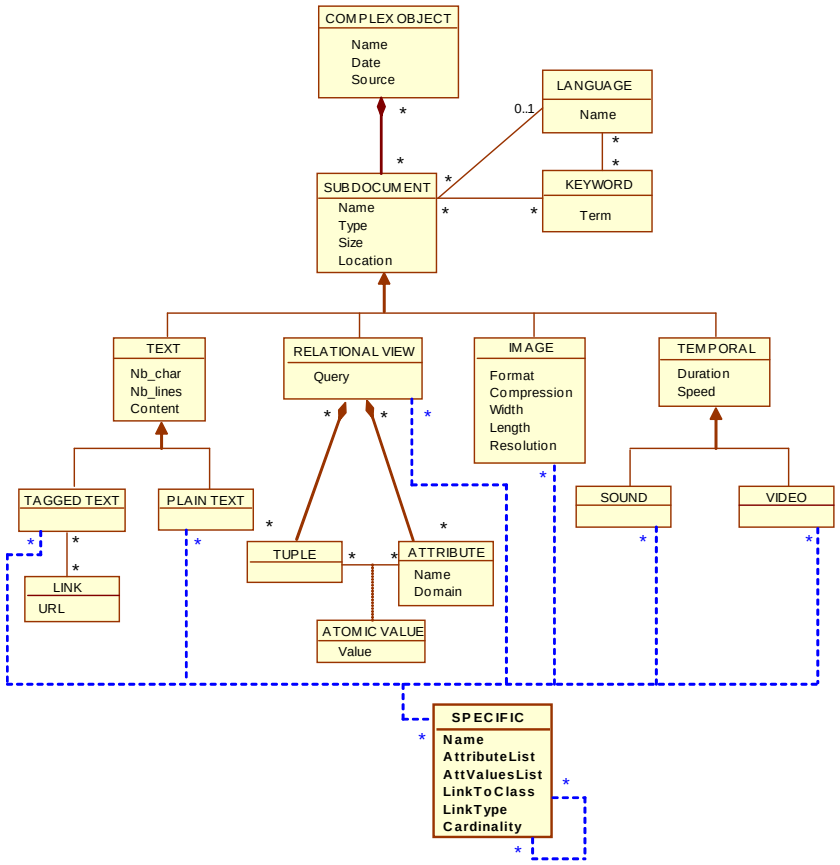

FIGURE 7.2 – Modèle UML générique pour la représentation de données complexes

D'autre part, pour mieux préparer les données complexes de l'ODS à l'analyse, il était nécessaire d'exploiter, en plus de leurs descripteurs de bas niveau, leurs descripteurs sémantiques. Ces derniers peuvent être obtenus par diverses techniques de fouille de données, de statistique, de traitement d'images ou du signal. En effet, depuis l'avènement des données complexes, la communauté scientifique de fouille de données s'intéresse à tous types de données. De là sont nées de nouvelles approches comme l'extraction de connaissances à partir de données texte (*text mining*), à partir d'images (*image mining*) ou à partir du web (*web usage mining*) ou de façon plus générale (*multimedia mining*). D'autres travaux portent sur la transformation d'un texte ou d'une image en un vecteur attributs-valeurs pour la construction de caractéristiques (*feature construction*) qui est considérée comme stratégique pour le développement de la fouille de données complexes en améliorant les résultats des méthodes d'extraction de connaissances.

C'est dans ce contexte que nous avons proposé une approche d'aide à la modélisation multidimensionnelle de données complexes en utilisant les techniques de fouille de données (Figure 7.3) [BTBD07].

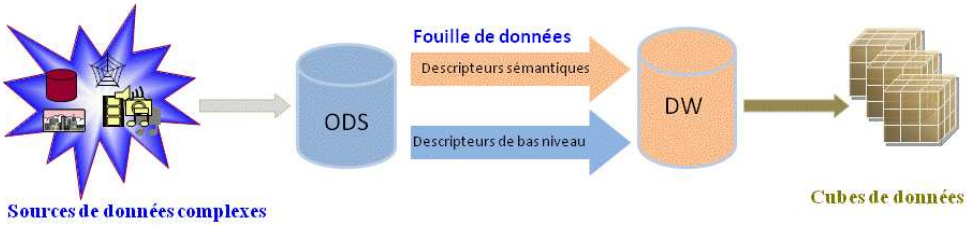

FIGURE 7.3 – La fouille pour l'aide à la modélisation multidimensionnelle de données complexes

La recherche d'informations pertinentes est une étape cruciale dans un processus décisionnel, notamment lorsque les données sont complexes. Une exploration par une technique de fouille des données peut, par exemple, contribuer à l'identification des faits intéressants à analyser ; ce qui permet de concevoir et de construire l'entrepôt de données correspondant. En effet, le choix des mesures, des dimensions et de leurs hiérarchies n'est pas une tâche évidente.

Sachant que les descripteurs de bas niveau n'offrent pas ou peu d'informations sur les données complexes, l'utilisateur est contraint de construire d'autres variables porteuses de sémantique en fonction de ses objectifs d'analyse. Dans ce cas, les techniques de fouille peuvent l'aider à trouver des caractéristiques pertinentes à analyser en mettant en évidence des correlations et des relations causales entre les variables. Les connaissances extraites peuvent être intégrées dans le modèle de l'entrepôt, soit sous forme de métadonnées, soit en étendant le schéma de l'entrepôt.

À titre d'exemple, nous avons considéré des données complexes représentant un corpus de 200 images de villes et de paysages et des textes. Ces images sont décrites par des descripteurs de bas niveau (résolution, taille, couleur, texture, etc.). Pour mieux décrire les images avec des informations d'ordre sémantique, il était nécessaire de connaître les objectifs de l'utilisateur. Dans notre cas, l'utilisateur s'intéresse à l'influence des caractéristiques de la *couleur* et de la *texture* des images sur le discernement entre les images représentant des villes et celles représentant des paysages. Le recours aux méthodes telles que les arbres de décision nous ont permis de découvrir des correlations intéressantes entre les caractéristiques des images et leur type *ville* ou *paysage*. Il en ressort par les expériences que nous avons menées, que la caractéristique la plus pertinente pour dissocier entre les villes et les paysages est la variable L2Norm_G qui mesure le "poids" de la couleur verte dans les images. En déclinant la variable L2Norm_G sur les trois canaux de couleurs rouge, vert et bleu (RGB), nous obtenons les variables suivantes : L2Norm_R, L2Norm_G et L2Norm_B que nous pourrons considérer comme les mesures des faits à analyser. Ces mesures ainsi que les dimensions choisies par l'utilisateur permettent de créer le schéma

de l'entrepôt des images. Le modèle ainsi obtenu peut être suffisant pour répondre aux objectifs d'analyse globaux et/ou peut être complété par l'utilisateur si ses besoins évoluent (cf. Chapitre 3).

Grâce à notre approche, nous avons montré qu'au-delà de leur capacité d'analyse, les techniques de fouille de données peuvent contribuer à la modélisation multidimensionnelle des données complexes en vue de la construction de cubes de données pertinents.

7.3.2 Système multiagent pour l'intégration de données complexes

Pour assurer une bonne intégration des données complexes dans l'ODS, nous avons proposé une approche d'ETL automatique basée sur les systèmes multiagents (SMA) [BBD03]. Notre approche s'inscrit dans l'environnement distribué qu'est le web, qui est un excellent fournisseur de données complexes. Une fois l'environnement identifié, il fallait définir les différents éléments nécessaires à la création d'un SMA capable de réaliser le processus d'intégration de données complexes dans une base de données relationnelle. On peut décomposer le processus d'intégration de données complexes en un ensemble de tâches effectuées par des programmes. Ces tâches peuvent être assimilées à des services offerts par des acteurs, définis dans un système destiné à accomplir un tel processus d'intégration, communiquant entre eux et évoluant dans un environnement distribué. Un système multiagent est constitué d'un ensemble de processus informatiques se déroulant en même temps, donc de plusieurs agents vivant au même moment, partageant des ressources communes et communiquant entre eux. Le point clé des systèmes multiagents réside dans la formalisation de la coordination entre les agents. Dans ce contexte et en respectant la modélisation du processus d'intégration des données complexes, nous avons défini les éléments suivants.

• *Les objets*. Plusieurs types d'objets sont à prévoir pour l'intégration de données complexes. Tout d'abord, il s'agit des données complexes elles-mêmes qu'il faut récupérer à partir du Web. Viennent ensuite les structures de données à créer, telles que le modèle UML ou la DTD.

• *Les agents*. Ce sont les différents acteurs qui interviennent dans le processus d'intégration des données. Ce sont des programmes intelligents capables de percevoir, produire, consommer, transformer et manipuler les objets définis ci-dessus.

• *Les communications*. Ce sont les échanges nécessaires effectués entre les différents agents pour mener à bien le processus d'intégration des données complexes.

Ensuite nous avons défini de manière précise les tâches suivantes nécessaires au bon déroulement du processus d'intégration des objets complexes.

• *La collecte des données*. Cette tâche est gérée par des agents dont le rôle consiste à récupérer les caractéristiques des données complexes pour pouvoir les transmettre ensuite aux agents responsables de la structuration des données.

• *La structuration des données.* Cette tâche est effectuée par les agents qui s'occupent de l'organisation des données complexes selon un modèle bien défini et transmettent ce dernier aux agents responsables du stockage.

• *Le stockage des données.* Cette tâche est gérée par des agents qui s'occupent de l'alimentation d'une base de données à partir du modèle fourni par les agents de structuration.

Après avoir défini tous les éléments nécessaires à la création d'un SMA, nous avons développé un système multiagent pour l'intégration de données complexes baptisé SMAIDoC [1] (Système MultiAgent Pour l'Intégration de Données Complexes) basé sur une plateforme d'agents génériques [BBDC03]. Le fonctionnement du système SMAIDoC s'articule autour de cinq agents qui se chargent de l'intégration de données complexes dans une base de données relationnelle (Figure 7.4). Lorsque l'utilisateur choisit un site dans lequel se trouvent les données complexes qui l'intéressent, l'agent *MenuAgent* ordonne aux agents *DataAgent* et *WrapperAgent* de migrer. L'agent *DataAgent* collecte les données ainsi que les métadonnées et les transmet séquentiellement à l'agent *WrapperAgent* qui instancie progressivement la structure UML. Ce dernier transmet la structure UML créée à l'agent *XMLCreator*. Ce dernier traduit la structure UML en une DTD et génère des documents XML valides. Pour terminer, l'agent *XMLCreator* transmet les documents XML à l'agent *XML2RDBAgent* qui se charge du stockage des documents XML dans la base de données relationnelle. Le processus décrit ci-dessus se répète autant de fois que nécessaire.

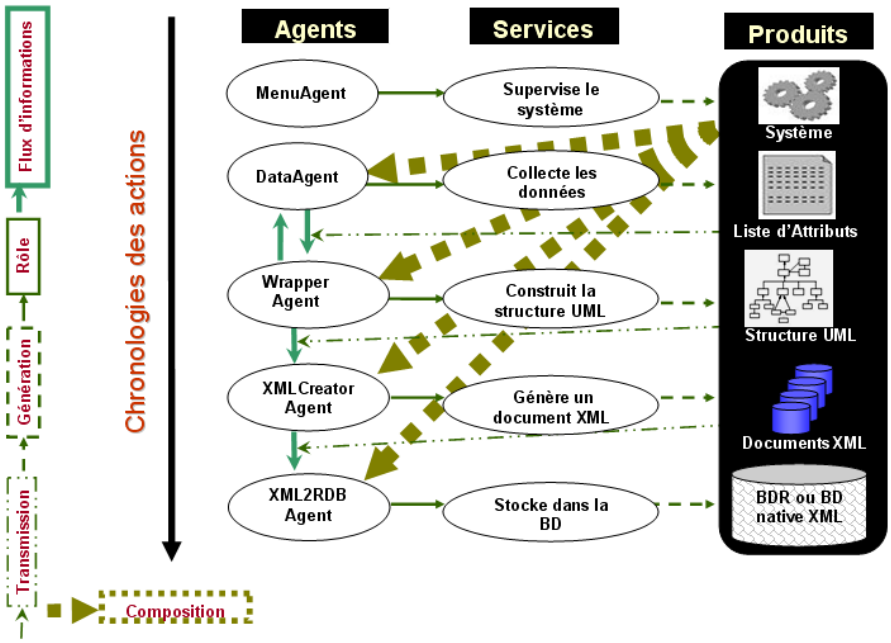

FIGURE 7.4 – Architecture du système SMAIDoC

1. http ://eric.univ-lyon2.fr/ bentayeb/logiciels.html

7.3.3 Bilan

Le processus d'intégration que nous avons présenté a pour finalité de modéliser des données complexes provenant de sources diverses et de types variés dans un format unifié (en l'occurrence XML), et de les stocker dans une base de données relationnelle. Pour modéliser de façon multidimensionnelle les données complexes, nous avons exploité leurs descripteurs de bas niveau ainsi que leurs descripteurs sémantiques obtenus grâce aux techniques de fouille de données. En effet, les techniques de fouille peuvent par exemple identifier les mesures des faits à analyser selon les besoins de l'utilisateur et permettre par conséquent de construire des contextes d'analyse pertinents. D'autre part, le choix du langage XML, outre ses nombreux avantages, permettrait par exemple d'extraire des connaissances à partir à la fois de la structure XML des documents (*XML StructureMining*) et de leur contenu (*XML Content Mining*). L'association de la fouille de structure et de la fouille de contenu devrait permettre d'améliorer la qualité des résultats obtenus [DBLB04].

Quant à l'utilisation des systèmes multiagents dans le processus d'intégration de données complexes, elle apporte une flexibilité à notre démarche d'entreposage de données complexes. En effet, notre approche repose sur une architecture évolutive dans laquelle on peut ajouter, modifier ou supprimer des services, voire créer de nouveaux agents. Ce travail ouvre de nombreuses perspectives, notamment dans l'extension des possibilités de SMAIDoC aux tâches de recueil et d'analyse des données complexes. Grâce à l'architecture évolutive de SMAIDoC, cette extension peut être réalisée. Il est possible de donner à l'agent de collecte de données (DataAgent) la capacité de recueillir des données en conversant avec des moteurs de recherche du web et d'exploiter les réponses de ces derniers. D'autre part, il est possible de créer de nouveaux agents dont les services respectifs peuvent être la modélisation multidimensionnelle des données complexes ou encore l'analyse à l'aide de techniques OLAP ou de fouille de données.

L'approche de modélisation de données complexes présentée dans cette section est basée sur les descripteurs de bas niveau et les descripteurs sémantiques des données. En exploitant les données complexes par la biais de leurs descripteurs uniquement, nous perdons de vue la notion de donnée complexe en tant qu'objet ayant une unité sémantique. Pour pallier ce problème, nous proposons dans la section suivante une nouvelle approche de modélisation de données complexes.

7.4 Modèle multidimensionnel d'objets complexes

Les travaux de recherche consacrés à la modélisation multidimensionnelle de données complexes sont assez nombreux et les solutions apportées sont variées et adaptées selon le type de complexité étudiée (données médicales, spatiales, Web, etc.).

Ainsi, nous ne prétendons pas ici traiter tout type de données complexes mais nous avons l'ambition d'analyser les données complexes dans leur globalité en tenant compte de leur sémantique. Nous considérons la donnée complexe comme un agrégat de données hétérogènes qui, une fois réunies, forment une entité sémantique. De plus, les liens à l'intérieur d'une même donnée complexe ou entre les données complexes elles-mêmes sont également explicités. Par ailleurs, pour renforcer le rôle de l'utilisateur au sein du SID, nous assurons le traitement symétrique entre faits et dimensions [PJ99]. Pour cela, nous utilisons un concept unique baptisé *objet complexe* qui représente une donnée complexe généralisant ainsi le concept de *meaningful fact* présenté par Nassis et al. [NRDR04] puisque nous l'utilisons aussi bien pour modéliser les faits que les dimensions. Peu de travaux proposent des modèles de données dans lesquels le traitement symétrique entre faits et dimensions est respectée. Le modèle qui s'en rapproche le plus est le modèle en Galaxie [RTTZ07].

