

Vers une optimisation de la performance des entrepôts de données XML

Doukifli Boukraâ*, Riadh Ben Messaoud**, Omar Boussaid**

*Université de Jijel, BP 98 Ouled Aïssa, 18000 Jijel, Algérie
d_boukraa@mail.univ-jjel.dz

**Laboratoire ERIC, Université Lumière Lyon2
5 avenue Pierre Mendès-France, 69676 Bron Cedex, France
rbenmessaoud@eric.univ-lyon2.fr, omar.boussaid@univ-lyon2.fr

Résumé. Les données nécessaires à des fins décisionnelles sont de plus en plus complexes. Elles ont des formats hétérogènes, proviennent de diverses sources et se présentent sur des supports variés. Le langage XML permet de pallier la complexité des données selon une description unifiée des données complexes, d'où l'émergence des entrepôts de données XML. Cependant, au regard de l'organisation semi-structurée du formalisme XML, les schémas multidimensionnels classiques des entrepôts sont à redéfinir. Cette redéfinition porte sur l'aspect logique, l'aspect physique et, par conséquent, sur les performances d'un entrepôt de données XML. Dans cet article, nous proposons un schéma physique d'un entrepôt de données XML. Nous présentons également une première solution pour l'optimisation des performances de ce dernier. La solution est mise en œuvre à travers un cas pratique et est testée par des expérimentations.

1 Introduction

Les *entrepôts de données* (*data warehouses*) ont apporté une solution adéquate et efficace au problème du stockage et de la gestion des données. Un entrepôt est une base centralisée de grands volumes de données, historisées, organisées par sujet et consolidées à partir de diverses sources d'informations (Inmon, 1996; Kimball, 1996). En plus de sa vocation de stockage, la modélisation d'un entrepôt est complètement dédiée à l'analyse de ses données. En effet, les données d'un entrepôt sont sélectionnées pour construire des magasins de données (*data marts*) dédiées à une activité particulière. Les données sont alors organisées de façon multidimensionnelle selon des modèles *en étoile* ou en *flocons de neige* (Chaudhuri et Dayal, 1997). Ces modèles sont largement employés pour préparer les données à l'analyse. Ils permettent également de produire des vues de données communément appelées *cubes de données*.

D'autre part, avec l'avènement des nouvelles technologies de communication et plus précisément Internet, les entreprises recueillent des masses de données de plus en plus importantes. Ces données étant généralement hétérogènes car elles proviennent de différentes sources. Elles sont dites complexes car elles sont de formats différents et se présentent sur des supports variés. Par exemple, dans le domaine médical, le dossier d'un patient contient des informations

générales sur le patient (âge, sexe, poids, taille, etc.), ainsi que des images de scanner, des interrogatoires sous forme d'enregistrements sonores ou des compte-rendus manuscrits de médecins. En vue d'exploiter de telles données à des fins décisionnelles, il est nécessaire de les structurer et de les homogénéiser. Le langage XML (*eXtensible Markup Language*) s'avère une solution appropriée à ce travail préparatoire sur les données complexes. XML est une norme de W3C¹ et est considéré comme un standard dans la description et l'échange des données. Il représente les données de façon semi-structurée. Sa capacité d'auto-description et sa structure arborescente donnent à ce formalisme une grande flexibilité et une puissance suffisante pour décrire des données complexes, hétérogènes et provenant de sources éparpillées. Les données sont alors stockées dans des documents XML formés conformément à une grammaire associée, exprimée sous forme de DTD (*Document Type Definition*) ou de *Schéma XML*.

Par ailleurs, les entreprises ont largement adopté XML pour la représentation de leurs données complexes. De nos jours, XML gagne tellement du terrain qu'il devient naturel de réfléchir à de nouveaux processus d'entreposage des données XML. Dans ce contexte, plusieurs travaux ont abordé le sujet des entrepôts de données XML traitant des aspects de la *modélisation logique* comme de la *modélisation physique*. Cependant, à notre connaissance, peu de travaux traitent du problème de l'optimisation des *performances* des entrepôts de données XML. À l'heure actuelle, nous ne disposons pas de consensus qui définit d'une manière stable un modèle reconnu d'entrepôts de données basés sur XML. La performance d'un entrepôt de données XML dépend fortement de sa modélisation physique. Par analogie aux entrepôts classiques, le modèle physique d'un entrepôt de données XML doit pouvoir répondre à des requêtes décisionnelles dans des temps raisonnables. Par conséquent, afin de remplir son rôle de base décisionnelle, un tel modèle physique doit inclure des mécanismes adaptés à l'interrogation des données XML.

