# Intégration de bases de données hétérogènes via XML

Myriam Lamolle, Nedra Mellouli

# Intégration de Bases de Données hétérogènes via XML

**Myriam Lamolle, Nédra Mellouli**

*LINC-IUT de Montreuil, Université Paris8*
*140, rue de la Nouvelle France – 93100 Montreuil*
*(Adresse institutionnelle complète)*
*{m.lamolle, n.mellouli}@iut.univ-paris8.fr*

RÉSUMÉ. *Les bases de données hétérogènes et les bases de connaissances prennent, de nos jours, une part prépondérante dans les nouvelles applications liées au web Les requêtes permettant de répondre aux demandes d'information des utilisateurs deviennent très complexes. De plus, l'extraction des données les plus pertinentes devient de plus en plus difficile dès que les sources de données sont hétérogènes et nombreuses. Dans ce contexte, nous proposons une approche générique basée sur les techniques XML. Cette approche prend en compte l'échange de données et de métadonnées au travers de règles XML dans la perspective de bases de données.*

ABSTRACT. *Actually, the use of heterogeneous database and knowledgebase increases with the new web applications. The requests allowing to replying to users become very complex. Then, the extraction of the most relevant data becomes more and more difficult when the data sources are heterogeneous and numerous. In this context, we suggest a generic approach based upon XML techniques. This approach takes into account the data and metadata mapping through XML rules in the perspective of database.*

MOTS-CLÉS : *bases de données hétérogènes, intégration de bases de données, définition de schéma XM, métadonnées XML.*

KEYWORDS: *heterogeneous database, database integration, XML Schema Definition, XML metadata.*

## 1. Introduction

Les bases de données hétérogènes et les bases de connaissances prennent, de nos jours, une part prépondérante dans les nouvelles applications liées au web. Les requêtes permettant de répondre aux demandes d'information des utilisateurs deviennent très complexes. De plus, l'extraction des données les plus pertinentes devient de plus en plus difficile dès que les sources de données sont hétérogènes et nombreuses (Verso, 1997). Dans ce contexte, nous proposons une approche générique basée sur les techniques XML. Cette approche prend en compte l'échange de données et de métadonnées au travers de règles XML dans la perspective de bases de données (Manolescu *et al*, 2000).

Notre approche est parente de projets en cours pour résoudre les problèmes relatifs à l'intégration de bases de données hétérogènes. Nous allons présenter ces différents projets dans la seconde partie de cet article. Puis, nous présenterons dans le chapitre suivant notre propre solution basée sur une architecture générique XML. Cette architecture permet de gérer l'évolution des schémas des bases de données et la génération de métaschémas XML contenant la sémantique portée par ces schémas de bases de données. Pour ceci faire, nous extrayons les schémas des bases de données et nous mettons à jour ces schémas en utilisant des fichiers XSD (XML Schema) et des règles de correspondance entre métaschémas sous forme de fichiers XSL. Nous pensons qu'il est essentiel d'avoir un unique format de représentation (XML) pour extraire, manipuler, mettre à jour les données et les métadonnées comme nous le verrons dans le chapitre quatre. Enfin, nous énoncerons les évolutions futures à intégrer dans notre architecture.

## 2. Etat de l'art

Il existe dans la littérature différents systèmes et différents projets plus ou moins avancés qui ont des points communs avec notre travail. Nous citons par exemple le projet Xyleme, Castor, Verso, Tsimmis, TopLink, BusinessObject, etc. Dans cet article, nous nous plaçons par rapport à trois projets très cités dans la littérature à savoir Xyleme (Xylème), Castor (Castor) et Verso (Verso, 1997). En effet, Xyleme est un prototype d'un entrepôt de données dynamiques. Il permet l'intégration d'informations hétérogènes de sources différentes et distantes. Ces informations doivent être structurées en documents XML. Or, actuellement le nombre de documents XML sur le web est assez limité, et les moteurs de recherche s'appuient sur des bases de données. Dans ce contexte, notre travail peut être complémentaire dans le sens où il peut transformer des bases de données en documents XML. Parmi les projets existants permettant de réaliser ce type de transformations, nous trouvons Castor et Verso. Le projet Castor (Castor) permet le *« mapping »* entre bases de données de n'importe quel type et un objet Java, c'est-à-dire que chaque attribut est représenté par une classe Java, manipulée par deux opérateurs : *Get* et *Set*. L'outil Castor (qui est un logiciel libre) paraît intéressant dans le but de réaliser des