Notre objectif est de proposer un modèle multidimensionnel qui puisse répondre à la fois aux spécificités des objets complexes que nous considérons (prise en compte des liens entre les objets et à l'intérieur d'un même objet), aux exigences de la modélisation multidimensionnelle et à la nécessité de pouvoir appliquer l'analyse en ligne.

7.4.1 Principe général

Notre idée principale derrière la modélisation d'objets complexes réside dans le fait de considérer l'espace multidimensionnel comme un ensemble d'objets reliés entre eux par des relations. De tous les modèles proposés dans la littérature, le modèle objet nous a semblé le plus approprié pour représenter les objets complexes ainsi que leurs liens internes et externes. A titre d'exemple, analyser une activité de publications de chercheurs d'un laboratoire revient à observer les articles publiés. Ces derniers sont alors des objets à observer qu'il faut représenter dans un modèle orienté analyse comme des entités à part entière. L'analyse de ces objets peut porter aussi bien sur leur contenu et/ou leur structure que sur leurs relations.

Un objet complexe est plus qu'une simple classe d'objets selon le paradigme objet ; il peut être apparenté à un diagramme de classes selon la terminologie UML. Par exemple un dossier patient peut être défini comme un objet complexe composé de rapports, de radios, de mesures numériques, etc.

Notre objectif est de pouvoir observer les objets complexes les uns par rapport aux autres grâce aux relations qui les relient entre eux. Les liens intra-objets permettent de naviguer à l'intérieur d'un même objet et enrichir ainsi l'analyse en ligne. Par ailleurs, chaque objet complexe du modèle peut jouer le rôle de fait ou de dimension. Le choix se fait au moment de l'analyse lorsque l'utilisateur affecte le rôle de fait ou de dimension aux objets choisis pour créer le cube d'objets complexes.

Pour modéliser de façon multidimensionnelle les objets complexes, nous avons besoin d'un niveau d'abstraction plus élevé que celui des diagrammes de classes qui constituent le premier niveau de modélisation. Alors que les diagrammes de classes permettent de représenter les liens intra-objets, nous utilisons les diagrammes de packages pour pouvoir représenter les liens inter-objets. Un package regroupe alors plusieurs classes qui forment une entité sémantique en un seul objet complexe. Les dépendances entre les packages représentent les liens inter-objets. Les packages constituent un deuxième niveau de modélisation qui a l'avantage de faciliter l'affectation des rôles aux objets.

Grâce à l'*opérateur de projection cubique* que nous avons défini, l'utilisateur peut créer des cubes d'objets complexes à partir du modèle multidimensionnel. Il sélectionne, au niveau diagramme de packages, un package pour jouer le rôle de sujet d'analyse ; ce qui implique la projection du modèle multidimensionnel sur les autres packages pour constituer les axes d'analyse. Ensuite, au niveau de la couche diagramme de classes, les attributs jouant le rôle de mesures avec les fonctions d'agrégat associées sont fixés.

7.4.2 Exemple illustratif

Pour illustrer nos propositions concernant la modélisation multidimensionnelle des objets complexes, nous présentons un exemple issu d'une étude de cas réelle adaptée de "*XMark benchmark project*" [1] concernant les ventes aux enchères (*auctions*). Un diagramme de classes UML des ventes aux enchères est donné dans la Figure 7.5. Tout au long de ce chapitre, nous utilisons la terminologie anglosaxonne pour l'exemple.

Une vente aux enchères (*auction*) correspond à un article (*item*) qui appartient à une ou plusieurs catégories (*categories*) et qui est ou a été vendu (*sold*) par une personne (*person*). Une *auction* peut être ouverte (*open*) ou fermée (*closed*). Une *auction* ouverte peut être observée (*watched*) par plusieurs personnes et peut être sujet de plusieurs offres (*bids*). Une *auction* fermée est attribuée à un acheteur (*buyer*) et est annotée (*annotated*) une fois. Les données des *auctions* peuvent être des sujets d'analyse pour des applications décisionnelles. Par exemple, il est intéressant de savoir quelles sont les catégories d'articles les plus concernées par les ventes aux enchères, ou quel est l'impact de la distance entre les localisations des ventes et les adresses des clients acheteurs sur la vente, ou encore quelle est l'évolution du prix de vente des enchères.

1. http ://www.xml-benchmark.org/

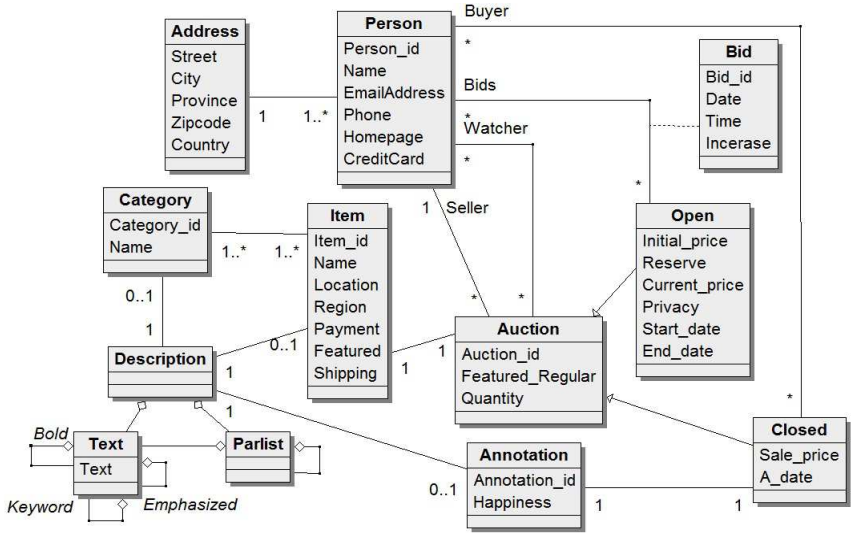

FIGURE 7.5 – Diagramme de classes des ventes aux enchères (*auctions*)

7.4.3 Caractéristiques du modèle multidimensionnel de données complexes

Le modèle multidimensionnel de données complexes des ventes aux enchères doit pouvoir répondre aux premiers besoins exprimés par les utilisateurs, mais doit également permettre des analyses plus élaborées. Aussi, nous avons pu identifier les caractéristiques suivantes que doit supporter un modèle multidimensionnel de données complexes.

1) *Faits et membres de dimensions complexes.* Le modèle de données doit permettre de définir des faits et des dimensions complexes (structure complexe). Par exemple, si on veut analyser les *Auction*[2] par rapport aux *Item*, le modèle de données des ventes aux enchères montre que *Auction* est composé d'une classe et de deux sous-classes alors que *Item* est composé de quatre classes reliées entre elles par huit relations.

2) *Hiérarchies à l'intérieur des données complexes.* Le modèle de données doit permettre d'observer les hiérarchies à l'intérieur d'une donnée complexe lorsqu'on souhaite traiter les éléments qui la composent. En outre, les membres d'une hiérarchie peuvent être des attributs et/ou des classes. Dans l'exemple de *Auction*, nous pouvons observer une hiérarchie dans *Address* qui peut être composée de *city* et de *region*.

3) *Hiérarchies des données complexes.* Les données complexes peuvent être organisées en hiérarchies. Dans notre exemple, *Item* et *Category* sont des données complexes et forment une hiérarchie.

4) *Traitement symétrique des faits et dimensions.* Les données complexes peuvent jouer le rôle de fait ou de dimension selon les besoins de l'utilisateur au moment de l'ana-

2. Nous utilisons les majuscules pour différencier les noms des données complexes des noms de classes

lyse. Depuis l'apparition des entrepôts de données dans les années 90, cette symétrie entre faits et dimensions a été largement recommandée [Cod93, PJ99].

5) *Mesures simples et mesures complexes.* Le modèle doit être capable de considérer des mesures simples et des mesures complexes selon que la mesure est attribuée à un attribut simple ou à un attribut complexe respectivement. Par exemple, le prix d'une *Auction* est une mesure simple alors que la description d'un *Item* peut être utilisée comme une mesure complexe.

Pour prendre en compte toutes les caratéristiques citées dans la section 7.4.2, nous proposons un modèle multidimensionnel d'objets complexes et un opérateur de projection cubique pour extraire des cubes d'objets complexes. Nous utilisons pour cela le paradigme objet qui, de par son expressivité, fournit des mécanismes puissants permettant de décrire les données comme des objets ainsi que leurs liens.

7.4.4 Formalisation

Nous présentons dans cette section notre modèle multidimensionnel d'objets complexes. Il est défini au niveau de la couche de diagramme de classes pour décrire les données et au niveau de la couche des diagrammes de packages pour décrire les objets complexes afin de faciliter la création de cubes d'objets complexes. Notre modèle d'objets complexes est fondé sur quatre concepts que nous allons détailler ci-après.

7.4.4.1 Objet complexe

Un *objet complexe* (CO - *Complex Object*) est un ensemble de classes d'objets qui forment une même entité sémantique. Il peut être représenté par un diagramme de classes au premier niveau de modélisation ou par un package au deuxième niveau de modélisation. Pour chaque package représentant un objet complexe, nous choisissons sa *classe représentative* parmi toutes les classes formant l'objet complexe.

Un objet complexe est caractérisé par *des attributs simples* (SOA - *Simple Object Attribute*), qui sont des attributs de la classe représentative et par *des attributs complexes* (COA - *Complex Object Attribute*) qui sont les autres classes de l'objet complexe. En considérant les classes qui composent un objet complexe comme des attributs complexes, cela permet de manipuler les attributs d'objets complexes de manière uniforme. De plus, cela facilite la définition des concepts de notre modèle et le traitement des mesures complexes. La Figure 7.6 présente et décrit le méta-modèle de l'objet complexe.

- La classe *Object_Class* représente une classe objet de CO.
- La classe *Class_Attribute* représente les attributs simples d'une classe.
- L'association *Class_to_attribute_link* lie une classe d'objets à ses attributs simples.

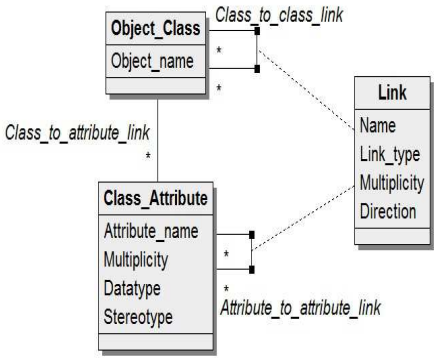

FIGURE 7.6 – Méta-modèle d'objet complexe

- L'association *Class_to_class_link* représente les liens entre les classes qui composent l'objet complexe CO (par exemple : association, héritage).
- L'association *Attribute_to_attribute_link* représente les liens entre les attributs simples de CO.

Définition 7.1 *Types de base*
Nous définissons les types de base suivants.

1) *Attribute* représente un attribut de classe
2) *Class*={Att_i : Attribute/ i=1,...} représente une classe
3) *Link*={association, composition, aggregation, ...} est un ensemble de relations entre les classes.
4) *Multiplicity*={*, 0..1, 1, 1..*} représente la cardinalité des relations
5) *Direction* ={forward, backward, none} représente la direction de navigation (le sens) d'une relation.

Définition 7.2 *Objet complexe*
Formellement, un objet complexe (CO) est défini comme un triplet :
$Obj = (ID^{Obj}$: Attribute, SA^{Obj}, $SR^{Obj})$ où :

- ID^{Obj} est l'identifiant de CO.
- $SA^{Obj} = \{A_i^{Obj} / i \in \mathbb{N}\}$ est l'ensemble des attributs de CO,
  tel que $A^{Obj} = (SOA$: Attribute$|COA$: Class).
- $SR^{Obj} = \{R_j^{Obj} / j \in \mathbb{N}\}$ est l'ensemble des relations entre classes et/ou entre attributs de CO où :
  R^{Obj}=(L : Link, $SrcM$: Multiplicity, $\{TgtM$: Multiplicity$\}$,
  D : Direction, Src : Class|Attribute,
  $\{Tgt_k$: Class | Attribute /k = 1,...$\}$).

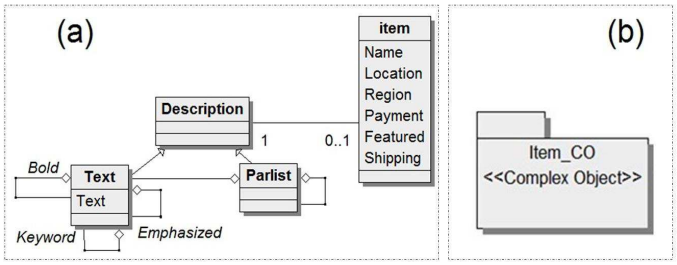

FIGURE 7.7 – Exemple d'objet complexe représentant les items

Notons que l'ensemble des relations possibles à l'intérieur d'un objet complexe n'est pas limité aux seuls liens définis dans la modélisation orientée-objet. En effet, nous étendons ces liens à d'autres relations spécifiques tels que les liens de référence (*references*), ou de navigation (*navigation*), etc. Ce qui peut entraîner certaines extensions de modélisation pour le diagramme de classes d'un objet complexe.

Exemple 7.1

Dans le diagramme de classes des *auctions* (Figure 7.5), nous identifions six données complexes que nous modélisons comme des objets complexes à savoir : Person_CO, Auction_CO, Item_CO, Category_CO, Annotation_CO et Bid_CO. Ces objets sont représentés par les classes Person, Auction, Item, Category, Annotation et Bid respectivement. Nous nous limitons ici à la présentation de l'objet complexe Item_CO. La Figure 7.7(a) présente Item_CO au niveau du diagramme de classes. Par ailleurs, dans la Figure 7.7(b), nous encapsulons la structure complexe de Item_CO pour pouvoir la présenter au niveau de la couche diagramme de packages. Nous définissons le stéréotype $\ll ComplexObject \gg$ et l'associons au package Item_CO pour le distinguer des autres packages. Nous pouvons définir Item_CO de la façon suivante :

$Item\_CO = (Item\_id, SA^{Item\_CO}, SR^{Item\_CO})$ où $SA^{Item\_CO} = \{Name, Location, Region, Payment, Featured, Shipping, Description, Text, Parlist\}$ et $SR^{Item\_CO} = \{Item\_desc, Desc\_parlist, Desc\_text, emphasize, keyword, bold, Parlist\_parlist, Parlist\_text\}$.

Un exemple de formalisation des relations est $Item\_desc$ =(association, 1, {0..1}, none, Item, {Descritption}).

7.4.4.2 Relation complexe

Le concept de *relation complexe* (CR - *Complex Relation*) définit les liens entre les objets complexes par opposition aux relations que l'on trouve à l'intérieur d'un même objet complexe. Une *relation complexe* peut être définie entre deux classes représentatives

de deux objets complexes ou entre les autres classes.

La notion de relation complexe est aussi importante que la notion d'objet complexe lui-même du point de vue de la vision multidimensionnelle que nous souhaitons donner à notre modèle d'objets complexes. Dans ce contexte, une relation complexe peut représenter un axe d'analyse possible selon lequel un objet complexe peut être analysé. Typiquement, une relation complexe peut relier plus de deux objets complexes, ce qui définit l'arité de la relation (*relationship arity*). Toutefois, nous choisissons de décomposer chaque CR d'arité supérieure à deux en un ensemble de relations binaires. La décomposition de relations n-aires en relations binaires est justifiée par le fait que dans un modèle multidimensionnel, les axes d'analyse sont définis entre le *fait* d'une part et *une dimension* d'autre part. En outre, nous définissons une relation complexe selon deux niveaux : un niveau classe et un niveau objet complexe. Au niveau classe, la CR relie deux classes appartenant au même objet complexe. Au niveau objet complexe, la CR relie deux objets complexes. Notons que dans le cas où deux COs sont reliés via plus d'une CR, chaque CR est représentée séparément.