Boussaid et al. (2006a,b) ont défini une démarche méthodologique, appelée *X-Warehousing*, pour la modélisation multidimensionnelle des données complexes en se basant sur le formalisme XML. Dans cette démarche, XML est perçu comme un langage de modélisation logique dédié à la préparation des contextes d'analyse multidimensionnelle dans les données complexes. Dans cet article, nous nous basons sur le modèle logique de cette approche et proposons un modèle physique qui permet un gain préliminaire en performance. Notre proposition consiste à générer, pour un cube de données, un document XML contenant les faits. Ce dernier est entouré d'un ensemble de documents XML, représentant les dimensions du cube. La liaison entre les faits et les dimensions est assurée par un jeu de *clés virtuelles* représentées par des *attributs ID*. Cette solution constitue un premier pas vers une optimisation des performances des entrepôts de données XML.

2 État de l'art

Les travaux liés aux entrepôts de données XML abordent le problème selon différents aspects. Néanmoins, il est possible de distinguer trois types de problématiques : la *construction* des entrepôts de données XML, leur *exploitation* et leur *gestion*.

Les travaux de Nassis et al. (2005) traitent de l'ingénierie des besoins pour la conception des entrepôts XML à travers un modèle à trois niveaux (*besoins, documents et contraintes*).

¹<http://www.w3.org/>

Dans (Boussaid et al., 2006a,b), les auteurs proposent d'intégrer les données complexes, issues de documents XML, et ce pour les préparer à la modélisation dimensionnelle. En utilisant le langage *XQuery*, Rusu et al. (2005) présentent un processus de construction de l'entrepôt XML qui commence par le nettoyage des données puis par la construction de documents XML intermédiaires (dimensions et faits) et leur liaison. Golfarelli et al. (2001) introduisent un processus de modélisation d'un entrepôt XML à partir des DTDs des documents XML sources conformément à un *Modèle Dimensionnel de Faits*. Les travaux de Vrdoljak et al. (2003) sont similaires aux précédents, mais considèrent les *Schémas XML* plutôt que les DTDs. En plus des sources de données XML, Zhang et al. (2005) utilisent l'historique des requêtes des utilisateurs afin de construire l'entrepôt par la découverte des chemins de requêtes les plus fréquentes.

Sur le plan de la modélisation conceptuelle, la plupart des travaux utilisent UML pour modéliser les concepts dimensionnels de l'entrepôt. Nassis et al. (2004) proposent un diagramme de classes représentant les faits, appelé *XFact*, et des packages UML regroupant les dimensions, appelées *Virtual Dimensions* qui sont des vues XML sur les faits. Jensen et al. (2001) utilisent UML, à travers un processus de transformation, pour décrire des données XML et relationnelles dans un même modèle. Dans le même contexte, Li et An (2005) utilisent UML pour intégrer les sources de données XML, puis transforment le diagramme de classe UML en *Schéma XML*. Ce dernier est utilisé comme méta-données pour les requêtes OLAP.

Sur le plan logique, les *Schémas XML* et les DTDs sont utilisés pour décrire les entrepôts. Pokorný (2001) définit un schéma en étoile appelé *XMLStar Schema*. Ce dernier comporte des hiérarchies de dimensions liées par le mécanisme d'intégrité référentielle. Hümmel et al. (2003) et Binh et al. (2001) utilisent directement des documents XML pour décrire les schémas de l'entrepôt. Ces travaux sont motivés par l'échange des données entre les entrepôts de données classiques. Ainsi, l'utilisation de XML est justifiée par le besoin de décrire des entrepôts ayant des schémas hétérogènes en vue de fournir un mécanisme commun pour l'exploration de leurs structures.

