Modélisation d'un entrepôt de documents XML

Fatma Abdelhedi*, Landry Ntsama*,**
Gilles Zurfluh*

*IRIT-SIG, Université de Toulouse 1 Capitole
2 rue du Doyen Gabriel Marti, 31042 Toulouse, France
prenom.nom@irit.fr,
http://www.irit.fr/-Equipe-SIG-
**landry.ntsama@ut-capitole.fr

Résumé. Le format XML est aujourd'hui omniprésent dans les organisations et sur le Web. Il facilite le transport et l'échange de données complexes et hétérogènes représentant une information précieuse très peu, voire pas du tout exploitée. Les technologies OLAP et les systèmes d'entrepôts de données actuels permettent l'analyse et le stockage des données transactionnelles issues des bases données relationnelles. Cependant, ces outils ne sont pas adaptés à l'analyse des documents XML du fait de leur structure hiérarchique, ou de leur contenu étant souvent textuel. Nous proposons dans cet article une approche permettant de construire un entrepôt de document XML « centré document » dont le schéma conceptuel est modélisé en utilisant le formalisme UML ; nous présentons aussi une architecture pour l'intégration physique de ces documents dans un environnement XML natif.

1 Introduction

Le format XML facilite la description et l'échange des données complexes provenant de sources hétérogènes. De plus sa structure balisée et auto-descriptive (Fankhauser et Klement, 2003) permet une représentation aisée de différents types de données (structurés et semi-structurés). C'est aujourd'hui un standard de communication tant au sein des organisations que sur le Web. Les systèmes OLAP et les entrepôts de données permettent respectivement l'analyse multidimensionnelle et le stockage orienté sujet des données transactionnelles. Les données stockées sont décrites par un schéma multidimensionnel (Golfarelli et al., 1998), qui permet à un décideur de réaliser des analyses OLAP sur les cubes de données définis dans l'entrepôt. Les systèmes OLAP permettent ainsi une analyse interactive en représentant les données suivant plusieurs perspectives ou dimensions. L'analyse OLAP sur des données relationnelles (structurées) est un processus aujourd'hui parfaitement maîtrisé. Mais ce processus s'avère inadapté pour des documents XML qui présentent une structure hiérarchique profonde et diffèrent des données plates tant au niveau de leur modèle de données que de leur modèles de requêtes (Bordawekar et Lang, 2005).

Nous proposons une nouvelle approche permettant de construire un entrepôt sur des documents XML « centré document » (Pérez et al., 2008) en les intégrant au niveau physique dans

un environnement XML natif. L'objectif est de permettre à un décideur de réaliser des analyses multidimensionnelles sur une collection de documents XML. Dans cette approche, le modèle conceptuel est construit à partir du XML Schéma (XSchema) (W3C-Consortium, 2013) représentant la classe de documents analysée ; il est défini sur la base du formalisme de modélisation orientée objet UML (Unified Modeling Language) ((OMG), 2013). Ce modèle permet de représenter la structure de la classe de documents sous la forme d'un schéma en étoile (ou en flocon) (Kimbal et Ross, 2002). Nous présentons donc dans cet article une architecture pour une intégration physique des documents XML « centré document » dans un entrepôt, notre principale contribution étant la définition d'une approche de modélisation conceptuelle d'un entrepôt de documents XML, basé sur le formalisme de modélisation orienté objet UML à partir d'un XSchema.

La suite de l'article est organisée comme suit. Dans la section 2 nous faisons le point sur les travaux relatifs à notre problématique ; la section 3 présente notre architecture et définit notre démarche de modélisation d'un entrepôt ; et enfin nous concluons dans la section 4 en présentant quelques perspectives liées à nos travaux.

2 Travaux associés

L'analyse OLAP des documents XML soulève des problématiques distinctes de celles liées aux systèmes OLAP traditionnels. Dans (Bordawekar et Lang, 2005), les auteurs passent en revue ces problématiques liées à l'implémentation d'un système OLAP pour les documents XML. De même ils proposent une approche basée sur l'utilisation de « ***modèle d'arbres abstraits*** » pour l'analyse OLAP des documents XML. Ils décrivent de nouveaux opérateurs pour supporter des « agrégations structurées » sur des documents XML, par le biais du langage XQuery.