Définition 7.3 *Relation complexe*

1) Au niveau des classes, une relation complexe CR est définie comme suit :
   R=(L : Link, $SrcM$: Multiplicity, $TgtM$: Multiplicity, D : Direction, Src : Class, Tgt : Class).
2) Au niveau de l'objet complexe, une relation complexe CR est définie comme suit :
   $R = (Obj_s^R, Obj_t^R)$ où Obj_s^R représente l'objet complexe source de R et Obj_t^R représente l'objet complexe cible.

Exemple 7.2

Un exemple de CR est montré dans la Figure 7.8(a). Il existe une relation complexe entre Auction_CO et Item_CO. La CR est modélisée comme une dépendance (*dependency*) entre les packages, définie par le stéréotype $\ll ComplexRelationship \gg$. Nous utilisons les stéréotypes afin de pouvoir distinguer les CR des autres dépendances entre les packages. Nous avons également utilisé un attribut de stéréotype $\ll name \gg$ pour pouvoir différencier les CRs multiples entre elles lorsqu'elles relient la même paire de COs. Le diagramme de classes dans la Figure 7.8(b) montre que l'origine d'une CR est une association entre les classes Item et Auction.

7.4.4.3 Hiérarchie d'attributs

Dans la section 7.4.4.1, nous avons défini la notion de relation comme faisant partie de la définition d'un objet complexe. Dans cette section, nous nous focalisons sur un

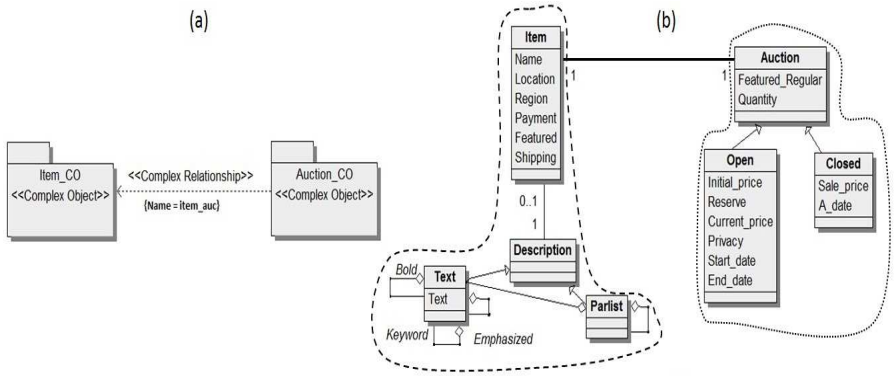

FIGURE 7.8 – Exemple de relations entre objets complexes

certain type de relations qui permet d'organiser en hiérarchies les attributs simples et/ou les attributs complexes d'un objet complexe. Nous appelons une telle organisation une hiérarchie d'attributs (AH - *Attribute Hierarchy*). Cependant, une AH ne contient pas forcément tous les attributs d'un objet complexe. De ce fait, une AH définit un ordre partiel dans l'ensemble des COAs. Une AH ne peut être définie qu'au niveau de la couche diagrammes de classes puisqu'elle est interne à l'objet complexe.

Définition 7.4 *Hiérarchie d'attributs*
Une hiérarchie d'attributs AH est définie par $AH^{Obj} = \{A_i^{Obj} \in SA^{Obj} \cup \{ID^{Obj}\}/i \in \mathbb{N}\} \cup \{All^A\}$ tel que All^A représente un attribut "artificiel" ayant le niveau le plus agrégé de la hiérarchie.

En outre, nous définissons une fonction $Level_A(A_i^{AH}, AH^{Obj})$ qui retourne le niveau de granularité de chaque attribut de AH. Nous supposons que le niveau de l'attribut ayant la granularité la plus fine dans la hiérarchie est égal à 0. Enfin, nous notons $AttObj(AH^{Obj}) = Obj$ la fonction qui associe la AH^{Obj} à l'objet complexe Obj.

Exemple 7.3

La Figure 7.9 montre un exemple de AH associée à Person_CO. La AH est composée d'attributs person_id, city, country et All^A avec leur niveau de granularité respectif 0, 1, 2 et 3. Nous définissons le stéréotype $\ll AttributeHierarchy \gg$ et l'associons à la AH pour distinguer les membres de la hiérarchie des autres COAs descriptifs de CO. Nous utilisons également les attributs de stéréotype *name* et *level* pour faire référence au nom de la hiérarchie à laquelle appartient l'attribut et à son niveau de granularité au sein de cette même hiérarchie. Notons que dans le cas où un membre d'une hiérarchie d'attributs correspond à une classe d'objets, c'est toute la classe qui est définie par le stéréotype $\ll AttributeHierarchy \gg$.

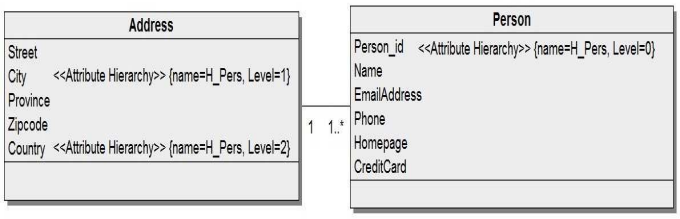

FIGURE 7.9 – Exemple de hiérarchie d'attributs associée à Person_ID

7.4.4.4 Hiérarchie d'objets

Une hiérarchie d'objet (OH - *Object Hierarchy*) est analogue à une hiérarchie d'attributs AH. Alors que la AH organise les attributs d'un objet complexe CO, la OH organise les objets complexes eux-mêmes en hiérarchie. De même que la AH, la OH définit un ordre partiel dans l'ensemble des COs. D'autre part, puisque la OH organise les COs, elle est définie seulement au niveau de la couche diagramme de packages.

Définition 7.5 *Hiérarchie d'objets*
Une hiérarchie d'objets est définie de la manière suivante : $OH = \{Obj_i / i \in \mathbb{N}\} \cup \{All^{Obj}\}$ où All^{Obj} représente un CO "artificiel" ayant le niveau le plus agrégé dans la hiérarchie et joue le même rôle que All^{A}. Nous définisson également une fonction $Level_{Obj}(Obj, OH)$ qui retourne le niveau de chaque CO de OH. Nous supposons que le niveau le plus fin de la hiérarchie est égal à 0.

Exemple 7.4

La Figure 7.10(a) montre un exemple de hiérachie d'objets OH composée de Item_CO et Category_CO au niveau du diagramme de packages. Nous définissons le stéréotype $\ll ObjectHierarchy \gg$ et l'associons aux membres de OH afin de distinguer l'organisation hiérarchique des COs des autres CRs. Nous utilisons également les atributs de stéréotype *name* et *level* de la même manière que dans une AH. La Figure 7.10(b) montre l'origine de OH qui est une association entre les classes Item et Category.

7.4.4.5 Le schéma multidimensionnel

Après avoir défini les quatre concepts de modélisation CO, CR, OH et AH, nous définissons le schéma multidimensionnel d'objets complexes (COMM - *Complex Object Multidimensional Model*). Le modèle COMM est composé d'un ensemble d'objets complexes COs qui sont reliés entre eux par un ensemble de relations complexes CRs. En outre, de la même manière que certains objets complexes COs peuvent s'organiser en hiérarchies (OHs), certains attributs de COs peuvent être organisés sous forme de hiérarchie (AHs).

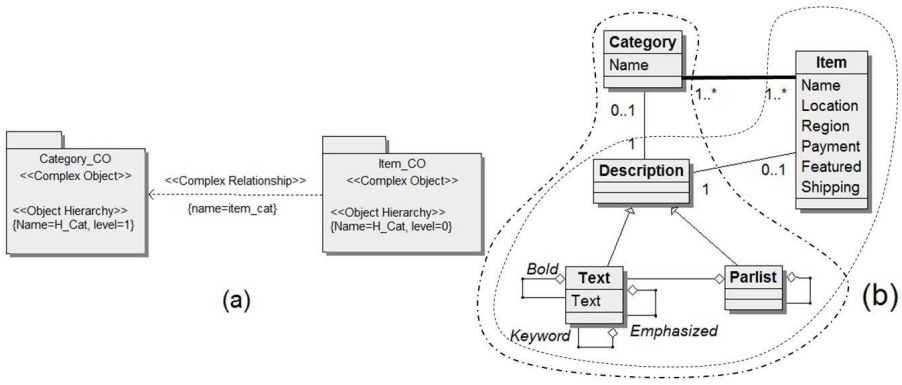

FIGURE 7.10 – Exemple de hiérachie d'objets

Définition 7.6 *Schéma multidimensionnel d'objets complexes*
Le schéma multidimensionnel d'objets complexes est défini par $COMM$=(SO, SR, SAH, SOH) où $SO = \{Obj_i/i \in \mathbb{N}\}$, $SR = \{R_j/j \in \mathbb{N}\}$, $SAH = \{AH_k/k \in \mathbb{N}\}$ et $SOH = \{OH_m/m \in \mathbb{N}\}$.

Exemple 7.5

La Figure 7.11 présente et décrit le modèle COMM des *auctions* au niveau de la couche diagramme de packages. Les packages Item_CO, Category_CO et Annotation_CO importent des classes à partir d'un package commun Description. C'est ce qu'on appelle un *objet abstrait* dans la terminologie UML. Les dépendances multiples entre les packages Auction_CO et Person_CO représentent les CRs. La classe association Bid est modélisée par un CO baptisé Bid_CO, ce qui donne lieu à deux relations complexes binaires. La première relie Auction_CO à Bid_CO et la deuxième relie Bid_CO à Person_CO. Nous notons *Auction_COMM* le modèle multidimensionnel des *auctions*.

7.4.5 Cube d'objets complexes

Les concepts de CO, CR, OH et AH sont le socle du modèle multidimensionnel et de l'analyse en ligne d'objets complexes que nous proposons. Chaque CO peut jouer le rôle de fait ou de dimension, chaque CR peut être vue comme un axe d'analyse alors que les AHs et les OHs permettent d'agréger les données. Par conséquent, dans le but de répondre à des besoins spécifiques des utilisateurs, lorsque l'objet complexe correspondant au fait complexe est choisi parmi les différents packages, nous procédons à la dérivation du modèle d'analyse à partir du modèle multidimensionnel d'objets complexes. Nous appelons ce modèle d'analyse *cube d'objets complexes* (COC - *Complex Object Cube*). Les données de COC peuvent être générées à la demande à partir des données de COMM, puis matérialisées pour être réutilisées plus tard pour l'analyse OLAP. Nous soulignons ici tout l'intérêt

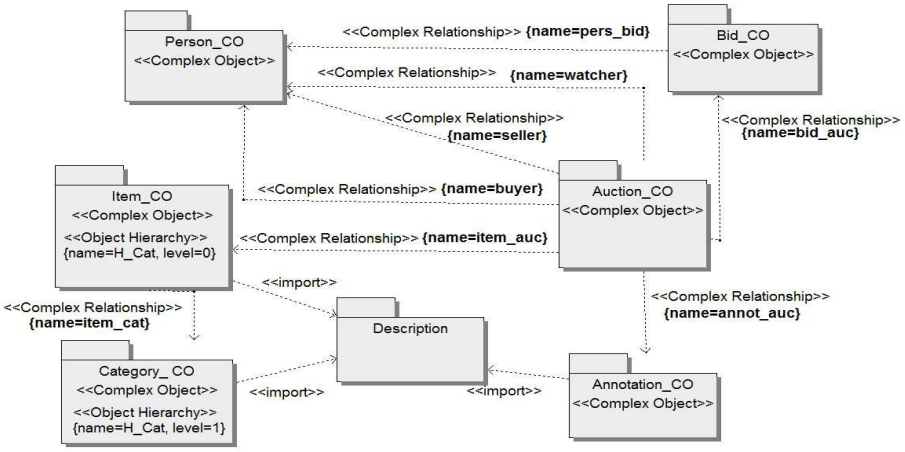

FIGURE 7.11 – Modèle multidimensionnel d'objets complexes des *auctions*

de cette approche de construction de cubes d'objets complexes puisqu'elle vient en appui à la personnalisation des analyses des utilisateurs.

Nous avons également proposé un langage composé d'un ensemble d'opérateurs afin de dériver des cubes à partir de COMM et de manipuler les COCs [BBB10b, BBB10a]. Le langage est composé de (1) un *langage de définition de données* (LDD) qui permet de créer et modifier la structure des cubes et (2) un *langage de manipulation de données* qui permet de réaliser des analyses OLAP. Nous nous focalisons ici sur l'opérateur le plus important de LDD, appelé *projection cubique*, dont la définition formelle peut être trouvée dans [BBB10a]. Nous présentons ici l'opérateur de projection cubique qui permet d'assurer le passage des concepts CO et CR aux concepts de fait et de dimension. Il agit sur les deux niveaux de modélisation du modèle multidimensionnel COMM de la manière suivante.

- Au niveau de la couche diagramme de packages, la projection cubique est réalisée par projection de COMM sur un objet complexe, pour lequel le rôle de *fait* est assigné. Le fait est dit complexe (CF) puisqu'il correspond à un objet complexe CO. D'autres CO sont projetés et désignés pour être des axes d'observation jouant le rôle de dimensions. Une dimension est composée par un CO ou par un ensemble de COs organisés en hiérarchie. L'organisation hiérarchique d'une dimension est obtenue à partir de la définition des OHs de COMM. Les axes d'analyse de COCs sont obtenus à partir de CRs qui lient directement le fait complexe aux autres objets complexes. Pour terminer, la projection cubique préserve les définitions des AHs qui sont assignées aux COs dans le COMM. En résumé, la projection cubique procède à l'élagage de COMM en gardant uniquement le CF, ses COs qui lui sont directement liés, possiblement organisés en OHs et toutes les AHs associés aux objets du COC.
- Au niveau du diagramme de classes, nous assignons le rôle de *mesure* à un attribut de CF et nous définissons *la fonction d'agrégat* correspondante. Une mesure peut

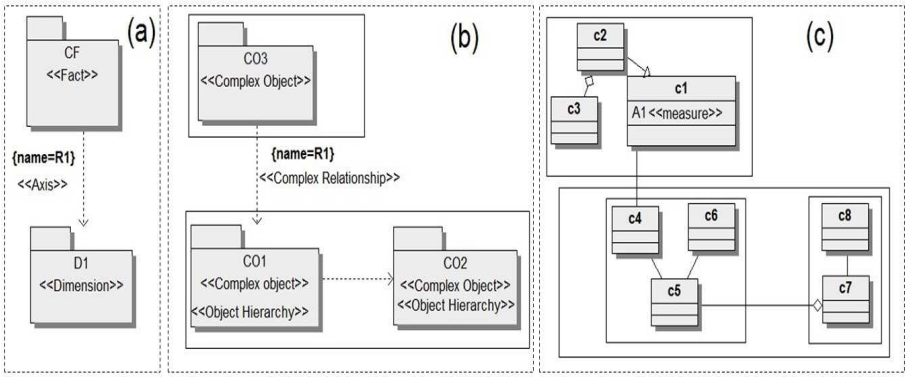

FIGURE 7.12 – Représentation à trois niveaux d'un cube d'objets complexes

être assignée à un attribut simple ou à un attribut complexe de CF.

Un COC peut être représenté par deux couches de modélisation comme un COMM. Toutefois, puisque dans le modèle du COC, le fait et les dimensions sont explicitement nommés, une couche supplémentaire est nécessaire pour améliorer la lisibilité du COC. Les trois couches de COC sont présentées dans la Figure 7.12.

