Etat de l'art : Le temps dans les entrepôts de documents XML

Mourad Hsan, Faiza Ghozzi, Rafik Bouaziz

Etat de l'art : Le temps dans les entrepôts de documents XML

Mourad HSAN*, Faiza GHOZZI**, Rafik BOUAZIZ*

* Faculté des Sciences Economiques et de Gestion de Sfax
mouradhsan@yahoo.fr
** Institue Supérieur d'Informatique et de Multimédia de Sfax

Résumé : Les documents électroniques (rapports, notes, articles…) évoluent et présentent des contenus de plus en plus riches et complexes. Ainsi, ils font l'objet de changements rapides a priori imprédictibles. Se pose alors le problème de détection de ces changements. En effet, il faut détecter les changements induits par les évolutions, en comparant les anciennes et les nouvelles versions des documents et en évaluant la similarité entre ces versions. Le besoin de pouvoir versionner et analyser ces documents augmente constamment, il convient donc de disposer des outils adéquats. Les entrepôts de documents présentent une solution d'intégration et d'historisation de ces documents. L'adjonction de l'aspect temporel dans ces entrepôts vise à proposer des solutions pour le versionnement des documents. L'objectif de cet article est de présenter et d'analyser l'état de l'art des travaux de recherche menés dans les entrepôts de documents XML. En particulier, il présente l'intégration de la notion du temps dans ce type d'entrepôts.

Mots-clés : Entrepôts de documents, XML, version, RELAX NG.

1. Introduction et motivation

Aujourd'hui, les entrepôts de données (ED), construits à partir des données transactionnelles essentiellement numériques, sont bien connus dans les systèmes décisionnels proposant des outils d'analyse en ligne (OLAP). Ils deviennent de plus en plus populaires du fait que seulement 20% des données d'un système d'information d'entreprise sont numériques et les 80% restants des documents sont principalement constitués de données textuelles Tseng et al. (2006). Or, les systèmes OLAP ne sont pas encore performants pour l'analyse de ce type de données à cause du manque d'outils et de méthodes adaptés.

Par ailleurs, les travaux décrits dans Tseng et al. (2006) soulignent l'importance des données textuelles pour la prise de décision. Ces données sont obtenues généralement du web, vu comme un point potentiel d'émission de données (Hümmer et al, 2003). Elles sont téléchargées sous forme de document HTML, XML, etc. Afin de pouvoir manipuler ces documents, des chercheurs ont proposé une analyse simple des documents, sans aborder l'analyse multidimensionnelle (khrouf, 2001 ; Djemal, 2010 ; Portier et al, 2010). D'autre ont étudié l'entreposage de documents (Golfarelli et al, 2001 ; Baril et al, 2003 ; Hümmer et al, 2003 ; Boussaid et al, 2006).

L'analyse des documents nécessite de détecter les changements induits par les évolutions, en comparant les anciennes et les nouvelles versions des documents et en évaluant la similarité de ces versions. Le besoin de pouvoir versionner et analyser ces documents augmente constamment, il convient donc de disposer des outils adéquats. Dans cet article, nous dressons un panorama des travaux de recherche établis dans ce domaine que nous avons classés en trois grands axes : le premier axe se focalise sur les documents XML temporels, le deuxième se focalise sur les entrepôts XML et le dernier traite l'historisation dans ce type d'entrepôts. Le reste de l'article est organisé comme suit : La section 2 présente un tour d'horizon des travaux antérieurs portant sur l'analyse des documents. La section 3 étudie les travaux relevant de ces trois axes. La section 4 présente une discussion de ces travaux. Finalement la conclusion et les perspectives font l'objet de la section 5.

2. Analyse de documents : Etat de l'art

Au niveau web, les documents représentent une capitalisation de connaissances et la volumétrie de ces documents accroît de plus en plus. Dans ce contexte, les données électroniques sont publiées à grande échelle. Pour faire face aux problèmes posés par cette publication, XML joue un rôle important dans l'échange de données sur le Web et au sein des entreprises (W3C-XML, 2006). Le langage XML permet de représenter à la fois du texte non structuré, du texte ayant une structure partielle et jusqu'à des données fortement structurées telles que des tuples d'une source relationnelle (Abiteboul, 2003).

Un document peut être de deux types (Rusu et al, 2006 ; 2009) : un document statique dont le contenu et les structures ne changent pas dans le temps (par exemple, un document XML contenant les articles publiés dans un livre) et un document dynamique dont, par opposition à un document statique, le contenu et les structures changent dans le temps. Il représente la couverture temporelle pour la version spécifique du document. La détection des changements dans un document dynamique est basée sur certains processus opérationnels (par exemple, le contenu d'un magasin en ligne peut changer d'horaire, quotidien ou hebdomadaire, en fonction du comportement client).

En outre, les documents XML peuvent être classés en deux grandes catégories : les documents orientés données qui utilisent XML pour transporter des données et ils sont caractérisés par la non importance de la structure physique et par un contenu fortement structuré avec des champs clairement séparés et bien identifiés. Les documents orientés documents, appelé aussi orientés présentation, qui utilisent XML pour ses capacités similaires à SGML, comme dans un manuel d'utilisateur, une page web statique en XHTML ou des brochures. Et a l'opposé des documents orientés données, la structure physique de ce type de document est importante. Par exemple, pour un manuel d'utilisateur, l'ordre des chapitres est important. Ces deux catégories sont caractérisées par un contenu mixte qui est principalement composé de texte et non de champs.

