Vers l'entreposage des données complexes : structuration, intégration et analyse

Omar Boussaid — Fadila Bentayeb — Jérôme Darmont — Sabine Rabaseda

BDD - ERIC - Université Lumière Lyon 2
Bâtiment L, 5 avenue Pierre-Mendès-France - France
69676 BRON Cedex – FRANCE
Tél. : +33 (0)4 78 77 23 77   Fax : +33 (0)4 78 77 23 75

{boussaid, darmont, rabaseda}@univ-lyon2.fr, bentayeb@eric.univ-lyon2.fr

RÉSUMÉ. *Les technologies utilisées dans les processus décisionnels, comme les entrepôts de données, l'analyse multidimensionnelle en ligne et la fouille de données, sont désormais très efficaces pour traiter des données simples numériques ou symboliques. Cependant, diverses sources, dont le Web, présentent des données sous des formes variées et hétérogènes : textes, images, sons, vidéos ou bases de données, de surcroît exprimées dans différentes langues, stockées dans différents formats, etc. Ces données, qualifiées de complexes, sont largement porteuses d'information et donc intéressantes à traiter au sein d'un processus décisionnel. Nous présentons dans cet article une démarche globale ainsi que quelques solutions que nous avons mises en oeuvre dans ce contexte, principalement au niveau des premières phases du processus décisionnel (intégration, entreposage et auto-administration des données complexes). Nous explorons également quelques pistes concrètes concernant la modélisation multidimensionnelle et l'analyse en ligne des données complexes.*

ABSTRACT. *Decision-support technologies as data warehouse, OLAP, and data mining have became very efficient to process simple numeric and symbolic data. However, various data sources, including the Web, contain various and heterogeneous data: texts, images, sounds, videos, or databases, which may furthermore be expressed in various formats and languages. These so-called complex data carry pieces of information and thus are interesting to include into a decision-support process. We present in this paper a global approach to warehouse complex data, as well as a couple of solutions we designed in this context that mainly deal with the first phase of the process (complex data integration, warehousing, and auto-administration). We also explore a couple of actual research perspectives regarding complex data multidimensional modelling and on-line analysis.*

MOTS-CLÉS : *Entrepôts de données, Analyse en ligne, Fouille de données, Données complexes, Intégration de données, Modélisation multidimensionnelle, Administration de données*

KEYWORDS: *Data warehouses, OLAP, Data mining, Complex data, Data integration, multidimensional modelling, Data administration*

1. Introduction

Le processus décisionnel ou les systèmes d'information décisionnels sont nés d'un besoin exprimé par les entreprises (au sens large du terme, entreprises privées, publiques, institutions, organisations...), besoin non satisfait par les systèmes de bases de données traditionnels. De ce besoin sont apparus dans les années quatre-vingt dix les entrepôts de données (*data warehouses*), qui sont des bases de données décisionnelles. Ces dernières ont eu une répercussion importante aussi bien dans le monde industriel que dans la communauté de la recherche scientifique. Le processus décisionnel est apparu comme une technologie clef pour les entreprises désirant améliorer l'analyse de leurs données et leur système d'aide à la décision. En intégrant la notion d'entrepôt de données, le processus décisionnel apporte une première réponse au problème de la croissance continuelle des données. De plus, il supporte efficacement les processus d'analyse en ligne (*On-Line Analytical Processing - OLAP*) et de fouille de données (*data mining*), permettant ainsi l'analyse des données. Les technologies d'entreposage de données, d'analyse en ligne et de fouille de données font l'objet de nombreuses recherches scientifiques et ont maintenant largement fait leurs preuves dans les domaines de la gestion de données et de l'extraction de connaissances à partir de données dites "simples" (données numériques ou symboliques exprimées dans un tableau de type individus-variables).

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 (textes, images, son, vidéos, bases 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 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. L'intégration des données complexes exige une modélisation permettant de prendre en considération les différents aspects de ces données. Or, il n'existe pas de modèle universel pour les données multimédia, et de manière générale, pour toutes les formes de données complexes. Le recours à un format unifié pour intégrer les diverses sources de données dans une base cible est rendu possible grâce à XML. En effet, ce langage permet non seulement de véhiculer les données, mais aussi de les décrire de façon précise. D'autre part, les bases de donnnées semi-structurées présentent un défi intéressant. Le mode de stockage peut s'opérer en natif XML ou dans des bases de données classiques (relationnelles, orientées objets ou relationnelles-objets). Elles disposent également d'outils d'interrogation efficaces.

La structuration et le stockage des données complexes n'est qu'une première étape d'une démarche dont la finalité est l'extraction de l'information pour alimenter des processus décisonnels. Par conséquent, les donnés complexes doivent être préparées à l'analyse. L'entreposage des données complexes peut s'avérer utile. Cependant, comment peut-on construire un entrepôt de données complexes ? Peut-on y appliquer les opérateurs classiques d'OLAP ? Certains travaux dans le domaine de la fouille de textes (*text mining*) ou d'images (*image mining*) ont permis de valider des techniques déjà opérationnelles.

L'objet de cet article est de montrer les différents problèmes que posent la structuration, l'intégration et l'analyse des données complexes dans un but décisonnel ainsi que d'apporter quelques éléments méthodologiques globaux et de présenter rapidement les solutions que nous avons proposées jusqu'ici.

Dans la section 2 de cet article, nous exposons un état de l'art sur les travaux dans les domaines de la modélisation des documents multimédia, des bases de données semi-structurées et de l'entreposage des données. Nous évoquons également certains travaux sur la fouille de textes ou d'images. Nous présentons ensuite dans la section 3 la démarche d'entreposage des données complexes que nous préconisons en évoquant les différents problèmes engendrés. La section 4 est consacrée aux différents travaux que nous avons réalisés pour aborder certains problèmes relatés dans la section précédente. Nous concluons et présentons nos perspectives de recherche dans la section 5.

2. Etat de l'art

2.1. *Données complexes et modèles de documents*

Les données complexes englobent entre autres les données multimédia. Elles commencent à intéresser plusieurs communautés de chercheurs. Les travaux dans le domaine du multimédia portent aujourd'hui essentiellement sur l'édition et la présentation des documents multimédia. Le processus de production de tels documents est basé sur une chaîne complète intégrant trois phases [THU 03]. La première, désignée par l'analyse, consiste à extraire automatiquement et/ou manuellement des informations pertinentes. Celles-ci sont alors spécifiées dans des formats définis : c'est la deuxième phase dite de description. Ces deux étapes permettent à la dernière, "applications multimédia", d'effectuer des traitements plus larges et plus efficaces des média.

La composition de documents multimédia est une application multimédia qui consiste à récupérer des fragments de médias appropriés pour produire une présentation multimédia [LAY 97]. D'autre part, les bases de données multimédia sont également un exemple d'application multimédia. Leur émergence est aujourd'hui évidente. L'intégration, la structuration, l'archivage et la recherche d'information ont besoin de modèles de description. La description des médias doit également porter sur le contenu, dont l'information sert déjà aux techniques d'indexation et de recherche. L'extraction des informations sur le contenu fait l'objet de nombreux travaux de recherche. Elle représente un axe fondamental entre autres, dans l'édition et la présentation des documents multimédia. Un document multimédia décrit une composition de la présentation multimédia dont la logique est formulée à travers un modèle de document. Il n'existe pas de modèle général permettant d'exprimer tous les aspects d'un document multimédia. Il est alors spécifié selon différents axes [THU 03]. Le *modèle du contenu* doit représenter les informations sur la sémantique du média. Celles-ci ne sont pas extraites de façon automatique. Des techniques telles que l'indexation ou l'annotation permettent d'extraire des fragments de sémantique, mais elles ne sont prises en compte par aucun modèle multimédia. Le *modèle logique* présente les informations sur la structure logique d'un média, qui sont issues des modèles temporel et spatial. Le *modèle temporel* permet de définir et de gérer la synchronisation entre les différentes parties d'un document par le biais de relations temporelles. Il n'exprime pas en général la sémantique d'un document. Le *modèle spatial* permet de décrire les informations sur les objets dans un média (couleur, luminosité, texture, disposition...). Il définit également des relations spatiales entre les objets. Le *modèle hyperlien* permet de définir des références sur des parties internes d'un document ou sur d'autres documents externes. Le *modèle d'animation* peut être défini par programmation. Un animateur peut changer l'état d'un objet dans le temps. Des animations prédéfinies peuvent également être employées.

Les descriptions actuelles de contenu sont insuffisantes pour la composition multimédia. Elles ne couvrent pas les structures logiques et temporelles d'un document multimédia. Le Web sémantique, contrairement aux outils actuels du Web qui sont limités seulement à l'affichage des données, tente de donner à l'information plus de sémantique bien définie afin de permettre aux utilisateurs (hommes ou machines) de mieux coopérer. Les informations visuelles expriment les caractéristiques de couleur, de luminosité, de texture, de forme et de position. Ces descripteurs visuels sont bien spécifiés par des formats standards (MPEG, JPEG, GIF...). Ils représentent des caractéristiques de bas niveau qui peuvent très bien être extraites automatiquement. En revanche, les informations issues du contenu sont très peu spécifiées. Elles nécessitent le recours à d'autres techniques, de reconnaissance de formes, de fouille de données ou d'annotation. Une démarche semi-automatique permet d'identifier et d'extraire différentes parties dans un média. Elles sont complétées par des annotations manuelles ou semi-automatiques [MAT 90, KAH 02]. Celles-ci permettent alors un rajout d'informations de haut niveau sémantique, sous forme de texte par exemple. Ces renseignements sont d'ailleurs gérés sous formes de métadonnées. Le recours à cette technique permet de manipuler efficacement les documents multimédia. Il existe de grands standards pour exprimer des descriptions uniformes et interopérables (Dublin Core, RDF, MPEG7...). Ils sont suffisamment généraux et permettent ainsi de décrire l'information dans n'importe quel domaine. Aujourd'hui, MPEG7 offre de grands espoirs dans la description du multimédia. De nombreux travaux de recherche y sont consacrés. C'est une norme générique composée de descripteurs, de schémas de description et d'un langage de définition de descriptions. Il permet non seulement de décrire des caractéristiques de bas niveau, mais également des descriptions sémantiques (personne, animal, maison...).