1) La première couche (a) est équivalente à un schéma en étoile. Elle est définie avec un diagramme de packages dans lequel un package correspond soit à un fait soit à une dimension. Nous distinguons les faits des dimensiosn en utilisant les stéréotypes $\ll$ *Fact* $\gg$ et $\ll$ *Dimension* $\gg$. Nous définissons le stéréotype $\ll$ *Axis* $\gg$ et l'associons aux dépendances entre les packages et nous ajoutons un attribut de stéréotype *name* pour différencier les multiples axes d'analyse entre le fait et une même dimension, connus sous le nom de *dimension roles*. Les noms des axes sont les mêmes que dans les noms des CR qui relient le CF aux autres COs.
2) La deuxième couche (b) donne des détails concernant le contenu de chaque dimension en termes de COs et de CF. Cette couche est similaire à la couche de diagramme de packages de COMM. Toutefois, elle est limitée aux COs et CRs les plus pertinents de COC. Le CF est obtenu par projection de COMM sur CO_3 tandis que la dimension D_1 est organisée en hiérachie de CO_1 et CO_2.
3) La troisième couche (c) donne des détails de chaque CO et présente ses diagrammes de classes correspondants. Le diagramme de classes de chaque CO est le même que dans la couche de diagramme de classes COMM excepté pour le CF où nous définissons le stéréotype $\ll$ *measure* $\gg$ et l'associons à l'attribut qui correspond à la mesure.

Exemple 7.6

Supposons que nous souhaitons calculer la moyenne des prix de ventes à la clôture des ventes aux enchères. Pour répondre à ce besoin, nous procédons de la manière suivante.

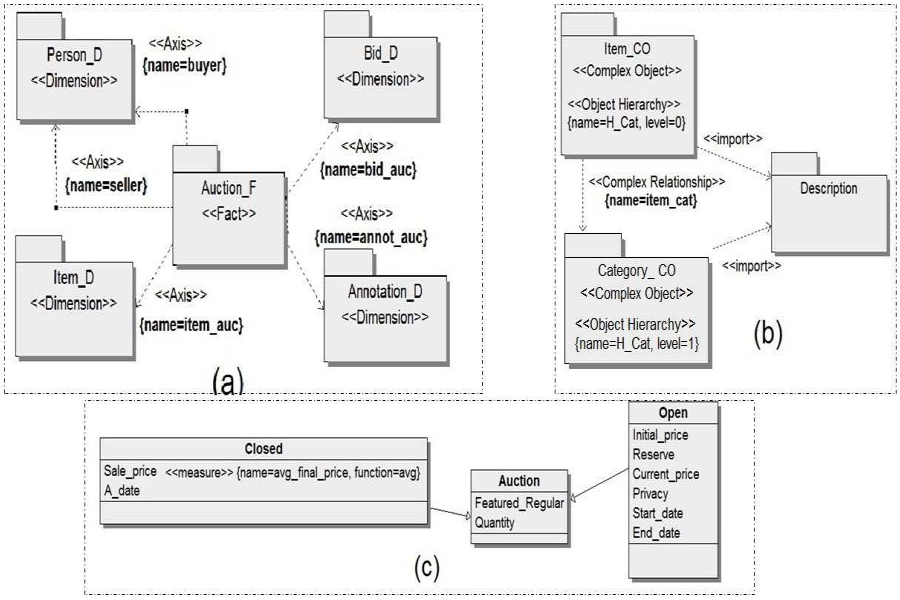

FIGURE 7.13 – Exemple de cube d'objets complexe des *auctions*

Nous projetons *Auction_COMM* sur l'obejt complexe Auction_CO, puis nous nous focalisons sur le diagramme de classes qui le représente pour définir la mesure et la fonction d'agrégat. Dans la Figure 7.13, nous trouvons les trois niveaux de représentation du cube complexe. La première couche (a) montre le modèle en étoile composé du CF Auction_F et de quatre dimensions, dont l'une possède deux rôles. En (b), nous zoomons à l'intérieur de la dimension Item_D pour découvrir l'organisation hiérarchique de la dimension, composée de Item_CO et Category_CO. En (c), nous présentons le diagramme de classes de Auction_CO. La mesure *avg_final_price* est associée à l'attributs simple Sale_price dont les valeurs sont agrégées en utilisant la fonction d'agrégat avg.

7.4.6 Implémentation

Pour valider notre modèle multidimensionnel d'objets complexes, un prototype a été développé, baptisé *Auction_COMM* représentant un entrepôt de données des ventes aux enchères, et faisant partie d'une architecture globale d'entreposage de données (Figure 7.14). L'entrepôt au format XML a été construit en tranformant le document XML généré à partir des données de *XMark project*, au travers d'un processus ETL mis en place. Le modèle d'entrepôt est présenté dans la Figure 7.11. Au niveau logique, le modèle multidimensionnel est défini en utilisant XML Schema.

Plus d'informations sur les mappings du niveau conceptuel au niveau logique sont rapportés dans les travaux de Boukraâ et. al [BMB09]. Le stockage de données en XML natif a été réalisé sous le SGBD Oracle 11g2 DB. Chaque objet complexe est modélisé

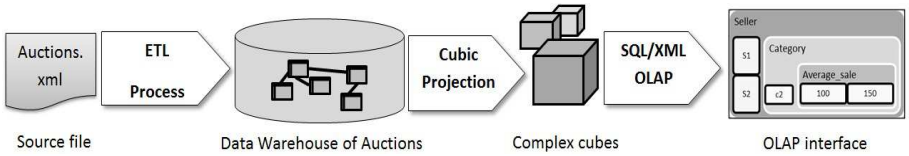

FIGURE 7.14 – Processu d'entreposage et d'analyse en ligne des données complexes *auctions*

comme une table d'objets de type XMLType. Chaque ligne de la table correspond à une instance de l'objet complexe. Les hiérarchies d'attributs AHs et les hiérarchies d'objets OHs sont implémentées à l'intérieur de leur objet complexe correspondant.

Le cube complexe est stocké de la même manière que COMM : le CF ainsi que chaque membre de dimension sont stockés sous forme de table objet. Nous avons également implémenté l'opérateur de projection cubique et avons dérivé un cube complexe avec deux mesures *Avg_sale_price* et *Avg_current_price*. Pour atteindre cet objectif, nous avons encodé un fichier de métadonnées en XML correspondant à la définition formelle de *auction_COMM*. Le fichier de métadonnées est utilisé par les procédures stockées PL/SQL qui parcourent les données du schéma multidimensionnel COMM et peuplent les tables d'objets complexes. En outre, pour appliquer les opérateurs OLAP sur le cube d'objets complexes obtenu, nous avons écrit quelques requêtes décisionnelles en SQL/XML où la partie écrite en XML permet d'accéder aux attributs des objets complexes via les chemins XML et la partie écrite en SQL couvre les chemins XML et permet d'effectuer des regroupements de données selon les AHs et les OHs.

Pour terminer et afin d'améliorer les performances de notre système, nous avons proposé une méthode d'optimisation de requêtes basée sur la fragmentation verticale en utilisant les règles d'association [BBB11]. Nos expérimentations montrent que cette démarche permet de réduire le temps de traitement de requêtes décisionnelles XQuery de façon significative.

7.5 Conclusion

L'avènement des données complexes (multiformats, multistructures, ayant une sémantique différente, etc.) a relancé de nouveaux défis concernant les processus d'entreposage, d'analyse en ligne et de fouille de données. Les données complexes renferment beaucoup d'informations qui peuvent être extraites en appliquant des techniques sophistiquées et adaptées qu'il faut intégrer dans le processus global d'entreposage. De nouvelles approches d'intégration, de modélisation et d'analyse en ligne de données complexes ont alors émergé.

Les nombreux travaux de recherche dans ce domaine montrent la difficulté de trouver un consensus pour entreposer et analyser les données complexes. Nous avons tenté, dans

un premier temps, de décrire un processus complet d'entreposage de données complexes pour l'aide à la décision, tout en soulevant les problèmes liés à leur intégration, leur structuration et leur modélisation multidimensionnelle. Notre approche d'entreposage des données complexes présente des avantages tels que l'utilisation d'un format unifié, XML, pour décrire des données de nature différente et le recours à la fouille de données pour extraire des connaissances nécessaires à la construction de contextes d'analyse pertinents.

Pour pallier les insuffisances des modèles multidimensionnels classiques, tant au niveau de la prise en compte des données complexes que de l'implication de l'utilisateur, nous avons défini dans un deuxième temps, un nouveau modèle multidimensionnel de données complexes centré utilisateur selon l'approche orientée objet. Dans ce modèle, une donnée complexe est représentée par un objet selon le paradigme objet. L'utilisation d'un concept unique, - objet complexe - dans le modèle multidimensionnel, nous a permis d'une part d'expliciter les liens sémantiques intra- et inter-objets, et d'autre part de résoudre le problème du traitement symétrique des faits et des dimensions. En effet, la modélisation objet permet de traiter de façon symétrique les faits et les dimensions. Ce dernier point aide à la personnalisation des contextes d'analyse des utilisateurs.

L'originalité de notre modèle multidimensionnel d'objets complexes réside dans le choix d'une architecture à deux niveaux. Au niveau diagramme de packages, nous modélisons l'univers par un ensemble d'objets complexes, reliés entre eux par des relations et pouvant être organisés en hiérarchies. Au niveau diagramme de classes, nous modélisons la structure et la sémantique de chaque objet complexe par un diagramme de classes et nous explicitons les hiérarchies entre les attributs de manière à pouvoir les exploiter dans les analyses futures.

Nous avons par ailleurs défini un opérateur de projection cubique qui permet à un utilisateur de définir des cubes complexes selon ses propres besoins d'analyse. Tout d'abord, l'utilisateur choisit l'objet complexe qui représentera le fait à observer, puis, pour obtenir les objets complexes jouant le rôle de dimensions, le modèle multidimensionnel est projeté sur le fait complexe choisi.

Comparé aux modèles multidimensionnels existants, le modèle de cube d'objets complexes que nous obtenons à patir du modèle multidimensionnel d'objets complexes apporte les nouveautés suivantes. D'une part, l'objet-fait (ou l'objet-dimension) étant par définition un objet complexe, il conserve toute sa complexité lors de la projection cubique, notamment en termes de liens entre ses différents composants. Cela étend les possibilités d'analyse du cube d'objets complexes. Par exemple, il devient possible d'exploiter les liens de composition pour décomposer l'objet-fait (ou l'objet-dimension) en sous-objets-faits (sous-objets-dimensions) et d'analyser chacun à part. D'autre part, un objet-fait est également décrit par un ensemble d'attributs qui peuvent être complexes à leur tour. Ainsi, il devient possible d'observer des mesures complexes, représentant des sous-objets de l'objet-

fait. Les fonctions d'analyse sont alors à définir en adéquation avec la nature des mesures à analyser.

Nous avons aussi démontré l'efficacité de notre approche en mettant en œuvre un processus d'expérimentation qui a impliqué le développement d'un prototype sous le SGBD Oracle en utilisant le stockage objet en XML. Nous avons conçu et créé un entrepôt de données de ventes aux enchères au format XML généré à partir des données de *XMark project*, au travers d'un processus ETL mis en place. Enfin, nous avons implémenté l'opérateur de projection cubique qui permet de dériver des cubes complexes à partir de l'entrepôt de données.

Ce travail ouvre de nombreuses perspectives tant au niveau de la modélisation conceptuelle, logique que physique. Une des évolutions possible réside dans l'amélioration de l'automaticité des différentes étapes de modélisation multidimensionnelle de données complexes. Par exemple, en ce qui concerne le passage du niveau diagramme de classes au niveau diagrammes de packages, il est intéressant de proposer au concepteur une approche semi-automatique pour l'aider à définir le diagramme de packages (regroupement de classes en objet complexe, choix de la classe représentative d'un objet complexe, etc.). De façon similaire, développer un outil graphique capable d'aider l'utilisateur à définir des cubes complexes à partir du modèle multidimensionnel représente une piste de recherche prometteuse. Par ailleurs, il faut certainement définir de nouvelles métriques (mesures) pouvant exploiter le caractère complexe des données permettant ainsi de créer des cubes complexes plus pertinents.

Pour terminer, nous pouvons dire que dans le contexte des entrepôts de données complexes, le problème de la performance constitue un problème majeur et demeure plus que jamais un enjeu crucial. Les performances des SGBD natifs XML étant actuellement limitées en termes de temps de réponse et de volume des données, il est nécessaire de trouver des moyens pour les optimiser. Dans ce contexte, nous avons comme objectif de poursuivre nos travaux sur l'optimisation et l'évaluation des performances des entrepôts de donnée complexes. D'abord, dans la continuité de nos travaux qui utilisent la fragmentation verticale, il est intéressant d'intégrer dans le modèle de coût le nombre de jointures à réaliser et la taille des fragments obtenus. Ensuite, nous pensons que l'idée de combiner une méthode d'indexation avec la configutaion de fragments obtenue constitue une piste de recherche intéressante.

7.6 Publications

Dans cette section, nous présentons nos nombreuses publications réalisées dans le domaine de l'entreposage et de l'analyse en ligne de données complexes. Certaines de ces publications, notamment les *chapitres d'ouvrages*, sont des travaux de synthèse qui couvrent

de façon plus large les thèmes abordés dans ce chapitre.

Revues internationales

[1] D. Boukraâ, O. Boussaid, **F. Bentayeb**, S. Loudcher, "OLAP Operators For A Complex Object-Based Multidimensional Model", Journal of Data Mining and Business Intelligence, 2010, 34-46.

[2] O. Boussaid, J. Darmont, **F. Bentayeb**, S. Loudcher, "Warehousing complex data from the Web", International Journal of Web Engineering and Technology, 2008, 408-43 (Invited paper).

[3] O. Boussaid, A. Tanasescu, **F. Bentayeb**, J. Darmont, "Integration and Dimensional Modelling Approaches for Complex Data Warehousing", Journal of Global Optimization, Vol. 37, No. 4, April 2007, 571-591.

Revues nationales

[4] O. Boussaid, **F. Bentayeb**, J. Darmont, S. Rabaséda, "Vers l'entreposage des données complexes : structuration, intégration et analyse", Ingénierie des Systèmes d'Information, Vol. 8, No. 5-6, 2003, 79-107.

[5] F. Clerc, A. Duffoux, C. Rose, **F. Bentayeb**, O. Boussaid, "SMAIDoC : Un Système Multi-Agents pour l'Intégration des Données Complexes", Revue des Nouvelles Technologies de l'Information, No. 1, 2003, 13-24.

Chapitres d'ouvrages d'audience internationale

[6] **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", IGI Book : Business Intelligence Applications and the Web : Models, Systems and Technologies, (In. Marta E. Zorrilla, Jose-Norberto Mazón, Óscar Ferrández, Irene Garrigós, Florian Daniel, Juan Trujillo). To appear.

[7] H. Mahboubi, J. Ralaivao, S. Loudcher, O. Boussaïd, **F. Bentayeb**, J. Darmont, "X-WACoDa : An XML-based approach for Warehousing and Analyzing Complex Data", Advances in Data Warehousing and Mining, IGI Publishing, Hershey, PA, USA, August 2009, 38-54 (In L. Bellatreche, Ed., Data Warehousing Design and Advanced Engineering Applications : Methods for Complex Construction).

[8] J. Darmont, O. Boussaid, **F. Bentayeb**, S. Rabaséda, Y. Zellouf, "Web multiform data structuring for warehousing", Multimedia Systems and Applications, Vol. 22, Kluwer Academic Publishers, 2003, 179-194.

Conférences internationales

[9] D. Boukraâ, O. Boussaid, **F. Bentayeb**, "Vertical Fragmentation of XML Data Warehouses using Frequent Path Setse", 13th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 11), Toulouse, France, August, 2011. To appear.

[10] A. Tanasescu, O. Boussaid, **F. Bentayeb**, "Preparing Complex Data for Warehousing", 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 05), Cairo, Egypt, January 2005 (Proceedings on CD).