La structuration des documents est une solution pour organiser, agencer et par conséquent faciliter l'interrogation du contenu documentaire (Djemal, 2010). Elle consiste à identifier chacun des éléments d'un document et elle peut être considérée comme étant un ensemble d'éléments organisés hiérarchiquement et/ou un enchaînement temporel d'éléments. Dans le cadre de cette solution, plusieurs travaux ont été proposés dans la littérature. (Chatti et al, 2004 ; 2006) propose un modèle, intitulé « **MSDM :** (**M**uli-**S**tructured **D**ocument **M**odel) », qui permet d'intégrer un ensemble de structures au sein d'un même document tout en assurant l'exploitation de ces structures conjointement et dont une instance, ***MultiX***, s'exprime dans le formalisme XML qui permet de sérialiser un document multi-structuré sous la forme d'un document XML bien formé. Ce modèle est constitué de deux structures : une Structure de Base (SB), qui représente le contenu d'un document et permet de le partager entre plusieurs structures, d'assurer leur cohérence et d'éviter le problème de la redondance. La deuxième structure est Documentaire (SD), qui décrit le contenu et elle est définie pour l'exploitation du document. La SD permet de décrire l'ensemble de descripteurs structurés (règles) utilisés dans un document et permet aussi de référencer leur contenu. Il permet aussi de définir des relations de correspondances entre les structures (SB et SD) afin de réaffecter du contenu aux éléments à partir de la structure de base et représenter des relations spécifiques entre leurs éléments. (Portier et al, 2009) proposent un modèle particulier « **MultiX^2** », qui représente une instance de MSDM et une méthodologie de construction de documents multi-structurés.

Quant aux travaux de (Bruno et al, 2006), ils proposent un modèle, intitulé « **MSXD** : (**M**ulti**S**tructured **X**ML **D**ocuments) », et un schéma, pour les documents multi-structurés. Ils permettent de définir des ***structures*** concurrentes sur les mêmes données textuelles en prenant en compte les annotations ajoutées par un utilisateur. Le modèle MSXD proposé est basé sur

le principe de la duplication du contenu en représentant les structures les unes indépendamment des autres ce qui engendre le problème de chevauchement des éléments et de cohérence. Il permet de définir plusieurs segmentations textuelles du même texte et une structure hiérarchique pour chacune de ces segmentations par la représentation par plusieurs documents XML et il permet de plus aux utilisateurs d'annoter les structures. Une instance de MSXD peut être vue comme un index sur la valeur textuelle et entre les structures.

(Djemal, 2010) propose un modèle, intitulé « **MVDM** : (**M**ulti **V**iew **D**ocument **M**odel) », basé sur la notion de fragmentation en décrivant séparément les différentes entités qui forment un document ainsi que leurs relations, de façon à pouvoir traduire plusieurs types de liens entre ces entités. Il permet de représenter les différentes structures qui peuvent être perçues dans un document mono ou multimédia d'une façon claire. Ce modèle intègre deux niveaux de description : un *niveau spécifique* permettant de décrire chaque document au travers des entités qui le composent et un *niveau générique* permettant de définir des typologies de documents au travers du regroupement de structures similaires. Les auteurs proposent aussi un outil, intitulé « **MDOCREP** : (**M**ultistructured **DOC**ument **REP**ository) », qui assure l'extraction et la classification des structures de documents et l'interrogation et la restitution multidimensionnelle de ces documents à partir de leurs différentes structures. Bien que les auteurs proposent des solutions pour la représentation des différentes structures pour un document mono ou multimédia, ils n'ont pas trouvé une solution pour gérer des structures concurrentes qui se chevauchent sur un même contenu, ainsi que la cohérence des informations représentées par l'ensemble des structures. Ils n'ont pas géré le cas où les informations subissent des changements, soit dans un même document, soit dans des documents différents.

Pour définir la structure d'un document, deux langages ont été définis dans la littérature : Définition du Type de Document (DTD), écrite selon la norme EBNF (Extended Backus Naur Form) et il permet de définir la structure logique d'un document XML en termes d'éléments et d'attributs. Le deuxième langage est le schéma XML qui permet de fournir une structure d'un document XML spécifique (Rusu et al, 2004a) et de produire un document qui est un fichier de description de structure (**X**ML **S**chema **D**escription (**XSD**)). Ces deux langage déterminent quels éléments peuvent être contenus dans un document XML, quels éléments peuvent être imbriqués dans d'autres, quelle valeur par défaut leurs attributs peuvent avoir, etc. (Zoubir, 2008).

3. Etat de l'art des Entrepôts XML Temporels

Nous avons classé les travaux, existants dans la littérature, en trois axes : le premier axe se focalise sur les documents XML temporels, le deuxième se focalise sur les entrepôts XML et le troisième traite l'historisation dans ce type d'entrepôts.

3.1 Documents XML temporels

Pour gérer les évolutions dans les documents XML, (Chien et al, 2001 ; 2002) proposent la création de documents multiversions basée sur un schéma comportant un ensemble de nœuds pour les éléments des documents XML, afin de classer les éléments et décomposer les documents en plusieurs sous documents dans plusieurs dossiers liés. Cette approche est basé sur l'extension du schéma du système de contrôle des versions (**S**ystem for **V**ersion **C**ontrol), qui représente le document de base (la version courante du document), stocke la première version de document et maintient les futures versions. Cette extension est traitée par la séparation des objets d'un document en ensemble de versions. L'atout de cette approche réside dans l'ajout de chaque nouvelle version au référentiel existant, sans modifier les informations précédemment stockées. Cependant, elle n'utilise pas des fonctions de transformation comme le cas dans les entrepôts de données numériques.

Dans la littérature, il existe d'autres systèmes qui assurent le contrôle des versions tels que le système de contrôle de code source (**S**ource **C**ode **C**ontrol **S**ystem : SCCS). Ce dernier insère des opérations d'édition dans le document original et associe un estampillage (ou d'identifications de versions) à chaque segment de document (objet) pour indiquer sa durée de vie. Il est efficace puisqu'il a des coûts de stockage minime car il stocke uniquement les informations de changement d'une version.

(Wang et al, 2003) proposent une approche basée sur des techniques efficaces pour la gestion des documents historiques et de soutenir des requêtes temporelles multiversions puissantes sur de tels documents en adaptant le schéma de base SCCS pour produire un document XML bien structuré. Elle consiste à représenter des *versions successives* d'un document comme un document XML qui implémente un modèle de données groupées temporellement a travers une technique qui stocke ces versions dans une structure intitulé **V-document** qui augmente progressivement à chaque nouvelle version. Ce modèle représente une extension d'une DTD par l'ajout de deux nouveaux attributs, « ***VStart*** » et « ***Vend*** ». L'approche permet aussi d'exprimer, en utilisant les langages de requêtes XML tels que XQuery, des requêtes complexes sur le contenu d'une version particulière, et sur l'évolution temporelle des éléments des documents et leur contenu. Elle permet de gérer les changements subis par un document, tels que la suppression, la modification et l'insertion et de manipuler les attributs. Cependant, cette approche traite l'évolution temporelle des éléments des documents et leur contenu, elle reste insuffisante du fait que l'utilisateur doit choisir une version particulière.