Dans (Hümmer et al., 2003), les auteurs proposent un modèle de document appelé ***XCube*** et basé sur le format XML. Ce modèle permet de faciliter l'échange et la manipulation de cubes de données contenus dans les entrepôts de données traditionnels. C'est un modèle dynamique qui est défini par trois schémas XML pour décrire les données échangées, à savoir ***XCube-Fact*** pour décrire l'ensemble des faits, ***XCubeDimension*** pour décrire les dimensions et leurs hiérarchies, et ***XCubeSchema*** pour décrire le schéma multidimensionnel.

Les auteurs de (fei Jiang et al., 2007) proposent une structure de données appelée ***IX-Cube*** (Iceberg XML Cube) permettant de faciliter l'application de requêtes multidimensionnelles OLAP sur des données XML. Ce modèle est défini à partir d'un arbre XML représentant la structure des documents dans la collection.

(Boussaïd et al., 2006) proposent un outil appelé ***X-Warehousing*** pour l'entreposage et la construction des cubes XML en utilisant essentiellement des technologies XML. Ils utilisent le modèle conceptuel multidimensionnel (MCM) pour représenter les besoins d'analyses de l'utilisateur. Ce MCM est traduit en XSchema pour représenter la structure logique de l'entrepôt. Ils utilisent des « arbres d'attributs » pour comparer les documents de la source au XSchema obtenu, afin d'identifier les documents dont la structure est proche des besoins du décideur.

L'ensemble des travaux ci-dessus concerne essentiellement l'analyse des documents XML « centrés-données » (Pérez et al., 2008), toutefois il est envisageable de les adapter à notre contexte qui est l'analyse OLAP des documents XML « centrés-documents ». Seulement, certains de ces travaux concernent le traitement des données relationnelles et utilisent le format

XML pour définir un cube sur ces données (Hümmer et al., 2003; Niemi et al., 2002). D'autres travaux, bien que définissant un cube à partir d'une source XML, présentent les limites suivantes :

- Ils ne proposent pas de modèle multidimensionnel représentant la structure des documents analysés
- Ils se basent sur les instances de documents constituant la source pour définir les modèles d'arbres XML qu'ils utilisent, or nous basons notre approche sur les XSchemas associés aux documents de la source
- Les modèles d'arbres qu'ils ont définis représentent la structure logique servant à définir les cubes de données XML, or nous utilisons un XSchema pour représenter la structure logique de notre entrepôt.

Les auteurs de (Park et al., 2005) proposent un outil nommé ***XML-OLAP*** pour l'analyse multidimensionnelle des documents XML « centré document ». Cet outil construit un cube XML à partir d'un entrepôt dans lequel les faits et les dimensions sont représentés par des documents XML différents. Le cube est construit et analysé au moyen des langages XQuery utilisé pour le calcul des mesures et l'identification des dimensions. Toutefois ils utilisent l'approche définie par (Nassis et al., 2005) pour le modèle conceptuel du cube.

3 Entrepôts de documents XML : modélisation conceptuelle

Notre objectif est de permettre l'analyse multidimensionnelle de documents XML issus du Web en reprenant les principes d'entreposage basés sur les faits et dimensions (Kimbal et Ross, 2002). Il s'agit donc de construire un entrepôt de documents décrit par un schéma en étoile (ou en flocon) et permettant aux décideurs d'effectuer des opérations multidimensionnelles (OLAP). La Figure 1 représente l'architecture globale de notre système. Cette architecture permet l'intégration physique des documents XML « centré document » dans un entrepôt, ainsi que leur analyse OLAP.

Nous avons développé un modèle multidimensionnel qui permet de décrire un entrepôt de documents XML sous la forme d'un schéma en étoile nommé StarCD. Ce schéma d'objets complexes s'appuie sur des diagrammes de classes UML pour représenter le fait et ses dimensions. Il est élaboré à partir de l'analyse des XSchemas des documents XML composant la source. En effet, cette source contient une collection de documents de même thématique, donc présentant des structures proches. Nous considérons dans la suite de l'article que chaque document dans cette collection est valide, et donc associé à un XSchema propre.