L'exploitation efficace des documents multimédia nécessite la manipulation des informations sur la sémantique des média. Quel que soit le domaine de recherche, l'extraction des caractéristiques sémantiques des documents multimédia est un véritable défi. C'est dans cette optique que nous situons notre approche d'intégration, de structuration et d'analyse, non seulement des données multimédia, mais plus globalement des données complexes.

2.2. *Bases de données semi-structurées*

2.2.1. *Généralités*

Les données semi-structurées telles que les documents XML sont caractérisées par l'absence de schéma fixe, bien que les données possèdent implicitement une certaine structure. La difficulté est alors de pouvoir extraire cette structure. Notons que les données semi-structurées sont souvent stockées dans des systèmes de fichiers pourvus de moyens limités pour l'organisation, la recherche et l'exploitation des données. Par ailleurs, les SGBD classiques sont inappropriés aux informations semi-structurées. En effet, ces dernières ne peuvent être gérées par les SGBD que si elles sont traduites de façon conforme au modèle sous-jacent du système utilisé.

D'autre part, le langage XML offre plusieurs fonctionnalités des SGBD telles que le stockage, les schémas, les langages de requêtes, les interfaces de programmation, etc. Cependant, il manque d'outils importants qui existent dans les SGBD, comme des techniques de stockage efficaces, les index, la sécurité, les transactions, l'intégrité des données, les accès multi-utilisateurs, les déclencheurs, les requêtes basées sur plusieurs documents, etc.

2.2.2. *Stockage de documents XML*

Nous dénombrons deux principales approches pour stocker des données semi-structurées. La première consiste à étendre un SGBD existant en permettant l'importation et l'exportation de documents XML. Les systèmes qui l'ont adoptée incluent tous les principaux SGBD relationnels (Oracle, SQL Server, DB2...) et orientés-objets (Matisse, Objectivity/DB, Versant...). Ils pourraient être qualifiés de SGBD "compatibles XML". Ce type de systèmes est principalement employé lorsqu'XML sert de standard d'échange de données.

La seconde approche consiste à construire un système dédié tel qu'une base de données "native XML" comme Tamino (qui a d'ailleurs popularisé ce terme). Ces systèmes peuvent être subdivisés en deux catégories [BOU 03a] : ceux basés sur le texte, qui stockent les documents XML dans des fichiers ou des BLOBs (*Binary Large Objects*) et les systèmes basés sur un modèle, qui construisent une structure de données interne (par exemple, un ensemble de tables relationnelles) d'après le document à stocker (processus de *mapping* [AND 00, KAP 00]). Cette approche peut également exploiter XML comme standard d'échange de données, mais elle permet surtout de gérer plus efficacement des documents en tant que tels (manuels d'utilisation, pages web statiques, etc.).

2.2.3. *Langages de requêtes*

Il existe quatre moyens principaux d'effectuer des requêtes sur des documents XML. Ils varient notamment selon le type de stockage employé.

1) Le langage XSLT [CLA 99], basé sur les feuilles de style XSL, permet de transformer la structure de documents, par exemple pour correspondre au modèle interne d'une base de données native XML et *vice versa*. Les temps de traitements peuvent cependant être importants si les transformations à effectuer sont nombreuses.

2) Les langages les plus courants qui fournissent un résultat sous forme de documents XML sont basés sur des modèles (*templates*). Des requêtes de type SELECT sont encapsulées dans un modèle, qui est ensuite interprété. Le résultat obtenu est un document XML de même structure que le modèle. Ces langages sont très flexibles, mais ne sont quasiment utilisés que pour exporter des données relationnelles dans des documents XML.

3) Les langages de requêtes basés sur SQL utilisent des instructions SQL modifiées dont les résultats sont ensuite transformés en documents XML. Les extensions XML de SQL sont en cours de standardisation par l'ISO et l'ANSI sous le nom de SQL/XML [MEL 02].

4) Contrairement aux langages de requêtes basés sur des modèles ou SQL, qui ne sont utilisés que dans un contexte relationnel, les langages de requêtes XML peuvent s'appliquer à tout document XML, y compris ceux qui sont stockés dans une base de données relationnelle, mais suivant un modèle XML. XQuery [BOA 03] est par exemple capable d'opérer un *mapping* vers une base de données relationnelle ou relationnelle-objet. XPath [BER 03] est très proche de XQuery, mais s'avère plus limité dans un contexte relationnel car il ne supporte pas les requêtes sur plusieurs tables. En revanche, cette limitation est levée dans un contexte relationnel-objet.

2.3. *Les entrepôts de documents XML*

Plusieurs travaux existent déjà sur l'intégration des documents XML, dont certains sont plutôt orientés vers l'entreposage. Dans [BAR 03], les auteurs proposent une architecture d'entrepôt pour les documents XML (*XML warehouse*) permettant de calculer des vues de documents XML à partir d'un fragment d'un document ou en combinant des fragments de différents documents par un mécanisme de jointure. Les auteurs présentent un langage de spécification de l'entrepôt de documents XML pour définir des vues pouvant être matérialisées.

Un entrepôt de données est constitué d'un ensemble de vues matérialisées [GAR 93, GUP 95]. Elles sont calculées à partir des tables sources de l'entrepôt. Lorsque ces dernières sont mises à jour, il est nécessaire de répercuter ces modifications sur les vues matérialisées. De nombreux travaux proposent des algorithmes relatifs à la maintenance incrémentale et à la réduction des coûts de maintenance des vues matérialisée [CER 91, Y.Z 95, BEL 98]. Sélectioner des vues matérialisées revient à résoudre un problème d'optimisation d'un ensemble de vues selon une fonction de coût d'évaluation de requêtes ou de maintenance de vues. De nombreux algorithmes ont été développés pour trouver des solutions optimales [YAN 97, SAR 97, GUP 99], qui d'ailleurs lient souvent la sélection des vues à celle des index. Dans les entrepôts de documents XML, le traitement de ces problèmes représente une voie de recherche intéressante.

D'autre part, les travaux de [JEN 01] proposent une démarche pour intégrer des documents XML (ainsi que des vues relationnelles) seulement à un niveau logique. Leur but est de modéliser ces données de façon multidimensionnelle sous la forme d'un diagramme de classes UML en schéma en étoile et/ou en flocon de neige, avec une ou plusieurs classes de faits, des classes dimensions et des classes hiérarchies de dimensions. Partant du modèle UML, les données sont finalement stockées dans des tables relationnelles et sont ainsi prêtes à des analyses en ligne à l'aide de serveurs OLAP.

Le rapprochement entre XML et les entrepôts de données est très prometeur. Le passage par XML semble être une voie priviligiée pour entreposer des données complexes. Il reste cependant à déterminer comment il est possible de les anlayser ?

2.4. *Techniques d'exploration des données complexes*

Initialement, les méthodes de fouille de données ont été développées pour extraire de la connaissance à partir de données "simples", c'est-à-dire des données symboliques ou numériques exprimées dans un tableau individus-variables ou attributs-valeurs d'une base de données. Une première extension des méthodes de fouille de données est apportée par les arbres de décision flous [Mar98] qui permettent d'intégrer le côté imprécis et/ou incertain des données. Depuis quelques années, la communauté scientifique de fouille de données s'intéresse à tous les formats différents que peuvent revêtir les données : textes, images, sons, vidéos, bases de données...De là sont nées de nouvelles approches comme l'extraction de connaissances à partir de données textuelles [WIT 99, GHA 00, NAH 00, ZIG 00, BER 02, CLE 03a], à partir d'images [DEL 95, DJE 00a] ou à partir du Web [ENG 01]. Une généralisation des différentes approches de d*ata mining*, *text mining*, *image mining* donne naissance à ce qui est appelé à l'heure actuelle le *multimedia mining* [DJE 00b].

D'autres travaux portent sur la transformation d'un texte ou d'une image en un vecteur attributs-valeurs [TEY 01, JAL 02, FOS 02]. Cette démarche entre dans le cadre d'un sujet plus large : la construction de caractéristiques (*feature construction*) [LIU 98, FLA 00], qui est à l'heure actuelle considérée comme stratégique pour le développement de la fouille de données complexes en améliorant les résultats des méthodes d'extraction de connaissance.

De plus, il devient nécessaire d'envisager l'utilisation conjointe des techniques d'analyse en ligne et de fouille de données. Cette idée de couplage est relativement récente dans la communauté scientifique, et trois approches semblent se dégager.

– La première consiste à modifier les opérateurs d'analyse OLAP afin qu'ils puissent simuler des techniques de fouille de données [GOI 97, HAN 97, HAN 98, CHE 00, GOI 01].

– La deuxième consiste à adapter la structure multidimensionnelle des données d'un cube OLAP de façon à pouvoir utiliser des techniques de fouille sur ces données [PIN 01a, PIN 01b, GOI 01, CHE 01, LAU 01].

– La troisième envisage de modifier les algorithmes des méthodes de fouille afin de traiter des données multidimensionnelles [SAR 98, PAL 00, SAR 01].