(Rusu et al, 2005 ; 2006) proposent une approche pour stocker les modifications d'un document XML au cours d'une période de temps. Elle considère principalement les parties modifiées des documents lors de son versionnement et ne traite pas les parties inchangées. Les auteurs construisent un seul document XML, intitulé « ***delta consolidé (CΔ)*** », qui permet d'extraire tous les changements soutenus par deux documents XML versionnés successifs dans la période de temps T_1-T_n même de différentes tailles, d'éliminer les éventuels doublons et permet aussi la gestion des versions à tout moment. Ce document introduit un nouvel élément temporel « *tampon* » pour enregistrer les modifications à chaque estampillage pour chaque élément modifié tels que le moment, le type de changement et les valeurs pour les éléments insérés ou modifiés, des détails des changements sur les valeurs des éléments initiaux et regroupe toutes les informations intéressantes et utiles extraites à partir des versions du document. Cependant, cette approche produit un seul document qui englobe tous les changements d'un document, mais il est insuffisant puisqu'il traite seulement deux versions successives et ne traite pas les parties inchangées ce qui donne peut être des résultats erronés.

(Kanhabua et al, 2008 ; 2009) proposent une approche de prétraitement fondée sur la sémantique qui améliore la qualité de l'estampillage et étendent un modèle de langage temporel en intégrant des connaissances plus internes et externes. Les auteurs proposent trois nouvelles méthodes pour améliorer l'approche JRH (**J**ong / **R**ode / **H**iemstra), qui est basé sur un modèle de langage statistique pour la détermination de l'estampillage des documents par l'attribution d'une probabilité selon les statistiques de l'utilisation des mots au fil du temps. La première méthode consiste à insérer des mots pour en compléter ou en dénaturer le sens en classant un mot dans l'une des deux classes en fonction des caractéristiques survenant dans le temps : répétitifs, qui sont liés à des événements périodiques (Noël, Jeux Olympiques et coupe de monde) et sont censés apparaître périodiquement dans le temps (chaque année ou tous les quatre ans), ou non répétitifs, qui n'apparaissent pas régulièrement mais peut toujours apparaître dans de nombreuses périodes de temps. La deuxième consiste à indiquer l'importance d'un terme dans une partition de temps spécifique (Entropie Temporelle : TE) et la troisième consiste en la recherche des statistiques externes à partir de Google Zeitgeist

(GZ), qui donne essentiellement des statistiques sur les tendances des termes et intègre un score supplémentaires afin d'augmenter la probabilité d'une partition du temps provisoires.

3.2 Entrepôt XML

En entreposage de données XML, les données XML sont transformées selon une structuration multidimensionnelle en générant des fichiers intermédiaires qui sont converties au format du magasin de données pour en alimenter les structures. Il existe deux types d'entrepôts dépendent du type de données qui y seront stockées dans les documents XML : les entrepôts de données XML pour les documents orientés données et les entrepôts de documents XML pour les documents orientés documents (Tournier, 2007).

Une quantité considérable de travail a été effectuée au cours des dernières années pour trouver des solutions efficaces au problème de l'entreposage des documents XML. Il existe deux principales approches : Les approches ascendantes qui sont guidées par le schéma de la source de données représentant les documents XML, associée a un schéma qui peut être un XML schéma ou une DTD. Les approches descendantes qui sont guidées par les besoins des utilisateurs décisionnels. Dans ce qui suit nous présentons succinctement les travaux qui nous semblent les plus pertinents.

3.2.1 Approches ascendantes

(Pokorny, 2001) propose une approche, intitulée « XML-Star » basée sur les schémas en étoiles simples avec l'utilisation des hiérarchies explicites. Une hiérarchie est dite explicite si elle contient un ensemble des contraintes référentielles. Cette approche permet la modélisation et l'interrogation d'un entrepôt de données à partir des sources de données XML. Elle se base sur l'utilisation de vues XML, qui aident à vérifier l'intégrité référentielle et accélèrent l'exécution des requêtes, par la reconstruction des hiérarchies des dimensions à partir de fragment de diverses données XML sources. Les dimensions constituant l'entrepôt doivent être conformes à une DTD qui est déterminée par la fusion des fragments de DTD des documents XML orientés données intervenant dans la construction de la vue. Chaque dimension (un document XML) est modélisée sous forme de séquence de DTDs qui sont logiquement associées, au niveau conceptuel et pour la construction d'une hiérarchie de dimensions, l'idée de cette approche est de trouver les relations logiques entre les dimensions voisines dans cette hiérarchie. Une fois les données qui caractérisent chaque dimension sont déterminées, il ne reste plus qu'à représenter les faits par une DTD. Dans cette approche, l'auteur n'a pas bien expliqué comment construire les faits.

(Golfarelli et al, 2001) ont adapté un modèle dimensionnel d'information qui représente l'entrepôt de données relationnel basé sur la modélisation en étoile et les arbres d'attributs qui peuvent être construits automatiquement en naviguant dans les dépendances fonctionnelles entre les entités du schéma source. Les auteurs ont proposé une approche *semi-automatique* pour construire les schémas conceptuels des magasins de données directement à partir d'une source XML orientée données en tenant compte des spécificités de ces sources. Cette approche permet de traduire les sources XML dans un schéma relationnel équivalent afin de concevoir un entrepôt. Elle commence par le schéma logique des sources opérationnelles.
L'approche réside dans la détermination des hiérarchies des faits par la navigation dans les dépendances fonctionnelles. Pour construire un schéma de fait à partir d'un schéma E/R elle suit les étapes suivantes : créer un graphe pour le DTD, choisir les faits et pour chaque fait construire l'arbre d'attributs, restructurer cet arbre afin de nettoyer les nœuds qui n'ont pas d'intérêt et enfin définir les dimensions et les mesures qui doivent être sélectionnées parmi les enfants de la racine de l'arbre. Cependant, cette approche utilise les DTD, ce qui le cas de plusieurs schémas, tout en signalant qu'il est possible d'utiliser les schémas XML.

