Article

Vers l'interopérabilité des systèmes d'information hétérogènes

Laïla BENHLIMA, Département Informatique Ecole Mohammadia d'Ingénieurs BP 765 Agdal, Rabat, Maroc, benhlima@emi.ac.ma.

Dalila CHIADMI, Département Informatique Ecole Mohammadia d'Ingénieurs BP 765 Agdal, Rabat, Maroc, chiadmi@emi.ac.ma.

Date de publication : 27 décembre 2006

Résumé

Des applications émergentes telles que le e-gouvernement, le e-learning, les bibliothèques électroniques, etc. nécessitent l'accès à diverses sources d'information. Ces dernières sont généralement hétérogènes, que ce soit au niveau syntaxique ou sémantique. En effet, des conflits sémantiques surviennent puisque les systèmes n'utilisent pas la même interprétation de l'information qui est définie différemment d'une organisation à l'autre.

Dans cet article, nous focalisons sur l'hétérogénéité sémantique entre systèmes d'information et nous proposons un système de médiation fondée sur des ontologies, pour permettre l'intégration de sources hétérogènes et réparties. Les ontologies permettent aux systèmes d'utiliser une terminologie consensuelle. Nous utilisons l'approche Global As View (GAV) pour le mapping entre les différentes ontologies. Ce mapping sera utilisé lors du processus de réécriture de requêtes. Les ontologies, ainsi que le mapping, sont décrits avec le langage OWL. Nous utilisons XQuery, le langage d'interrogation de documents XML comme langage de requêtes au niveau de notre médiateur et une algèbre XML pour le traitement des requêtes.

Abstract

Emerging applications such as e-gov, e-learning, digital libraries, etc. need access to different information sources. These are generally syntactically and semantically heterogeneous. Semantic conflicts occur since systems don't use the same interpretation of the information. This is because the way organisations define data semantics is generally different from the way other organisations define it.

In this paper, we focus on semantic heterogeneity between information systems and we propose an ontology-based mediation to enable distributed and heterogeneous data sources integration. Ontology allows systems to agree on the terms they use. We use a Global As View (GAV) approach for the mapping between ontologies. This mapping will be used in query rewriting process. Ontologies and mapping are described using OWL language. We use XQuery, the query language for XML documents as query language in our mediator and XML Algebra to process the queries.

Table des matières

Texte intégral

1 INTRODUCTION

On assiste depuis quelques années à l'émergence de nouvelles applications qui ont besoin de partager des informations entre différents systèmes. C'est le cas du e-gouvernement, du e-learning, du e-commerce, de la bioinformatique ou encore des bibliothèques électroniques. Or, dans ce contexte, les systèmes d'informations (SI), conçus et développés par des organisations différentes, constituent généralement des sources de données autonomes et hétérogènes.

De ce fait, l'interopérabilité entre ces systèmes d'information est complexe puisque les applications doivent être adaptées pour pouvoir déterminer, pour chaque requête, les sources de données pertinentes, la syntaxe requise pour l'interrogation, la terminologie (concept) propre à la source, et pour pouvoir combiner les fragments de résultats issus de chaque source en vue de construire le résultat final. Ce processus d'adaptation peut être plus ou moins complexe : exploitation des résultats d'une source pour interroger une autre, élimination des redondances, etc.

L'intégration virtuelle des sources de données hétérogènes, autonomes et réparties est une solution pour l'interopérabilité entre différents systèmes d'information, puisqu'elle simplifie l'accès aux données. L'intégration virtuelle est fondée sur la médiation et plus particulièrement sur le couple médiateur-adaptateur. Le médiateur a pour rôle de masquer l'hétérogénéité et la répartition des sources de données. Quant à l'adaptateur, il a pour fonction d'adapter les requêtes aux formats des sources de données.

L'hétérogénéité est non seulement due aux différents formats de structuration des systèmes d'information mais également aux multiples interprétations que des systèmes autonomes peuvent avoir de la même donnée. Ainsi, l'intégration de sources de données hétérogènes, autonomes et réparties passe par la résolution de ces conflits sémantiques et différences syntaxiques.

Nous nous intéressons particulièrement à l'intégration sémantique qui représente un défi aussi bien pour les chercheurs que pour les industriels. En effet, des millions de dollars sont dépensés pour des projets d'intégration (Park et Ram, 2004), (Doan et Halevy, 2004). Par ailleurs, l'intégration sémantique suscite de plus en plus d'intérêt avec l'émergence du web sémantique. Grâce à ce dernier, les systèmes devraient pouvoir échanger des informations et des services dans un contexte sémantiquement riche. Plus généralement, le web sémantique vise à fournir des données compréhensibles aussi bien par les sujets humains que par les machines. Ces dernières pourront effectuer des traitements automatiques par des modules logiciels grâce à un enrichissement sémantique des données.

Nous proposons une solution pour l'intégration de systèmes d'informations hétérogènes, fondée sur la médiation et les ontologies pour résoudre les conflits sémantiques et syntaxiques. Nous visons l'hétérogénéité sémantique aussi bien au niveau des schémas (structures) qu'au niveau du contenu (données), et ceci, tout en préservant l'autonomie des sources de données. Pour faciliter l'accès à l'information, nous utilisons un langage de requêtes standard, à savoir XQuery (Chamberlin et al., 2002), à travers une interface adaptative. Le traitement des requêtes passe par une algèbre XML. Quant aux ontologies, elles sont représentées à l'aide du langage OWL (Smith, M.K. et al., 2003).

La suite de l'article est organisée comme suit : dans la section 2, nous focalisons sur l'hétérogénéité des systèmes d'information, ainsi que sur les systèmes d'intégration et les ontologies. Dans la section 3, nous passons en revue quelques systèmes d'intégration existants, avant de présenter notre système de médiation WASSIT en section 4. Quelques composants du médiateur sont décrits en section 5 qui présente également un cas d'utilisation afin d'illustrer les différents traitements réalisés par notre système. La section 6 détaille les

modules d'analyse et de réécriture. Enfin, nous décrirons l'implémentation en section 7 avant de conclure par un aperçu des perspectives de ce travail.

2 HÉTÉROGÉNÉITÉ, ONTOLOGIES ET INTÉGRATION

2.1 Des systèmes d'information hétérogènes

Etant conçus par des communautés différentes, les systèmes d'information sont autonomes et hétérogènes. L'hétérogénéité se situe à deux niveaux : syntaxique et sémantique.

L'hétérogénéité syntaxique : se retrouve dans les formats de stockage des données (XML, relationnel, objet, etc.), dans les langages d'interrogation (XQuery, SQL, OQL, etc.), dans les protocoles d'accès (HTTP, etc.), dans les interfaces, etc.

L'hétérogénéité sémantique : représente les différences entre les interprétations du monde réel selon le contexte et l'utilisation des données. Ceci a généralement lieu durant le processus de conception des systèmes d'information. Les conflits sémantiques peuvent survenir au niveau des schémas et des données (Park et Ram, 2004).

Les conflits au niveau des données résultent de l'utilisation de domaines de données différents selon le type d'application (e-gov, e-commerce, bioinformatique, etc.). En effet, des données similaires peuvent être représentées et interprétées différemment dans chaque domaine.

Les conflits des schémas sont, quant à eux, caractérisés par des différences dans les structures logiques ou méta données.

Dans ce contexte, Goh (Goh, Bressan et al, 1999) a identifié quatre principaux types de conflit : conflits de nom, conflits d'échelle, conflits d'indéterminisme (*confounding conflicts*) et conflits de représentation. Nous les commentons dans ce qui suit.