3. Entreposage de données complexes

3.1. *Notre approche méthodologique*

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'extraction des connaissances à partir de données complexes, il faut intégrer puis représenter les données complexes sous une forme adaptée aux techniques d'analyse en ligne ou de fouille de données. Partant de ce constat, nous proposons un processus d'entreposage et d'analyse des données complexes (Figure 5).

Notre démarche permet de concevoir des processus décisionnels utilisant des données complexes. Deux grands axes se dégagent : (1) la structuration des données complexes avec la modélisation et l'intégration des données dans une base de données ; (2) l'analyse des données complexes par des techniques de fouille de données, d'analyse en ligne ou par les deux approches combinées.

3.1.1. *Structuration des données complexes*

Notre idée consiste à modéliser et à intégrer les données complexes dans une source de données structurée comme une base de données relationnelle décisionnelle. L'approche que nous préconisons permet de considérer une donnée complexe comme un objet complexe pouvant être décrit par un modèle conceptuel UML. Le diagramme de classes UML (Figure 2) est traduit en une définition de schéma XML représentant le modèle logique de la donnée complexe. Cette phase de structuration est finalisée par la construction d'un modèle physique représentant des documents XML stockés dans une base de données relationnelle. Ainsi, nous représentons toutes sortes de donnée

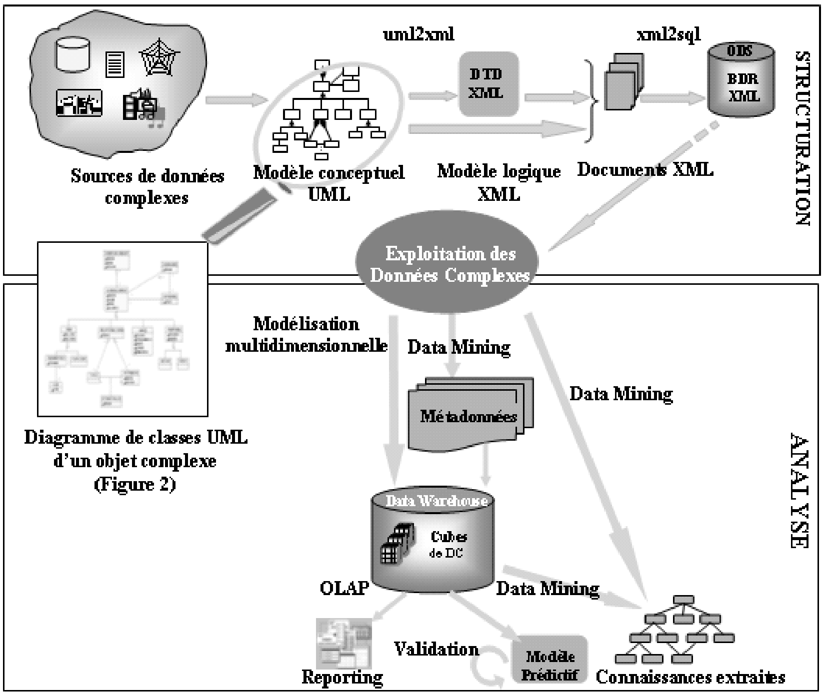

FIG. 1. *Processus d'entreposage et d'analyse des données complexes*

complexe dans un format unifié par un document XML. Nous développons notre démarche d'intégration des données complexes dans la section 3.2.1. Celle-ci est définie de manière générale. Le modèle conceptuel UML est générique. Il comporte les descripteurs de bas niveau des données complexes (comme la taille, la résolution d'une image ou le nombre de lignes ou de mots d'un texte). Lorsque ce modèle est utilisé dans différentes applications, il est nécessaire de le compléter par des descripteurs sémantiques propres au domaine d'application. L'ensemble des descripteurs de bas niveau et sémantiques va ainsi permettre de construire un modèle de données spécifique.

Le dernier aspect de l'intégration et de la structuration de données porte sur la gestion de la base de données décisionnelle et de son évolution. Il concerne les stratégies à adopter face à des données externes nouvelles. Quelles sont les données à intégrer ? Faut-il les insérer dans le schéma de base de données existant ou restructurer la base de données de façon dynamique ? Nous reviendrons sur ces aspects dans la section 4.4.

3.1.2. *Analyse des données complexes*

Les approches possibles pour l'analyse des données complexes englobent la fouille de données et l'analyse en ligne. De plus, il est possible d'envisager une imbrication de ces deux approches.

– *Fouille de données complexes.*

Partant des données complexes structurées dans une base de données relationnelle, il est possible d'appliquer des techniques de fouille de données. Une généralisation des différentes approches de *data mining*, *text mining*, *image mining*, etc. pourrait être de faire de la fouille dans les données modélisées en XML (*XML mining*) afin de s'affranchir de la nature originelle des données tout en étant capable de faire de la fouille sur des données de types différents. Le *XML mining* pourrait être une réponse au problème de *multimedia mining*.

– *Analyse en ligne de données complexes.*

OLAP permet d'agréger les données et de naviguer dans des structures mulidimensionnelles. Si l'on souhaite pouvoir utiliser les opérateurs OLAP d'analyse en ligne sur les données complexes, une étape complémentaire de modélisation multidimensionnelle des données complexes est nécessaire (section 4.2).

– *Fouille et analyse en ligne de données complexes.*

La combinaison de ces deux techniques permet d'enrichir et d'affiner l'information extraite à partir des données. L'utilisation des structures multidimensionnelles (telles que les cubes de données) apporte un plus à l'application des outils de la fouille de données et contribue à un meilleur ciblage des objectifs recherchés. Les opérateurs classiques d'OLAP ne sont pas applicables sur les données complexes. Cependant, le recours aux techniques de fouille de données peut être envisagé pour définir de nouveaux opérateurs, comme nous le présentons dans la section 4.2.2.

Ainsi, la première phase du processus d'entreposage des données complexes consiste à intégrer des données provenant de sources diverses dans une base de données relationnelle. De là, il est déjà possible d'y appliquer des techniques de fouille de données pour extraire des connaissances. La deuxième phase du processus proposé permet de sélectionner dans la base de documents XML les données que l'on souhaite explorer et de les enrichir avec des métadonnées. Elles peuvent être modélisées sous forme multidimensionnelle en créant des cubes de données. L'exploration peut alors se faire par l'analyse en ligne ou par différentes techniques de fouille de données.

La modélisation multidimensionnelle des données est généralement bien maîtrisée pour des données classiques. Cependant, elle est à réinventer lorsque la nature, les sources et les supports des données sont divers et distribués, lorsque la définition des données évolue dans le temps, etc.

3.2. *Problèmes liés à l'entreposage de données complexes*

Il est évident que les techniques classiques d'entreposage de données, d'analyse en ligne et de fouille de données, initialement conçues pour des données simples, ne sont pas adaptées aux données complexes. Plusieurs questions, correspondant à des problèmes de recherche non encore ou partiellement résolus, se posent en termes de structuration, d'intégration, de modélisation, d'analyse multidimensionnelle et de fouille de données complexes.

3.2.1. *Intégration de données complexes*

Plusieurs approches d'intégration de données existent. Dans l'approche basée sur les médiateurs [GOA 00, ROU 02], il est question de maintenir les données dans leurs sources d'origine et de construire des vues abstraites à partir desquelles un médiateur tente de satisfaire des requêtes d'utilisateurs. Le médiateur réalise l'intégration des données en fournissant une vue homogène et globale du système à l'utilisateur. Sa tâche est de reformuler les requêtes posées par l'utilisateur en fonction des différents contenus des sources de données accessibles [CLU 99, REY 02].

Par ailleurs, l'approche basée sur la technologie de l'entreposage des données [INM 96, KIM 00] consiste à construire à partir de différentes sources de données une nouvelle base appelée entrepôt de données. Dans ce cas, l'intégration correspond au processus d'ETL (*Extracting Transforming and Loading*) chargé de nettoyer et de transformer les données qui sont hétérogènes, avant leur chargement dans l'entrepôt. Le modèle de données est multidimensionnel et caractérise un contexte d'analyse. Les requêtes sont en général complexes [HAR 96]. Elles nécessitent des traitements sur les données pour les résumer et faciliter leur interprétation. De plus, elles nécessitent des moyens sophistiqués de navigation dans les données à travers les différentes dimensions et ce grâce aux opérateurs OLAP [COD 93].

Ces approches d'intégration sont valides et efficaces quand il s'agit de données simples (numériques ou symboliques). Cependant comment peut-on intégrer des données dont les types sont divers ? Comment peut-on les stocker et les interroger ?

3.2.2. *Modélisation multidimensionnelle des données complexes*

Dans la démarche d'intégration que nous proposons (section 4.1), les données complexes sont représentées sous forme de document XML. Ceci donne l'avantage d'utiliser le même format quel que soit le type d'origine. Les documents XML sont construits selon une grammaire (DTD -Document type Defintion- ou schéma XML) disposant d'une suite bien identifiée d'attributs qui constitue un espace de représentation des données complexes. Ces dernières sont alors spécifiées par des vecteurs d'attributs. Ainsi, il est possible de les archiver dans une base de données relationnelle ou native XML. Cette approche est plutôt destinée à une exploitation transactionnelle des données complexes. Il est facile de les gérer, de les mettre à jour ou de les interroger. Cependant, elle est peu appropriée à une démarche analytique. En s'inspirant des méthodes utilisées dans le processus d'entreposage des données classiques, il s'avère nécessaire de rajouter une couche supplémentaire de modélisation des données complexes. Le modèle multidimensionnel, et plus précisément les modèles en étoile, offrent un cadre d'analyse des données. Ils permettent d'observer des faits à travers d'indicateurs (mesures) et d'axes d'analyse (dimensions). Une des premières difficultés de cette voie est le choix des données à modéliser. Dans une approche classique d'entreposage, les données traduisent les performances d'une activité. L'analyse OLAP consiste alors à résumer l'information et à la représenter dans un espace multidimensionnel (cube de données) dans lequel l'utilisateur peut naviguer dans les données et effectuer ainsi une démarche exploratoire. Les données complexes n'expriment pas les informations sur une activité de la même manière, du moins pas aussi directement. Il est possible de trouver dans un rapport d'activité d'une entreprise des renseignements chiffrés textuels ou graphiques, par exemple. Une extraction préalable de ces données (interrogation du contenu d'un document multimédia) est nécessaire. Le choix des données à observer est un véritable handicap pour l'analyse de données de type image, vidéo ou sonore.