(Hümmer, 2003) propose une approche, intitulée « ***XCube*** », qui utilise trois schémas XML de base, pour le stockage, l'échange et l'interrogation des données d'un ou plusieurs entrepôts. Ces schémas représentent le schéma multidimensionnel, les dimensions et les valeurs des faits. Le premier schéma, intitulée *XCubeSchema*, décrit un seul ou plusieurs cubes qui représentent le format central de la structure et stocke la description du schéma multidimensionnel d'un cube par la modélisation de ses dimensions et ses mesures. Le deuxième schéma, intitulé *XCubeDimension*, modélise la structure des dimensions par la définition des niveaux hiérarchiques. Ce document contient des nœuds appartenant à la classification des niveaux définie sur *XCubeSchema*. La troisième structure, intitulé *XCubeFact*, définit la collection des cellules des cubes de données de l'élément cube. Elle permet de décrire les données du fait d'un cube. L'auteur a présenté chaque module à part afin de rendre possible leur réutilisation dans plusieurs cubes dans une même application et l'analyse des différentes spécificités multidimensionnelles.

(Tseng et al, 2006) proposent une architecture pour l'entrepôt de document où les documents proviennent des données sources internes et externes. Ces données subissent un prétraitement (e.g. résumé des textes) pour extraire les éléments qui seront enregistrés sous forme de métadonnées. Le système construit un cube de document, présenté sous forme de schéma en étoile, par la création des dimensions et des indexes. Les auteurs distinguent trois nouveaux types de dimensions : la dimension ordinaire qui comporte des données permettant le balayage du contenu du document (e.g. les mots clés), la dimension métadonnées qui englobe les métadonnées extraites des documents (e.g. le titre, l'éditeur, la date) et la dimension catégorie qui contient des données externes relatives au document. Ces trois dimensions sont liées à une table de fait qui se compose d'un ensemble d'attributs comportant une clé composite obtenue de l'ensemble des clés étrangères des dimensions décrites auparavant.

Pour l'automatisation de la conception, (Vrdoljak et al, 2003) considèrent que l'utilisation des DTD comme structure logique des sources de données est dépassée et proposent un processus de conception, similaire à celle dans (Golfarelli et al. en 2001), qui construit le schéma de fait à partir des spécifications des schémas XML. Il étudie le prétraitement et la construction d'un graphe de dépendance fonctionnelle de ce schéma et la création d'un schéma logique pour chaque fait choisi. Cependant, ce processus est *semi-automatique* du fait qu'il nécessite l'intervention du concepteur qui doit se référer à ses propres connaissances pour décider si les relations dans les hiérarchies sont multiples et s'elles sont intéressantes pour l'agrégation. De plus, elle ne traite pas l'aspect temporel pour les éléments multidimensionnels.

Dans la même tentative, (Hachaichi et al, 2010) présentent une méthode automatique de conception de schémas multidimensionnels à partir de DTD. Cette méthode traite les éléments les plus pertinents à l'analyse dans ces schémas afin de guider le décideur à exprimer ses besoins. Elle est basée sur trois étapes implémentées dans un outil intitulé ***CAME-XML*** (**C**onception **A**ssistée de **M**agasin de données en **E**toile à partir de source **XML**) :

- Le prétraitement qui simplifie une DTD initiale, restructure cette DTD simplifiée en des arbres de transition décrivant les liens structurels entre les éléments de la DTD et enrichit ces arbres par les types déterminés en examinant des documents XML conformes à leur DTD ;
- La construction de schémas de magasins de données (MD). Dans cette étape les auteurs appliquent sur les arbres de transition enrichis un ensemble de règles d'extraction automatiques pour chaque concept multidimensionnel. Ces règles maintiennent la traçabilité de chaque concept extrait et attribuent pour chacun un niveau de pertinence reflétant son potentiel analytique.

- L'Ajustement/Validation. Cette étape est assistée par le niveau de pertinence attribué par la deuxième étape.

En outre, la méthode proposée dans (Hachaichi et al, 2010) assiste le concepteur dans la définition des procédures ETL en préservant les correspondances entre les éléments des schémas de MD et les sources XML. Cependant, cette méthode ne gère pas les versions des données.

3.2.2 Approches descendantes

(Zhang et al, 2003) proposent une méthodologie pour matérialiser les entrepôts de données XML en analysant les modèles de requêtes fréquentes extraites à partir de requêtes historiques émises par les utilisateurs. Cette méthodologie permet de manipuler les sources de données XML distribuées représentées sous forme de DTD en commençant par déterminer quelles sources de données sont plus fréquemment consultées par les utilisateurs. Pour intégrer ces données XML, les auteurs ont utilisé la technique de classification hiérarchique qui permet d'obtenir et maintenir les documents XML.
Cette approche utilise en entrée deux arbres de sources différents, celui provenant de sources XML et celui provenant des requêtes proposées par les utilisateurs. Ces arbres sont confrontés afin de générer des documents XML intégrés.

(Rusu et al, 2004 a et b) proposent une approche qui permet de construire, à partir d'un document XML initial orienté données, des fichiers XML structurés de manière multidimensionnelle (fait, dimension, …) et leur liaison. Cette approche couvre plusieurs processus tel que le nettoyage et l'intégration des données, par la création des documents XML intermédiaires afin d'extraire les informations utiles et nécessaires et pour traiter le manque d'informations. Pour valider le contenu des documents XML intermédiaires, les auteurs utilisent les schémas XML, si un schéma existe, il faut vérifier l'existence de toutes les conditions de schéma telles que le nom des éléments, le type des données XML et vérifier si la hiérarchie des éléments est respectée. Un autre processus est l'agrégation des données, par l'examen des données existantes dans la base de données, afin de créer des dimensions.
Après la création et la vérification des documents XML intermédiaires, les auteurs proposent l'étape de construction d'une table de fait qui permet de lier ces documents en établissant les relations entres les clés. Cependant, la majorité des étapes dans cette approche sont accomplies manuellement par un expert du domaine des documents XML (Hachaichi et al, 2010). De plus, elle ne traite pas l'aspect temporel pour les éléments multidimensionnels.