- Les conflits de nom sont liés aux différences dans la désignation de concepts. Le cas le plus fréquent est la présence de **synonymes**, d'**homonymes**, d'**hyperonymes** et d'**hyponymes** (Mena, Illarramendi et al., 1996). Les **synonymes** sont deux mots distincts ayant le même sens. C'est l'exemple de "publication" et "article" qui capturent la même information sur les articles de recherche publiés. Les **homonymes** sont des mots partageant la même graphie et la même prononciation mais n'ayant pas le même sens. Par exemple, "mémoire" peut faire référence à des entités différentes: "mémoire informatique" et "mémoire de thèse". Les **hyperonymes** et les **hyponymes** indiquent des niveaux d'une hiérarchie désignés par le concept plus "général" ou le concept moins "général". C'est le cas de "personne" qui est un hyperonyme de "employé" car c'est un terme plus "général".

- Les conflits d'échelle ont lieu quand des systèmes de référence différents sont utilisés pour mesurer une grandeur. C'est l'exemple de "Fahrenheit" ou "Celsius" pour la température.

- Les conflits d'indéterminisme surgissent quand les concepts semblent avoir le même sens alors qu'ils sont différents. Ceci peut être dû à des contextes temporels différents.

- Les conflits de représentation surviennent quand les schémas de deux sources décrivent différemment un même concept. Par exemple, le nom d'un étudiant peut être représenté par deux champs "prénom" et "nom" dans une source et par un seul champ "identité" dans une autre source. Nous pouvons également citer l'exemple d'un concept défini comme une classe dans une source de données et comme attribut dans une autre, etc.

Masquer l'hétérogénéité sémantique revient à faire communiquer les systèmes d'information via une connaissance commune qui permet d'expliciter et de préciser le sens des données pour être interprétées correctement par différents systèmes. Cette connaissance peut être capturée grâce à des ontologies formelles.

2.2 Ontologie

La définition la plus commune présente une ontologie comme étant "une spécification explicite et formelle d'une conceptualisation partagée" [1] (Gruber, 1993). En d'autres termes, une ontologie est une description formelle d'un domaine de discours. C'est une conceptualisation dans le sens où elle fournit un vocabulaire

formalisé de concepts, de leurs relations, et des hypothèses d'un domaine. Elle peut ainsi être interprétée aussi bien par les humains que par les applications. La conceptualisation est partagée car une ontologie fournit une base de compréhension commune d'un ou de plusieurs domaines, et peut être utilisée par plusieurs communautés. Une ontologie peut également contenir des instances ou des individus. Elle constitue aussi une base d'inférence qui permet de déduire de nouveaux faits en appliquant des règles sur des faits existants.

L'ontologie est utilisée dans un système d'intégration pour décrire formellement la sémantique des concepts utilisés dans les sources d'information. Il existe trois approches pour l'intégration d'information fondée sur l'ontologie (Wache, Vössel et al., 2001) :

- les systèmes à ontologie unique qui fournit un vocabulaire partagé par toutes les sources,
- les systèmes à ontologies multiples, où chaque source d'information est décrite par sa propre ontologie,
- les systèmes hybrides tirent avantage des deux approches précédentes; chaque source est décrite par sa propre ontologie, et toutes les ontologies des sources (ontologies locales) sont reliées à une ontologie globale permettant de partager un vocabulaire commun.

Les ontologies connaissent un regain d'intérêt depuis l'avènement du web sémantique (Berners-Lee, Hendler et al. 2001). Cet intérêt s'est traduit par une prolifération d'ontologies de domaines, construites indépendamment les unes des autres par différents groupes, créant ainsi des ontologies hétérogènes. L'interopérabilité entre systèmes s'appuyant sur différentes ontologies est possible grâce aux systèmes d'intégration de l'information.

2.3 Intégration de l'information

Les systèmes d'intégration de l'information fournissent une vue unifiée de multiples systèmes hétérogènes, autonomes et répartis, facilitant ainsi l'accès à l'information. Ceci est réalisé par l'utilisation d'un schéma global ou d'une ontologie globale, qui fournit une vue réconciliée (consensuelle) des sources locales. Il existe deux approches pour l'intégration : l'intégration physique et l'intégration virtuelle. La première consiste à créer un entrepôt de données à partir des sources locales, dupliquant ainsi les données. L'intégration physique a l'avantage de fournir des temps d'accès rapides mais nécessite un support de stockage volumineux et fiable et des outils spécifiques, appelés ETL (*Extract, Transform and Load*), pour le traitement préalable des données. La solution de l'intégration virtuelle, que nous avons adoptée, permet d'avoir une vue *fraîche* des données, sans avoir à les dupliquer ni à les transformer. L'intégration virtuelle de sources hétérogènes et autonomes est fondée sur la médiation (Wiederhold, 1992). Cette dernière repose sur deux niveaux : le médiateur et les adaptateurs. Le médiateur a pour fonction d'offrir une vue unifiée des différentes sources de données grâce au schéma global, cachant en cela leur hétérogénéité et leur répartition. Il offre un protocole d'accès et un langage de requêtes commun à toutes ces sources. Quant à l'adaptateur, il adapte la requête exprimée dans le langage commun au langage de la source, tout en utilisant le bon protocole d'accès. Etant donné que la requête utilisateur est exprimée en fonction du schéma global, alors qu'elle doit être exécutée par les sources locales, un mapping ou correspondance entre ce schéma global et les schémas locaux (des sources) est nécessaire. Ce mapping constitue un traitement clé dans le processus général. Il sera utilisé pour réécrire la requête initialement exprimée en fonction du schéma global, en des sous-requêtes exprimées, chacune, en fonction du schéma local de la source qui l'exécutera.

Deux approches existent pour définir le mapping entre le schéma global et les schémas des sources: *Local As View* (LAV) et *Global As View* (GAV) (Halevy, 2001).

Dans l'approche GAV, le schéma global est exprimé à l'aide de vues sur les schémas locaux, à l'inverse de l'approche LAV qui nécessite la description des sources locales en fonction du schéma global.

Les approches LAV et GAV ont chacune des avantages et des inconvénients. Ainsi, la LAV favorise l'extensibilité du système d'intégration puisque l'ajout ou la suppression des sources est simple, chaque source étant décrite indépendamment des autres. Mais, la réécriture dans ce cas est un problème complexe. Quant à l'approche GAV, elle favorise, la performance du système quand l'utilisateur pose fréquemment des requêtes complexes puisque les algorithmes de réécriture de requêtes sont plus simples. Cependant, l'ajout ou la suppression d'une source de données nécessite la mise à jour du schéma global pour l'adapter au nouvel état du système.

En plus de la LAV et de la GAV, il faut mentionner l'approche GLAV qui est une combinaison des deux approches (Lenzerini, 2002).

3 TRAVAUX CONNEXES

De nombreux travaux de recherche ont été menés sur les ontologies, dans le domaine de l'ingénierie des connaissances d'une part, et sur l'intégration de l'information, dans le domaine des bases de données d'autre part. Dans ce dernier contexte, ce sont essentiellement les problèmes liés à l'hétérogénéité syntaxique qui ont été résolus. C'est seulement depuis quelques années que l'hétérogénéité sémantique est traitée dans les systèmes d'intégration par l'utilisation d'ontologies. Ceci a donné lieu à divers systèmes tels que TSIMMIS (Garcia-Molina, Papakonstantino et al. 1997), SIMS (Arens, Knoblock et al., 1996), MOMIS (Beneventano et Bergamashi, 2004), KRAFT(Visser, Beer et al., 1999), XYLEME (Delobel, Reynaud et al., 2003), PICSEL (Reynaud et Giraldo, 2003), OBSERVER (Mena, Illarramendi et al., 1996) et PIAZZA (Halevy, Ives et al., 2003). Pour montrer la diversité des systèmes d'intégration, nous en présentons quelques uns en mettant l'accent sur le modèle de données utilisé et le formalisme sous-jacent ainsi que sur l'architecture d'ontologie adoptée et le type de mapping (GAV, LAV).