L'analyse OLAP des entrepôts XML est introduite dans plusieurs travaux. Park et al. (2005) proposent un langage, inspiré du langage *MDX* de *Microsoft*, pour l'interrogation des entrepôts de données XML. Cela permet de construire des cubes, appelés *XQ-Cube*, constitués de données agrégées. Dans (Wang et al., 2005), les auteurs définissent deux nouveaux opérateurs d'agrégation qui étendent *XQuery* : *GXagregation* et *XCube*.

Enfin, d'autres travaux se sont focalisés sur la gestion des données des entrepôts XML. Le projet Xylème (2001), dédié à l'entreposage des documents XML à partir du Web, traite de la gestion des rafraîchissements et des performances des requêtes. Ce dernier aspect est mis en œuvre par un mécanisme d'index, appelé *MyIndex*, et un opérateur d'interrogation, appelé *PatternScan*.

Certains travaux, notamment (Park et al., 2005) et (Rusu et al., 2004) abordent la conception physique des entrepôts XML sous d'autres angles (processus de construction, analyse OLAP). À notre connaissance, aucun travail ne traite la problématique de la conception physique d'un entrepôt de données XML afin de réduire les temps de réponse des requêtes décisionnelles. Dans cet article, nous proposons un modèle physique basé sur le modèle logique proposé dans (Boussaid et al., 2006a,b). Le modèle consiste à découper un cube de données XML en un document des faits et plusieurs documents de dimensions. La liaison entre les deux types de documents est assurée par un mécanisme de *clés primaires* et de *clés étrangères*. Cette solution, retenue parmi d'autres, permet un gain préliminaire en performances de l'entrepôt.

3 Modélisation logique d'un cube de données XML

Dans cette section, nous reprenons le modèle logique du *cube XML* proposé dans le cadre de la méthodologie *X-Warehousing* de Boussaid et al. (2006a,b). Selon cette dernière, un modèle logique d'un cube de données est décrit par un schéma XML. En respectant un certain nombre de règles, le schéma XML est perçu comme un langage de modélisation logique d'un fait OLAP. Nous résumons ces règles par les points suivants :

- le nom du fait correspond au nom de l'élément racine du document XML ;
- le nom d'une mesure correspond au nom d'un attribut dans l'élément racine du document XML ;
- la valeur d'une mesure correspond à la valeur que prend l'attribut associé à cette mesure dans le document XML ;
- le nom d'une dimension correspond au nom d'un sous-élément de l'élément racine du document XML ;
- le nom d'un niveau hiérarchique d'une dimension correspond au nom d'un sous-élément de l'élément associé à cette dimension dans le document XML ;
- la modalité que prend un fait dans un niveau hiérarchique d'une dimension correspond à la valeur que prend un attribut dans l'élément associé à ce niveau hiérarchique dans le document XML.

Par exemple, dans le cas d'une application médicale dont l'objectif est de prévenir le cancer du sein, on peut établir un contexte d'analyse en ligne concernant les régions suspectes détectées dans des radios de mammographies. Dans un tel contexte, une région suspecte est décrite par un ensemble de descripteurs catégoriels qui renseignent sur ses propriétés médicales. Elle est aussi décrite par ses propriétés spatiales qui déterminent sa position et sa frontière sur une radio. Ainsi, comme le montre le modèle logique de la figure 1 (a), nous proposons d'analyser les régions suspectes selon les dimensions suivantes :

- D_1 : le type de la lésion (*Lesion_type*) ;
- D_2 : l'indice d'évaluation de la lésion (*Assessment*) ;
- D_3 : l'indice de subtilité de la lésion (*Subtlety*) ;
- D_4 : la pathologie (*Pathology*) ;
- D_5 : la date d'examen de la patiente qui correspond à la région suspecte (*Date_of_study*) ;
- D_6 : la date de numérisation de la radio de la région suspecte (*Date_of_digitization*) ;
- D_7 : la frontière de la région suspecte marquée par les médecins radiologues sur la radio (*Boundary*) ;
- D_8 : l'image radio numérisée de la région suspecte (*Scanner_image*) ;
- D_9 : la patiente qui correspond à la région suspecte (*Patient*) ;
- D_{10} : la machine utilisée pour la numérisation de l'image radio (*Digitizer*).