La modélisation des données complexes n'est pas une tâche facile. La formulation des faits est loin d'être évidente. C'est d'ailleurs le principal obstacle. Le modèle en étoile est bien adapté aux informations à caractère numérique. Comment peut-on l'appliquer sur des données complexes ? La construction d'entrepôts de textes, d'images, de vidéos et de documents sonores est-elle envisageable ? Peut-on affiner encore les magasins (*data marts*) selon le type des données complexes ? Est-il pertinent de construire un cube (*data cube*) dans lequel les données sont issues de différents documents ? Toutes ces interrogations révèlent la nécessité de repenser autrement le processus d'entreposage dans le cas des données complexes.

3.2.3. *Analyse des données complexes*

La vocation d'OLAP est de permettre d'agréger les très nombreuses données pour résumer l'information qu'elles contiennent et de représenter celle-ci sous différents angles permettant à l'utilisateur de naviguer dans les données et de les explorer. Les opérateurs OLAP sont définis pour des données classiques c'est-à-dire numériques. Ils sont par conséquent inadaptés quand il s'agit de documents textes, images, vidéos ou sonores. Le recours à d'autres techniques par exemple de fouille de données, peut s'avérer intéressant. Les algorithmes classiques de ces méthodes s'appliquent à des données représentées sous forme de tableaux "attributs-valeurs". Faut-il alors représenter les données complexes sous la dite forme ou faut-il adapter les algorithmes classique de fouille aux données complexes ?

3.2.4. *Administration des données complexes*

L'utilisation courante de bases de données requiert un administrateur qui a pour rôle principal la gestion des données au niveau logique (définition de schéma) et physique (fichiers et disques de stockage), ainsi que l'optimisation des performances de l'accès aux données. Avec le déploiement à grande échelle des systèmes de gestion de bases de données et la complexité et la diversité sans cesse croissante des données, minimiser cette fonction d'administration est devenu indispensable [WEI 02].

L'une des tâches importantes d'un administrateur est la sélection d'une structure physique appropriée pouvant améliorer les performances du système en minimisant les temps d'accès aux données [FIN 88]. Ce travail d'optimisation des performances de l'administrateur se porte en grande partie sur la sélection d'index et de vues matérialisées [GUP 99, AGR 01]. Ces structures jouent un rôle particulièrement important dans les bases de données décisionnelles telles que les entrepôts de données, qui présentent une volumétrie très importante et qui sont interrogés par des requêtes complexes. De plus, optimiser l'accès à des objets complexes et/ou volumineux est indispensable pour obtenir des temps de réponse acceptables lors des interrogations de la base de données.

4. Quelques solutions

Dès l'apparition de la technologie des bases de données, les éditeurs de SGBD ont porté un intérêt particulier aux données multimédia, en dotant leurs logiciels d'outils permettant le stockage et la gestion de ces données [KHO 96]. Néanmoins, l'interrogation de ces données par leur contenu a nécessité l'intégration d'outils spécifiques dans les SGBD. La technique la plus fréquemment utilisée consiste à extraire des caractéristiques de ces données au moyen de techniques telles le traitement d'images. Nous avons montré que les caractéristiques extraites à partir des données complexes à l'aide de techniques de traitement d'images représentent des descripteurs de bas niveau et peuvent être utilisées et exploitées de manière efficace dans le processus d'intégration de ces données dans un entrepôt [DAR 02].

En fait, dans l'approche que nous préconisons, nous nous sommes intéressés à l'ensemble des données complexes et nous avons opté pour l'utilisation de la technologie d'entreposage. Nous avons réalisé un certain nombre de travaux pour aborder les différents problèmes évoqués dans la section précédente. Avant de les présenter, rappelons que notre démarche consiste en une proposition d'un cadre méthodologique et d'outils génériques pour l'intégration, la structuration, le stockage et la modélisation multidimensionnelle des données complexes et enfin de leur exploration par des techniques d'analyse en ligne et de fouille de données. Cela consiste en un processus complet d'aide à la décision. Cependant, il s'avère complexe à cause de ses nombreuses tâches citées plus haut. C'est pourquoi nous avons conçu un système multi-agents où les différentes tâches en question sont assimilées à des services offerts par les différents agents (section 4.3). La complexité des nombreuses tâches du processus que nous proposons est ainsi prise en charge par un travail collaboratif développé par les différents agents du système.

Les résultats des différents travaux que nous présentons concernent quelques pistes de recherche intéressantes, mais elles ne sont en rien exhaustives. Nous avons souvent recourru aux techniques de fouille de données. En effet, au-delà de leur capacité d'analyse, celles-ci doivent pouvoir contribuer à la modélisation multidimensionnelle des données complexes. Par exemple, des techniques de fouille de données telles que les arbres de décision pourraient explorer les données complexes, déjà structurées dans une base de données relationnelle, pour aider à leur modélisation multidimensionnelle. Dans le processus décisionnel complet des données complexes qui vient d'être décrit, il apparaît clairement que les techniques de fouille de données peuvent être utilisées "classiquement" pour extraire des connaissances à partir des données complexes -sachant que cette utilisation est loin d'être triviale et constitue en soi un champ de recherche-, mais aussi comme :

– des techniques d'aide à la modélisation multidimensionnelle des données avec l'extraction de métadonnées,

– des techniques permettant de définir de nouveaux opérateurs OLAP pour les données complexes,

– des méthodes d'extraction de connaissances à partir de données multidimensionnelles.

4.1. *Intégration des données complexes*

4.1.1. *Principe*

Cette phase de notre démarche consiste à intégrer physiquement des données complexes et hétérogènes, éventuellement issues de sources variées, dans une base de données relationnelle jouant le rôle d'un sas à un entrepôt de données. Pour cela, nous proposons un processus de modélisation en trois phases (modèle conceptuel, logique, puis physique) [DAR 02]. Il est important de noter que notre objectif n'est pas seulement de stocker les données, mais aussi de les préparer pour l'analyse, ce qui est encore plus complexe qu'une tâche d'ETL classique.

Nous avons sélectionné XML en tant que formalisme pivot des volets logique et physique de notre processus de modélisation, et ce pour plusieurs raisons. Tout d'abord, XML encapsule à la fois les données et leur schéma, soit implicitement, soit dans une définition de ce schéma. Cette représentation se retrouve dans les entrepôts, qui stockent à la fois des données et des métadonnées qui les décrivent. Comme la complexité de nos données impose lors de leur modélisation de les décrire par des informations supplémentaires, XML est donc particulièrement adapté à nos besoins.

Nous bénéficions par ailleurs de la flexibilité, de l'extensibilité et de la richesse du modèle de données semi-structurées. De plus, puisque des documents XML peuvent facilement être stockés dans une base de données conventionnelle [AND 00, KAP 00], nous tirons aussi avantage de la structuration des données et de l'efficacité de requêtage des systèmes relationnels. Par ailleurs, les bases de données natives XML se développent rapidement, ce qui nous permettrait de migrer nos données dans ces systèmes facilement si nécessaire. De cette manière, nous profitons des avantages du modèle structuré et du modèle semi-structuré en adoptant un format XML.

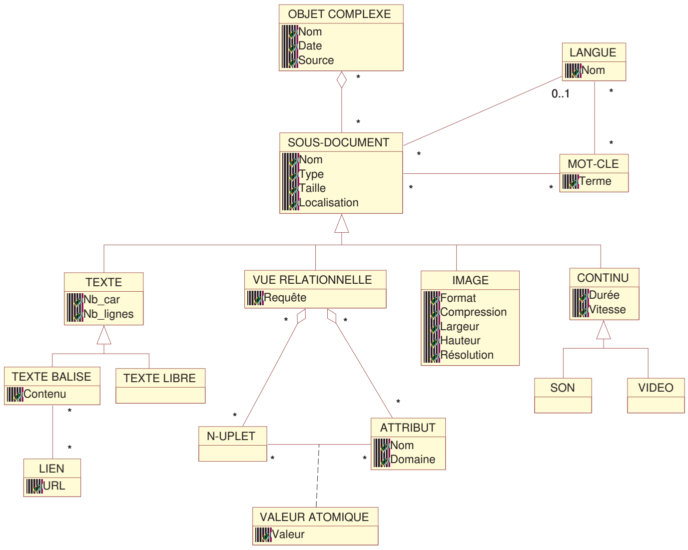

FIG. 2. *Modèle conceptuel d'un objet complexe*

4.1.2. *Modèle conceptuel*

Notre modèle conceptuel UML unifié (Figure 2) représente un objet complexe qui est une superclasse des données à traiter [MIN 01]. Un objet complexe peut être composé d'un ou de plusieurs sous-documents pouvant avoir chacun une langue et des mots-clefs associés. Chaque sous-document a un type identifié qui peut ête un texte libre ou balisé avec des liens, une vue relation, une image, un son ou une vidéo. Un objet complexe peut permettre de décrire, par exemple, une page web constituée de différentes portions de textes, d'images et de données issues de bases de données. Il faut souligner que notre objectif est ici de proposer une structure de données générale : la liste des attributs de chacune des classes de ce diagramme est volontairement non-exhaustive. Nous nous sommes en effet focalisés sur les principales caractéristiques des données complexes qui peuvent être exploitées lors d'une interrogation de la base de données.