SIMS est un système qui vise l'intégration de sources de données hétérogènes et de bases de connaissances qu'il représente en utilisant le langage LOOM, basé sur la logique de description. SIMS adopte l'architecture à ontologie unique et utilise le mapping GAV. Le médiateur dans SIMS est spécialisé dans un seul domaine d'application.

MOMIS est un système d'intégration qui se base sur son propre langage orienté objet dénommé ODL_I^3. Il utilise une ontologie unique globale appelée GVV (Global Virtual View) qui est générée semi-automatiquement. MOMIS adopte l'approche GAV pour le mapping entre l'ontologie globale et les sources locales.

OBSERVER est un système qui permet l'interopérabilité entre différentes sources, en utilisant pour cela de multiples ontologies pour décrire les sources de données. Il se base sur CLASSIC, un langage de logique de description. Il n'y a pas d'ontologie globale dans OBSERVER ; le mapping entre les multiples ontologies est réalisé à l'aide de tables de correspondance. Cependant, les relations entre ontologies sont limitées à des relations lexicales basiques telles que les synonymes, hyponymes et hyperonymes.

PICSEL est un système qui a connu plusieurs versions. Il vise à construire un médiateur à base de connaissances. Le langage utilisé est ALN-CARIN, un formalisme à base de règles et de logique de description. Dans sa dernière version, PICSEL adopte l'approche hybride comme architecture d'ontologie et utilise la LAV pour le mapping entre l'ontologie globale et les ontologies locales. Ce mapping est généré semi automatiquement.

Quant à XYLEME, c'est un système d'intégration physique (approche entrepôt) avec XML comme modèle de données. Il adopte l'approche hybride dans son architecture. Les ontologies globale et locales sont exprimées à l'aide d'arbres, avec un mapping GAV/LAV. Le mapping est réalisé semi automatiquement par la génération de tables de correspondance entre les chemins de l'ontologie globale et les chemins des ontologies locales.

Enfin, PIAZZA est un système qui intègre des sources qui sont, soit des ontologies, soit des données semi structurées (décrites par schéma XML ou DTD). Il s'appuie sur le modèle de données XML et une adaptation de XQuery comme langage de requêtes. Ce système suit une architecture peer-to-peer avec des ontologies et/ou schémas multiples. Le mapping entre les différentes ontologies est bidirectionnel (une source est décrite en fonction d'une autre et vice versa). Pour une requête donnée, le schéma d'un nœud (source de données) peut être considéré comme étant le schéma global et ainsi, de proche en proche, les sources concernées seront interrogées.

La plupart de ces systèmes utilisent une architecture à ontologie unique. Cette approche est intéressante dans le cas où les sources auraient une vue similaire du domaine. Si ce n'est pas le cas, il est difficile de trouver une ontologie consensuelle minimale. D'un autre côté, une source ne peut être changée sans reconsidérer et l'ontologie globale et les autres sources pour s'assurer qu'il n'y a pas de conflits. Ceci ne privilégie pas l'autonomie des sources.

Pour les autres systèmes qui sont basés sur des ontologies multiples, la comparaison d'ontologies est difficile en l'absence d'un vocabulaire commun. De plus, ils ont besoin de définir le mapping entre plusieurs couples d'ontologies.

XYLEME et PICSEL sont les seuls à utiliser l'approche hybride. Mais le premier adopte l'approche entrepôt qui ne favorise pas la fraîcheur des données. Quant au second, il utilise son propre formalisme de langage de description.

Nous proposons un système d'intégration virtuelle qui se base sur une approche d'ontologie hybride pour garantir l'interopérabilité. Ceci permet de préserver l'autonomie des sources tout en bénéficiant d'un

vocabulaire commun. Contrairement à PICSEL, nous avons privilégié les standards, à savoir XML comme modèle de données avec son langage de requêtes XQuery, et OWL (Smith, Welty et al., 2003) comme langage de description des ontologies. Nous avons choisi de traiter les requêtes à l'aide d'une algèbre XML qui permet de modéliser formellement une requête, portant sur des données semi structurées, en vue de faciliter son traitement. Cette algèbre fournit également des règles d'équivalence pour l'optimisation des requêtes. Nous avons choisi l'approche GAV pour décrire le mapping entre l'ontologie globale et les ontologies locales et ce, pour des raisons de performance du processus de réécriture.

4 NOTRE SYSTÈME DE MÉDIATION

Notre système de médiation, WASSIT (frameWork d'intégrAtion de reSSources par la médIaTion), permet d'intégrer aussi bien les sources relationnelles que XML, OWL ou objet. Notre système est basé sur le modèle de données XML car ce dernier permet de représenter aussi bien les données structurées, que les données peu ou pas structurées. Chaque source locale est décrite au niveau d'un catalogue à l'aide d'une ontologie locale qui permet d'avoir une vue sur la source. Dans le cas où la source serait décrite à l'aide de schéma XML, l'ontologie sera générée semi automatiquement au format OWL. Pour représenter un univers de discours commun et consensuel des systèmes à intégrer, nous fournissons une ontologie globale de domaine qui capture les connaissances consensuelles et inclut un vocabulaire de concepts avec une spécification précise et formelle de leur signification. L'ontologie globale est également représentée en OWL. L'utilisation d'ontologie nous permet de profiter des capacités d'inférences qu'elle offre, moyennant l'utilisation d'un outil de raisonnement. Ces capacités seront exploitées, notamment, lors de la construction de l'ontologie globale ou encore lors de l'optimisation de requêtes. En effet, la combinaison des règles de l'ontologie avec les informations sur les capacités des sources permettra de construire des requêtes optimisées équivalentes. Quant au mapping entre l'ontologie globale et les ontologies locales, il est basé sur l'approche GAV. L'ontologie globale adresse l'hétérogénéité sémantique au niveau des schémas. Pour cela, nous utilisons les opérateurs fournis par le langage OWL, notamment pour faire correspondre les concepts de l'ontologie globale avec ceux des ontologies locales.

Afin de résoudre l'hétérogénéité sémantique au niveau des données, nous avons opté pour l'utilisation d'un thésaurus, dérivé de la base de données lexicale WordNet (Fellbaum, 1998). Cette dernière est une ontologie pour la langue anglaise. Aussi faut-il l'adapter au domaine d'application et utiliser un système adéquat pour le traitement des autres langues. Ceci permet de retrouver les synonymes, les hyperonymes et les hyponymes.

XQuery est le langage d'interrogation que nous avons adopté dans WASSIT. C'est le standard émergent du W3C pour l'interrogation des documents XML.

La requête exprimée en XQuery est traitée par le médiateur en utilisant une représentation intermédiaire basée sur une algèbre XML qui sera décrite en 4.2.

Un langage d'ontologie doit fournir, en plus d'une sémantique formelle et bien définie, un support pour le raisonnement, et des moyens pour le mapping entre ontologies. C'est le cas du langage OWL que nous décrivons brièvement en 4.3.