Une région suspecte est observée selon la mesure du *périmètre* de sa frontière (*Boundary_length*). Cette mesure indique l'importance de la région suspecte. En effet, plus le périmètre d'une région est grand, plus la surface de cette région est importante. Une région à grande surface peut représenter un risque d'une tumeur maligne. En plus du périmètre, pour une région suspecte, nous proposons aussi d'utiliser une deuxième mesure dérivée. Elle renseigne sur le *nombre des régions* suspectes dans le dossier de la patiente concernée (*Number_of_regions*).

D'après la figure 1 (a), le nom du fait *région suspecte* (*Suspicious_region*) correspond au nom de l'élément racine déclaré par le schéma XML. Les mesures de ce fait, *périmètre de la*

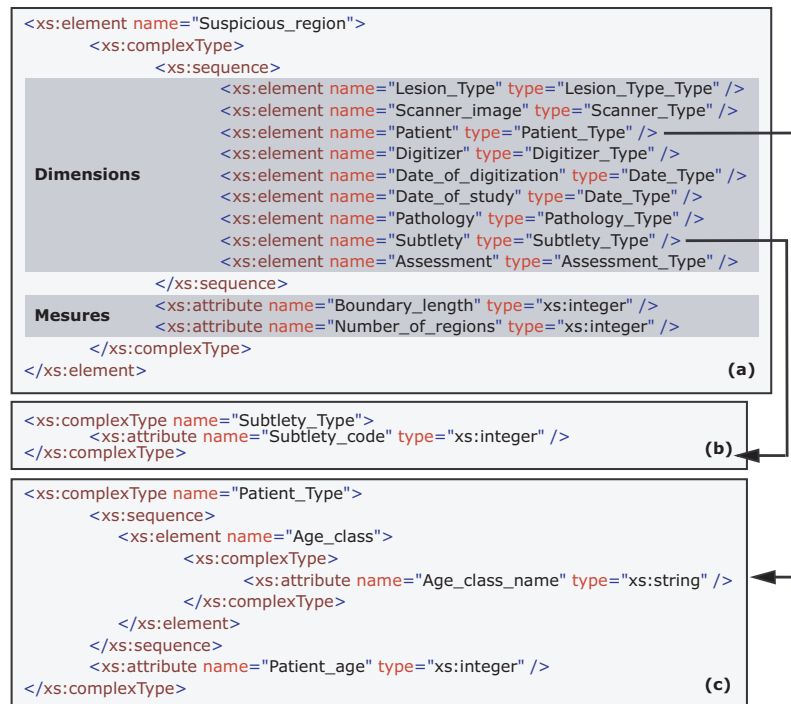


FIG. 1 – Modèle logique du cube XML des régions suspectes

frontière (*Boundary_length*) et nombre de régions (*Number_of_regions*), sont déclarées en tant qu'attributs, de type entier, dans l'élément racine.

Les dimensions d'une région suspecte sont représentées par des sous-éléments de l'élément racine. Chacun de ces sous-éléments est associé à un *type complexe (complexType)* qui définit la hiérarchie d'une dimension. Par exemple, la figure 1 (b) représente le type complexe qui définit la hiérarchie de la dimension de l'indice de *subtilité (Subtlety)*. Cette dernière comporte un seul niveau hiérarchique. La figure 1 (c) représente un autre type complexe associé à la dimension de la *patiente (Patient)* qui définit deux niveaux hiérarchiques : l'*âge (Patient_age)* et la *classe d'âge* de la patiente (*Age_class*).

4 Modèle physique proposé

En employant un schéma XML, l'approche *X-Warehousing* permet de décrire le modèle logique d'un cube XML. Un documents XML valide au regard de ce modèle logique représente un fait à analyser. Ainsi, sur le plan physique, le cube XML est représenté par l'ensemble de tels documents XML. Dans ce cas, il y aura autant de documents XML que de faits. En plus, les instances de chaque dimension et ses hiérarchies seront incorporées dans le même document

Performance des entrepôts de données XML

XML. S'agissant d'un entrepôt de données, le nombre des documents est appelé à augmenter significativement, ce qui rend cette solution inadéquate, à cause de :

- la redondance des informations des dimensions dans plusieurs cubes XML ;
- l'analyse OLAP de l'entrepôt nécessiterait des temps de réponses prohibitifs. En effet, le calcul d'agrégats, courant dans l'analyse OLAP, sera difficile à mettre en œuvre car ce dernier devra agir sur des documents séparés.