[11] A. Tanasescu, O. Boussaid, **F. Bentayeb**, "Towards Complex Data Warehousing : A New Approach for Integrating and Modeling Complex data", 5th International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences (MCO 04), Metz, France, July 2004, 619-626.

[12] O. Boussaid, **F. Bentayeb**, A. Duffoux, F. Clerc, "Complex Data Integration based on a Multi-Agent System", 1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS 03), Prague, Czech Republic, September 2003 ; Proceedings LNAI, Vol. 2744, 201-212.

[13] O. Boussaid, **F. Bentayeb**, J. Darmont, "A Multi-Agent System-Based ETL Approach for Complex Data", 10th ISPE International Conference on Concurrent Engineering : Research and Applications (CE 03), Madeira, Portugal, July 2003, 49-52.

[14] J. Darmont, O. Boussaid, **F. Bentayeb**, "Warehousing Web Data", 4th International Conference on Information Integration and Web-based Applications and Services (iiWAS 02), Bandung, Indonesia, September 2002, 148-152.

Conférences nationales

[15] D. Boukraâ, O. Boussaïd, **F. Bentayeb**, "Opérateurs OLAP pour des cubes d'objets complexes : construction, visualisation et analyse", 6èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 10), Djerba, Tunisie, Juin 2010, 49Ű63 ; Revues des Nouvelles Technologies de l'Information, Cépaduès Editions, Toulouse.

[16] D. Midouni, J. Darmont, **F. Bentayeb**, "Approche de modélisation multidimensionnelle des données complexes : Application aux données médicales", 5èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 09), Montpellier, Juin 2009 ; Revues des Nouvelles Technologies de l'Information, Vol. B-5, Cépaduès Editions, Toulouse, 155-166.

[17] A. Duffoux, O. Boussaid, S. Lallich, **F. Bentayeb**, "Fouille dans la structure de documents XML", 4èmes Journées Francophones d'Extraction et de Gestion des

Connaissances (EGC 04), Clermont-Ferrand, Janvier 2004 ; Revue des Nouvelles Technologies de l'Information, Vol. 2, 519-524.

[18] F. Clerc, A. Duffoux, C. Rose, **F. Bentayeb**, O. Boussaid, "SMAIDoC : Un Système Multi-Agents pour l'Intégration des Données Complexes", XXXVèmes Journées de Statistique, Session spéciale Entreposage et Fouille de Données, Lyon, Juin 2003, 337-340.

Chapitre 8

Conclusion générale

8.1 Bilan et contributions

Dans ce mémoire, nous avons rapporté nos principaux travaux menés depuis 2001, dans un contexte local au laboratoire ERIC (encadrement d'étudiants en Master et en thèse), international (Algérie, Tunisie), et industriel (LCL-Le Crédit Lyonnais). Ces travaux ont été menés autour des thématiques liées aux entrepôts de données complexes centrés utilisateur. Plus précisément, nous avons abordé les thèmes de la personnalisation dans les entrepôts de données complexes et de l'exploitation de la sémantique dans le processus d'entreposage que ce soit par le biais des techniques de fouille de données ou par le biais des ontologies.

Nous nous sommes plus particulièrement intéressée à l'utilisateur dans le système décisionnel. Nous avons cherché à lui offrir des méthodes et outils efficaces et des plus naturels possible tant au niveau de l'adaptation de l'entrepôt selon ses besoins, qu'au niveau de la recommandation de requêtes décisionnelles. Nous sommes arrivée à la conclusion qu'un "bon" entrepôt de données centré utilisateur doit s'affranchir de certaines contraintes liées aux modèles d'entrepôts de données classiques.

En s'appuyant sur l'intuition que des connaissances sur les données entreposées et leur usage (requêtes) peuvent contribuer à aider l'utilisateur dans son exploration et sa navigation dans les données, nous avons proposé une première approche de personnalisation basée sur les connaissances explicites des utilisateurs. Ces connaissances sont intégrées dans l'entrepôt, sous forme de règles d'agrégation qui sont transformées en axes d'analyse nouveaux, en empruntant le concept d'évolution de schéma. Dans ces travaux, nous avons relâché la contrainte du schéma fixe de l'entrepôt en permettant d'ajouter ou de supprimer un niveau de hiérarchie dans une dimension. Bien entendu, ces mises à jour sont suivies de vérification de la cohérence à la fois dans les données et dans la structure. Ces travaux s'inscrivent dans le courant des systèmes adapatatifs puisque l'entrepôt s'adapte aux

besoins de l'utilisateur.

D'autre part, pour optimiser les accès aux données, nous avons proposé une approche de fouille de données en ligne capable de concurrencer les méthodes de fouille opérant en mémoire avec l'avantage en plus de traiter des bases sans limitation de taille. De plus, la fouille en ligne que nous avons proposée étend les possibilités d'analyse des SGBD, de l'OLAP vers la structuration, l'explication et la prédiction. Par exemple, en utilisant la méthode des k-means, nous avons pu découvrir de nouvelles structures naturelles porteuses de sémantique. Celles-ci peuvent constituer de nouveaux niveaux de hiérarchie de dimension permettant d'élaborer des scénarios d'analyse pertinents non prévus par le schéma de l'entrepôt initial. Ces travaux s'inscrivent dans le courant des systèmes de recommandation.

Nous avons également proposé une approche d'analyse en ligne à la demande basée sur un dispositif d'intégration de données en utilisant un système de médiation. Les sources de données sont décrites par des ontologies locales et le médiateur est décrit par une ontologie globale obtenue par fusion des ontologies locales. L'avantage d'une telle architecture décisionnelle est double. D'abord, l'utilisation des ontologies permet dans un système de médiation de résoudre à la fois l'hétérogénéité structurelle et sémantique des données. D'autre part, cette architecture permet un accès en temps réel aux données des sources locales pour construire des cubes de données à la demande. Nous avons fait le choix de ne développer, dans ce document, que l'algorithme de fusion des ontologies qui utilise la méthode de classification *CAH* et le mécanisme d'inférence *OWL*. Il s'agit d'extraire les classes de concepts les plus similaires à partir de plusieurs ontologies locales, puis de trouver leur subsumant, afin de construire l'ontologie globale. Nous avons également défini une mesure de similarité sémantique adaptée. Le calcul de la similarité entre deux concepts est basé sur la comparaison de leurs termes, de leurs attributs et de leurs relations avec leur voisinage respectif. L'approche globale de construction des cubes à la demande est développée dans la thèse de N. Maïz [Mai10].

Nous avons ensuite étendu nos travaux pour la prise en compte de données complexes. C'est un sujet auquel nous nous sommes particulièrement intéressée depuis plusieurs années. Nous sommes arrivée à la conclusion qu'il n'y a pas de modèle multidimensionnel de données complexes unique qui peut répondre à tous les types de données complexes et à tous les types d'analyse souhaités. Par ailleurs, lorsqu'il faut intégrer le volet utilisateur qui constitue à lui seul un verrou scientifique à part entière, plusieurs concepts des entrepôts de données sont alors à revoir comme nous avons pu le démontrer dans nos travaux sur la personnalisation.

Dans ce contexte, nous avons proposé une démarche globale d'intégration, de modélisation et d'analyse en ligne de données complexes. Le processus d'intégration que nous avons présenté a pour finalité d'aider à la modélisation multidimensionnelle de données

complexes provenant de sources diverses dans un format unifié, XML, et de les stocker dans une base de données relationnelle. Pour mieux préparer les données à l'analyse, le recours aux techniques de fouille de données s'est avéré nécessaire puisqu'il a permis d'identifier les faits pertinents à analyser en fonction des besoins de l'utilisateur. Pour assurer une bonne intégration des données complexes, nous avons proposé une approche d'ETL automatique basée sur les systèmes multiagents qui apporte une flexibilité à notre démarche d'entreposage de données complexes.

Plus récemment, nous avons proposé un modèle multidimensionnel d'objets complexes centré utilisateur en utilisant le paradigme objet qui permet de décrire à la fois les données et leurs relations sémantiques. De façon plus précise, nous nous sommes intéressée à la modélisation et à l'exploitation des liens sémantiques intra- et inter-objets complexes. Notre contribution principale dans ce domaine porte sur le choix d'une architecture à deux niveaux d'abstraction. Le niveau diagramme de classes qui définit les données, et le niveau diagramme de packages qui définit les objets complexes ainsi que leurs relations externes (liens inter-objets). Pour analyser les objets complexes, nous avons défini un opérateur de projection cubique qui permet à l'utilisateur de choisir ses objets, de leur affecter le rôle de sujet ou d'axe d'observation, et enfin de créer le cube d'objets complexes personnalisé.

Pour terminer, nous avons toujours eu le souci de démontrer l'efficacité de nos propositions en mettant en œuvre un processus d'expérimentation qui a impliqué à chaque fois leur implémentation sur des systèmes existants. Chacune de nos contributions a donné lieu au développement d'un prototype logiciel. Notre contribution concernant la personnalisation dans les entrepôts de données a été appliquée en partie dans le cadre de partenariat avec l'entreprise LCL-Le Crédit Lyonnais dans le cadre d'une convention CIFRE.

8.2 Projet de recherche

Les entrepôts de données constituent un terrain fertile pour effectuer de nouvelles recherches. Aussi, les perspectives associées à ce domaine de recherche sont nombreuses. Certaines font d'ores et déjà partie de nos prospections. D'autres sont des perspectives à plus long terme.

8.2.1 Encore plus de sémantique dans les entrepôts de données...

Depuis quelques années, nous explorons le domaine de la personnalisation dans les entrepôts de données complexes. Dans le cadre de nos travaux, nous avons utilisé le concept d'évolution de schéma pour adapter l'entrepôt de données aux nouveaux besoins de l'utilisateur. Nous avons cependant remarqué que la plupart des résultats d'analyse obtenus sont liés au contexte d'utilisation. L'une de nos premières préoccupations dans ce domaine

est d'intégrer la notion de contexte dans les entrepôts de données afin de l'exploiter dans la phase d'interprétation des résultats. Le contexte peut alors être intégré soit dans le profil de l'utilisateur soit dans le modèle même de l'entrepôt sous forme de dimensions contextuelles. En se basant sur notre approche d'évolution de schéma pour la personnalisation des analyses, des travaux proposant des dimensions contextuelles ont été proposés [PFLP10].

Par ailleurs, nous avons eu recours à la fouille de données pour la recommandation d'axes d'analyse pertinents. Dans la continuité de ces travaux et toujours avec le souci de remettre l'utilisateur au sein du système décisionnel, nous menons actuellement des recherches pour l'exploitation de fichiers logs de requêtes en utilisant les itemsets fréquents pour la recommandation interactive de requêtes décisionnelles. L'objectif étant de proposer un système de recommandation de requêtes décisionnelles interactif basé sur l'usage de l'utilisateur. De manière générale, notre idée est d'enrichir le profil utilisateur avec des données sémantiques. Ces dernières peuvent être définies explicitement par l'utilisateur ou peuvent être extraites à partir de ses usages (requêtes), de ses différents contextes d'utilisation, etc.

Dans le cadre des entrepôts d'objets complexes, plusieurs pistes de recherche existent. Tout d'abord, définir de nouveaux opérateurs OLAP pour naviger dans le cube d'objets complexes. Ensuite, créer des métriques (mesures) adaptées au type d'objets complexes entreposés. Par ailleurs, pour faire évoluer l'OLAP vers une analyse sémantique des données, nous orientons nos recherches vers une combinaison des principes de l'OLAP, de la fouille de données et de la recherche d'information : l'OLAP sémantique.

Ces travaux à venir font l'objet de la thèse de R. Khémiri (thèse en co-tutelle avec la Tunisie financée en partie par l'Institut Français de Coopération -IFC-) que nous co-encadrons.

8.2.2 Evaluation de la qualité d'une analyse personnalisée

La personnalisation et la qualité de l'information constituent un enjeu majeur pour l'industrie informatique. Que ce soit dans le contexte des systèmes d'information d'entreprise, du commerce électronique, de l'accès au savoir et aux connaissances ou même des loisirs, la pertinence de l'information délivrée et son adaptation aux usages et préférences des clients constituent des facteurs clés du succès ou du rejet de ces systèmes. C'est dans ce contexte que nous soulignons les enjeux de l'évaluation qualitative de la personnalisation dans les systèmes décisionnels. En effet, l'un des facteurs clé de la personnalisation est la qualité des informations délivrées. On peut par exemple étendre la définition du profil pour donner la possibilité aux utilisateurs de décrire leurs préférences sur la qualité du processus d'exécution des requêtes et sur la qualité des données délivrées. Une des métriques

standard de mesure de la qualité d'un processus est le temps de réponse qui est le temps nécessaire au processus d'exécuter la requête et d'afficher le résultat. Concernant la qualité des données délivrées, il s'agit d'évaluer la qualité d'une analyse proposée en réponse à un besoin, notamment la proximité des résultats calculés avec les résultats attendus. Dans ce cas, il faut définir des critères de qualité adaptés. Ces derniers peuvent être inspirés des travaux menés en qualité des données et qualité des processus mais peuvent être définis également par les utilisateurs du système eux-mêmes. C'est dans ce cadre que nous envisageons d'engager nos recherches à venir pour améliorer le processus de personnalisation dans les systèmes décisionnels. Nous avons d'ores et déjà initié des travaux pour l'évaluation de la qualité des niveaux de hiérarchie obtenus par classification automatique dans le cadre de nos travaux sur la recommandation de nouveaux axes d'analyse à l'utilisateur.

Le but étant de définir un modèle d'évaluation de la qualité qui inclut tous les aspects de la personnalisation. Ainsi, nous sommes personnellement impliquée dans un projet de l'Agence Nationale de la Recherche (ANR) (Contenus Numériques et Interactions, édition 2011) avec des collègues de l'Université de Toulouse et de Tours (Antenne de Blois) qui porte sur *la qualité de l'analyse en ligne centrée multi-utilisateurs*.

8.2.3 Entrepôts de données texte

Dans la continuité de nos travaux sur les entrepôts de données complexes, nous avons initié des travaux de recherche visant à proposer un modèle d'entrepôt de données texte. Dans ce contexte, il faut redéfinir la notion de mesure, et étendre la notion de hiérarchie pour mieux prendre en compte les liens structurels et sémantiques entre termes. Récemment, une structure de cubes de données texte a été proposée en empruntant les principales mesures liées à la recherche d'information. L'idée est d'étendre ces travaux à d'autres mesures pouvant être le résultat d'une combinaison entre la RI, la fouille de données et l'OLAP. Notre objectif consiste également à construire des entrepôts de données texte centrés utilisateur. Autrement dit, laisser à l'utilisateur la possibilité de fixer le rôle des données texte (axes ou mesures) au moment de l'analyse. Cela donne plus de flexibilité au modèle et répond plus aux attentes des utilisateurs.

Nous souhaitons également étendre nos travaux sur la recommandation de requêtes à la combinaison de la RI avec l'OLAP pour mieux exploiter les données texte. Nous pouvons par exemple emprunter la notion de profil de document qui correspond à la description d'un document qui est souvent représentée par une liste de mots-clés pondérés décrivant le contenu sémantique du document. Les mots-clés et leurs poids sont obtenus en général par une opération d'indexation. Nous pouvons rajouter à cela des annotations qualitatives qui peuvent aider à mieux cibler les analyses des données texte au travers de leur contenu sémantique. En rapprochant les données texte entreposées, leurs annotations ainsi que leur

profil avec les centres d'intérêts des utilisateurs regroupés dans leur profil, nous pouvons réduire l'espace de recherche dans l'entrepôt et garantir une meilleure adaptation des réponses obtenues.

Par ailleurs, nous nous intéressons à l'analyse de documents texte au format XML. Il s'agit par exemple de créer des opérateurs d'agrégation basés sur la fouille de données et/ou des mesures de similarité à définir. La comparaison entre les documents XML peut porter à la fois sur leur structure et sur leur contenu.