4.1 Architecture de WASSIT

Notre système de médiation, WASSIT comporte quatre niveaux (Benhlima, Chiadmi et Zellou, 2003): le niveau interface, le niveau médiateur, le niveau adaptateur et le niveau des sources locales (figure 2.). Dans la suite, nous décrirons succinctement le médiateur et l'adaptateur ainsi que le traitement général d'une requête.

4.1.1 Médiateur

Au niveau médiateur, la requête passe d'abord par la phase d'**analyse et de génération de l'arbre** et de **réécriture**, puis par les phases d'optimisation, d'exécution et d'intégration et enfin la présentation des résultats. Outre les modules de traitement de la requête, le médiateur dispose d'un catalogue et d'un cache (cf. figure 1).

La requête générée à travers l'interface QBE est exprimée en fonction des concepts de l'ontologie globale. Elle est ensuite analysée et représentée en algèbre XML sous forme d'arbre. Le module de réécriture utilise le mapping pour connaître les sources locales. Puis, chaque concept global est remplacé par son correspondant local. La sélection sur le contenu est enrichie en utilisant le thésaurus.

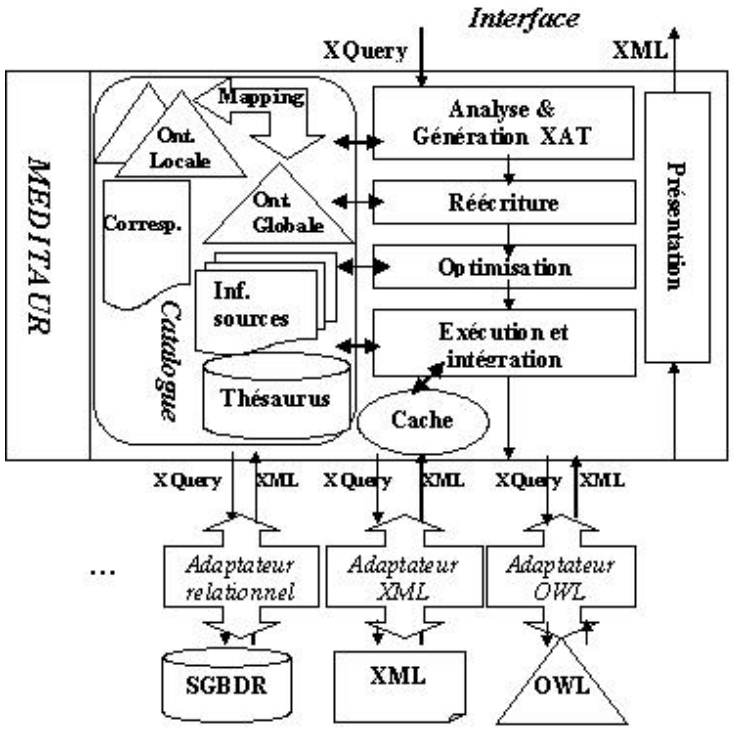

Figure 1. Architecture de WASSIT

Les sous requêtes sont ensuite optimisées avant d'être exécutées puis intégrées selon le plan d'exécution généré à partir du mapping. Le résultat est enfin présenté en utilisant le modèle XSL, généré à partir des choix effectués par l'utilisateur au niveau de l'interface.

4.1.2 Adaptateur

L'hétérogénéité syntaxique est résolue au niveau des adaptateurs. En effet, ces derniers adaptent la sous requête exprimée en XQuery au langage de requête de la source et envoient la requête à la source en utilisant le protocole adéquat. Les résultats renvoyés par la source sont alors transformés en XML avant d'être acheminés vers le médiateur.

4.2 Langage de description d'ontologie

Le langage que nous avons choisi pour représenter la connaissance dans WASSIT est le langage OWL. Ce dernier est composé de trois sous langages d'expressivité croissante: OWL Lite, OWL DL et OWL Full. Dans notre système de médiation, nous avons utilisé OWL DL et OWL Lite. Ce dernier est suffisant pour représenter le thésaurus puisqu'il permet d'exprimer des hiérarchies et des contraintes simples. Quant à OWL DL, il comporte toutes les constructions du langage mais avec des restrictions sur la hiérarchie. Il possède en plus la puissance d'expressivité des logiques de description (Description Logic). Nous avons choisi OWL DL pour représenter les ontologies parce qu'il est suffisamment expressif pour décrire le mapping entre ontologies et qu'il assure la complétude (toutes les inférences sont calculables) et les mécanismes de raisonnement décidable (tous les calculs se terminent en un temps fini).

OWL est basé sur RDF (*Resource Description Framework*) et RDFS (*Resource Description Framework Schema*) (Brickley et Guha, 2003). RDF est un langage de description des ressources et de leurs relations alors que RDFS fournit un vocabulaire permettant de décrire les classes des ressources RDF avec des hiérarchies de type généralisation des classes et de leurs propriétés.

OWL est le successeur de DAML+OIL (Connolly, Van Harmelen et al., 2001). C'est un langage déclaratif, défini formellement. Il permet de représenter des concepts (owl:class), les relations entre concepts (owl:ObjectProperty), les types de données (owl: DatatypeProperty) et aussi les cardinalités. De plus, OWL permet de caractériser les relations comme la transitivité (owl:TransitiveProperty) et la symétrie

(owl:SymetricProperty). Il supporte les hiérarchies de spécialisation/généralisation (rdfs:subClassOf). Les concepts équivalents peuvent être reliés (owl:equivalentClass ou owl:sameClass). L'union et l'intersection de classes est possible (owl:unionOf, owl:intersectionOf).

Une ontologie OWL peut être représentée sous différents formats: N-Triple, syntaxe abstraite ou sérialisation RDFS/XML.

4.3 Choix de l'Algèbre XML

Nous avons choisi d'utiliser une algèbre XML pour pouvoir modéliser formellement les requêtes portant sur des données semi structurées. Il existe plusieurs travaux sur l'algèbre XML (Beech, Malhotra et al., 1999), (Fernandez, Simenon et al. 2001). L'algèbre XAT (Wadjinny et Chiadmi, 2005), que nous avons adoptée pour le traitement des requêtes dans WASSIT, a été développée par Niagara (Galanis et al., 2201) et adaptée par XEAR (Zhang et Rundensteiner, 2002).

L'algèbre XAT est suffisamment puissante et complète pour tenir compte de toutes les spécificités des données semi-structurées, notamment la navigation et le formatage. Elle dispose de plusieurs opérateurs. Certains sont spécifiques à XML comme format et follow, d'autres sont issus de l'algèbre relationnelle comme select, produit cartésien, union, etc.

Certains opérateurs sont unaires (follow, source, etc.), d'autres sont binaires (union, join, etc.).

Ainsi, à chaque clause de la requête XQuery est associé un opérateur algébrique. La liste suivante énumère quelques opérateurs de l'algèbre :

Source : opérateur qui exprime la référence à un document XML. Cet opérateur est unaire et a en paramètre le nom du document.

Follow : permet de se déplacer dans l'arborescence des documents XML. Il a en paramètre l'expression PATH.

Format : permet de formater les données entre deux balises. Il a en paramètres le nom de la balise et la variable à formater.

L'algèbre XAT dispose également de règles d'équivalence pour l'optimisation des requêtes.

5 ENRICHISSEMENT SÉMANTIQUE

Avant de décrire plus en détail les composants qui contribuent à l'enrichissement sémantique, nous présentons un scénario d'utilisation simplifié pour illustrer l'hétérogénéité entre deux sources d'informations.