(Nassis et al, 2004) proposent une méthodologie de conception d'entrepôt de documents XML orientés document, intitulée « ***XDW*** » basée sur les concepts orientés objets et utilise XML lui-même en plus du schéma XML. Cette méthodologie possède trois niveaux. Le premier niveau représente les exigences utilisateur qui décrivent les besoins des utilisateurs de l'entrepôt et le modèle OO des besoins (acteurs, cas d'utilisation, objets, …). Le deuxième niveau décrit le référentiel des Faits XML (***XFact***). Ce référentiel représente le niveau conceptuel de l'entrepôt XML. ***XFact*** est composé d'un schéma XML construit du modèle conceptuel XDW et ses documents XML et des dimensions, en utilisant les vues conceptuelles de XML qui fournissent des perspectives à la hiérarchie des documents entreposés dans XFact. Le troisième niveau assure la transformation du modèle conceptuel en schéma XML.
En effet, cette méthodologie utilise des packages des diagrammes UML afin de grouper logiquement et de construire des vues conceptuelles hiérarchiques pour améliorer la sémantique du XDW. Les données XML sont stockées sous la forme de documents XML structurés selon une grammaire XML au niveau logique. Elles sont représentées par un diagramme de classes UML au niveau conceptuel.

(Niemi et al, 2002) proposent un système basé sur un langage XML pour la collecte des données. Ce système est bien défini et ciblé puisque les besoins d'informations sont bien connus et les mesures, les dimensions et les contraintes pour le cube sont bien connus à partir des requêtes et des documents XML. Il est basé sur la modélisation en étoile. Les auteurs ont étudié comment un cube OLAP peut être présenté en utilisant XML par une présentation formelle de ses différents éléments en utilisant un modèle relationnel. Ils ont défini les données proposées par les utilisateurs en utilisant le langage MDX qui est dédié pour les bases de données multidimensionnelles OLAP.

(Boussaid et al, 2006) propose une démarche méthodologique *semi-automatique*, intitulée « ***X-Warehousing*** ». Cette démarche permet la modélisation multidimensionnelle des données complexes par la construction des cubes de données avec des documents XML sources orientés données et/ou orienté documents. Ces cubes sont faciles à lire et contiennent toute l'information. Cette démarche utilise en entrée, soit une collection de documents XML soit un modèle conceptuel multidimensionnel, qui décrit les besoins des utilisateurs, et décrit une collection de documents XML en sortie qui est représentée à l'aide de schémas XML.

3.3 Entrepôt XML temporel

On peut avoir souvent l'espoir de retrouver ou de récupérer tout ce qu'on perd. Le temps constitue le champ de l'existence d'une information donc ensemble d'informations, l'espace où il est, où il profite de son existence et où il fait profiter autrui. Un système d'entreposage XML efficace doit répondre aux besoins des utilisateurs au niveau de son contenu ou sur les versions des documents entreposés (Rusu et al, 2006). Nous présentons, dans ce qui suit, un diaporama des travaux dans la littérature sur la conception des entrepôts de données XML temporels.

(Marian et al, 2000) proposent une méthode pour contrôler les documents XML obtenus à partir du web dans un entrepôt des données web XML. Cette méthode traite les changements entre *deux versions consécutives*. Elle gère les versions du système Xylème. Les auteurs commencent par une séquence des versions des données XML. L'idée principale de cette méthode est la calcule des changements d'une nouvelle version et la modification de l'historique par l'ajout d'une nouvelle version.

(Norvag, 2002) étudie les conflits dans un entrepôt de données XML en particulier ceux liés à la cohérence et l'approximation du temps. Il propose une solution pour stocker des informations complètes sur une période au niveau d'un entrepôt de données XML. L'indexation et l'interrogation de ces informations sont basées sur le temps du document qui est similaire au temps de validité dans les bases de données XML temporelles.

(Rusu et al, 2006) proposent une méthodologie pour l'entreposage des documents XML dynamiques. Cette méthodologie étudie les spécificités des documents XML dynamique (multiversions) et leurs stockage, avec le moins de redondance possible. Elle permet de gérer des versions des informations essentielles de documents, pour permettre l'analyse des requêtes riche. Les auteurs proposent de construire un entrepôt de données XML, en un schéma en étoile, seulement pour les informations qui se trouvent dans le delta consolidé qui constitue le noyau de cette méthodologie. En effet, les auteurs proposent de créer les dimensions, le document de fait et de lier ce document à ses dimensions via un identifiant de nœud (LNI). Le document de fait inclut seulement les détails des changements produisent au T_i. LNI représente de façon unique chaque enregistrement d'une dimension, donc il peut être considéré comme une clé primaire pour les données relationnelles (Rusu et al, 2009).
Dans cette méthodologie, les auteurs étendent la notion du LNI pour l'exploitation des entrepôts de données XML. Ils représentent cet identifiant avec le format XML, pour assurer la position dans la hiérarchie de dimension XML. Ce nœud peut être soit un nœud séparé ou

un attribut d'un autre nœud. Les auteurs extraient les éléments à stocker pendant qu'ils existent à T_0 (c.à.d. la version initiale du document avant tout changement) et ajoutent des enregistrements au document de fait contenant les données de noyau à entreposer, ainsi que des liens aux dimensions. Et pour chaque version T_i, où i = 1,2, … n, ils ajoutent les changements (insertion, suppression, modification) au schéma en étoile.
En 2009, les auteurs ont proposé d'enrichir cette méthodologie par l'adjonction d'un processus qui traite les informations sur les changements dans le delta consolidé après nettoyage et intégration des données utiles qui est étudié et validé dans (Rusu et al, 2008). Toutes les dimensions dans l'entrepôt seront des documents XML, et non pas des tables relationnelles. Une dimension est dite particulière du fait qu'elle n'existe pas dans un entrepôt de documents XML statiques, car pour ces derniers il n'y a pas d'informations qui subissent des changements historiques à stocker.