Ces travaux à venir font l'objet de la thèse de R. Aknouche (thèse financée par le gouvernement Algérien) que nous co-encadrons.

8.2.4 Amélioration de la visualisation dans les cubes OLAP

Dans un système décisionnel, la composante visuelle est importante pour l'analyse OLAP. L'objectif étant de construire un espace de représentation se prêtant mieux à l'analyse et facilitant la navigation dans les données pour l'utilisateur.

Dans ce domaine, peu de travaux existent même si des tentatives d'amélioration de la présentation des résultats à l'utilisateur émergent depuis peu. Ces travaux prennent en compte les préférences utilisateurs, les contraintes d'affichage, le contexte d'utilisation, l'éparsité des données, etc. Ce dernier point est important car les cubes OLAP présentent souvent des données éparses. Visuellement, l'information est éparpillée dans l'espace de représentation des données de façon aléatoire ; ce qui rend difficile l'exploitation du cube. En revanche, si les cellules pleines sont regroupées dans un même espace du cube, ce dernier offrirait des possibilités d'analyse et de comparaison des données plus aisées et plus rapides pour l'utilisateur et lui faciliterait l'interprétation des résultats. Dans ce contexte, il existe des travaux qui utilisent des méthodes factorielles qui donnent des résultats satisfaisants dans le cas où le cube présente une grande éparsité. D'autres méthodes utilisent des algorithmes génétiques qui améliorent considérablement les résultats.

Dans ce projet, nous avons d'ores et déjà entamé des travaux pour l'amélioration de la visualisation de cubes OLAP 2D et 3D en utilisant la sériation [1]. Les premiers résultats que nous avons obtenus nous encouragent à poursuivre dans cette voie. Notre objectif à présent est de combiner la méthode de sériation avec les préférences utilisateurs afin de satisfaire les centres d'intérêt de ces derniers tout en atténuant sigificativement l'effet négatif de l'éparsité du cube OLAP.

1. ré-ordonnancement des lignes et des colonnes d'un ensemble de données afin de les énumérer dans un ordre approprié (Bertin, 1983).

8.2.5 Analyse de données de transcriptions

CLAPI (Corpus de Langue Parlée en Interaction) est une base de données en ligne qui héberge des corpus enregistrés depuis le début des années 80. Elle a été mise en œuvre au cours d'une collaboration entre les laboratoires ICAR [2] - Interactions, Corpus, Apprentissages, Représentations - (Lyon 2 - ENS-LSH) et ERIC (Lyon 2) entre 2002 et 2005 dans le cadre d'une Action Concertée Incitative "Terrains, Techniques, Théories" (ACI-TTT) dont l'objectif était d'assurer la constitution, la gestion, la valorisation et la mise en ligne de bases de données complexes (audio, vidéo, textes annotés) rassemblant des corpus linguistiques oraux. Nous avons participé très activement à cette ACI et contribué à la mise en place de la base CLAPI.

En 2009, nous avons porté plus personnellement le projet "Bonus Qualité Recherche" (BQR) dans le cadre d'un appel d'offres de projets internes Lyon 2. C'est un projet interdisciplinaire (informatique-linguistique) qui vise à identifier automatiquement les phénomènes complexes qui composent une interaction en s'appuyant sur l'expertise des chercheurs en linguistique et en informatique des laboratoires ICAR et ERIC, respectivement. L'objectif est d'identifier des séquences distinctes au cours d'une interaction : ouverture et clôture des différents types d'interactions en français, remerciements, ouverture / développement / sortie de conflit, plaisanteries familières, séquences de plaintes, de confidences, séquences émotionnelles, diverses phases des interactions commerciales quotidiennes... En tant que porteur du projet, nous avons personnellement animé le groupe de travail composé de collègues informaticiens et linguistes afin de mieux comprendre la problématique et dégager des pistes de recherche sérieuses.

A l'issue de ce projet interne Lyon 2 et grâce aux premiers résultats encourageants que nous avons obtenus [HBB+11], nous avons comme objectif à moyen terme de monter un projet d'ANR sur cette thématique. Notre démarche de recherche porte principalement sur deux volets :

(1) Analyse des données des transcriptions d'interactions disponibles au format XML en utilisant l'analyse OLAP. Une première étude consiste à considérer la transcription comme le fait à observer dans un schéma multidimensionnel sur lequel on définit des contextes d'analyse. Par exemple, on peut observer des événements (occurrence de token, pause, chevauchement de parole, etc.) en fonction de diverses dimensions comme le locuteur qui en est à l'origine, le moment où l'événement se produit, etc. L'originalité de ce travail est de traduire les transcriptions dans des schémas multidimensionnels décrivant les spécificités de l'oral afin de les préparer à l'analyse. Notre choix de corpus s'est porté sur une séance d'une commission de conciliation cherchant à régler à l'amiable les conflits entre locataires et propriétaire, afin d'éviter le recours au tribunal, négociation par essence conflictuelle.

2. http ://icar.univ-lyon2.fr/

(2) Utilisation des outils du Traitement Automatique du Langage (TAL). Etudier le sens d'une production verbale (expression, phrase) dans les séquences d'interaction est un challenge intéressant, qui ouvre des perspectives importantes. Une telle étude permettrait une annotation sémantique du corpus en proposant de nouvelles balises décrivant des phénomènes linguistiques complexes. On appelle séquence toute action sémantique complète produite autour d'un matériel verbal (verbe, phrase, texte) : séquence émotionnelle, séquence argumentative, question-réponse, proposition-acceptation. L'identification des séquences n'est pas une tâche facile. Une méthode de détection de séquences semi-automatique impliquant l'utilisateur (expert) constituerait une voie de recherche plus appropriée.

8.2.6 Vers une nouvelle génération de SID

L'évolution des entrepôts et de l'analyse en ligne des données complexes ne peut se réaliser, à notre avis, que par le couplage avec d'autres technologies telles que la fouille de données, la recherche d'information, les ontologies, les systèmes d'annotations... qui permettent d'intégrer la sémantique dans tout le processus d'entreposage. Ceci implique de définir de nouveaux modèles multidimensionnels et de proposer une analyse OLAP plus élaborée facilitant ainsi le processus de personnalisation. Notre objectif est de poursuivre nos travaux dans cette voie et de faire évoluer les SID vers une nouvelle génération de systèmes décisionnels centrés utilisateur, supports de l'OLAP sémantique.

Le domaine de la personnalisation est assez récent dans le domaine des entrepôts de données mais existe depuis longtemps dans les domaines tels que la recherche d'information et les bases de données. Consciente de l'importance de la recherche fondamentale en informatique et forte des résultats personnellement déjà obtenus dans le domaine de la personnalisation, nous avons pour objectif de promouvoir ce domaine d'étude au sein de la communauté des entrepôts de données et d'en faire un axe de recherche important.

Nous allons également continuer d'entretenir et développer nos partenariats pour réaliser des applications, qui deviennent souvent elles-mêmes le terreau de nouvelles recherches. C'est en ce sens que nous continuons à collaborer avec d'autres laboratoires de recherche en vue de déposer auprès de l'ANR des projets pluri-disciplinaires.

Bibliographie

[Aas97] K. Aas. A survey on personalised information filtering systems for the world wide web. Technical report, Technical report, Norwegian Computing Center, 1997.

[ACWP01] J. Akoka, I. Comyn-Wattiau, and N. Prat. Dimension Hierarchies Design from UML Generalizations and Aggregations. In *XXth International Conference on Conceptual Modeling (ER 01), Yokohama, Japan*, volume 2224 of *LNCS*, pages 442–455. Springer, 2001.

[AMSea96] R. Agrawal, H. Mannila, R. Srikant, and et al. Fast Discovery of Association Rules. In *Advances in Kowledge Discovery and Data Mining*, pages 307–328, 1996.

[ASS00] A. Abelló, J. Samos, and F. Saltor. Benefits of an Object-Oriented Multidimensional Data Model. In *Objects and Databases, International Symposium, Sophia Antipolis, France, June 13, 2000, Proceedings*, pages 141–152, 2000.

[AT08] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *RecSys*, pages 335–336, 2008.

[AW00] R. Agrawal and E. L. Wimmers. A Framework for Expressing and Combining Preferences. In *SIGMOD Conference*, pages 297–306, 2000.

[BAGC04] D. Bourigault, N. Aussenac-Gilles, and J. Charlet. Construction de ressources terminologiques ou ontologiques à partir de textes. Un cadre unificateur pour trois études de cas. *Revue d'Intelligence Artificielle*, 18(1) :87–110, 2004.

[BBB10a] D. Boukraâ, O. Boussaid, and F. Bentayeb. Olap operators for complex object data cubes. In *ADBIS*, pages 103–116, 2010.

[BBB10b] D. Boukraâ, O. Boussaid, and F. Bentayeb. Opérateurs OLAP pour des cubes d'objets complexes : construction, visualisation et analyse. In *6èmes Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 10), Djerba, Tunisie.*, volume B-1 of *Revue des Nouvelles Technologies de l'Information*, pages 49–63. Cépaduès Editions, 2010.

[BBB11] D. Boukraâ, O. Boussaid, and F. Bentayeb. Vertical Fragmentation of XML Data Warehouses using Frequent Path Sets. In *DaWaK*, 2011. To appear.

[BBBL10] D. Boukraâ, O. Boussaid, F. Bentayeb, and S. Loudcher. OLAP Operators For A Complex Object-Based Multidimensional Model. *Journal of Data Mining and Business Intelligence (DMBI)*, pages 34–46, 2010.

[BBC+00] D. Beneventano, S. Bergamaschi, S. Castano, A. Corni, R. Guidetti, G. Malvezzi, M. Melchiori, and M. Vincini. Information Integration : The MOMIS Project Demonstration. In *The VLDB Journal*, pages 611–614, 2000.

[BBD03] O. Boussaid, F. Bentayeb, and J. Darmont. A multi-agent system-based ETL approach for complex data. In *Proceedings of the 10th ISPE International Conference on Concurrent Engineering (ISPE CE 2003), July 26-30, 2003, Madeira, Portugal*, pages 49–52, 2003.

[BBDC03] O. Boussaid, F. Bentayeb, A. Duffoux, and F. Clerc. Complex Data Integration Based on a Multi-agent System. In *HoloMAS*, pages 201–212, 2003.

[BBDR03] O. Boussaid, F. Bentayeb, J. Darmont, and S. Rabaséda. Vers l'entreposage des données complexes. Structuration, intégration et analyse. *Ingénierie des Systèmes d'Information*, 8(5-6) :79–107, 2003.

[BBF+09] F. Bentayeb, O. Boussaid, C. Favre, F. Ravat, and O. Teste. Personnalisation dans les entrepôts de données : bilan et perspectives. In *Journées Francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA), Montpellier*, RNTI, pages 7–22, 2009.

[BBGV01] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. The MOMIS Approach to Information Integration. In *ICEIS (1)*, pages 194–198, 2001.

[BD02] F. Bentayeb and J. Darmont. Decision tree modeling with relational views. In *XIIIth International Symposium on Methodologies for Intelligent Systems (ISMIS 02), France*, volume 2366 of *LNAI*, pages 423–431, Heidelberg, Germany, 2002. Springer Verlag.

[BDBR08] O. Boussaid, J. Darmont, F. Bentayeb, and S. Loudcher Rabaséda. Warehousing complex data from the web. *Int. J. Web Eng. Technol.*, 4(4) :408–433, 2008.

[BDFU07] F. Bentayeb, J. Darmont, C. Favre, and C. Udréa. Efficient On-Line Mining of Large Databases. *International Journal of Business Information Systems*, 2(3) :328–350, 2007.

[BDU04] F. Bentayeb, J. Darmont, and C. Udrea. Efficient Integration of Data Mining Techniques in Database Management Systems. In *8th International Database Engineering and Applications Symposium (IDEAS 04), Coimbra, Portugal*, pages 59–67. IEEE Computer Society, July 2004.

[BEK+04] B. Bebel, J. Eder, C. Koncilia, T. Morzy, and R. Wrembel. Creation and management of versions in multiversion data warehouse. In *SAC*, pages 717–723, 2004.

[Bel02] Z. Bellahsene. Schema Evolution in Data Warehouses. *Knowl. Inf. Syst.*, 4(3) :283–304, 2002.

[Ben08] F. Bentayeb. K-Means Based Approach for OLAP Dimension Updates. In *ICEIS (1)*, pages 531–534, 2008.

[BF09] F. Bentayeb and C. Favre. RoK : Roll-Up with the K-Means Clustering Method for Recommending OLAP Queries. In *DEXA*, volume 5690 of *LNCS*, pages 501–515. Springer, 2009.

[BFB08] F. Bentayeb, C. Favre, and O. Boussaïd. A User-driven Data Warehouse Evolution Approach for Concurrent Personalized Analysis Needs Integrated Computer-Aided Engineering. *Journal of Integrated Computer-Aided Engineering*, 15(1) :21–36, 2008.

[BGM+05] L. Bellatreche, A. Giacometti, P. Marcel, H. Mouloudi, and D. Laurent. A personalization framework for OLAP queries. In *DOLAP*, pages 9–18, 2005.

[BGMM06] L. Bellatreche, A. Giacometti, P. Marcel, and H. Mouloudi. Personalization of MDX Queries. In *XXIIèmes journées Bases de Données Avancées (BDA 06), Lille, France*, 2006.

[BK05] M. Bouzeghoub and D. Kostadinov. Personnalisation de l'information : aperçu de l'état de l'art et définition d'un modèle flexible de profils. In *In CORIA*, pages 201–218, 2005.

[BMB09] D. Boukraâ, R. Ben Messaoud, and O. Boussaïd. *Open and novel issues in XML database applications : future directions and advanced technologies*, chapter Modeling XML Warehouses for Complex Data : The New Issues, pages 108–135. IGI Global, Information Science Reference, USA/UK, 2009.

[BMBT03] M. Body, M. Miquel, Y. Bédard, and A. Tchounikine. Handling Evolutions in Multidimensional Structures. In *ICDE*, pages 581–, 2003.

[BMM+11] F. Bentayeb, N. Maiz, H. Mahboubi, C. Favre, S. Loudcher, N. Harbi, O. Boussaid, and J. Darmont. *Innovative Approaches for efficiently Warehousing Complex Data from the Web*. Business Intelligence Applications and the Web : Models, Systems and Technologies. IGI Global, Hershey, PA, USA, 2011. M.E. Zorrilla, J.N. Mazón, Ó. Ferrández, I. Garrigós, F. Daniel, J. Trujillo, Eds. ; to appear.

[BRS00] K. Bradley, R. Rafter, and B. Smyth. Case-Based User Profiling for Content Personalisation. In *AH*, pages 62–72, 2000.

[BSH99] M. Blaschka, C. Sapia, and G. Höfling. On Schema Evolution in Multidimensional Databases. In *DaWaK*, pages 153–164, 1999.

[BSSJ98] R. Bliujute, S. Saltenis, G. Slivinskas, and C. Jensen. Systematic Change Management in Dimensional Data Warehousing. In *IIIrd International Baltic Workshop on Databases and Information Systems, Riga, Latvia*, pages 27–41, 1998.

[BTBD07] O. Boussaid, A. Tanasescu, F. Bentayeb, and J. Darmont. Integration and Dimensional Modelling Approaches for Complex Data Warehousing. *Journal of Global Optimization*, 37(4) :571–591, April 2007.

[BTM06] S. Bimonte, A. Tchounikine, and M. Miquel. GeoCube, a Multidimensional Model and Navigation Operators Handling Complex Measures : Application in Spatial OLAP. In Tatyana M. Yakhno and Erich J. Neuhold, editors, *Proceedings of the 4th International Conference on Advances in Information Systems (ADVIS'06), Izmir, Turkey*, volume 4243 of *Lecture Notes in Computer Science*, pages 100–109. Springer, 2006.