De nombreux travaux ont traité de la modélisation multidimensionnelle des documents XML se basant sur UML, notamment (Li et An, 2005) ou encore (Nassis et al., 2005). Cependant, aucune des approches proposées dans ces travaux ne conserve la structure des documents analysés dans le schéma présenté au décideur. Les auteurs de (Nassis et al., 2005) proposent néanmoins dans leur modèle ***xFact*** une représentation de la dite structure. Toutefois celle-ci est difficilement identifiable par le décideur. Dans notre approche, nous supposons que le décideur connaît la structure des documents qu'il veut analyser, aussi nous lui permettons de retrouver cette structure de façon partielle dans le schéma de l'entrepôt, compte tenu de ses besoins d'analyse.

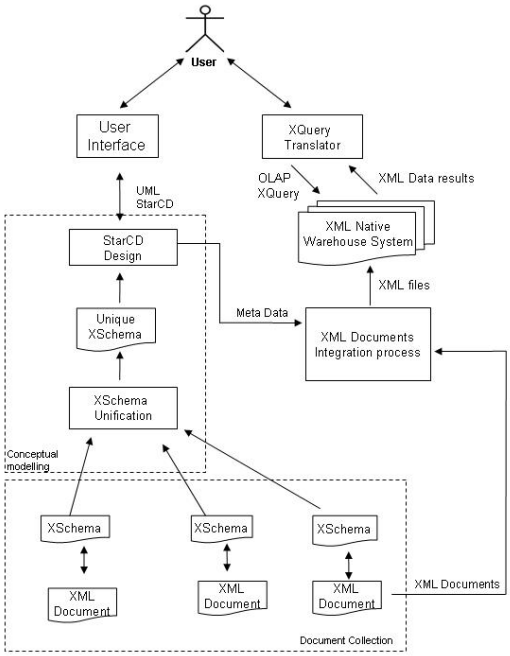

FIG. 1 – *Architecture globale.*

3.1 Unification des structures

Dans la Figure 1, le cadre « Conceptual Modeling » représente le processus de modélisation conceptuelle partant de l'unification des schémas de la source. En effet, bien qu'étant de même thématique, les documents ont généralement des structures différentes. Pour résoudre ce problème, un processus d'unification va traiter automatiquement l'ensemble des XSchemas pour en extraire une (ou plusieurs) structure(s) unifiée(s). Cette structure unifiée correspond à un XSchema dont la racine représente le sujet d'analyse. Le processus d'unification n'est pas détaillé dans cet article, aussi nous invitons les lecteurs à se référer à (Messaoud et al., 2011) ou (Janga et Davis, 2013) pour plus de précisions.

Une fois le processus d'unification réalisé, la source contient des documents regroupés en classes. Tous les documents appartenant à une même classe sont décrits par un XSchema unique ; le nom de la classe de document est situé au premier niveau du XSchema comme le montre la Figure 2. Celle-ci représente un XSchema unifié partiel relatif à une classe de document XML sur une collection d'articles scientifiques.

3.2 Modélisation UML multidimensionnelle

Tout XSchema unifié est transformé en un Diagramme de Classes UML (noté SourceCD pour Source Class Diagram) ; il se présente sous la forme d'un graphe arborescent dont la