transformations de bases de données relationnelles ou objets en véritable objets Java et donc en un système de gestion d'objets ; Castor pourrait pourquoi pas devenir un véritable système de gestion de bases de données objets, plus performant que ceux qui existent actuellement. Quand au projet Verso (Verso, 1997), il est capable de réaliser des migrations de bases de données relationnelles en bases de données objets et de réunir plusieurs bases de données objets. Le seul inconvénient de ce système étant d'utiliser un langage complètement nouveau.

## 3. Notre approche

L'évolution perpétuelle du Web a engendré la diversité des applications ainsi que des informations manipulées. Les données interrogées via des applications Web, peuvent prendre différentes formes, telles que des fichiers ordinaires non structurés, des entrepôts de données, des documents ou des données rigidement structurées en tant que bases de données. S'il s'agit de faire coopérer ces différentes et multiples formes, il est impératif de considérer la sémantique de ces données et de l'exploiter le plus possible.

Pour atteindre cet objectif, nous procédons d'une manière incrémentale. Tout d'abord, nous ciblons notre réflexion sur la fédération de données structurées à savoir des bases de données hétérogènes, objets ou relationnelles.

Pour décrire la sémantique du schéma des bases de donnés fédérées (Figure 1), nous nous basons sur la formalisation en XML du métaschéma[1], sous forme de fichiers XSD (Xylème). Le choix d'utiliser XML comme formalisme de représentation, outre son succès en étant la norme du Web, est lié à sa capacité de représenter tout type de données ; notamment, le formalisme XSD (Van Der Llist *et al*, 2001) est beaucoup plus riche que celui des DTD. Cependant, la génération des fichiers XSD implique des choix de construction des types utilisateurs. Ces choix sont très importants car l'efficacité de la transformation d'un métaschéma en un autre en dépend.

[1] Nous portons à l'attention des lecteurs que nous n'utilisons pas le terme de métaschéma défini par la norme W3C mais comme un schéma porteur de **méta**connaissances adjoint au schéma de la base de données

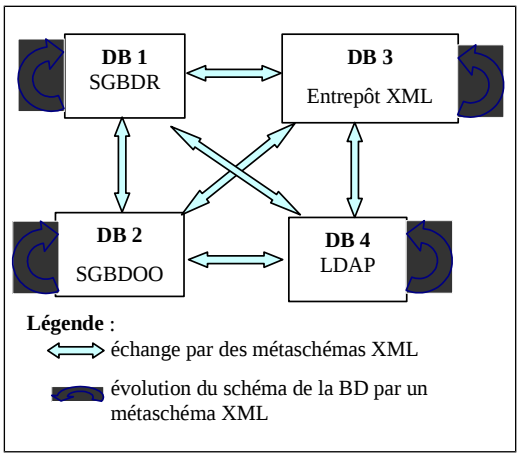


**Figure 1.** *Transfert d'informations via XML*

A l'heure actuelle, le métaschéma permet une description complète du schéma de la base à partir des catalogues système du SGBD correspondant. A court terme, nous envisageons d'enrichir ce métaschéma en rajoutant des connaissances introduites par les concepteurs des bases de données ; et à long terme, nous voulons faire évoluer notre projet vers une application intégrant les capacités des bases de connaissances.

L'association des métaschémas aux bases de données permet de réaliser les finalités suivantes :

– Une extraction partielle ou totale du schéma de la base de données,

– Une comparaison de deux types de bases de données grâce à leur métaschéma par le formalisme XSD,

– Une meilleure compréhension de la base de données sans faire recours à sa structure,

– Un échange facile entre des bases de données différentes, grâce aux règles de correspondances XSL définies entre les métaschémas (Figure 2),

– Une détection rapide des changements de schémas des bases de données,

– Une migration assez simple d'un type de bases de données à un autre,

– La transparence des sources, des types et des structures des données.