[Cas08] S. Castagnos. *Modélisation de comportements et apprentissage stochastique non supervisé de stratégies d'interactions sociales au sein de systèmes temps réel de recherche et d'accès à l'information*. PhD thesis, Université Nancy 2, 2008.

[CCRT07] G. Cabanac, M. Chevalier, F. Ravat, and O. Teste. An Annotation Management System for Multidimensional Databases. In *IXth International Conference on Data Warehousing and Knowledge Discovery (DaWaK 07), Regensburg, Germany*, volume 4654 of *LNCS*, pages 89–98. Springer, 2007.

[CD97] S. Chaudhuri and U. Dayal. An Overview of Data Warehousing and OLAP Technology. *SIGMOD Rec.*, 26(1) :65–74, 1997.

[CEP09] G. Chatzopoulou, M. Eirinaki, and N. Polyzotis. Query Recommendations for Interactive Database Exploration. In *SSDBM*, pages 3–18, 2009.

[CGFZ03] M. Cherniack, E. F. Galvez, M. J. Franklin, and S. B. Zdonik. Profile-Driven Cache Management. In *XIXth International Conference on Data Engineering (ICDE 03), Bangalore, India*, pages 645–656. IEEE Computer Society, 2003.

[CGMH+94] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom. The TSIMMIS Project : Integration of heterogeneous information sources. In *16th Meeting of the Information Processing Society of Japan*, pages 7–18, Tokyo, Japan, 1994.

[Cha98] S. Chaudhuri. Data mining and database systems : Where is the intersection ? *Data Engineering Bulletin*, 21(1) :4–8, 1998.

[Cho03] J. Chomicki. Preference formulas in relational queries. *ACM Trans. Database Syst.*, 28(4) :427–466, 2003.

[CMS03] G. H. Cheng, S. E. Madnick, and M. D. Siege. Semantic interoperability through context interchange : representing and reasoning about data conflicts in heterogeneous and autonomous systems. Working papers, Massachusetts Institute of Technology (MIT), Sloan School of Management, April 2003.

[Cod93] E. F. Codd. Providing OLAP (On-Line Analytical Processing) to User-Analysts : an IT mandate. Technical report, E.F. Codd and Associates, 1993.

[CR00] J-H. Chauchat and R. Rakotomalala. A new sampling strategy for building decision trees from large databases. In *7th Conference of the International Federation of Classification Societies (IFCS 00), Belgium*, pages 199–204, 2000.

[CRST06] Bee-Chung Chen, Raghu Ramakrishnan, Jude W. Shavlik, and Pradeep Tamma. Bellwether analysis : Predicting global aggregates from local regions. In *VLDB*, pages 655–666, 2006.

[CT98] L. Cabibbo and R. Torlone. A Logical Approach to Multidimensional Databases. In *VIth International Conference on Extending Database Technology (EDBT 98), Valencia, Spain*, volume 1377 of *LNCS*, pages 183–197. Springer, 1998.

[DBB02] J. Darmont, O. Boussaid, and F. Bentayeb. Warehousing Web Data. In *4th International Conference on Information Integration and Web-based Applications and Services (iiWAS 02), Bandung, Indonesia*, pages 148–152. SCS Europe Bvba, September 2002.

[DBLB04] A. Duffoux, O. Boussaid, S. Lallich, and F. Bentayeb. Fouille dans la structure de documents XML. In *EGC*, pages 519–524, 2004.

[DJ07] C. Domshlak and T. Joachims. Efficient and non-parametric reasoning over user preferences. *User Model. User-Adapt. Interact.*, 17(1-2) :41–69, 2007.

[DLST03] M. Dash, H. Liu, P. Scheuermann, and K. L. Tan. Fast hierarchical clustering and its validation. *Data Knowl. Eng.*, 44(1) :109–138, 2003.

[DS99] B. Dunkel and N. Soparkar. Data organization and access for efficient data mining. In *ICDE*, pages 522–529, 1999.

[EGV05] J. Euzenat, P. Guégan, and P. Valtchev. OLA in the OAEI 2005 Alignment Contest. In *Integrating Ontologies*, pages 97–102, 2005.

[EV01] M. Minuto Espil and A. A. Vaisman. Efficient Intensional Redefinition of Aggregation Hierarchies in Multidimensional Databases. In *IVth ACM International Workshop on Data Warehousing and OLAP (DOLAP 01), Atlanta, Georgia, USA*, pages 1–8. ACM Press, 2001.

[Fav07] C. Favre. *Évolution de schémas dans les entrepôts de données : mise à jour de hiérarchies de dimension pour la personnalisation des analyses*. PhD thesis, Université Lumière Lyon 2, Décembre 2007.

[FB05a] C. Favre and F. Bentayeb. Bitmap Index-based Decision Trees. In *XVth International Symposium on Methodologies for Intelligent Systems (ISMIS 05), New York, USA*, volume 3488 of *LNAI*, pages 65–73, Heidelberg, Germany, May 2005. Springer.

[FB05b] C. Favre and F. Bentayeb. Intégration efficace des arbres de décision dans les SGBD : utilisation des index bitmap. In *Vèmes Journées d'Extraction et de Gestion des Connaissances (EGC 05), Paris, France*, volume E-3 of *Revue des Nouvelles Technologies de l'Information*, pages 319–330. Cépaduès Editions, 2005.

[FBB06a] C. Favre, F. Bentayeb, and O. Boussaïd. A Rule-based Data Warehouse Model. In *23rd British National Conference on Databases (BNCOD 2006), Belfast, Northern Ireland*, volume 4042 of *LNCS*, pages 274–277, Heidelberg, Germany, July 2006. Springer.

[FBB06b] C. Favre, F. Bentayeb, and O. Boussaid. A Knowledge-driven Data Warehouse Model for Analysis Evolution. In *XIIIth ISPE International Conference on Concurrent Engineering : Research and Applications (CE 06), Antibes, France*, volume 143 of *Frontiers in Artificial Intelligence and Applications*, pages 271–278. IOS Press, 2006.

[FBB07a] C. Favre, F. Bentayeb, and O. Boussaïd. Dimension Hierarchy Updates in Data Warehouses : a User-driven Approach. In *9th International Conference on Enterprise Information ystems (ICEIS 07), Funchal, Madeira, Portugal*, pages 206 – 211, June 2007.

[FBB07b] C. Favre, F. Bentayeb, and O. Boussaid. *A Survey of Data Warehouse Model Evolution*, pages 129–136. Encyclopedia of Database Technologies and Applications, Second Edition. Idea Group Publishing, 2007.

[FBMB02] J. T. Fernández-Breis and R. Martínez-Béjar. A cooperative framework for integrating ontologies. *International Journal of Human-Computer Studies*, 56(6) :665–720, 2002.

[For65] EW. Forgy. Cluster analysis of multivariate data : efficiency versus interpretability of classification. In *Biometrics num 21*, pages 768–780, 1965.

[GH99] J. E. Gilbert and C.a Y. Han. Arthur : Adapting Instruction to Accommodate Learning Style. In *WebNet*, pages 433–438, 1999.

[GK97] R. M. Genesereth and A. M. Keller. Infomaster : an information integration system. pages 539–542, 1997.

[GKMV09] L. Gómez, B. Kuijpers, B. Moelans, and A. Vaisman. A Survey of Spatio-Temporal OLAP. *International Journal of Data Warehousing and Mining*, 5(3) :28–55, 2009.

[GLR00] F. Goasdoué, V. Lattès, and M-C. Rousset. The Use of CARIN Language and Algorithms for Information Integration : The PICSEL System. *International Journal of Cooperative Information Systems*, 9(4) :383–401, 2000.

[GLRV06] M. Golfarelli, J. Lechtenbörger, S. Rizzi, and G. Vossen. Schema versioning in data warehouses : Enabling cross-version querying via schema augmentation. *Data Knowl. Eng.*, 59(2) :435–459, 2006.

[GMN08] A. Giacometti, P. Marcel, and E. Negre. A framework for recommending OLAP queries. In *DOLAP*, pages 73–80, 2008.

[GMN09] A. Giacometti, P. Marcel, and E. Negre. Recommending Multidimensional Queries. In *DaWaK*, pages 453–466, 2009.

[GMR98] M. Golfarelli, D. Maio, and S. Rizzi. The Dimensional Fact Model : A Conceptual Model for Data Warehouses. *International Journal of Cooperative Information Systems*, 7(2-3) :215–247, 1998.

[GNOT92] David Goldberg, David A. Nichols, Brian M. Oki, and Douglas B. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Commun. ACM*, 35(12) :61–70, 1992.

[GR09] M. Golfarelli and S. Rizzi. Expressing OLAP Preferences. In *SSDBM*, pages 83–91, 2009.

[GRG98] J. Gehrke, R. Ramakrishnan, and V. Ganti. RainForest - A Framework for Fast Decision Tree Construction of Large Datasets. In *24th International Conference on Very Large Data Bases (VLDB 98), USA*, pages 416–427. Morgan Kaufmann, 1998.

[GRG00] J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest - a framework for fast decision tree construction of large datasets. *Data Mining and Knowledge Discovery*, 4(2/3) :127–162, 2000.

[Gru95] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing ? *Int. J. Hum.-Comput. Stud.*, 43(5-6) :907–928, 1995.

[GRV01] M. Golfarelli, S. Rizzi, and B. Vrdoljak. Data Warehouse Design from XML Sources. In *Proceedings of teh 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001), Atlanta, Georgia, USA*, 2001.

[GS02] I. Geist and K. U. Sattler. Towards data mining operators in database systems : Algebra and implementation. 2nd International Workshop on Databases, Documents, and Information Fusion (DBFusion 2002) - Information Integration and Mining in Databases and on the Web, 2002.

[GSY04] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match : an Algorithm and an Implementation of Semantic Matching. In *ESWS'04 : Proceedings of the First European Semantic Web Symposium*, pages 61–75, 10-12 May 2004.

[Ham50] R.W. Hamming. Error Detecting and Error Correcting Codes. *The Bell System technical journal*, XXVI(2) :147Ű160, 1950.

[HBB+11] K. Hajlaoui, R. Boujelben, F. Bentayeb, C. Etienne, and O. Boussaid. Fouille de l'oral tel qu'il est parlé. In *Conférence nationale en Terminologie & Ontologie : Théories et applications (TOTh 2011), Annecy, France*, 2011.

[HFW+96] J. Han, Y. Fu, W. Wang, K. Koperski, and O. Zaiane. DMQL : A data mining query language for relational databases. In *SIGMOD'96 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'96)*, Canada, 1996.

[HMV99a] C. Hurtado, A. Mendelzon, and A. Vaisman. Maintaining data cubes under dimension updates. In *Proc. 15th Int'l Conf. on Data Engineering, (ICDE'99)*, pages 346–355, March 1999.

[HMV99b] C. A. Hurtado, A. O. Mendelzon, and A. A. Vaisman. Updating OLAP Dimensions. In *ACM International Workshop on Data Warehousing and OLAP (DOLAP 99)*, pages 60–66, 1999.

[HR04] M-S. Hacid and C. Reynaud. L'intégration de sources de données. *Revue Information - Interaction - Intelligence (R I3)*, 4(2), 2004.

[Hua97] Z. Huang. Clustering large data sets with mixed numeric and categorical values. In *First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1997.

[IBM01] IBM. DB2 Intelligent Miner Scoring. http ://www-3.ibm.com/software/data/iminer/scoring, 2001.

[IDG05] IDG. Entreprises et décisionnel : état des lieux, objectifs et perspectives. http ://www.decisio.info/Entreprises-et-decisionnel.html, Résultats de l'enquête dans le document http ://www.decisio.info/IMG/pdf/ Enquete_CIO_SAS_230605.pdf, 2005.

[Inm96] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, 1996.

[Inm02] W. H. Inmon. *Building the Data Warehouse*. John Wiley & Sons, third edition, 2002.

[IT07] A. Inokuchi and K. Takeda. A method for online analytical processing of text data. In Mário J. Silva, Alberto H. F. Laender, Ricardo A. Baeza-Yates, Deborah L. McGuinness, Bjørn Olstad, Øystein Haug Olsen, and André O. Falcão, editors, *Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM 2007), Lisbon, Portuga*, pages 455–464. ACM, 2007.

[IV99] T. Imielinski and A. Virmani. Msql : A query language for database mining. *Data-Mining and Knowledge Discovery : An International Journal*, 3 :373–408, 1999.

[JMP01] M. R. Jensen, T. H. Møller, and T. B. Pedersen. Specifying OLAP Cubes On XML Data. In *Proceedings of the 13th International Conference on Scientific and Statistical Database Management, July 18-20, 2001, George Mason University, Fairfax, Virginia, USA*, pages 101–112. IEEE Computer Society, 2001.

[JRTZ08] H. Jerbi, F. Ravat, O. Teste, and G. Zurfluh. Management of context-aware preferences in multidimensional databases. In *ICDIM*, pages 669–675, 2008.

[JRTZ09] H. Jerbi, F. Ravat, O. Teste, and G. Zurfluh. Applying Recommendation Technology in OLAP Systems. In *ICEIS*, pages 220–233, 2009.

[KI04] G. Koutrika and Y. Ioannidis. Personalization of Queries in Database Systems. In *XXth International Conference on Data Engineering (ICDE 04), Boston, Massachusetts, USA*, pages 597–608. IEEE Computer Society, 2004.

[KI05] G. Koutrika and Y. Ioannidis. Personalized Systems : Models and Methods from an IR and DB Perspective. In *VLDB 05*, pages 1365–1365, 2005.

[Kie02] W. Kießling. Foundations of Preferences in Database Systems. In *VLDB*, pages 311–322, 2002.

[Kim96] R. Kimball. *The Data Warehouse Toolkit*. John Wiley & Sons, 1996.

[KKL05] Steven Keith, Owen Kaser, and Daniel Lemire. Analyzing Large Collections of Electronic Text Using OLAP. In *APICS 2005*, October 2005.

[Kob07] A. Kobsa. Generic User Modeling Systems. In *The Adaptive Web*, pages 136–154, 2007.

[Kor96] R. R. Korfhage. *Information Storage and Retrieval*. John Wiley & Sons, 1996.

[KR02] R. Kimball and M. Ross. *The Data Warehouse Toolkit : The Complete Guide to Dimensional Modeling*. John Wiley & Sons, deuxième édition, 2002.

[KRRT00] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. *Concevoir et déployer un data warehouse*. Eyrolles, 2000.

[KT07] E. Kondratas and I. Timko. CT-OLAP : temporal multidimensional data model and algebra for moving objects. In *ACM DOLAP*, pages 81–88, 2007.

[KVS06] K. Kotis, G. A. Vouros, and K. Stergiou. Towards automatic merging of domain ontologies : The HCONE-merge approach. *J. Web Sem.*, 4(1) :60–79, 2006.

[LA05] Y. Li and A. An. Representing UML Snowflake Diagram from Integrating XML Data Using XML Schema. In *Proceedings of the 2005 International Workshop on Data Engineering Issues in E-Commerce (DEEC '2005)*, pages 103–111, Tokyo, Japan, 2005. IEEE Computer Society.

[Leh98] W. Lehner. Modelling Large Scale OLAP Scenarios. In *VIth International Conference on Extending Database Technology (EDBT 98), Valencia, Spain*, volume 1377 of *LNCS*, pages 153–167. Springer, 1998.

[Len01] M. Lenzerini. Data Integration Is Harder than You Thought. In *CooplS01 : Proceedings of the 9th International Conference on Cooperative Information Systems*, pages 22–26, London, UK, 2001. Springer-Verlag.

[Lie95] H. Lieberman. Letizia : An Agent That Assists Web Browsing. In *IJCAI (1)*, pages 924–929, 1995.

[LL87] M. Lacroix and P. Lavency. Preferences ; Putting More Knowledge into Queries. In *VLDB*, pages 217–225, 1987.

[LM98] H. Lia and H. Motoda. *Feature Selection for knowledge discovery and data mining*. Kluwer Academic Publishers, 1998.