4.1.3. *Modèle logique*

Notre modèle conceptuel peut être traduit directement en schéma XML, qu'il soit exprimé à l'aide d'une DTD ou à l'aide du langage XML-Schema. Nous avons envisagé d'utiliser la méthode XMI [COV 01] pour nous assister dans ce processus de traduction UML-XML, mais étant donné la relative simplicité de nos modèles, nous avons pu procéder directement. A titre d'exemple, nous montrons dans les figures 3 et

4 comment les classes *SOUS-DOCUMENT*, *LANGUE* et *MOT-CLE* se traduisent en DTD et XML-Schema, respectivement.

```
<!ELEMENT SOUS-DOCUMENT (Nom, Type, Taille, Localisation, LANGUE?,
MOT-CLE*, (TEXTE | VUE_RELATIONNELLE | IMAGE | CONTINU))>
  <!ELEMENT Nom PCDATA #REQUIRED>
  <!ELEMENT Type PCDATA #REQUIRED>
  <!ELEMENT Taille PCDATA #REQUIRED>
  <!ELEMENT Localisation PCDATA #REQUIRED>
  <!ELEMENT LANGUE PCDATA #REQUIRED>
  <!ELEMENT MOT-CLE PCDATA #REQUIRED>
```

FIG. 3. *Extrait du modèle logique d'un objet complexe (DTD)*

```
<xsd :element name="SOUS-DOCUMENT">
  <xsd :complexType>
    <xsd :sequence>
      <xsd :element name="Nom" type="xsd :string"/>
      <xsd :element name="Type" type="xsd :string"/>
      <xsd :element name="Taille" type="xsd :int"/>
      <xsd :element name="Localisation" type="xsd :string"/>
      <xsd :element name="LANGUE" type="xsd :string"
      minOccurs="0" maxOccurs="1"/>
      <xsd :element name="MOT-CLE" type="xsd :string"
      minOccurs="0" maxOccurs="unbounded"/>
    </xsd :sequence>
    <xsd :choice>
      <xsd :element ref="TEXTE"/>
      <xsd :element ref="VUE_RELATIONNELLE"/>
      <xsd :element ref="IMAGE"/>
      <xsd :element ref="CONTINU"/>
    </xsd :choice>
  </xsd :complexType>
</xsd :element>
```

FIG. 4. *Extrait du modèle logique d'un objet complexe (XML-Schema)*

4.1.4. *Modèle physique*

Pour terminer, le modèle logique est instancié en modèles physiques qui sont des documents XML. A partir du modèle logique (DTD ou schéma XML) et des données complexes les documents XML valides sont générés. Ces derniers peuvent finalement être stockés soit dans une base de données native XML, soit dans une base de données relationnelle via un processus de *mapping*. En l'occurrence, nous avons utilisé le SGBDR MySQL pour sa simplicité et ses performances. Cette base de documents XML joue le rôle d'un sas de données qui pourrait alimenter une base de données orientée vers l'analyse. Elle constitue un véritable ODS (*Operational Data Storage*) contenant des données de production dans la terminologie des entrepôts de données, et qui représente le niveau de granularité des données le plus fin. L'ODS est un lieu de stockage temporaire typiquement utilisé dans un processus d'ETL avant l'alimentation de l'entrepôt de données proprement dit.

4.1.5. *Bilan et perspectives*

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. Le choix du langage XML, outre ses nombreux avantages, permettrait à la fouille dans les documents XML d'extraire des informations contenues dans des documents à l'origine de type texte, image, vidéo, son, etc.

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 lien sémantique entre les différents documents. Une de nos perspectives consiste à finaliser cette structuration en complétant le modèle relationnel de la base des documents XML avec des informations sur la sémantique de ces derniers.

4.2. *Modélisation multidimensionnelle et analyse en ligne des données complexes*

4.2.1. *Modélisation multidimensionnelle des données complexes*

La base de données relationnelle que nous préconisons dans notre approche et dans laquelle nous archivons sous forme de documents XML toutes sortes de données complexes permet d'abord une exploitation transactionnelle. Les différents attributs des documents XML étant structurés sous forme de tables relationnelles, il est facile de les mettre à jour et surtout de bénéficier de la puissance du langage d'interrogation SQL. Ces données peuvent alors être agrégées et alimenter une base de données multidimensionnelle (entrepôt, magasin ou cube de données).

Dans notre démarche, nous recommandons une couche supplémentaire de modélisation des données complexes pour mieux les préparer à l'analyse. Les modèles en étoile sont les mieux adaptés. Cependant la construction de tels modèles sur des données complexes n'est pas du tout évidente. Nous proposons dans nos travaux consacrés à cette question, un cadre méthodologique pour la modélisation multidimensionnelle des données complexes [TAN 03]. Un référentiel de données réunit l'ensemble des données définies dans les modèles élaborés lors de la phase d'intégration des données complexes. Il liste de façon exhaustive les données nécessaires dans le modèle multidimensionnel, décrit les caractéristiques de chacune des données en précisant son rôle dans le modèle à créer, aide dans le choix des éléments du modèle (descripteurs, indicateurs) conforme aux objectifs d'analyse et enfin vérifie la cohérence des don-

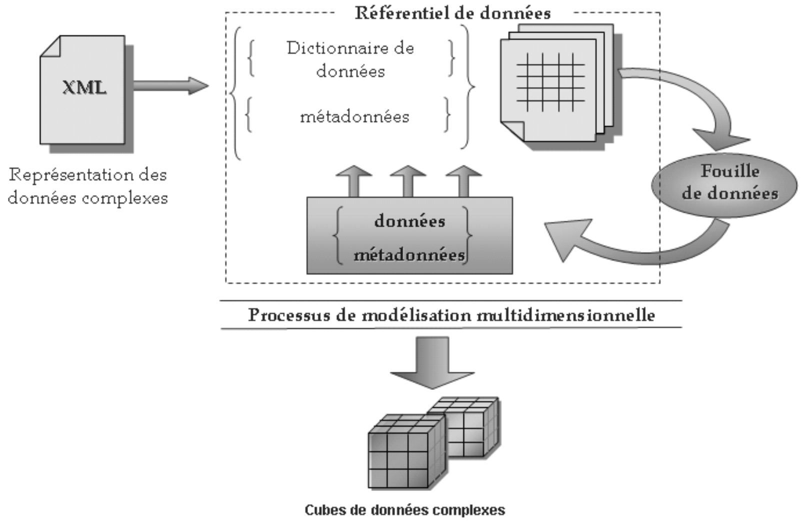

FIG. 5. *Modélisation multidimensionnelle des données complexes*

nées participant au modèle créé. Il est complété par des informations sur les données indiquant leur origine, leur nature et le rôle qu'elles peuvent jouer.

Les données complexes sont décrites par des attributs de bas niveau et possédent également des descripteurs sémantiques. Ces derniers peuvent être obtenus par diverses techniques de fouille de données, de statistique, de traitement d'images ou du signal. La sélection des données 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 à analyser et peut permettre d'enrichir le référentiel par de nouveaux descripteurs sémantiques.

En fonction des objectifs définis par l'utilisateur, les faits à observer sont identifiés et exprimés à l'aide d'indicateurs (mesures) et d'axes d'analyse (dimensions). Le cube de données construit correspond à une vue des données complexes et représente ainsi un espace d'analyse sur lequel il est possible d'effectuer une analyse en ligne (OLAP) ou d'appliquer des techniques de fouilles de données. La construction des cubes de données complexes se fait à la volée en fonction des besoins d'analyse de l'utilisateur. L'information ou la connaissance extraite lors des ces études est capitalisée sous forme de métadonnées et archivées dans le référentiel de données d'aide à la modélisation multidimensionnelle que nous définissons dans le cadre méthodologique cité ci-dessus.

La complexité des données nécessite des opérateurs spécifiques d'agrégation, de navigation ou même d'extraction de connaissance. Une nouvelle approche consiste à développer des opérateurs OLAP pertinents pour les données complexes, basés sur des techniques de fouille.

4.2.2. *Nouveaux opérateurs pour l'analyse en ligne des données complexes*

Les fonctions classiques d'agrégation d'OLAP sont basés sur des opérateurs prévus pour des données numériques avec de fortes propriétés d'additivité (somme, produit, moyenne, max, min, compte…) et sont donc inadaptés aux données complexes. Lorsque la granularité des données est le document XML, les techniques de classification (*clustering*) peuvent être utiles. Les classes des documents XML regroupés correspondent alors à des agrégats et résument ainsi l'information contenues dans les données complexes. Nous avons construit un nouvel opérateur d'analyse en ligne basé sur la CAH (Classification Ascendante Hiérarchique) pour résumer l'information contenue dans les données complexes dans le cadre des travaux de [BEN 03]. Grâce à une API (*Application Programming Interface*), l'utilisateur peut visualiser différents regroupements possibles. Il décide alors du nombre de classes (agrégats) qu'il souhaite en se basant sur un indicateur de qualité des classes que nous avons conçu. La démarche que nous avons adoptée respecte l'esprit d'OLAP. L'utilisateur navigue dans les données et extrait l'information correspondant à ses besoins. L'extension des opérateurs OLAP est faisable et permet des combinaisons avec des techniques de fouille de données pour réaliser des analyses exploratoires tout en exploitant des relations de causalité entre les informations contenues dans les données complexes.