L'autre solution, extrême également, consiste à stocker tous les faits et dimensions dans un même document XML. L'avantage de cette solution réside dans la présence des informations nécessaires à l'analyse dans le même document XML. Cependant, outre l'inconvénient de redondance signalé ci-dessus, cette solution ne résisterait pas au passage à l'échelle.

Nous présentons plus loin des résultats empiriques permettant d'illustrer les temps d'exécution d'une charge de requêtes pour chacune des configurations physiques précédentes.

La solution que nous proposons consiste à séparer les faits de dimensions. Ainsi, tous les faits seront regroupés dans un document XML, comme des sous-éléments d'un élément racine, désigné par un nom au pluriel. D'autre part, nous construisons autant de documents XML que de dimensions. Nous lions les documents XML des dimensions à celui des faits par le mécanisme de clé primaire/clé étrangère semblable à celui du modèle relationnel.

Soit (F, \mathcal{D}) un schéma en étoile représentant un cube de données, où F est un ensemble de faits et $\mathcal{D} = \{D_s, 1 \leq s \leq r\}$ est un ensemble de r dimensions. Le modèle physique du cube XML (F, \mathcal{D}) est un ensemble de documents XML comportant :

- r documents XML de dimensions. Nous désignons chaque document de dimension par $D_s.xml$ ($1 \leq s \leq r$) ;
- un document XML des faits, que nous désignons par $Faits.xml$.

En supposant qu'une dimension D_s contient un ensemble de n_s attributs notés $\{D_s.A_i, 1 \leq i \leq n_s\}$, le document XML $D_s.xml$, qui représente cette dimension, est structuré selon les éléments suivants :

- un élément racine $\langle Dim_s \rangle$;
- une dimension est un sous-élément $\langle D_s \rangle$ de l'élément racine $\langle Dim_s \rangle$;
- $\forall i \in \{1, \dots, n_s\}$, $D_s.A_i$ définit un sous-élément XML inclut dans $\langle Dim_s \rangle$;
- il existe un seul indice $k \in \{1, \dots, n_s\}$ tel que l'attribut $D_s.A_k$ joue le rôle de clé artificielle pour l'élément $\langle Dim_s \rangle$;
- la hiérarchie d'une dimension est construite par imbrications successives d'éléments sous l'élément $\langle Dim_s \rangle$ selon la granularité des niveaux de la hiérarchie.

En supposant qu'un fait dans F comporte m mesures $\{m_q, 1 \leq q \leq m\}$, le document XML $Faits.xml$, qui représente les faits, est structuré selon les éléments suivants :

- un élément racine $\langle Fait \rangle$;
- un fait F est un sous-élément $\langle F \rangle$ de l'élément racine $\langle Fait \rangle$;
- $\forall q \in \{1, \dots, m\}$, $F.M_q$ définit un attribut XML de l'élément $\langle F \rangle$;
- $\forall s \in \{1, \dots, r\}$, $\langle D_s \rangle$ est un sous éléments XML de l'élément $\langle F \rangle$;
- le contenu de l'élément $\langle D_s \rangle$ référence les valeurs d'attributs $D_s.A_k$ des éléments dans le document $D_s.xml$;
- par analogie au modèle relationnel, l'élément $\langle D_s \rangle$ dans $Faits.xml$ est une clé étrangère référençant la clé primaire $D_s.A_k$ dans le document $D_s.xml$.

```
<Produits>
.
.
  <Produit Produit_id = "PD_1" >
    < Description_produit> Yaourt </Description_produit>
    < Description_sous_categorie> Produits laitiers
      < Description_categorie> Produits alimentaires </Description_categorie>
    </Description_sous_categorie>
  </Produit>
.
.
</Produits>
```

FIG. 2 – Exemple du document *Produit.xml*

```
<Dates>
.
.
  <Date Date_id = "D_06_61" >
    <Jour>02-03-2006
      <Mois> Mars </Mois>
      <Année> 2006 </Année>
    </Jour>
  </Date>
.
.
</Dates>
```