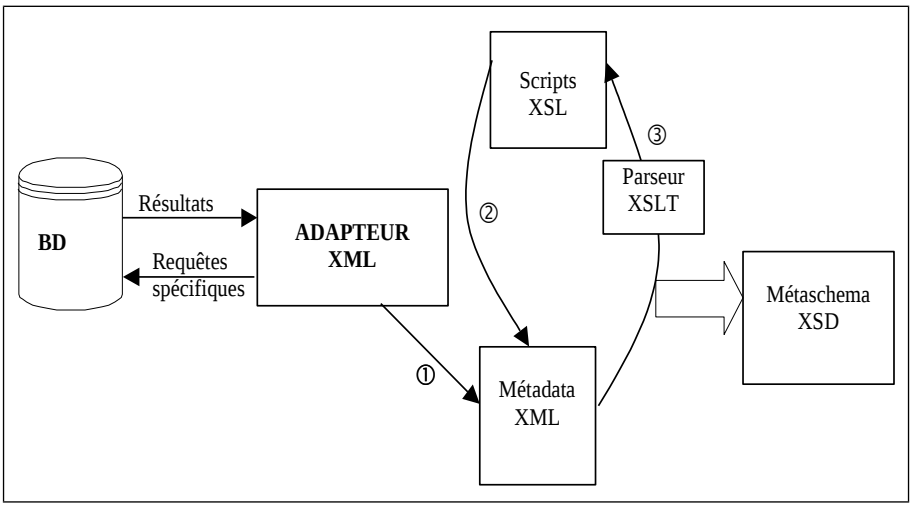


**Figure 2.** *Génération des métaschémas XSD*

Dans ce travail, nous nous focalisons sur deux grands axes de réflexion à savoir la détection des changements (Cobéna G *et al*., 2001) de schémas des bases de données et la liaison de métaschémas XML.

### 3.1. *Détection des changements de schémas des bases de données*

Nous constituons une collection de métaschémas XML au dessus du schéma des bases de données à fédérer. Chaque métaschéma mémorise le transfert entre la dernière version du schéma et la version antérieure de ce même schéma. Le but du métaschéma est de rendre plus facile l'ajout d'informations sur le schéma de la base qui requiert une nouvelle structuration. Pour l'instant, nous suggérons de préserver autant que possible le schéma initial de la base de données.

### 3.2. *Mise en relation de métaschémas XML*

Tout d'abord, nous construisons une collection de métaschémas XML autour d'un schéma de bases de données hétérogènes (Garcia-Molina *et al*, 1997). Un fichier de transformation concerne un couple de métaschémas. Les métaschémas et les fichiers de transformations permettent de constituer un graphe de la fédération des bases de données où les métaschémas sont les nœuds et les fichiers de transformations sont les arcs. Nous utiliserons les fichiers de transformations comme des opérateurs de passage et de liaison (Bernstein *et al*, 2000). Le but est d'atteindre un certain niveau de généricité de notre application (Bernstein, 2000).

## 4. La gestion des schémas de bases de données

La génération des métaschémas est obtenue selon deux cas. Le premier cas correspond à l'intégration d'une nouvelle base de données dans la fédération de bases de données existantes.

Ceci donne lieu à la création d'un nouveau fichier XSD (*schema.xsd* issu de la réponse *non* dans la figure 3) qui contient les principales métadonnées de cette nouvelle intégration comme nous l'avons expliqué dans le chapitre précédent.