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name = "Paper">
<xs:complexType>
<xs:sequence>
<!--Types simples -->
<xs:element name="Title" type="xs:string"/>
<!--Types complexes -->
<xs:element name = "CoAuthors">
<xs:complexType>
<xs:sequence>
<xs:element name="Author" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name = "Name" type="xs:string"/>
<xs:element name = "FirstName" type="xs:string"/>
<xs:element name = "Affiliation" type="xs:string" minOccurs="0"/>
<xs:element name = "Mail" type="xs:string" minOccurs="0"/>
</xs:sequence>
<!--Attributs auteur -->
<xs:attribute name = "H-Index" type="xs:integer" use ="required"/>
</xs:complexType>
</xs:element>
<xs:element name = "AffiliationGroup" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Abstract" type="xs:string" minOccurs="0"/>
<xs:element name="Résumé" type="xs:string" minOccurs="0"/>
<!--    -->
```

FIG. 2 – *XSchéma de la classe de document Paper.*

racine correspond à l'élément de plus haut niveau : la classe de documents à analyser. Tout « élément » du XSchema est transformé en une classe dans le SourceCD avec son nom et ses attributs ; les éléments imbriqués sont reliés à l'élément hiérarchiquement supérieur par des relations de composition. Les éléments associés au terme « Choice » sont traduits par des liens d'héritage. La Figure 3 montre la simplicité du modèle utilisé pour formaliser un XSchema.

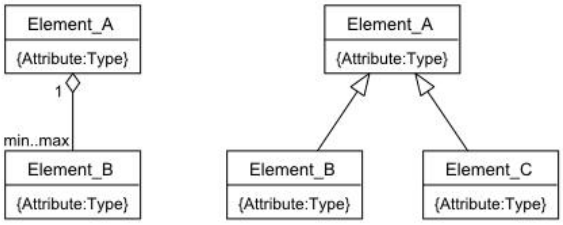

FIG. 3 – *Modèle UML.*

Notons que les liens Key et KeyRef entre éléments imbriqués sont traduits par des relations de référence, mais ils ne sont pas pris en compte dans nos travaux. La Figure 4 présente le diagramme de classes (SourceCD) décrivant une classe d'articles scientifiques et correspondant à l'extrait du XSchema de la Figure 2.

La transformation d'un XSchema en un diagramme de classe (SourceCD) a été automatisée. Elle fait correspondre les différents constituants du XSchema avec les concepts d'un diagramme de classe. La Figure 5 présente les règles de correspondance. Un XSchema et son

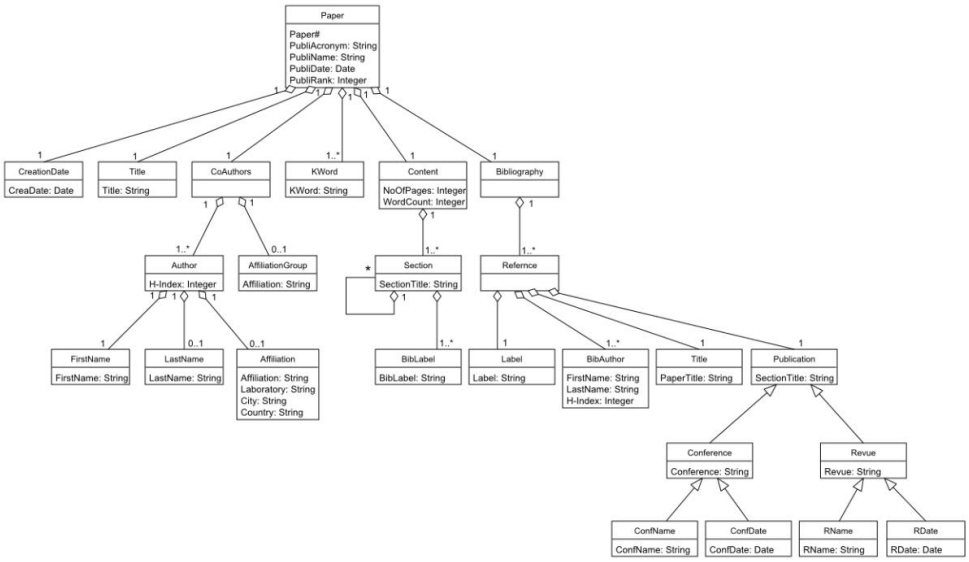

FIG. 4 – *SourceCD représentant une classe de documents d'articles scientifiques.*

SourceCD sont tous deux des arbres n-aires enracinés. Chaque élément du XSchema est analysé, décomposé le cas échéant et transformé en une classe dans le SourceCD.