5.1 Scénario d'utilisation

Nous considérons ici un exemple d'application qui permet d'intégrer deux bibliothèques virtuelles. En plus des références sur les documents, chaque bibliothèque stocke les ouvrages sous format électronique. Un fragment des SI de chacune des deux bibliothèques est donné en figure 2.

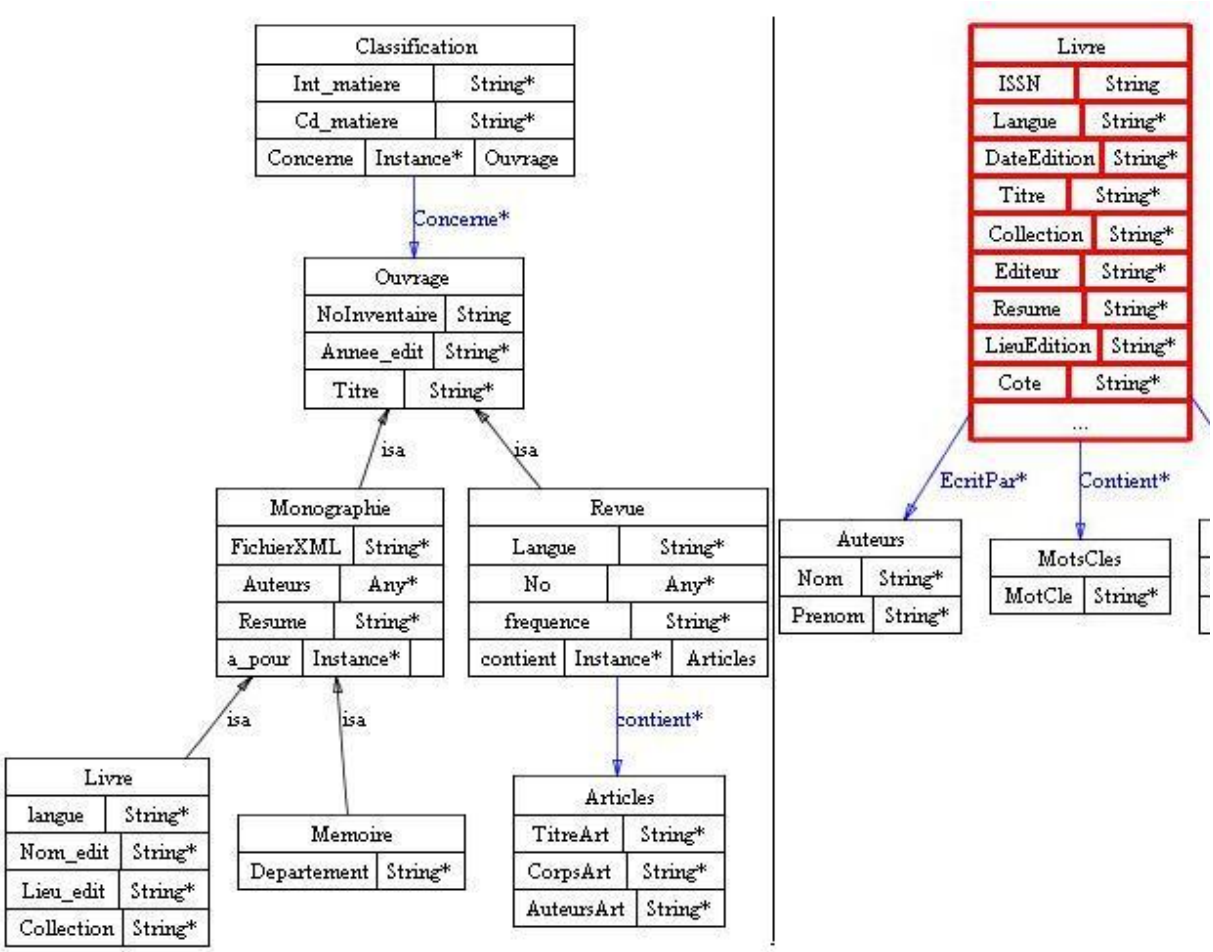

Figure 2. Fragment du Système d'information des deux bibliothèques

Les hétérogénéités entre les deux systèmes sont nombreuses ; nous citons à titre d'exemple les cas suivants :

- Dans le premier système, la classe Livre est une spécialisation de la classe Monographie, elle-même spécialisation de la classe Ouvrage. Dans le second système, la classe Livre regroupe tous les types d'ouvrages.
- L'auteur est un attribut dans le premier système alors que c'est une classe dans le second.
- Le premier système ne donne que l'année d'édition alors que le second donne la date d'édition complète.
- Pour le premier système, le contenu du document électronique est pointé par le nom de fichier, représenté par l'attribut FichierXML, d'une part et par la classe Articles, d'autre part. Quant au contenu du second système, il est représenté par la classe Section.

Dans la suite, nous considérons le cas d'une requête qui demande l'année d'édition, ainsi que le contenu des documents relatifs aux "bases de données réparties".

5.2 Composants pour l'enrichissement sémantique

Dans cette section, nous décrivons les composants utilisés dans WASSIT pour enrichir sémantiquement la requête, à savoir le catalogue et les modules d'analyse et de génération de l'arbre ainsi que la réécriture.

5.2.1 Catalogue

Utilisé par tous les modules du médiateur, il contient toutes les données nécessaires pour le traitement d'une

requête, à savoir les ontologies, les informations sur les sources, le mapping entre l'ontologie globale et les ontologies locales, la correspondance entre les concepts et enfin le thésaurus. Nous les commentons dans ce qui suit.

- **Ontologies** : Toutes les ontologies, aussi bien globale que locales sont décrites à l'aide du langage OWL DL et sont sérialisées en RDFS/XML.

Pour notre exemple, l'ontologie globale, construite semi automatiquement par le gestionnaire du médiateur, est représentée (cf. figure 3) en syntaxe abstraite.

```
Ontology (OntoBIB
        Class (Documents partial)
        Class (Revue partial Documents)
        Class (Auteurs complete)
        …
        ObjectProperty (Possede  domain(Documents) range (Section))
        ObjectProperty (EcritPar  domain(Monographie) range (Auteurs))
        DatatypeProperty (Titre domain (Documents)  range (xsd:string))
        DatatypeProperty (DateEdition domain (Documents)  range (xsd:date))
        DatatypeProperty (NomAuteur domain (Auteurs)  range (xsd:string))
        DatatypeProperty (IntituleSection domain (Section)  range (xsd:string))
        … )
```

Figure 3. Ontologie globale en syntaxe abstraite

- **Informations sur les sources** : C'est un document XML qui contient les données nécessaires à la localisation et à l'accès aux sources. Ainsi, pour chaque source, on peut connaître son URI, son type (XML, relationnel, texte, etc.), le protocole d'accès, le méta modèle de données, etc.

Certaines informations vont servir à choisir l'adaptateur adéquat.

- **Mapping** : décrit la liaison (GAV) entre les vues de l'ontologie globale et les vues des ontologies locales. Ainsi une vue globale peut être l'union de deux ou plusieurs vues locales. Elle peut être une intersection entre plusieurs vues locales, avec sélection de certains éléments ou restriction sur certains éléments. Le mapping est une ontologie sérialisée en RDFS/XML.