4.2.3. *Bilan et perspectives*

La modélisation multidimensionnelle des données complexes n'est pas simple, mais demeurre possible. Elle s'appuie sur un cadre méthodologique qui apporte une aide effective à l'utilisateur qui se doit de bien connaître les données. L'application des techniques de fouille de données à ce stade lui permet de les enrichir. L'information extraite lui prodigue des renseignements pertinents dans le ciblage des objectifs d'analyse. Le cadre multidimensionnel lui offre un contexte d'analyse. L'adaptation des opérateurs OLAP est nécessaire. Nous envisageons d'améliorer le nouvel opérateur d'agrégation que nous avons créé et d'étendre cette approche à d'autres opérateurs. D'autre part, nous travaillons sur l'utilisation de la multidimensionnalité des données dans les algorithmes d'extraction des connaissances.

4.3. ***Système multi-agents pour l'entreposage de données complexes***

4.3.1. *Motivation*

On peut décomposer le processus d'intégration de données complexes décrit en section 4.1 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 tel système peut être vu comme un système multi-agents dans lequel seront définis les agents capables de réaliser une telle intégration de données complexes.

4.3.2. *Système Multi-agents Agents : Définitions et objectifs*

Un système multi-agents 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 communicant entre eux. Le point clé des systèmes multi-agents réside dans la formalisation de la coordination entre les agents [KLU 01].

Ainsi, un SMA est un système composé d'un environnement ; d'un ensemble d'objets situés dans l'environnement pouvant être perçus, créés, détruits et modifiés par les agents ; d'un ensemble d'agents capables de percevoir, produire, consommer, transformer et manipuler les objets de l'environnement ; d'un ensemble de communications unissant les agents et enfin d'un administrateur chargé de contrôler l'activation des agents. Les SMA se doivent de respecter les normes de programmation définies par la FIPA (*Foundation for Intelligent Physical Agents*) [1].

4.3.3. *Application des SMA à l'intégration des données complexes*

Nous proposons dans le cadre de l'entreposage de données complexes et grâce aux SMA une approche intelligente pour l'intégration de données complexes [BOU 03b]. Notre approche se situe dans l'environnement distribué qu'est le Web, qui est un excellent fournisseur de données complexes. Une fois l'environnement identifié, il faut 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. Dans ce contexte et en respectant la modélisation du processus d'intégration des données complexes décrite dans la section 4.1, nous définissons les événements 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-même 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.

Il faut maintenant définir de manière précise les différentes tâches nécessaires au processus d'intégration des objets complexes.

1) *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.

1. FIPA. *The Foundation for Intelligent Physical Agents.* http ://www.fipa.org, 2002

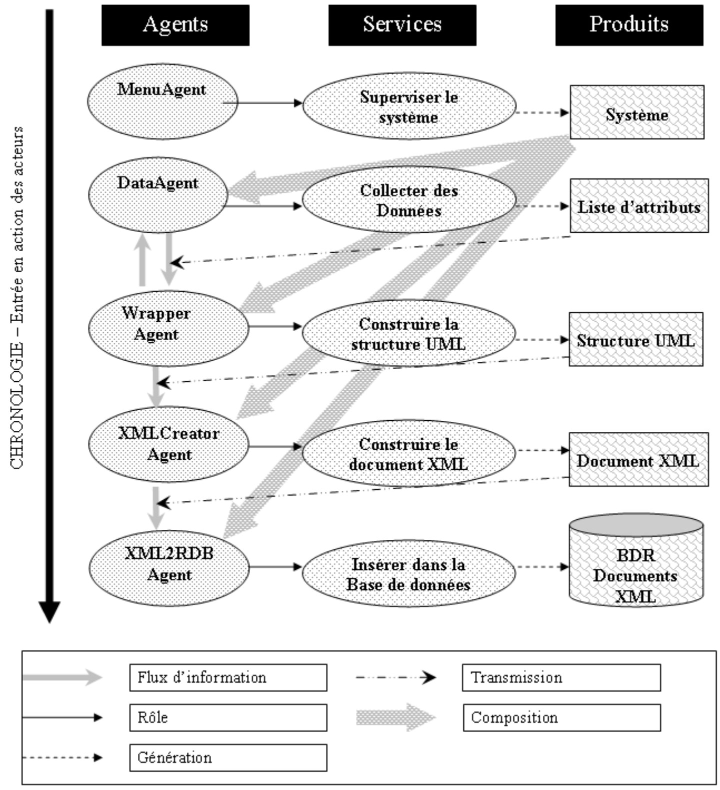

FIG. 6. *Architecture du système SMAIDoC*

2) *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.

3) *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.

Tous ces éléments ont été définis à l'aide d'une grammaire développée dans [BOU 03b].

Après avoir défini tous les éléments nécessaires à la création d'un SMA, nous avons développé un Système Multi-agents pour l'Intégration de Données Complexes baptisé SMAIDoC [2] [BOU 03c] basé sur une plate-forme d'agents génériques (Fig 6). Nous avons instancié cinq agents appelés respectivement *MenuAgent*, *DataAgent*,

2. SMAIDoC : "http ://bdd.univ-lyon2.fr/logiciels.php ?id=4"

WrapperAgent, *XMLCreator* et *XML2RDBAgent* offrant des *services* permettant l'intégration des données complexes. Chacun de ces services engendre des *produits*.

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 [CLE 03b]. Lorsque l'utilisateur choisit un site dans lequel se trouvent des données complexes, 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*. *XMLCreator* 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.

4.3.4. *Bilan et Perspectives*

L'utilisation des systèmes multi-agents dans le processus d'intégration de données complexes 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. Notre objectif consiste à étendre les possibilités de SMAIDoC aux tâches de recueil et d'analyse des données complexes. Cet objectif peut être atteint grâce à l'architecture évolutive de SMAIDoC. 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, nous pouvons 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 [CHA 97b] ou de fouille de données.

4.4. *Auto-administration des données complexes*

4.4.1. *Problématique*

Si le problème de la sélection d'un ensemble d'index optimal pour une base de données a été étudié depuis les années 70, le concept d'auto-administration n'a fait son apparition que dans la fin des années 90 [3]. Les systèmes auto-administratifs ont pour objectif de s'administrer et de s'adapter eux-mêmes, automatiquement, sans perte (ou même avec un gain) de performance par rapport à une gestion assurée par un opérateur humain. Actuellement, la recherche dans ce domaine se concentre dans ce qui est considéré comme une première étape sur la conception physique des bases de données (index, vues matérialisées, etc.).

3. http ://research.microsoft.com/dmx/autoadmin/

4.4.2. *Extraction de motifs fréquents pour la sélection automatique d'index*

Depuis quelques années, l'idée est avancée d'utiliser les techniques de fouille de données pour extraire des connaissances utiles des données elles-même pour leur administration [CHA 98]. Cependant, peu de travaux de recherche ont été entrepris dans ce domaine jusqu'ici. C'est pourquoi nous avons conçu et réalisé un outil qui utilise la fouille de données pour proposer une sélection (configuration) d'index pertinente.

Partant de l'hypothèse que l'utilité d'un index est fortement corrélée à la fréquence de l'utilisation des attributs correspondants dans l'ensemble des requêtes d'une charge donnée, la recherche de motifs fréquents [AGR 93] nous a semblé appropriée pour mettre en évidence cette corrélation et faciliter le choix des index à créer.

L'approche que nous proposons [AOU 03a, AOU 03b], dont le principe est représenté dans la figure 7, exploite le journal des transactions (fichier log) pour extraire une configuration d'index. Les requêtes présentes dans le journal constituent une charge, qui est traitée par un analyseur de requêtes qui extrait tous les attributs indexables, c'est à dire ceux qui sont présents dans les clauses WHERE, ORDER BY, GROUP BY et HAVING des requêtes SQL [CHA 97a].

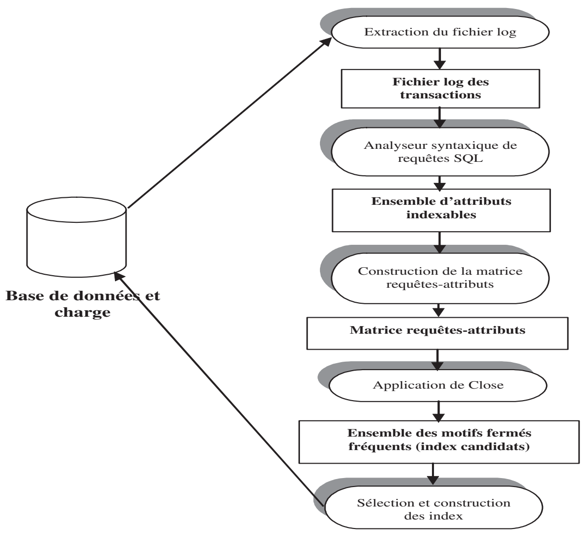

FIG. 7. *Séléction automatique d'index grâce à l'extraction de motifs fréquents*

Nous construisons ensuite une matrice dite "requêtes-attributs" qui permet d'associer à chaque requête les attributs indexables qu'elle comporte et qui représente le contexte d'extraction des motifs fréquents. Pour obtenir ces derniers, nous avons choisi l'algorithme Close [PAS 99a], car il calcule l'ensemble des fermés (au sens de la connexion de Galois [PAS 99b]) fréquents, qui est un générateur pour tous les motifs fréquents et leur support. Dans la plupart des cas, le nombre des fermés fréquents est sensiblement moins important que la totalité des motifs fréquents obtenue en sortie des algorithmes classiques tels qu'Apriori [AGR 94]. Dans notre contexte, l'utilisation de Close nous permet d'obtenir une configuration d'index candidats moins volumineuse (tout en demeurant aussi significative) en un temps de calcul plus court. Finalement, nous sélectionnons au sein de la configuration d'index candidats les plus pertinents à l'aide de diverses stratégies et les créons physiquement.