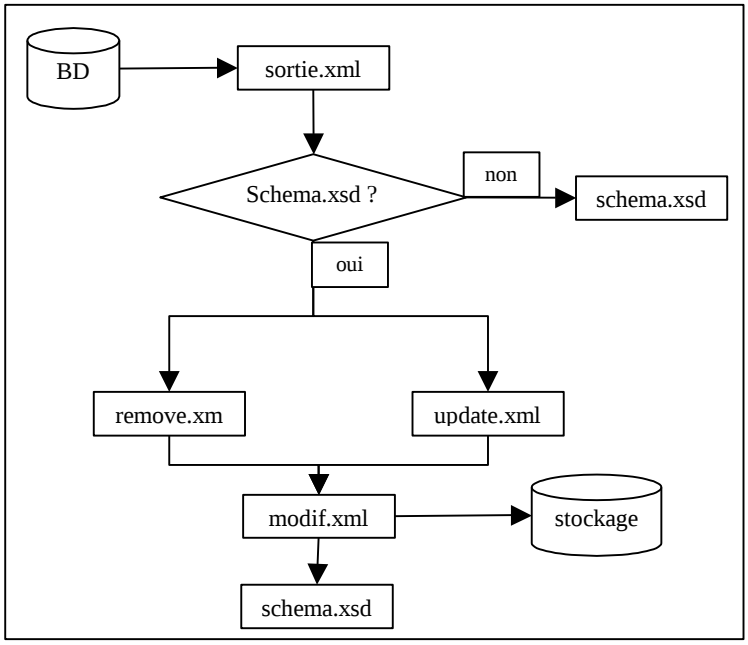


**Figure 3.** *Génération des métaschémas XML*

Le second cas correspond à l'évolution du schéma d'une base de données (versioning) déjà intégrée dans une fédération de base de données. Nous opérons alors une mise à jour du fichier XSD existant (schema.xsd issu de la réponse oui dans la figure 3). Cependant, cette mise à jour nécessite de détecter les changements opérés (Cobéna G *et al*., 2001) sur le schéma (sortie.xml dans la figure 3) et de mémoriser uniquement dans le schéma XSD existant ces changements (modif.xml dans la figure 3). Les différentes évolutions de schéma sont gardées par un stockage de tous ces fichiers de modification.

## 5. Conclusions et perspectives

Dans cet article, nous avons présenté une approche générique pour gérer des bases de données hétérogènes via XML. L'architecture que nous avons proposée nous permet de prendre en compte l'évolution des schémas des bases de données par des métaschémas et de gérer la fédération de bases de données par la mise en correspondance via des fichiers XSL des métachémas représentés par des fichiers XSD. Ces deux aspects facilitent l'intégration de nouvelles données dans une fédération de bases de données. Ceci est réalisé de façon incrémentale. Dans un premier temps, une extraction partielle ou totale du schéma de la base de données (relationnelle, objet-relationnelle, objet ou xml) est effectuée. Puis, nous traduisons ce schéma en un métaschéma. Enfin, nous établissons des correspondances entre les métaschémas générés constituant ainsi une cartographie de la fédération des bases de données. Cependant, l'établissement de ces correspondances ne peut être faite sans la collaboration des experts des différentes bases de données.

Les travaux futurs à réaliser sont l'adjonction de connaissances dans les métaschémas, puis la génération semi-automatique des correspondances entre métaschémas, enfin l'utilisation de ces métaschémas lors de la construction des requêtes.

## 6. Bibliographie

Bernstein P.A., Levy A. Y., Pottinger R.A., A Vision for Management of Complex Models. Technical Report MSR-TR-2000-53, http://www.research.microsoft.com/pubs/, June 2000.

Bernstein P.A., Panel: Is Generic Metadata Management Feasible? Proceedings of the 26th International Conference on Very Large Databases (VLDB), Cairo, Egypt, 2000.

Castor, http://castor.exolab.org/

Garcia-Molina H., Papakonstantinou Y., Quass D., Rajaraman A., Sagiv Y., Ullman J., and Widom J., « The TSIMMIS project: Integration of heterogeneous information sources », *Journal of Intelligent Information Systems*, 8(2):117-132, 1997.

Manolescu I., Florescu D., Kossmann D., Xhumari F., and Olteanu D., « Agora: Living with XML and Relational », *Proceedings of the 26th International Conference on Very Large Databases VLDB'00*, Cairo, Egypt, 2000.

XML Schema Part 0 : Primer. Available at http://www.w3c.org/TR/xmlschema-0/.

Cobéna G., Abiteboul S., Marian A., Detecting changes in XML Documents. Rapport interne du projet Xylème, INRIA-Rocquencourt, Mars 2001.

Van Der Llist E., *XML Schema, The W3C's Object-Oriented Descriptions for XML*, Edition O'Reilly, 2002

Verso, http://www.inria.fr/rapportsactivite/RA97/verso/verso.html., Rapport d'activité, 1997, INRIA.

Xylème, http://www-rocq.inria.fr/verso/research/xyleme/, Rapport d'activité, INRIA.