Le SourceCD construit sera traité automatiquement pour en extraire les mesures et dimensions potentielles. Seules les mesures simples (faisant intervenir un seul attribut) sont identifiées par l'algorithme de traitement. Par manque de place, l'algorithme de traitement ne sera pas abordé ici. Le fait ayant déjà été identifié correspond à la racine du SourceCD (classe Paper). D'après (Bordawekar et Lang, 2005) tout élément XML peut être analysé autant en tant que dimension que mesure. Ceci est dû au fait que la classification entre les dimensions et les mesures n'est pas rigide, comme c'est le cas pour les données plates. Nous retrouvons cette propriété dans notre modèle, où un élément identifié comme dimension par le décideur peut aussi être identifié comme mesure (à l'exception de l'élément Date, utilisé exclusivement comme dimension). Les dimensions et leurs hiérarchies sont extraites des classes imbriquées, ou des attributs issus de la classe racine. Nous travaillons uniquement avec des hiérarchies simples représentées par des liens de composition dans le StarCD. Nous introduisons néanmoins le concept de hiérarchie inversée dans lequel l'élément de plus haute granularité dans la hiérarchie est l'élément rattaché au fait. Ce type de hiérarchie permet de respecter certaines contraintes de représentation de la structure des documents dans le schéma multidimensionnel.

Traditionnellement les mesures sont des valeurs numériques pré calculées au niveau logique, suivant les agrégations appliquées sur les données transactionnelles pour définir le cube de données. Dans notre modèle, nous prenons aussi en compte les mesures textuelles relatives aux éléments présents dans la structure, bien que ne traitant pas le contenu des documents. En effet notre modèle repose essentiellement sur l'exploitation de la structure des documents analysés, il est donc important que cette structure soit représentée dans le schéma multidimen-

XSchema	SourceCD
Elément (atomique ou complexe)	Classe
Lien d'imbrication entre 2 éléments	Relation de composition
Lien de spécialisation	Lien d'héritage
Lien de référence	Relation d'association
Attribut d'un élément	Attribut d'une classe

FIG. 5 – *Correspondances entre XSchema et SourceCD.*

sionnel.

Les mesures et dimensions potentielles identifiées sont présentées au décideur afin qu'il identifie les éléments nécessaires à son analyse. Les choix du décideur sont ainsi validés par la création dans le système d'un fichier XSchema représentant la structure du schéma multidimensionnel final. Ce document sera traduit en un schéma UML nommé StarCD (Star Class Diagram) qui définira le schéma de l'entrepôt, et il sera aussi utilisé en entrée du système d'intégration (voir Figure 1) pour la sélection des données satisfaisant aux contraintes d'analyse. Les Figures 6 et 7 représentent respectivement un XSchema partiel défini pour l'analyse de la collection d'articles scientifiques et le StarCD correspondant.