Pour notre exemple, le mapping est l'union de deux sources. Ceci est exprimé dans la figure 4.

```
…<owl:imports  rdf:resource=http://www.emi.ac.ma/biblio/>
<owl:imports  rdf:resource=http://www.ensias.ma/bib/>
<owl :Class rdf :ID= "Documents">
        <owl:unionOf  rdf:parseType = "collection"
    <owl:Class  rdf:about ="&biblio;ouvrage">
    <owl:Class  rdf:about ="&bib;livre">
</owl:Class>  …
```

Figure 4. Mapping entre la vue globale "Documents" et les classes locales "Livre" et "Ouvrage"

- **Correspondance** : Permet de faire correspondre les synonymes et les hyperonymes entre les concepts de l'ontologie globale et ceux des ontologies locales. C'est à ce niveau que sont également résolus les conflits de représentation et les conflits d'échelle avec des fonctions de conversion.

Pour notre exemple, il faudra faire correspondre, entre autres, "CodeMatiere" de l'ontologie globale avec "cd_matiere" de l'ontologie Biblio et "Cote" de l'ontologie Bib. Il en est de même pour "NomAuteurs" qui sera mis en relation avec "Auteurs" de l'ontologie Biblio et "Nom" de l'ontologie Bib (cf. figure 5). Quant au conflit de représentation entre "Anne_edit" et "DateEdition", il sera résolu par une fonction de conversion "ExtraireAnne(date)".

```
...
<owl:DatatypeProperty rdf :about= "#NomAuteur"
        <owl:equivalentProperty rdf:resource="&Biblio;Auteurs"/>
    <owl:equivalentProperty rdf:resource="&bib;Nom"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf :about= "#CodeMatiere"
        <owl:equivalentProperty rdf:resource="&Biblio;cd_matiere"/>
    <owl:equivalentProperty rdf:resource="&bib;Cote"/>
</owl:DatatypeProperty> ...
```

Figure 5. Correspondance entre concepts de l'ontologie globale et des ontologies locales

- Thesaurus : En vue de traiter l'hétérogénéité sémantique au niveau des données, nous utilisons un thésaurus qui donne une hiérarchie de concepts avec leurs synonymes et hyperonymes pour le domaine d'application. Ce thésaurus est décrit à l'aide de OWL Lite et sérialisé en RDFS/XML. De cette façon, des réponses maximales seront fournies à l'utilisateur puisque la requête sera enrichie sémantiquement en ajoutant les synonymes et/ou les hyperonymes (ou hyponymes) des termes qui figurent dans des restrictions (condition *where)* sur certains concepts.

Ainsi, pour le terme "base de données réparties", il a pour synonyme "base de données distribuées" et pour hyperonyme "base de données".

5.2.2 Analyse et Génération de l'arbre

La requête reçue par le médiateur sera d'abord analysée en vue de détecter d'éventuelles erreurs syntaxiques ou références à un élément qui n'existe pas. Elle sera ensuite représentée à l'aide d'un arbre en se basant sur l'algèbre XML introduite précédemment. Chaque nœud de l'arbre correspond à un opérateur algébrique.

5.2.3 Réécriture

Il s'agit de décomposer la requête en sous requêtes à la fois mono source et sémantiquement enrichies. Rappelons que nous avons adopté l'approche GAV pour la définition de l'ontologie globale. Ainsi, les vues globales sont décrites en fonction des vues sur les sources locales. De ce fait, un processus de réécriture est nécessaire pour exprimer la requête générée au départ en fonction des vues globales en requêtes destinées aux sources locales. Ce module va donc consulter le mapping du catalogue afin de remplacer la vue globale par ses vues locales.

Le module de réécriture prend en entrée la requête représentée sous forme d'un arbre. En sortie de ce module, un sous arbre constitué d'opérateurs algébriques est construit pour chaque sous requête. Les nœuds du sous arbre ne concernent que les concepts relatifs à la source. En effet, dans le cas où un nœud ferait intervenir des concepts existant dans plusieurs sources (cas d'un nœud select avec plusieurs conditions), le nœud sera éclaté en plusieurs nœuds pour ne garder, dans chaque noeud, que les concepts concernant la source. Dans le cas où le paramètre d'un nœud n'aurait pas de concept correspondant dans une source, le nœud sera écarté pour la source concernée.

6 IMPLÉMENTATION

Les ontologies ont été créées à l'aide de l'éditeur open source Protégé editor (Horridge, M., 2004). Les modules du médiateur ont été implémentés en JAVA et c'est l'API Jena (Caroll, J. et al. 2003) qui nous a permis d'accéder aux ontologies.

La requête générée à travers l'interface a la forme donnée en figure 6. Elle est exprimée en fonction des concepts de l'ontologie globale.

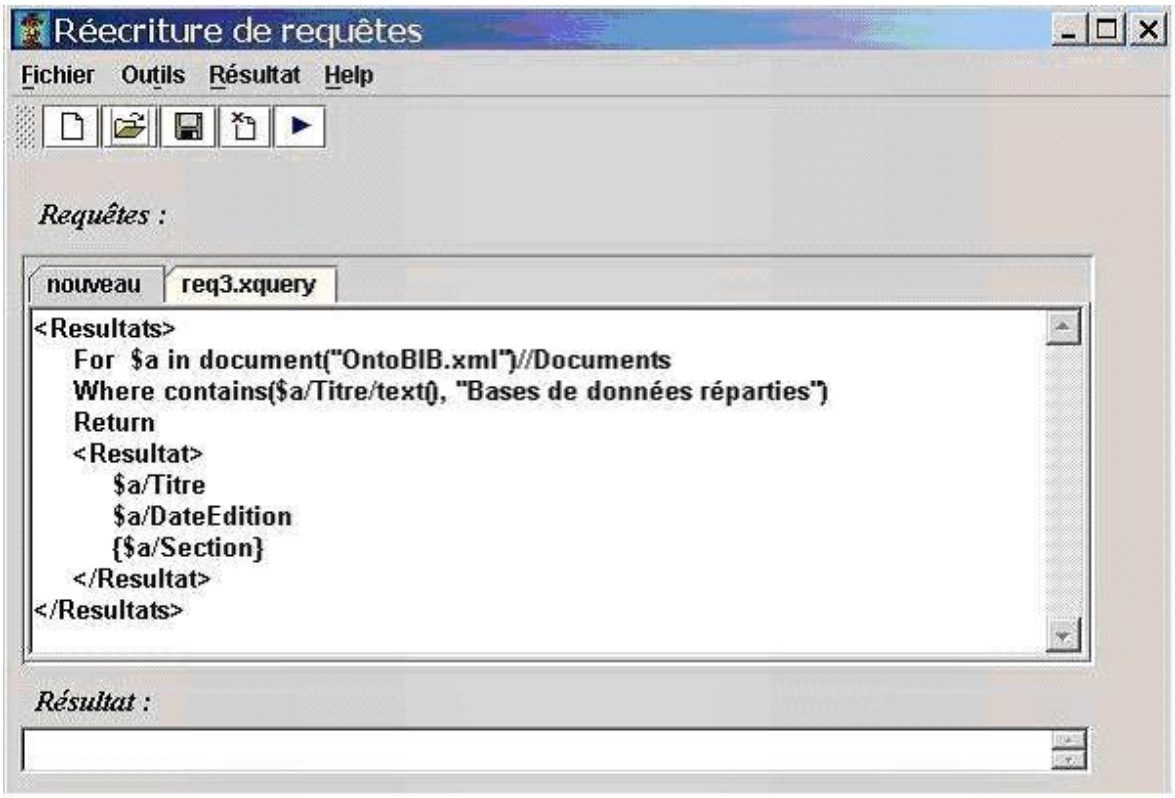

Figure 6. Requête Année édition et contenu des documents traitant les "Bases de données réparties"

Analyse et génération XAT

Le premier traitement effectué sur la requête est l'analyse syntaxique de la requête, suivie de la génération de l'arbre selon l'algèbre XAT. Chaque nœud de l'arbre est associé à un opérateur et est identifié par un nom. Dans la suite, nous en décrivons quelques-uns.

follow : Le rôle de l'opérateur *follow* est de représenter les expressions de chemin de XPath. La structure de cet opérateur est constituée d'un point d'entrée et d'une destination. Le point d'entrée est la position initiale dans l'expression de chemin. La destination contient les déplacements dans l'expression de chemin et correspond à un concept. L'accès à l'ontologie globale permet de récupérer les concepts disponibles pour cette vue. L'information des opérateurs *follow*, en particulier point d'entreé/destination, confronté avec les concepts de la vue globale permet de savoir si l'opérateur *follow* est valide. Si ce n'est pas le cas, la requête est rejetée.