4.4.3. *Bilan et perspectives*

Ces premières recherches sont prometteuses, tant au niveau de notre approche originale par rapport à celle retenue dans le projet AutoAdmin de Microsoft [CHA 97a] qu'à celui des résultats expérimentaux obtenus. Cependant, il reste beaucoup de travail à accomplir pour les appliquer au contexte spécifique des entrepôts de données, dont les index spécifiques (index *bitmap*, par exemple) diffèrents des index classiques dérivés des B-arbres habituellement implantés dans les SGBD relationnels que nous avons exploités jusqu'ici. Il reste également à prendre en compte la spécificité des données complexes, en termes d'indexation, ainsi que le couplage de notre approche avec d'autres techniques d'optimisation des performances (vues matérialisées, gestion de cache, regroupement physique, etc.). En effet, dans le contexte des entrepôts de données, c'est principalement en conjonction avec d'autres structures physiques (principalement les vues matérialisées) que l'indexation permet d'obtenir des gains de performance significatifs [GUP 99, AGR 01].

5. Conclusion et perspectives

A l'heure actuelle, l'enjeu pour les technologies des entrepôts de données, de l'analyse en ligne et de la fouille de données, est de pouvoir prendre en compte des données complexes, de formats, sources ou sémantiques différents. Ces différentes technologies ont largement fait leurs preuves quand il s'agit de données "classiques" mais de nouveaux problèmes théoriques et méthodologiques se posent quand il s'agit de données complexes.

Dans cet article nous avons tenté de décrire un processus complet d'entreposage de données complexes pour l'aide à la décision, tout en soulevant les problèmes liés à l'intégration, la structuration, la modélisation multidimensionnelle des données complexes et à leur exploration par des techniques d'analyse en ligne ou de fouille de données. Le processus d'intégration que nous proposons permet de modéliser les données complexes dans un format unifié en XML et de les stocker dans une base de données relationnelle. Il s'agit maintenant de compléter le modèle relationnel de la base pour prendre en compte le contenu sémantique des données. A des fins d'analyse, nous avons montré que la modélisation multidimensionnelle et l'analyse en ligne des données complexes n'étaient pas simples, mais néamoins possibles. Les méthodes de fouille de données constituent alors une aide pour la définition du modèle multidimensionnel et il s'avère nécessaire d'enrichir les opérateurs OLAP pour prendre en compte la complexité des données. L'utilisation des systèmes multi-agents dans le processus décisionnel apporte une certaine flexibilité à notre démarche d'entreposage de données complexes avec une architecture évolutive. Le système comporte d'ores et déjà la phase d'intégration, il s'agit maintenant de l'étendre à tous les aspects du processus d'entreposage des données complexes.

Nous avons proposés un cadre méthodologique pour l'intégration et la modélisation multidimensionnelle des données complexes, même s'ils ne constituent qu'une partie des réponses possibles aux problèmes évoqués. D'autres solutions existent et de nombreuses questions restent en suspens. Comment faire de la fouille de données sur des documents XML ou sur des données multidimensionnelles ? Comment enrichir encore les opérateurs OLAP ?

6. Bibliographie

[AGR 93] AGRAWAL R., IMIELINSKI T., SWAMI A. N., « Mining Association Rules between Sets of Items in Large Databases », *SIGMOD Record*, vol. 22(2), 1993, p. 207-216.

[AGR 94] AGRAWAL R., SRIKANT R., « Fast Algorithms for Mining Association Rules », *20th International Conference on Very Large Data Bases (VLDB 1994), Santiago, Chile*, 1994, p. 487-499.

[AGR 01] AGRAWAL S., CHAUDHURI S., NARASAYYA V. R., « Materialized View and Index Selection Tool for Microsoft SQL Server 2000 », *2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, USA*, , 2001.

[AND 00] ANDERSON R., BIRBECK M., KAY M., LIVINGSTONE S., LOESGEN B., MARTIN D., MOHR S., OZU N., PEAT B., PINNOCK J., STARK P., WILLIAMS K., *Professional XML Databases*, Wrox Press, 2000.

[AOU 03a] AOUICHE K., DARMONT J., GRUENWALD L., « Frequent itemsets mining for database auto-administration », *7th International Database Engineering and Application Symposium (IDEAS 03), Hong Kong*, July 2003, p. 98-103.

[AOU 03b] AOUICHE K., DARMONT J., GRUENWALD L., « Vers l'auto-administration des entrepôts de données », *Revue des Nouvelles Technologies de l'Information*, , n° 1, 2003, p. 1-12.

[BAR 03] BARIL X., BELLAHSENE Z., « *Designing and Managing an XML Warehouse* », p. 455-473, Addison Wesley, 2003, in XML Data Management.

[BEL 98] BELLAHSENE Z., « Structural View Maintenance in Data warehousing Systems », 14^{imes} *Journées Bases de Données Avancées, BDA98, Tunisie*, October 1998.

[BEN 03] BENMESSAOUD R., « Construction d'un opérateur d'analyse en ligne des données complexes basé sur une technique de fouille de données », Mémoire de DEA, Université Lumière Lyon 2, 2003.

[BER 02] BERRY M., « A Comprehensive Survey of Text Mining », *Second SIAM Conference on Data Mining, Text Mining Workshop*, 2002.

[BER 03] BERGLUND A., BOAG S., CHAMBERLIN D., FERNANDEZ M., KAY M., ROBIE J., SIMEON J., « XML Path Language (XPath) 2.0 », http ://www.w3.org/TR/xpath20/, 2003, W3C Working Draft.

[BOA 03] BOAG S., CHAMBERLIN D., FERNANDEZ M., FLORESCU D., ROBIE J., SIMEON J., « XQuery 1.0 : An XML Query Language », http ://www.w3.org/TR/xquery/, 2003, W3C Working Draft.

[BOU 03a] BOURRET R., « XML and Databases », http ://www.rpbourret.com/xml/XMLAndDatabases.htm, 2003.

[BOU 03b] BOUSSAID O., BENTAYEB F., DARMONT J., « A Multi-Agent System-Based ETL Approach for Complex Data », *10th ISPE International Conference on Concurrent Engineering : Research and Applications (CE 03), Madeira Island, Portugal*, July 2003, p. 49-52.

[BOU 03c] BOUSSAID O., BENTAYEB F., DUFFOUX A., CLERC F., « SMAIDoC : A Multi-Agent System-Based for Complex Data Integration », *First International Conference on Holotic and Multi-Agent System (Holomas 03), Prague, Republic Czech*, 2003, p. 201-212.

[CER 91] CERI S., WIDOM J., « Deriving Production Rules for Incremental View Maintenance », *Seventeenth International Conference on Very Large Data Bases, Barcelona, Spain*, September 1991, p. 577-589.

[CHA 97a] CHAUDHURI S., NARASAYYA V. R., « An Efficient Cost-Driven Index Selection Tool for Microsoft-SQL Server », *23rd International Conference on Very Large Data Bases (VLDB 1997), Athens, Greece*, 1997, p. 146-155.

[CHA 97b] CHAUDHURI S., DAYAL U., « An overview of data Warehousing and OLAP Technology », *In ACM-SIGMOD, Record 26(1)*, , 1997.

[CHA 98] CHAUDHURI S., NARASAYYA V. R., « AutoAdmin 'What-if' Index Analysis Utility », *1998 ACM SIGMOD International Conference on Management of Data, Seattle, USA*, 1998, p. 367-378.

[CHE 00] CHEN Q., U. DAYAL M. H., « An OLAP-based Scalable Web Access Analysis Engine », *Second International Conference on Data Warehousing and Knowledge Discovery, London*, September 2000.

[CHE 01] CHEN M., ZHU Q., CHEN Z. X., « An integrated interactive environment for knowledge discovery from heterogeneous data resources », *Information and Software Technology, 43 (8)*, , 2001, p. 487-496.

[CLA 99] CLARK J., « XSL Transformations (XSLT) Version 1.0 », http ://www.w3.org/TR/xslt, 1999, W3C Recommendation.

[CLE 03a] CLECH J., ZIGHED D., « Data Mining et analyse des cv : une expérience et des perspectives », *Extraction des Connaissances et Apprentissage (RSTI série RIA-ECA), 17(1-2-3)*, , 2003, p. 189-200.

[CLE 03b] CLERC F., DUFFOUX A., BENTAYEB F., BOUSSAID O., « Un Systeme Multi-Agent pour l'intégration des données complexes », *Revue des Nouvelles Technologies de l'Information*, , n° 1, 2003, p. 13-24.

[CLU 99] CLUET S., « Intégration de données hétérogènes », http ://www-rocq.inria.fr/c̃luet/, 1999.

[COD 93] CODD E., « Providing OLAP (On-Line Analytical Processing) to user-analysts : an IT mandate », rapport, 1993, E.F. Codd and Associates.

[COV 01] COVER R., « XML Metadata Interchange (XMI) », http ://xml.coverpages.org/xmi.html, 2001.

[DAR 02] DARMONT J., BOUSSAID O., BENTAYEB F., RABASEDA S., ZELLOUF Y., « *Web multiform data structuring for warehousing* », vol. 22 de *Multimedia Systems and Applications*, p. 179-194, Kluwer Academic Publishers, October 2002, In Multimedia Mining : A Highway to Intelligent Multimedia Documents, C. Djeraba, ed.

[DEL 95] DELANOY R. L., SASIELA R. J., « Machine learning for a toolkit for image mining », rapport n° 1017, 1995, Lincoln Laboratory, MIT.

[DJE 00a] DJERABA C., « Image Access and Data Mining : An Approach », *International Conference on Knowledge Discovery and Data Mining (PKDD 2000), Lyon, France*, September 2000, p. 375-380.