```
<!--Element Paper -->
  <xs:element name = "Paper" type="Fact">
  <xs:complexType>
    <!-- Measures -->
    <xs:attribute name = "ConfRank" type="xs:integer" />
    <xs:sequence>
      <!-- Measure Author -->
      <xs:element name = "CoAuthors">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Author" type="Measure">
              <xs:complexType>
                <xs:attribute name = "H-Index" type="xs:integer" />
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <!-- Measure Word -->
      <xs:element name = "KWord" type="Measure"/>
      <!--  -->
      <!-- Dimensions -->
      <!--Dimension PubliDate -->
      <xs:element name = "PubliDate" type="Dimesion">
          ...
      </xs:element>
      <!--Dimension CoAuthors -->
      <xs:element name = "CoAuthors" type="Dimesion">
          ...
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  </xs:element>
<!-- -->
```

FIG. 6 – *XSchema multidimensionnel.*

Dans le StarCD, les dimensions sont directement liées au fait par des relations d'association « By », et les mesures par des relations de composition. Dans le schéma StarCD représenté Figure 7, nous travaillons suivant deux dimensions (PubliDate et CoAuthors) et 3 mesures (Author, KWord, et ConfRank).

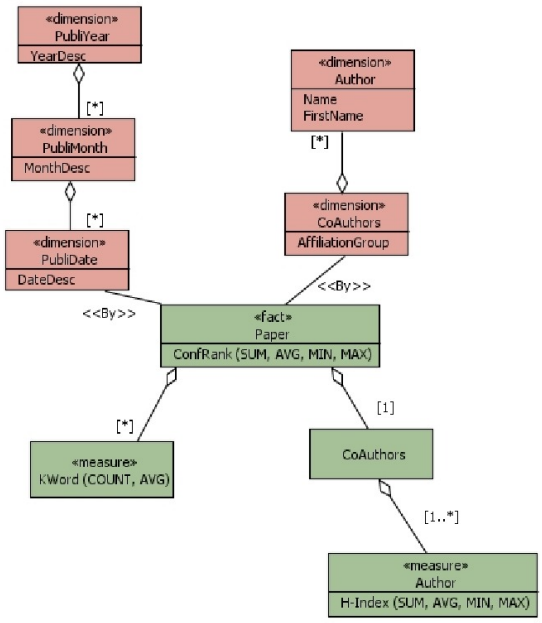

FIG. 7 – *Star Class Diagram (StarCD).*

3.3 Définitions formelles

Nous avons formalisé notre modèle multidimensionnel en définissant les notions de faits, mesures, dimensions et hiérarchies tels que nous les appliquons. Cette formalisation est la suivante :

$StarCD = (F, D)$ où :

- F correspond au fait, c'est-à-dire à l'élément racine du XSchéma définissant la classe de documents à analyser.
- $D = \{D_1, \ldots, D_n\}$ est un ensemble de dimensions associé au fait.

Le fait est caractérisé par un ensemble de mesures (numériques ou textuelles). Dans le StarCD, il représente la classe de documents de la source à analyser. Les mesures sont décrites sous forme de classes liées au fait par des liens d'agrégation, ou sous forme d'attributs intégrés au fait ; on retrouve ainsi la structure hiérarchique qui est présente dans le XSchéma de la source. Le fait est défini comme suit :

$F = (M, Agg)$ où :

- $M = \{M_1, \ldots, M_p\}$ est un ensemble de mesures.
- $Agg = \{ag_1, \ldots, ag_p\}$ l'ensemble des fonctions d'agrégation associées aux mesures, avec ag inclus dans $\{SUM, COUNT, AVG, MAX, MIN\}$.

Les dimensions sont quant à elles caractérisées par des paramètres organisés en hiérarchies. Ces hiérarchies précisent des niveaux de granularité allant du paramètre de plus haut niveau (All) au paramètre de plus bas niveau, qui correspond à la classe liée au fait. Elles sont définies

comme suit :
$D = (P, H)$ où :
- $P = \{p_1, \ldots, p_s\}$ est l'ensemble des paramètres de la dimension D D.
- $H = \{h_1, \ldots, h_s\}$ est l'ensemble des hiérarchies dans lesquelles sont organisés les paramètres, avec h défini comme suit :
  - $h = \{p_1, p_2, \ldots, All\}$ où $p1$ est le paramètre lié au fait (paramètre de plus bas niveau, et All étant le paramètre de plus haut niveau sur la hiérarchie.

Dans le StarCD, les dimensions sont décrites sous la forme d'un ensemble de classes ; chaque classe représente un paramètre qui permet de partitionner l'ensemble des instances du fait. Chaque dimension est une arborescence de classes dont la racine est liée au fait par une relation d'association «By». Les attributs faibles des dimensions correspondent aux attributs des classes. Avec ce modèle nous introduisons le concept de **hiérarchie inversée**. Dans ce type de hiérarchie, le paramètre de plus haute granularité est relié au fait pour respecter la structure du XSchéma de la source. Le paramètre « All » considéré comme paramètre de plus haut niveau dans une hiérarchie dite « normale » n'est plus pris en compte ici.

4 Conclusions et perspectives

Nous proposons dans cet article une approche de modélisation permettant d'élaborer le schéma conceptuel d'un entrepôt à partir d'une source de documents XML « centré document ». Cette approche est définie pour une pour une intégration physique dans un environnement XML natif. Elle est essentiellement basée sur l'exploitation de la structure des documents. Il est donc important que cette structure soit présentée au décideur afin de faciliter l'expression des requêtes d'analyse multidimensionnelle.