Source : L'opérateur *source* contient les informations relatives à la source de données de la requête à savoir le nom de la source et son type (XML, relationnel, etc.). Le nom de cet opérateur est utilisé par un opérateur *follow* comme point d'entrée. Le nom de la source permet de vérifier s'il s'agit bien d'une vue globale. Ceci est réalisé en accédant à l'ontologie globale. Si ce n'est pas le cas, la requête est rejetée.

Select : Cet opérateur a comme paramètre une expression booléenne qui peut inclure les opérateurs AND et OR. L'opérateur Select filtre les éléments selon cette expression.

L'arbre en sortie de ce module est présenté en figure 7.

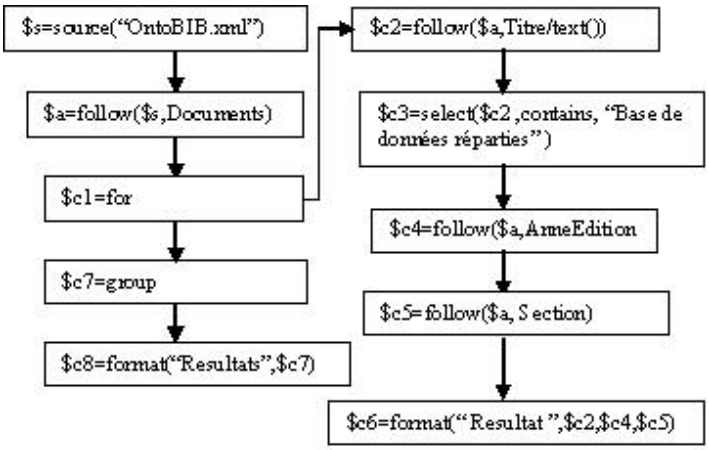

Figure 7. Arbre généré après analyse

Réécriture de requêtes

Le module de réécriture prend en entrée la requête représentée sous forme d'un arbre. Deux phases sont nécessaires à cette étape : prétraitement et génération des sous arbres.

L'arbre est parcouru en une première passe afin de déterminer, pour chaque type d'opérateur (*format, source, follow, select*, etc.), les informations qui figurent dans cet opérateur. Ceci permet de déterminer le nombre d'arbres et par suite, de requêtes en sortie. Une fois ce nombre connu, on détermine pour chaque source, les vues locales qui la concernent dans la requête ainsi que les chemins et par suite les concepts de cette source.

Dans la seconde phase, il s'agit de générer les arbres avec les nœuds comportant juste la portion d'information nécessaire.

Dans la suite, nous décrivons le traitement réalisé pour quelques opérateurs.

Source : Pour cet opérateur, il s'agit d'extraire le nom d'une source de l'ontologie globale. La consultation du mapping au niveau du catalogue permet de connaître les vues locales en liaison avec la vue globale. Le document XML Informations est également consulté afin d'extraire les sources (URI) des vues en question. Enfin, le nœud *source* généré a en paramètre le nom d'une source locale.

Follow : L'examen de l'opérateur *follow* permet de connaître les concepts de l'ontologie globale concernés pour chaque source. Ces derniers sont utilisés pour trouver les concepts équivalents dans chaque source ou ontologie locale concernée. Ceci est réalisé grâce au document Correspondance qui indique la correspondance entre les concepts de la vue globale et ceux des sources locales. Ainsi le nœud *follow* généré aura en paramètre un concept de l'ontologie locale.

Select : Pour cet opérateur, il s'agit d'extraire de l'expression paramètre les concepts qu'elle conditionne ainsi que les termes (constantes chaînes de caractères) utilisés dans la condition. L'accès au thésaurus permet de déterminer les synonymes, hyperonymes et hyponymes des termes extraits. Ces éléments serviront à enrichir l'expression paramètre du nœud *select* généré.

Pour notre exemple, cela aura pour effet de supprimer certains nœuds et d'en modifier d'autres :

- les nœuds *source* qui auront, chacun, en paramètre le nom de la source locale ;
- les nœuds *follow* avec en paramètre les chemins locaux ;
- les nœuds *select* avec, en paramètre, l'expression portant uniquement sur les noeuds *follow* qui interviennent dans la condition. Les termes du nœud *select* sont aussi enrichis par l'hyperonyme 'bases de données' et par le synonyme 'bases de données distribuées';
- les nœuds *format* avec, en paramètre, les noeuds *format* fils ou *follow* de ce nœud.