FIG. 3 – Exemple du document *Date.xml*

Pour illustrer les formalismes ci-dessus, nous présentons un exemple classique d'un entrepôt de données XML concernant l'activité des ventes d'une chaîne de magasins (Kimball, 1996). Cette activité est analysée selon trois dimensions : le *produit*, la *date* et le *magasin*. Elle est également observée selon deux mesures : la *quantité vendue* et le *montant des ventes*. Selon notre modèle physique, l'entrepôt de données XML comportera quatre documents XML :

- trois documents pour les dimensions : *produit.xml* (figure 2), *date.xml* (figure 3) et *magasin.xml* (figure 4) ;
- un document pour les faits *vente_journalière.xml* (figure 5).

Performance des entrepôts de données XML

```
<Magasins>
.
.
  <Magasin Magasin_id = "M_31_12">
    <Nom_magansin>Magasin du Saint Michel</Nom_magansin>
    <Adresse_magansin>51 rue Saint Michel
      <Code_postal>69003</Code_postal>
      <Ville_magansin>Lyon
        <Region_magansin>Rhône-Alpes</Region_magansin>
      <Ville_magansin>
    </Adresse_magansin>
  </Magasin>
.
.
</Magasins>
```

FIG. 4 – Exemple du document *Magasin.xml*

```
<Ventes_journalieres>
.
.
  <Vente_journaliere Quantite_vendue ="1370" Montant_vente="1233">
    <Produit> PD_1 </Produit>
    <Date> D_06_61 </Date>
    <Magasin> M_31_12 </Magasin>
  </Vente_journaliere>
.
.
</Ventes_journalieres>
```

FIG. 5 – Exemple du document *vente_journalière.xml*

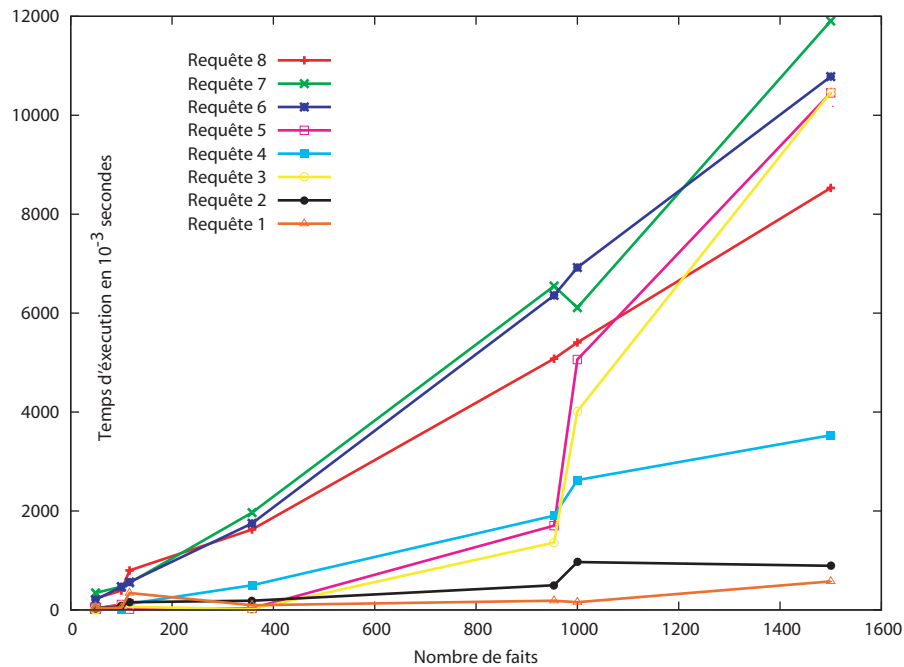


FIG. 6 – Résultat de l'exécution des requêtes

5 Expérimentations et discussion

5.1 Expérimentations

5.2 Discussion

6 Conclusion et perspectives

Références

- Binh, N. T., A. M. Tjoa, et O. Mangisengi (2001). MetaCube-X : An XML Metadata Foundation for Interoperability Search among Web Data Warehouses. In *Proceedings of the 3rd International Workshop on Design and Management of Data Warehouses (DMDW'2001)*, Interlaken, Switzerland, pp. 8. CEUR-WS.org.
- Boussaid, O., R. B. Messaoud, R. Choquet, et S. Anthoard (2006a). Conception et construction d'entrepôts XML. In *2^{ème} journée francophone sur les Entrepôts de Données et l'Analyse en ligne (EDA'2006)*, Revue des Nouvelles Technologies de l'Information, Versailles, France, pp. 3–21. Cepaduès Editions.