4. Discussion

L'approche de versionnement dans un ensemble de documents orientés données nécessite la gestion des versions de ses éléments afin de créer un entrepôt de données XML temporel. Nous présentons, dans ce qui suit, un ensemble de critères pour évaluer les travaux sur l'évolution des modèles dans les entrepôts de données XML :

- C1 : Format utilisé pour décrire le contenu des documents. La plupart des travaux utilisent le format XML tels que les travaux de (Golfarelli et al, 2001) et (Pokorny, 2001) ;
- C2 : Type de document. (Zoubir, 2008) a classé les documents XML en deux grandes catégories :
  - Les documents orientés données (DOD) qui sont caractérisés par un contenu fortement structuré avec des champs clairement séparés et bien identifiés. Nous citons ceux de (Pokorny, 2001) et (Golfarelli et al, 2001) ;
  - Les documents orientés présentation (DOP) qui sont caractérisés par un contenu mixte et une structure irrégulière. Nous citons ceux de (Boussaid et al, 2006), (Vrdoljak et al, 2003).
- C3 : Approches traitant la conception des entrepôts de données XML. Ces approches peuvent être classées en deux catégories (Zoubir, 2008) :
  - Les Approches Ascendantes (AA) qui sont guidées par le schéma de la source de données représentant les documents XML. Nous citons les travaux de (Pokorny, 2001) ;
  - Les Approches Descendantes (AD) qui sont guidées par les besoins des utilisateurs décisionnels. Nous citons les travaux de (Golfarelli et al, 2001).
- C4 : Langages utilisés pour définir la structure des documents XML. Ces langages assurent la validité d'un document à une structure et aux contraintes désirées (Zoubir, 2008).
  - (L1) La DTD (Document Type Definition). Nous citons les travaux de (Pokorny, 2001) et (Golfarelli et al, 2001) ;
  - (L2) le Schéma XML. Nous citons les travaux de (Hümmer et al, 2003).
- C5 : Modèles utilisés pour l'entreposage des documents.
  - Un modèle Entité Relation (ER) tel que (Golfarelli et al, 2001) ;
  - Des schémas multidimensionnels en étoile (SME) tels que (Pokorny, 2001) et (Hümmer et al, 2003) ;
  - Des schémas en constellation (SMC) tel que (Hümmer et al, 2003) ;

  - Des schémas en flocon de neige (SMF) tel que (Boussaid et al, 2006), ou en galaxie (SMG).
- C6 : Richesse des spécificités de l'entrepôt de données XML. L'adjonction de nouvelles notions dans les éléments multidimensionnels (dimensions, hiérarchie). Dans ce contexte nous citons les travaux de (Pokorny, 2001).
- C7 : Etendue du versionnement.
  - Le versionnement peut toucher le contenu et/ou le schéma ;
  - Comment gérer les versions des éléments d'un entrepôt de données XML temporel (faits, mesures, dimensions, hiérarchies, …).

  Dans ce contexte, nous citons les travaux de (Marian et al, 2000), (Norvag, 2002), (Rusu et al, 2006 ; 2008 ; 2009).
- C8 : Interrogation des données. A l'opposé des entrepôts de données classiques, peu de travaux ont proposé des langages pour analyser les entrepôts de données XML.

La comparaison des travaux selon les critères présentés auparavant dans le tableau 1 présenté ci-dessous.

Critiques / **Travaux**	**C1**	**C2**	**C3**	**C4**		**C5**		**C6**	**C7**	**C8**
				E	**S**	**E**	**S**			
Pokorny, 2001	+	DOD	AA	L1			SME	+	-	+
Hümmer et al, 2003	+	DOD	AA	L2			SME/ SMC	-	-	+
Boussaid et al, 2006	+	DOD/ DOP	AD		L2	SME/ SMF		-	-	+
Golfarelli et al, 2001	+	DOD	AD	L1/L2		ER		-	-	+
Vrdoljak et al, 2003	+	DOD	AA	L2			SME	-	-	-
Tseng et al, 2006	+	DOP	AA				SME	+	-	+
Zhang et al, 2003	+	DOD	AD	L1				+	-	-
Rusu et al, 2004	+	DOD/ DOP	AA		L2		SME	-	-	-
Rusu et al, 2006	+	DOD/ DOP	AA		L2		SME	-	+	+
Rusu et al, 2008 ; 2009	+	DOD/ DOP	AA		L2		SME	-	+	+
Nassis et al, 2004	+	DOP	AD	L2		SME		+	-	-
Niemi et al, 2002	+	DOD/ DOP	AD	L1/L2		SME		-	-	+

Tableau 1 : Comparaison des travaux

(Pokorny, 2001 ; Golfarelli et al, 2001) n'ont pas adopté l'historisation et le versionnement des sources de données temporelles mais ils supposent que les documents XML non temporels. Cependant (Tseng et al, 2006 ; Ravat et al, 2007) traitent des nouveaux types de dimensions sans adopter l'approche de versionnement. Nous retenons cinq types : la dimension « *ordinaire* » qui se compose des données permettant le balayage du contenu du document (e.g., les mots clés). La dimension « *méta-données* » qui représente les méta-données extraites des documents (e.g., la date, le titre, l'éditeur, …). La dimension « *catégorie* » qui contient des données externes du document permettant sa catégorisation (Tseng et al, 2006). La dimension « *structure* » qui modélise la structure générique (e.g. section, paragraphe, …) et la structure spécifique (e.g. introduction, définition, …). La

dimension « *complémentaire* » qui représente une dimension classique que nous trouvons dans l'environnement OLAP (e.g. produit, client) (Ravat et al, 2007).

L'approche proposée dans (Rusu et al, 2004 a et b) permet d'obtenir des documents valides basés sur des schémas XML et des données propres avec un faible niveau de redondance sans avoir des exigences spécifiques à partir des documents sources. Mais il reste le problème de gestion des versions des documents XML et plus particulièrement les versions des documents XML intermédiaires ainsi que ses schémas XML. En effet, (Nassis et al, 2004) captent des besoins décisionnels à un niveau conceptuel avec le pouvoir de UML et le deuxième niveau, XFact, de la méthodologie de ces auteurs représente un cliché instantané pour un contexte donné et non pas une suite de version.

Le système proposé dans (Boussaid et al, 2006) est bien défini et ciblé ce qui rendre la collection des bonnes données simple. Ce système devient de plus en plus difficile pour les besoins des utilisateurs et plus particulièrement qui subis des changements dans le temps ce qui oblige l'étude de la possibilité de l'historisation et le versionnement des documents XML.