[DJE 00b] DJERABA C., *Multimedia Mining : A highway to intelligent Multimedia Documents*, Kluwer Academic Publichers, 2000.

[ENG 01] ENGELS R., BREMDAL B., JONES R., « Semantic Web Mining Workshop », *Workshop at ECML/PKDD-2001, Freiburg, Germany*, September 2001.

[FIN 88] FINKELSTEIN S. J., SCHKOLNICK M., TIBERIO P., « Physical Database Design for Relational Databases », *TODS*, vol. 13, n° 1, 1988, p. 91-128.

[FLA 00] FLACH P., LAVRAC N., « The role of feature construction in inductive rule learning », *17th International Conference on Machine Learning, Workshop on Attribute-Value and Relational Learning : crossing the boundaries, Stanford*, 2000, p. 1-11.

[FOS 02] FOSCHI P., KOLIPPAKKAM D., LIU H., MANDVIKAR A., « Feature Extraction for Image Mining », *International Workshop on Multimedia Information Systems (MIS 2002)*, 2002.

[GAR 93] GARDARIN G., *Maîtriser les Bases de Données et les Langages*, Eyrolles, 1993.

[GHA 00] GHANI R., JONES R., MLADENIC D., NIGAM K., SLATTERY S., « Data Mining on Symbolic Knowledge Extracted from the Web », *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, USA*, August 2000.

[GOA 00] GOASDOUÉ F., LATTÈS V., ROUSSET M., « The use of CARIN language and algorithms for information integration : the PICSEL system », *International Journal of Cooperative Information Systems (IJCIS)*, vol. 9(4), 2000, p. 383-401.

[GOI 97] GOIL S., CHOUDHARY A., « High Performance Data Mining Using Data Cubes on Parallel Computer », *Journal of Data Mining and Knowledge Discovery, 1(4)*, , 1997, p. 391-417.

[GOI 01] GOIL S., CHOUDHARY A., « PARSIMONY : An Infrastructure for parallel Multidimensional Analysis and Data Mining », *Journal of Data Mining and Knowledge Discovery*, vol. 61(3), 2001, p. 285-321.

[GUP 95] GUPTA A., MUMICK I. S., « Maintenance of Materialized Views : Problems, Techniques and Applications », *Data Engeneering Bulletin*, , 1995.

[GUP 99] GUPTA H., « Selection and maintenance of views in a data warehouse », PhD thesis, Stanford University, 1999.

[HAN 97] HAN J., « OLAP Mining : An Integration of OLAP with Data Mining », *IFIP Conference on Data Semantics, Leysin, Switzerland*, October 1997, p. 1-11.

[HAN 98] HAN J., « Toward On-line Analytical Mining in Large Databases », *SIGMOD Record 27(1)*, , 1998, p. 97-107.

[HAR 96] HARINARAYAN V., RAJARAMAN A., ULLMAN J., « Implementing Data Cubes Effciently », *IEEE Transactions on Data Engineering*, 1996, p. 1-25.

[INM 96] INMON W., *Building the data Data Warehouse*, John Wiley & Sons, 1996.

[JAL 02] JALAM R., CHAUCHAT J., « Pourquoi les n-grammes permettent de classer des textes ? Recherche de mots-clefs pertinents à l'aide des n-grammes caractéristiques », *Sixth International Conference on Textual Data Statistical Analysis, France*, vol. 1, 2002, p. 381-390.

[JEN 01] JENSEN M., MOLLER T., PEDERSEN T., « Specifying OLAP Cubes on XML data », *SSDBM*, 2001.

[KAH 02] KAHAN J., KOIVUNEN M., PRUD'HOMMEAUX E., SWICK R., « Annotea : An Open RDF Infrastructure for Shared Web Annotations », *The WWW10 Inter. Conference*, Hong-Kong, May 2002.

[KAP 00] KAPPEL G., KAPSAMMER E., RETSCHITZEGGER W., « X-Ray – Towards Integrating XML and Relational Database Systems », *19th International Conference on Conceptual Modeling (ER 2000), Salt Lake City, USA*, 2000, p. 339-353.

[KHO 96] KHOSHAFIAN S., BAKER B., « *Multimedia and Imaging Databases* », Multimedia Systems and Applications, Morgan Kaufmann Publishers, 1996.

[KIM 00] KIMBALL R., MERZ R., *The data webhousing*, Eyrolles, 2000.

[KLU 01] KLUSCH M., « Information Agent technology for the Internet : A Survey », *Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration, D. Fensel (Ed.)*, vol. 36(3), 2001.

[LAU 01] LAURENT A., « De l'OLAP Mining au F-OLAP Mining », *Journées francophones d'Extraction et Gestion des Connaissances (EGC 2001), Revue Extraction des connaissances et apprentissage (ECA),France*, vol. 1(1-2), Hermès, Janvier 2001, p. 189-200.

[LAY 97] LAYAIDA N., « Madeus : Systeme d'édition et de présentation de documents structurés multimédia », Thèse de doctorat, Université Joseph Fourrier de Grenoble, 1997.

[LIU 98] LIU H., MOTODA H., *Feature Extraction, Construction and Selection : A Data Mining Perspective*, vol. 453, Kluwer Academic Publishers, 1998.

[MAT 90] MATSUYAMA T., HWANG V., *SIGMA : A Knowledge-Based Aerial Image Undeestanding System*, Plenum Press, 1990.

[MEL 02] MELTON J., « XML-Related Specifications (SQL/XML) », rapport n° WG3 :DRS-020 H2-2002-365, August 2002, ISO-ANSI.

[MIN 01] MINIAOUI S., DARMONT J., BOUSSAID O., « Web data modeling for integration in data warehouses », *First International Workshop on Multimedia Data and Document Engineering (MDDE 01), Lyon, France*, July 2001, p. 88-97.

[NAH 00] NAHM U. Y., MOONEY R. J., « Using Information Extraction to Aid the Discovery of Prediction Rules from Text », *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,Boston,USA*, August 2000.

[PAL 00] PALPANAS T., « Knowledge Discovery in Data Warehouses », *SIGMOD Record 29(3)*, , 2000, p. 88-100.

[PAS 99a] PASQUIER N., BASTIDE Y., TAOUIL R., LAKHAL L., « Discovering Frequent Closed Itemsets for Association Rules », *7th International Conference on Database Theory (ICDT 1999), Jerusalem, Israel*, vol. 1540 de *LNCS*, 1999, p. 398-416.

[PAS 99b] PASQUIER N., BASTIDE Y., TAOUIL R., LAKHAL L., « Efficient mining of association rules using closed itemset lattices », *Information Systems*, vol. 24(1), 1999, p. 25-46.

[PIN 01a] PINTO H., HAN J., ET AL. U. D., « Multi-dimensional Sequential Pattern Mining », *Conference on Information and Knowledge Management (CIKM'01), Atlanta, USA*, November 2001.

[PIN 01b] PINTO H., HAN J., ET AL. U. D., « PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth », *International Conference on Data Engineering, Heidelberg, Allemagne*, April 2001, p. 215-224.

[REY 02] REYNAUD C., GIRALDO G., « Vers l'automatisation de la construction de systèmes de médiation pour le commerce électronique », *Journées de l'Action Spécifique Web Sémantique, Paris*, 2002.

[ROU 02] ROUSSET M., « Knowledge Representation for Information Integration », *XIIIth International Symposium on Methodologies for Intelligent Systems (ISMIS 2002), Lyon, France*, vol. 2366 de *LNAI*, Springer Verlag, June 2002, p. 1-3.

[SAR 97] SARAWAGI S., « Indexing OLAP Data », *Data Engineering Bulletin*, vol. 20, n° 1, 1997, p. 36-43.

[SAR 98] SARAWGI S., MEGIDDO R. A. N., « Discovery-driven Exploration of OLAP Data Cubes », *Sixth International Conference on Extending Database Technology (EDBT), Valencia, Spain*, Mars 1998.

[SAR 01] SARAWAGI S., « iDiff : Informative summarization of differences in multidimensional aggregates », *Data Mining And Knowledge Discovery*, vol. 5(4), 2001, p. 213-246.

[TAN 03] TANASESCU A., « Modélisation multidimensionnelle de données complexes », Mémoire de DEA, Université Lumière Lyon 2, 2003.

[TEY 01] TEYTAUD O., JALAM R., « Kernel based text categorization », *12th International Joint Conference on Neural Networks (IJCNN-IEEE), Washington, USA*, 2001.

[THU 03] THUONG T. T., « Modélisation et traitement du contenu des médias pour l'édition et la présentation de documents multimédias », Thèse de doctorat, Institut National Polytechnique de Grenoble, 2003.

[WEI 02] WEIKUM G., MONKEBERG A., HASSE C., ZABBACK P., « Self-tuning Database Technology and Information Services : from Wishful Thinking to Viable Engineering », *28th International Conference on Very Large Data Bases (VLDB 2002), Hong Kong, China*, 2002, p. 20-31.

[WIT 99] WITTEN I., BRAY Z., ET AL M. M., « Text Mining : a new frontier for lossless compression », *Data Compression Conference, Los Alamitos, CA*, IEEE, 1999, p. 198-207.

[YAN 97] YANG K., KARLAPALEM J., LI Q., « Algorithms for materialized view design in data warehousing environment », *Proceedings of the 23rd VLDB*, Athens, Greece, 1997, p. 136–145.

[Y.Z 95] Y. ZHUGE H. GARCIA-MOLINA J. H., WIDOM J., « View Maintenance in a Warehousing Environement », *International Conference on Management of Data*, ACM SIGMOD, June 1995, p. 316-327.

[ZIG 00] ZIGHED D., RAKOTOMALALA R., *Graphes dinduction - apprentissage et data mining*, Hermès, 2000.