- Boussaid, O., R. B. Messaoud, R. Choquet, et S. Anthoard (2006b). X-Warehousing : an XML-Based Approach for Warehousing Complex Data. In *Proceedings of the 10th East-European Conference on Advances in Databases and Information Systems (ADBIS'2006)*, Thessaloniki, Greece, pp. 39–54. Springer-Verlag.
- Chaudhuri, S. et U. Dayal (1997). An Overview of Data Warehousing and OLAP Technology. *SIGMOD Record* 26(1), 65–74.
- Golfarelli, M., S. Rizzi, et B. Vrdoljak (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.
- Hümmer, W., A. Bauer, et G. Harde (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.
- Inmon, W. H. (1996). *Building the Data Warehouse*. John Wiley & Sons.
- Jensen, M. R., T. H. Moller, et T. B. Pedersen (2001). Specifying OLAP Cubes on XML Data. *Journal of Intelligent Information Systems* 17(2-3), 255–280.
- Kimball, R. (1996). *The Data Warehouse Toolkit*. John Wiley & Sons.
- Li, Y. et A. An (2005). Representing UML Snowflake Diagram from Integrating XML Data Using XML Schema. In *Proceedings of the 2005 International Workshop on Data Engineering Issues in E-Commerce (DEEC '2005)*, Tokyo, Japan, pp. 103–111. IEEE Computer Society.
- Nassis, V., R. Rajugan, T. S. Dillon, et W. Rahayu (2004). Conceptual Design of XML Document Warehouses. In *Proceedings of the 6th International Conference Data Warehousing and Knowledge Discovery (DaWaK 2004)*, Zaragoza, Spain, pp. 1–14. Springer.
- Nassis, V., R. Rajugan, T. S. Dillon, et W. Rahayu (2005). Conceptual and Systematic Design Approach for XML Document Warehouses. *International Journal of Data Warehousing and Mining* 1(3), 63–86.
- Park, B.-K., H. Han, et I.-Y. Song (2005). XML-OLAP : A Multidimensional Analysis Framework for XML Warehouses. In *Proceedings of the 7th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2005)*, Copenhagen, Denmark, pp. 32–42. Springer.
- Pokorný, J. (2001). Modelling Stars Using XML. In *Proceedings of the 4th ACM International Workshop on Data Warehousing and OLAP (DOLAP 2001)*, Atlanta, Georgia, USA, pp. 24–31. ACM Press.
- Rusu, L. I., J. W. Rahayu, et D. Taniar (2004). On Building XML Data Warehouses. In *Proceedings of the 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2004)*, Exeter, UK, pp. 293–299. Springer.
- Rusu, L. I., J. W. Rahayu, et D. Taniar (2005). A Methodology for Building XML Data Warehouses. *International Journal of Data Warehousing and Mining* 1(2), 67–92.
- Vrdoljak, B., M. Banek, et S. Rizzi (2003). Designing Web Warehouses from XML Schemas. In *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2003)*, Prague, Czech Republic, pp. 89–98. Springer.
- Wang, H., J. Li, Z. He, et H. Gao (2005). OLAP for XML Data. In *Proceedings of the*

- 5th International Conference on Computer and Information Technology (CIT'2005)*, Dallas, Texas, USA, pp. 233–237. IEEE Computer Society.
- Xylème, L. (2001). A Dynamic Warehouse for XML Data of the Web. *IEEE Data Engineering Bulletin* 24(2), 40–47.
- Zhang, J., W. Wang, H. Liu, et S. Zhang (2005). X-Warehouse : Building Query Pattern-driven Data Warehouse. In *Proceedings of the 14th International Conference on World Wide Web (WWW 2005)*, Chiba, Japan, pp. 896–897. ACM Press.

Summary