(Norvag, 2002) ont proposé seulement comment résoudre les problèmes de stockage et non pas comment créer et/ou extraire les éléments (fait, dimensions, hiérarchies, …) d'un entrepôt de données XML temporel. Il traite les aspects du temps dans les bases de données : le temps d'application et de validités.

La méthodologie, proposée par Rusu et al, en 2006, assure un faible degré de redondance des données entreposées. Elle préserve l'information critique puisque seulement le document initial et les changements temporels sont inclus et elle représente des requêtes temporelles faciles à être appliqué. Mais il existe des modifications qui peuvent engendrer des résultats erronés tels que, par exemple, la suppression d'un élément engendre la suppression automatique de son parent qui peut être utilisé dans une autre version.

Les modèles, proposées par Rusu et al, en 2008 et 2009, pour décomposer les documents XML afin de créer un schéma en étoile XML, n'étudient pas l'utilisation des fonctions de transformation comme le cas des entrepôts de données classiques afin d'avoir une correspondance entre les versions.

Peu de travaux ont traité le versionnement des entrepôts de données XML. Nous constatons aussi que les travaux que nous avons présentés auparavant traitent seulement des transformations entre deux versions successives et nécessitent l'extension des contraintes proposés dans (Ghozzi et al, 2004). Et comme le montre le tableau 1, toutes les propositions utilisent soit les DTD soit les schémas XML. De plus, toutes les méthodes actuelles nécessitent une intervention manuelle pour l'identification des éléments d'un entrepôt de document.

5. Conclusion et perspectives

L'entreposage de données XML aide à expliquer un phénomène qui s'est produit. Par exemple, l'entrepôt de données XML pourrait expliquer pourquoi un auteur publie dans une conférence et ne publie pas dans d'autres. Ceci est dû soit au fait que cette conférence n'est pas importante soit qu'elle a changée son état à une date. Dans cet article, nous avons dressé un état de l'art pour l'entreposage de données XML temporelles. Dans le contexte des recherches sur les entrepôts de données XML temporels, nous avons classé les travaux, présentés dans la littérature, en trois axes principaux : les documents XML temporels, les entrepôts de données XML et l'historisation dans ce type d'entrepôts.

Les documents sur le net peuvent toucher plusieurs domaines. Ils comportent des données qui sont sujet à plusieurs évolutions soit au niveau de la structure soit au niveau du contenu et la consultation de ces documents à une date donnée peut devenir obsolète à une date ultérieure. Nous considérons les documents orientés données dans ces domaines comme par

exemple le e* (e-commerce, e-earling, etc.) qui peuvent être modifiés. Ces documents constituent des sources de plus en plus utilisées aussi bien pour le stockage que pour le transport et les échanges transactionnels (Hachaichi et al, 2010). Les évolutions de ces documents peuvent être perdues. A une date postérieure, certaines informations n'ont pas appris dans les analyses. On distingue plusieurs exemple tels que :

- A une date D, un atelier A devient une conférence C. Cependant, une requête comme « déterminer le nombre d'articles dans l'atelier A après la date D » ne permet pas de retourner des résultats corrects dans un entrepôt non temporel car elle aura un résultat égal à NULL puisque l'atelier A a été modifié à la date D.
- Cet exemple étudie aussi le problème d'insertion et de suppression du fait que si un nouvel article P, publié dans l'atelier A, passe à la conférence C à l'instant T, le nombre d'articles sera changé. On peut même ajouter un nouvel élément dans un document XML. Si on supprime un élément, la structure d'un document sera changée et une nouvelle version sera prise en charge. Par exemple, les éléments d'un employé (*rue, ville et état*), appartenant à l'élément *adresse* et à l'instant T, deviennent indépendants de l'élément *adresse*. Ce problème est similaire à l'ajout et la suppression d'une hiérarchie.

Dans nos travaux futurs, nous constatons qu'on peut utiliser le schéma RELAX NG (**RE**gular **LA**nguage for **X**ML **N**ext **G**eneration), au lieu des DTD et schémas XML, qui est une norme internationale de l'ISO / CEI[i], considéré comme une alternative à XML Schéma. Il représente un langage de description de document XML issu de la fusion de TreX[ii] (**T**ree **R**egular **E**xpressions for **X**ML) et de Relax Core[iii] (**Re**gular **La**nguage description for **X**ML) et il peut faire référence à des bibliothèques externes de types de données. Ce schéma est simple, facile à apprendre. Il utilise à la fois une syntaxe XML et une syntaxe non-XML, il ne change pas l'ensemble des informations d'un document XML, il supporte des espaces de noms XML, il traite les attributs de manière uniforme avec des éléments autant que possible, il supporte les contenus non-ordonnée d'une façon illimité ainsi les contenus mixtes. RELAX NG utilise une base théorique solide et peut s'associer avec un langage de typage des données distinctes.

Nous constatons que l'entreposage des données XML temporelles, à des fins d'analyse reste un axe de recherche à traiter. Nous nous intéressons plus particulièrement à la modélisation conceptuelle. Afin de résoudre les différents problèmes engendrés par les aspects d'évolution, nous proposons une démarche mixte (hybride) basées sur les données sources des documents XML et sur les besoins d'analyse. Dans cette démarche nous proposons de créer un système de gestion d'Entrepôt de Données XML MultiVersions (EDXMV) basé sur le versionnement des schémas des EDX et de leurs membres de dimension. La première étape est la création d'un schéma global décrivant les changements dans documents XML sources et les besoins d'analyse en utilisant le schéma RELAX NG (RELAX NG de la e* par exemple), puis dans la deuxième étape la génération d'un modèle multidimensionnel en étoile passant par la récupération des concepts multidimensionnels qui subissent des évolutions (le fait, les dimensions) d'une manière automatique en utilisant des règles de récupération.

Références

- Abiteboul S. (2003), « Managing an XML Warehouse in a P2P Context », 15th Intl. Conf. on Advanced Information Systems Engineering (CAiSE), LNCS 2681, Springer, p. 4–13, 2003.
- Baril, X. et Bellahsène Z. « XML Data Management: Native XML and XML-Enabled Database Systems » (First ed.), Chapter Designing and Managing an XML Warehouse, pp. 455–474. Addison Wesley Professional.