[LM02] S. Luján-Mora. Multidimensional Modeling using UML and XML. In *12th Workshop for PhD Students in Object-Oriented Systems, 16th European Conference on Object-Oriented Programming, June 10-14 2002, Málaga, Spain*, volume 2548 of *LNCS*, pages 48–49, 2002.

[LMT] S. Luján-Mora and J. Trujillo. A Comprehensive Method for Data Warehouse Design. In *5th International Workshop on Design and Management of Data Warehouses (DMDW'03)*, volume 77 of CEUR Workshop Proceedings, pages 1.1–1.14.

[LMTS02] S. Luján-Mora, J. Trujillo, and I-Y. Song. Multidimensional Modeling with UML Package Diagrams. In S. Spaccapietra, S. T. March, and Y. Kambayashi, editors, *Proceedings of the 21st International Conference on Conceptual Modeling Conceptual Modeling (ER'02), Tampere, Finland*, volume 2503 of *LNCS*, pages 199–213. Springer, 2002.

[LRO96] A. Y. Levy, A. Rajaraman, and J. J. Ordille. The World Wide Web as a Collection of Views : Query Processing in the Information Manifold. In *VIEWS*, pages 43–55, 1996.

[LW67] G.N. Lance and W.T. Williams. A general theory of classificatory sorting strategies : 1. Hierarchical systems. *Computer Journal*, 9 :373–380, 1967.

[Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings 5th Berkeley Symposium*, pages 281–297, 1967.

[Mad04] M. Madlej. Fouille en ligne dans les entrepôts de données. Technical report, Université Lumière Lyon 2, 2004.

[Mai10] N. Maiz. *Intégration de données par médiation basée sur les ontologies pour l'analyse en ligne (OLAP) à la demande*. PhD thesis, Université Lumière Lyon 2, Juillet 2010.

[MAR96] M. Mehta, R. Agrawal, and J. Rissanen. Sliq : A fast scalable classifier for data mining. In *EDBT*, pages 18–32, 1996.

[MBB06] N. Maiz, O. Boussaid, and F. Bentayeb. Ontology-Based Mediation System. In *ISPE CE*, pages 181–189, 2006.

[MBB07] N. Maiz, O. Boussaid, and F. Bentayeb. Hybrid Architecture of OWL-Ontology for Relational Data Sources Integration. In *IRMA*, pages 857–860, 2007.

[MER06] S. Massmann, D. Engmann, and E. Rahm. COMA++ : Results for the Ontology Alignment. In *Contest OAEI 2006 Ontology Matching proceedings*, 2006.

[Mes06] R. Ben Messaoud. *Couplage de l'analyse en ligne et de la fouille de données pour l'exploration, l'agrégation et l'explication des données complexes*. PhD thesis, Université Lumière Lyon 2, Décembre 2006.

[MFBB10] N. Maiz, M. Fahad, O. Boussaid, and F. Bentayeb. Automatic Ontology Merging by Hierarchical Clustering. In *10th International Conference on Knowledge Management and Knowledge Technologies (I-KNOW 10), Graz, Austria*, Special Issue, Journal of Universal Computer Science (J.UCS), pages 81–93, 2010.

[Mil95] G. A. Miller. WordNet : A Lexical Database for English. *Communincations of the ACM*, 38(11) :39–41, 1995.

[MP08] José A. Macías and Fabio Paternò. Customization of Web applications through an intelligent environment exploiting logical interface descriptions. *Interacting with Computers*, 20(1) :29–47, 2008.

[MPC96] R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. In *The VLDB Journal*, pages 122–133, 1996.

[MRBB04] R. Ben Messaoud, S. Rabaséda, O. Boussaid, and F. Bentayeb. OpAC : opérateur d'analyse en ligne basé sur une technique de fouille de données. In *Extraction et gestion des connaissances (EGC'2004), Actes des quatrièmes journées Extraction et Gestion des Connaissances, Clermont Ferrand, France, 20-23 janvier 2004, 2 Volumes*, pages 35–46, 2004.

[MRMB07] R. Ben Messaoud, S. Loudcher Rabaséda, R. Missaoui, and O. Boussaid. OLEMAR : an On-Line Environment for Mining Association Rules in Multidimensional Data. *Advances in Data Warehousing and Mining, IGI Global*, 2 :1–35, 2007.

[MT06] J. N. Mazón and J. Trujillo. Enriching Data Warehouse Dimension Hierarchies by Using Semantic Relations. In *XXIIIrd British National Conference on Databases*

(BNCOD 06), Belfast, Northern Ireland, UK, volume 4042 of *LNCS*, pages 278–281. Springer, 2006.

[MV00] A. O. Mendelzon and A. A. Vaisman. Temporal Queries in OLAP. In *VLDB*, pages 242–253, 2000.

[MW03] T. Morzy and R. Wrembel. Modeling a Multiversion Data Warehouse : A Formal Approach. In *Vth International Conference on Enterprise Information Systems (ICEIS 03), Angers, France*, pages 120–127, 2003.

[MW04] T. Morzy and R. Wrembel. On Querying Versions of Multiversion Data Warehouse. In *VIIth ACM International Workshop on Data Warehousing and OLAP (DOLAP 04), Washington, Columbia, USA*, pages 92–101. ACM Press, 2004.

[MZ04] E. Malinowski and E. Zimányi. OLAP Hierarchies : A Conceptual Perspective. In *XVIth International Conference on Advanced Information Systems Engineering (CAiSE 04), Riga, Latvia*, volume 3084 of *LNCS*, pages 477–491. Springer, 2004.

[NM01] N. F. Noy and M. A. Musen. Anchor-PROMPT : Using non-local context for semantic matching. pages 63–70, 2001.

[NRDR04] V. Nassis, R. Rajugan, T. S. Dillon, and J. Wenny Rahayu. Conceptual Design of XML Document Warehouses. In Yahiko Kambayashi, Mukesh K. Mohania, and Wolfram Wöß, editors, *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2004)*, volume 3181 of *LNCS*, pages 1–14. Springer, 2004.

[NSF+05] A. Nabli, A. Soussi, J. Feki, H. Ben-Abdallah, and F. Gargouri. Towards an Automatic Data Mart Design. In *VIIth International Conference on Enterprise Information Systems (ICEIS 05), Miami, Florida, USA*, pages 226–231, 2005.

[Ora01] Oracle. Oracle 9i Data Mining. White paper, 2001.

[PACW06] N. Prat, J. Akoka, and I. Comyn-Wattiau. A UML-based data warehouse design method. *Decision Support Systems*, 42(3) :1449–1473, 2006.

[PFLP10] Y. Pitarch, C. Favre, A. Laurent, and P. Poncelet. Context-aware generalization for cube measures. In *DOLAP*, pages 99–104, 2010.

[PG99] A. Pretschner and S. Gauch. Ontology Based Personalized Search. In *XIth IEEE International Conference on Tools with Artificial Intelligence (ICTAI 99), Chicago, Illinois, USA*, pages 391–398. IEEE Computer Society, 1999.

[PHS05] B-K. Park, H. Han, and I-Y. Song. XML-OLAP : A Multidimensional Analysis Framework for XML Warehouses. In A. Min Tjoa and Juan Trujillo, editors, *DaWaK*, volume 3589 of *Lecture Notes in Computer Science*, pages 32–42. Springer, 2005.

[PJ99] T. B. Pedersen and C. S. Jensen. Multidimensional Data Modeling for Complex Data. In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Austrialia*, pages 336–345, 1999.

[Qui86] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1 :81–106, 1986.

[RA10] Oscar Romero and Alberto Abelló. Automatic validation of requirements to support multidimensional design. *Data Knowl. Eng.*, 69(9) :917–942, 2010.

[RALT06] S. Rizzi, A. Abelló, J. Lechtenbörger, and J. Trujillo. Research in Data Warehouse Modeling and Design : Dead or Alive ? In *IXth ACM International Workshop on Data Darehousing and OLAP (DOLAP 06), Arlington, Virginia, USA*, pages 3–10. ACM Press, 2006.

[RB01] E. Rahm and Ph. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4) :334–350, 2001.

[RB07] O. Rakotoarivelo and F. Bentayeb. Evolution de schéma par classification automatique pour les entrepôts de données. In *IIIèmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 07), Poitiers, France*, volume B-3 of *Revue des Nouvelles Technologies de l'Information*, pages 99–112. Cépaduès Editions, 2007.

[RBF+02] M-C. Rousset, A. Bidault, C. Froidevaux, H. Gagliardi, F. Goasdou, C. Reynaud, and B. Safar. Construction de Médiateurs pour Intégrer des Sources d'information multiples et hétérogènes. *Revue I3*, 2 :09–59, 2002.

[Riz07] S. Rizzi. OLAP Preferences : a Research Agenda. In *DOLAP 07*, pages 99–100, 2007.

[RMZ02] G. Ramesh, W. Maniatty, and M. Javeed Zaki. Indexing and Data Access Methods for Database Mining. In *VIIth ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 02), Madison, Wisconsin, USA*, pages 2–9, 2002.

[RT08] F. Ravat and O. Teste. Personalization and OLAP Databases. *Annals of Information Systems, New Trends in Data Warehousing and Data Analysis*, 3 :71–92, novembre 2008.

[RTTZ07] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh. A Conceptual Model for Multidimensional Analysis of Documents. In *ER*, pages 550–565, 2007.

[RTZ06] F. Ravat, O. Teste, and G. Zurfluh. A Multiversion-Based Multidimensional Model. In *VIIIth International Conference on Data Warehousing and Knowledge Discovery (DaWaK 06), Krakow, Poland*, volume 4081 of *LNCS*, pages 65–74. Springer, 2006.

[RTZ07] F. Ravat, O. Teste, and G. Zurfluh. Personnalisation de bases de données multidimensionnelles. In *XXVème Congrès Informatique des Organisations et Systèmes d'Information et de Décision (INFORSID 07), Perros-Guirec, France*, pages 121–136, 2007.

[SAM98] S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-Driven Exploration of OLAP Data Cubes. In *VIth International Conference on Extending Database Technology (EDBT 98), Valencia, Spain*, volume 1377 of *LNCS*, pages 168–182. Springer, 1998.

[Sar00] S. Sarawagi. User-Adaptive Exploration of Multidimensional Data. In *VLDB*, pages 307–316, 2000.

[SE05] P. Shvaiko and J. Euzenat. A Survey of Schema-Based Matching Approaches. pages 146–171, 2005.

[Sea03] S. Searby. Personalisation - an overview of its use and potential. *BT Technology Journal*, 21(1) :13–19, 2003.

[SM01] G. Stumme and A. Maedche. FCA-MERGE : Bottom-Up Merging of Ontologies. In *IJCAI*, pages 225–234, 2001.

[SO98] M. Specht and R. Oppermann. ACE - Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia*, 4 :141–162, 1998.

[SSB03] Wee K. Ng Sourav S. Bhowmick, Sanjay K. Madria. *Web Data Management : A Warehouse Approach.* Springer Verlag, New York, USA, 2003.

[STA98] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating Mining with Relational Database Systems : Alternatives and Implications. In *ACM SIGMOD International Conference on Management of Data (SIGMOD 98), USA*, pages 343–354. ACM Press, 1998.

[STY01] S. Soni, Z. Tang, and J. Yang. Performance Study Microsoft Data Mining Algorithms. Technical report, Microsoft Corp., 2001.

[SW00] M. Stern and B. Park Woolf. Adaptive Content in an Online Lecture System. In *AH*, pages 227–238, 2000.

[TBB05] A. Tanasescu, O. Boussaid, and F. Bentayeb. Preparing complex data for warehousing. In *AICCSA*, page 30, 2005.

[TC06] F. S. C. Tseng and A. Y. H. Chou. The concept of document warehousing for multidimensional modeling of textual-based business intelligence. *Decision Support Systems*, 42(2) :727–744, 2006.

[Tes00] O. Teste. Elaboration d'entrepôts de données complexes. In *Actes du XVIIIème Congrès INFORSID'00, Lyon, France*, pages 229–245, 2000.

[TLM03] J. Trujillo and S. Luján-Mora. A UML Based Approach for Modeling ETL Processes in Data Warehouses. In *Conceptual Modeling - ER 2003, 22nd International Conference on Conceptual Modeling, Chicago, IL, USA, October 13-16, 2003, Proceedings*, volume 2813 of *Lecture Notes in Computer Science*, pages 307–320, 2003.

[TLMS03] Juan Trujillo, Sergio Luján-Mora, and Il-Yeol Song. Applying UML For Designing Multidimensional Databases And OLAP Applications. In *Advanced Topics in Database Research, Vol. 2*, pages 13–36. 2003.

[Toi96] H. Toivonen. Sampling large databases for association rules. In *International Conference on Very Large Data Bases*, pages 134–145. Morgan Kaufman, 1996.

[TPRT10] R. Tournaire, J-M. Petit, M-C. Rousset, and A. Termier. Combining Logic and Probabilities for Discovering Mappings between Taxonomies. In *KSEM*, pages 530–542, 2010.

[Tru99] J. Trujillo. The GOLD Model : An OO Multidimensional Data Model for Multidimensional Databases. In *Object-Oriented Technology, ECOOP'99 Workshop Reader, ECOOP'99 Workshops, Panels, and Posters, Lisbon, Portugal, June 14-18, 1999, Proceedings*, volume 1743 of *LNCS*, pages 24–30. Springer, 1999.

[TSM01] T. Thalhammer, M. Schrefl, and M. Mohania. Active Data Warehouses : Complementing OLAP with Analysis Rules. *Data and Knowledge Engineering*, 39(3) :241–269, 2001.

[UBDB04] C. Udréa, F. Bentayeb, J. Darmont, and O. Boussaid. Intégration efficace de méthodes de fouille de données dans les SGBD. In *4èmes Journées Francophones d'Extraction et de Gestion des Connaissances (EGC 04), Clermont-Ferrand*, Revue des Nouvelles Technologies de l'Information, pages 83–94. Cépaduès Editions, 2004.

[VBR03] B. Vrdoljak, M. Banek, and S. Rizzi. Designing Web Warehouses from XML Schemas. In Y. Kambayashi, M. K. Mohania, and W. Wöß, editors, *5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'03), Prague, Czech Republic*, volume 2737 of *Lecture Notes in Computer Science*, pages 89–98. Springer, 2003.

[VM00] A. Vaisman and A. Mendelzon. Temporal queries in olap. In *Proc. VLDB 2000*, September 2000.

[WHK+01] Stephen T.C. Wong, Kent Soo Jr Hoo, Robert C. Knowlton, Kenneth D. Laxer, Xinhau Cao, Randall A. Hawkins, William P. Dillon, and Ronald L. Arenson. Design and Applications of a Multimodality Image Data Warehouse Framework. *The journal of the American Medical informatics Association*, 2001.

[Wid99] J. Widom. Review - An Overview of Data Warehousing and OLAP Technology. *ACM SIGMOD Digital Review*, 1, 1999.

[Wie92] G. Wiederhold. Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3) :38–49, 1992.

[WVV+01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. Ontology-based integration of information — a survey of existing approaches. In H. Stuckenschmidt, editor, *IJCA'01 : Workshop : Ontologies and Information Sharing*, pages 108–117, 2001.

[WZL03] H. Wang, C. Zaniolo, and C. Luo. ATLAS : A Small but Complete SQL Extension for Data Mining and Data Streams. In *VLDB*, pages 1113–1116, 2003.

[Xyl01] Lucie Xyleme. A dynamic warehouse for XML Data of the Web. *IEEE Data Eng. Bull.*, 24(2) :40–47, 2001.

[ZR96] D. A. Zighed and R. Rakotomalala. SIPINA-W(c) for Windows : User's Guide. Technical report, ERIC laboratory, University of Lyon 2, France, 1996.

[ZR00] D. A. Zighed and R. Rakotomalala. *Graphes d'induction. Apprentissage et Data Mining.* Hermes Science Publication, 2000.