Nous envisageons de développer l'approche présentée dans cet article dans la base de données XML native eXist-DB. Ses mécanismes d'indexation permettent en effet une gestion optimale des collections de documents. Optant pour une intégration physique, une étude de la complexité du chargement de l'entrepôt est à envisager. Nous souhaitons orienter notre réflexion vers la recherche de solution permettant d'optimiser le temps de traitement pendant le chargement. Ceci implique aussi l'étude des problèmes liés à la mise à jour de l'entrepôt, en cas de changement dans les données de la source, ou de modification dans le modèle conceptuel. Par ailleurs, nous envisageons une étude comparative de performance de la solution que nous proposons, par rapport à un système relationnel possédant des fonctionnalités de gestion de documents textuels.

Références

Bordawekar, R. et C. A. Lang (2005). Analytical processing of xml documents: opportunities and challenges. *SIGMOD Record 34*, 27–32.

Boussaïd, O., R. BenMessaoud, R. Choquet, et S. Anthoard (2006). X-warehousing : Anxml-based approach for warehousing complex data. *In 10th East European Conferenceon Advances in Databases and Information Systems (ADBIS06) 4152*, 39–54.

Fankhauser, P. et T. Klement (2003). Xml for data warehousing changes and challenges. *Procedings of DaWaK 03*, 1–3.

fei Jiang, F. M., J. Pei, et A. W.-C. Fu (2007). Ix-cubes: iceberg cubes for data warehousing and olap on xml data. *CIKM*, 905–908.

Golfarelli, M., D. Maio, et S. Rizzi (1998). The dimensional fact model: a conceptual model for data warehouses. *International Journal of Cooperative Information Systems 07*, 215–247.

Hümmer, W., A. Bauer, et G. Harde (2003). Xcube: Xml for data warehouses. *DOLAP*, 33–40.

Janga, P. et K. C. Davis (2013). Schema extraction and integration of heterogeneous xml document collections. *Model and Data Engineering - Third International Conference, MEDI 2013. Proceedings 8216*, 176–187.

Kimbal, R. et M. Ross (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Moedeling*. Wiley Computer.

Li et A. An (2005). Representing uml snowflake diagram from integrating xml data using xml schema. *Proceedings of: 2005 International Workshop on Data Engineering Issues in E-Commerce (DEEC 2005)*.

Messaoud, I. B., J. Feki, K. Khrouf, et G. Zurfluh (2011). Unification of xml document structures for document warehouse (docw). *ICEIS 2011 - Proceedings of the 13th International Conference on Enterprise Information Systems 1*, 85–94.

Nassis, V., R. Rajagopalapillai, T. S. Dillon, et W. Rahayu (2005). Conceptual and systematic design approach for xml document warehouses. *Proceedings of the 2005 international conference on Computational Science and Its Applications*, 914–924.

Niemi, T., M. Niinimäki, J. Nummenmaa, et P. Thanisch (2002). Constructing an olap cube from distributed xml data. *DOLAP*, 22–27.

(OMG), O. M. G. (2013). Unified modelling languafe (uml). *http://www.uml.org/*.

Park, B., H. Han, et I. Song (2005). Xml-olap: A multidimensional analysis framework for xml warehouses. *Data Warehousing and Knowledge Discovery 3589*, 32–42.

Pérez, M., R. Berlanga, M. J. Aramburu, et T. B. Pedersen (2008). Data warehouses with web data: A survey, knowledge and data engineering. *IEEE Transactions on In Knowledge and Data Engineering 20*, 940–955.

W3C-Consortium (2013). Xml schema. *http://www.w3.org/XML/Schema*.

Summary

The XML format is today omnipresent on the Web and also in organizations' information systems. It allows to transport and exchange complex and heterogeneous data representing valuable information, which is barely exploited, even not at all. The current data warehouses systems and OLAP technologies allow the analysis and the storage of transactional data extracted from relational databases. However, these tools are not well suited for the XML documents analysis because of their hierarchical structure or their content being generally textual. We propose in this article an approach that allows building a XML "document-centric" warehouse whose conceptual schema is modeled using the UML formalism ; we also present architecture for a physical integration of these documents in a XML native environment.