- Boussaid O., Ben Messaoud R., Choquet R., Anthoard S., « X-Warehousing: an XML-Based Approach for Warehousing Complex Data », 10th East-European Conference on Advances in Databases and Information Systems ADBIS'06, Thessaloniki, Greece, September 2006; LNCS, Vol. 4152, Springer, Heidelberg, Germany, p 39-54.
- Bruno, E. et Murisasco, E. « MSXD : A Model and a Schema for Concurrent Structures Defined over the Same Textual Data ». DEXA 2006: 172-181
- Chien S.Y., Tsotras V.J., Zaniolo C., " Efficient Schemes for Managing Multiversion XML Documents", The VLDB Journal (2002) / Digital Object Identifier (DOI) 10.1007/s00778-002-0079-4.
- Chatti, N., Calabretto, S., et Pinon, J. M. (2004). « Vers un environnement de gestion de documents à structures multiples ».
- Chatti, N. (2006). « Documents multi-structurés : De la modélisation vers l'exploitation » Thèse de doctorat, L'institut National Des Sciences Appliquées De Lyon.
- Djemal K., (2010). « De la modélisation à l'exploitation des documents à structures multiples », Thèse de doctorat, Université Paul Sabatier, Toulouse III.
- Hachaichi Y., Feki J., Ben-Abdallah H. (2010). « Modélisation multidimensionnelle de documents XML centrés-données » Journal of Decision Systems (JDS). Ed. Lavoisier, vol. 19 – No.3/2010, pages 313-345, 2010. ISSN 1246-0125. ISBN 978-2-7462-1976-2.
- Hümmer W., Bauer A., et Harde G. (2003). « XCube : XML for Data Warehouses ». In Proceedings of the 6th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2003), New Orleans, Louisiana, USA, pp. 33–40. ACM Press.
- Khrouf K., Soulé-Dupuy C. (2001). « Conception d'entrepôts de documents décisionnels ». INFORSID 2001, 387-401.
- Kanhabua, N., Nørvåg, K. « Improving Temporal Language Models for Determining Time of Non-timestamped Documents ». ECDL 2008 : 358-370
- Kanhabua, N., Nørvåg, K. « Using Temporal Language Models for Document Dating ». ECML/PKDD (2) 2009: 738-741
- Marian, A., Abiteboul, S., Mignet, L. « Chance-centric Management of Versions in an XML Warehouse ». BDA 2000
- Nassis, V., Rajugan, R., Dillon, T., Rahayu, W. « Conceptual Design of XML Document Warehouses ». DaWaK 2004: 1-14
- Niemi, T., Niinimäki, M., Nummenmaa, J., Thanisch, P. « Constructing an OLAP cube from distributed XML data ». DOLAP 2002: 22-27
- Pokorny J., « Modelling Stars Using XML ». In Proc. The 4th ACM Intl Workshop on Dara Warehousing and OLAP (DOLAP01), pp. 24-31, Atlanta, 2001;
- Portier PE., Calabretto S. « Méthodologie pour la création de documents multistructurés ». INFORSID 2009: 211-226
- Portier P.E., Calabretto S., (2010) « Multi-structured documents and the emergence of annotation vocabularies ». The Markup Conference 2010, Montréal, Canada.
- Ghozzi F., Ravat F., Teste O. et Zurfluh G. (2004), « Contraintes pour modèle et langage multidimensionnels ». Dans RSTI-ISI : Fouille, transactions, évaluation dans les bases de données, Hermes –Lavoisier, Vol. 9, N. 1, p.9-34, 2004.
- Golfarelli, M., Rizzi S., et Vrdoljak B. (2001). « Data Warehouse Design from XML Sources ». In Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001), Atlanta, Georgia, USA, pp. 40–47. ACM Press.
- Rusu, L.I., Rahayu, W., Taniar, D. « On Building XML Data Warehouses ». In Proc. Intelligent Data Engineering and Automated Learning – IDEAL 2004, 5th International Conference, pp. 293-299, Exeter, UK, 2004 a.

- Rusu, L.I., Rahayu, W., Taniar, D. « On data cleaning in building XML data warehouses », In : Proceedings of the 6th International Conference on Information Integration and Web-based Applications & Services (iiWAS'2004), pp. 797–807 (2004b) ;
- Rusu, L.I., Rahayu J.W., Taniar D. (2005). « Maintaining Versions of Dynamic XML Documents ». WISE 2005: 536-543
- Rusu, L.I., Rahayu J.W., Taniar D. (2006). « Mining Changes from Versions of Dynamic XML Documents », Workshop on Knowledge Discovery in XML Documents (KDXD), p. 3-12, 2006.
- Rusu, L.I., Rahayu J.W., Taniar D. (2009). Partitioning methods for multi-version XML data warehouses. Distributed and Parallel Databases 25(1-2): 47-69 (2009)
- Tseng F., Chou A. (2006). « The concept of document warehousing for multi-dimensional modeling of textual-based business intelligence », journal of Decision Support Systems (DSS), vol.42(2), Elsevier, p. 727–744, novembre 2006.
- Tournier R. (2007). « Analyse en ligne (OLAP) de documents ». Thèse de doctorat en informatique, Université Paul Sabatier, Toulouse (France), 2007.
- Vrdoljak, B., Banek, M., Rizzi, S. « Designing Web Warehouses from XML Schemas ». DaWaK 2003: 89-98
- Zhang J., Ling T.W., Bruckner R.M. and Tjoa A.M., "Building XML Data Warehouse Based on Frequent Patterns in User Queries", Data Warehousing and Knowledge Discovery, 5th International Conference DaWak, Prague, Czech Republic, LNCS 2737, Springer, 2003, pp. 99–108 Sept.3-5, 2003.
- Wang, F., Zaniolo, C. « Temporal Queries in XML Document Archives and Web Warehouses ». TIME 2003: 47-55.

[i] http://relaxng.org/

[ii] Trex, conçus par James Clark, est un langage de validation des documents XML (http://www.thaiopensource.com/trex/)

[iii] RELAX Core, conçus par MURATA Makoto, a été approuvé comme un rapport technique de l'ISO / CEI en Mai 2001 (http://www.xml.gr.jp/relax/).