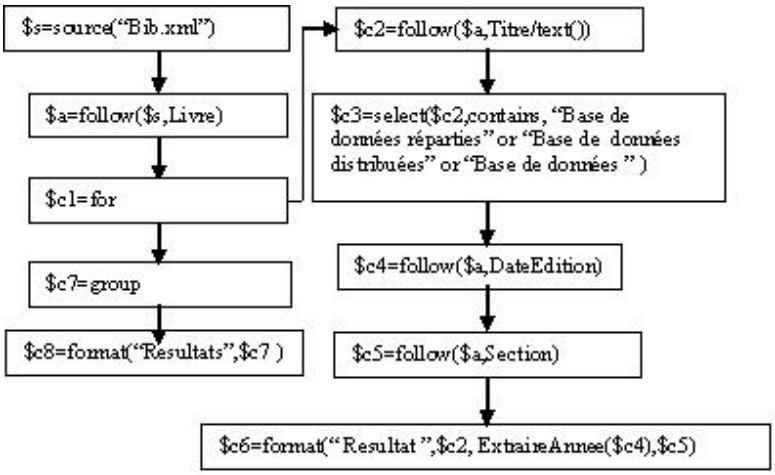

Figure 8. Arbre relatif à la sous requête de la source Bib

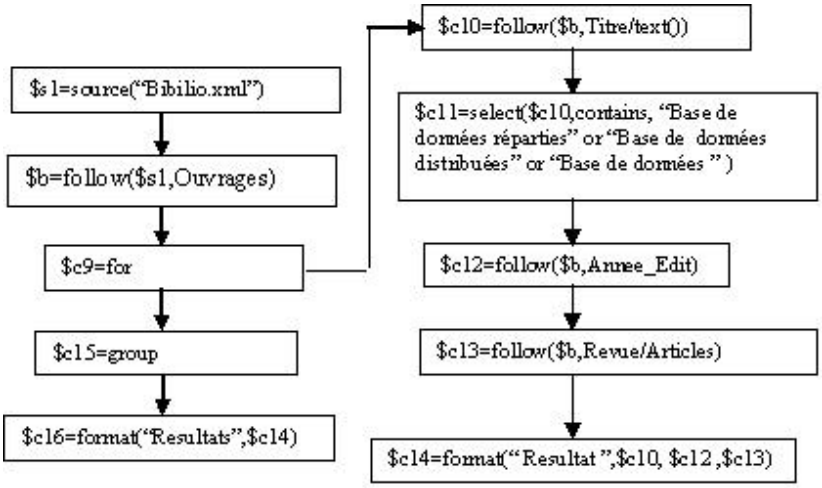

Figure 9. Arbre relatif à la sous requête de la source Biblio

7 CONCLUSION

Nous avons proposé un système de médiation pour l'intégration de sources hétérogènes et réparties. L'hétérogénéité sémantique (schéma et données) est traitée au niveau du médiateur de WASSIT. La résolution de conflits schématiques est réalisée grâce à la mise en correspondance entre les concepts de l'ontologie globale et ceux des ontologies locales représentant les sources hétérogènes. Quant à l'hétérogénéité au niveau du contenu, elle est traitée grâce à l'utilisation d'un thésaurus de domaine. Ainsi, la requête est enrichie par des synonymes, des hyponymes et des hyperonymes pour avoir des réponses maximales. Notre système est fondé sur le modèle de données XML et son langage de requête XQuery. Il utilise les ontologies OWL pour représenter le domaine de connaissances. L'utilisation de l'algèbre XAT nous permet de modéliser formellement les requêtes par le biais d'opérateurs algébriques.

Nous avons implémenté le catalogue, le module d'analyse et de génération ainsi que le module de réécriture. Outre l'implémentation des autres modules, nos recherches portent sur plusieurs axes :

En vue de faciliter la gestion du médiateur, nous travaillons à développer un outil pour la génération semi-automatique de l'ontologie globale et du mapping entre ontologies. Cet outil devrait également réaliser la transformation des schémas des sources (XML schema, DTD, relationnel, objet, etc.) vers une ontologie OWL.

La création semi-automatique d'un thésaurus de domaine à partir d'une ontologie générale telle que WordNet ou Cyc fait partie de nos travaux actuels.

Pour pouvoir traiter le multilinguisme au niveau des sources de données, nous travaillons à mettre en place un service web qui fait appel à un traducteur durant la phase d'enrichissement sémantique.

L'optimisation des requêtes est également un de nos axes de recherche. Elle porte sur différents niveaux : utilisation des règles d'équivalence pour réécrire les requêtes, génération de plans d'exécution optimisés, etc. L'intégration d'un outil de raisonnement, tel que Racer (Haarslev, V.,2003), dans le médiateur permettrait également d'optimiser les requêtes, étant données des règles d'inférence.

Enfin, un adaptateur de XQuery vers SQL a été réalisé. Par la suite, nous pensons automatiser la génération d'adaptateurs en utilisant le pattern factory.

Bibliographie

Arens, Y. Knoblock, C.A., Shen, W. (1996). Query reformulation for dynamic information integration. *Journal of intelligent information systems*. 99-130.

Beech D., Malhotra A. et al. (1999). A formal Data Model and Algebra for XML.

Beneventano, D. et Bergamashi, S. (2004). The Momis Methodology for integrating heterogeneous data sources. *In IFIP*.

Benhlima, L. Chiadmi, D. et Zellou, A. (2003). XML-based integration for e-learning. *Proceeding of SEBD'2003*, Cetraro, Italie 24-27 juin, 313-322.

Berners-Lee, T.,Hendler, J. et Lassila,O.(2001). The semantic web. *Scientific American*.Vol5,34-43.

Brickley, D., Guha, R. (2003). RDF vocabulary Description Language 1.0 RDF Schema. http://www.w3.org /TR/rdf-schema, W3C Working group

Caroll, J., Dickinson, I. et al. (2003). *Jena : Implementing the semantic web recommendations*, Technical Report HPL-2003-146, 24 Dec, http://jena.sourceforge.org

Chamberlin,D. et al. (2002). XQuery 1.0: An XML query language,. http://www.w3.org/TR/xquery

Connolly, D. F. Van Harmelen, Horrocks, I, Mc-Guniess, D.L, Stein, L.A (2001). DAML+OIL reference Description. http://www.w3.org/TR/daml+oil reference

Delobel, C., Reynaud, C., Rousset, M6C., Sirot, J6P., Vodislav, D. (2003). Semantic integration in Xyleme : a uniform tree-based approach. *Data & knowledge engineering* 44, 267-298.

Doan, A. et Halevy, A.Y. (2004). Semantic integration research in the database community : a brief survey. *American association for artificial Intelligence* (www.aaai.org).

Fellbaum, C.D. (1998). *WordNet: An Electronic Lexical Database*, MIT Press.

Fernandeze M., Simenon J. and Walder P. (2001). A semi-Monad for semi-structured Data. *In International Conference on Database Theory.*

Galanis L. et al. (2001). *Following the Paths of XML Data: An Algebraic Framework for XML Query Evaluation.* Technical Report.

Garcia-Molina H., Papakonstantino, Y., Quass D., Rajman A., Sagir Y., Ullman J., et al. (1997). The TSIMMIS approach to mediation : Data models and Languages. *Journal of Intelligent Information System*s.

Goh, C.H., Bressan, H., Madnick, S.E, Siege, M.D. (1999). Context Interchange: New features and formalisms for the intelligent Integration of Information. *ACM Trans. On Inform. Syst*. 17 270-293.

Gruber, T.R. (1993). *Toward principle for the design of ontologies used for knowledge sharing*. Technical report KSL 93-04 Standford University.

Haarslev, V., Moller, R.: Racer (2003). An OWL Reasoning Agent for the semantic web, *in the proceeding of International Workshop on Applications, Products and Services of Web-based Support Systems*, Halifax, Canada. 91-95.

Halevy, A.Y. (2001). Answering queries using views. *In VLDB journal,* Vol. 10 No. 4 270-294.

Halevy, A. Y., Ives, Z. G., Mork, P. & Tatarinov, I. (2003). Piazza: Data Management Infrastructure for Semantic Web Applications, *in Proceedings of the twelfth international conference on World Wide Web*, Budapest, Hungary, 556- 567.

Horridge, M., Knublach, H., Rector, A., Stevens, R., Wroe, C. (2004). A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE *Tools Edition 1.0*

Lenzerini, M. (2002). Data integration : a theoretical perspective, *in PODS* 233-246.

Mena, E. Illarramendi, A, Kashyap, V. Sheth, A.P. (1996). *Observer : An approach for query processing in global information systems based on interoperation across pre-existing ontologies*, Kluwer Academic Publisher.

Park, J., Ram, S. (2004). Information Systems Interoperability : what lies beneath ? *ACM Trans. Inform. Syst.,* Vol. 22, No4, 595-632

Reynaud, C., Giraldo, G. (2003). An application to the mediator approach to services over the web. *In Concurrent Engineerin*g.

Smith, M.K. Welty, C. McGuiness, D.L (2003). OWL Web Ontology Language Guide, http://www.w3.org /TR/2003/PR-owl-guide-20031215.

Visser, P.R.S., Beer, M., Bench-Capon, T., Diaz, B.M., Shave, M.J.R. (1999). Resolving ontological heterogeneity in the kraft project. In DEXA'99, Septembre. 668-677.

Wache, H., Vögle, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H. et al. (2001). Ontology-based integration of information – a survey of existing approaches*. In proceedings of the international workshop on ontologies and information sharing.* Aout, 108-117.

Wadjinny, F., Chiadmi, D. (2005) Taxonomie des algèbres XML, Wotic.

Wiederhold, G. (1992). Mediators in the Architecture of Future Information Systems. In *IEEE Computer Conference* 38-49.

Zhang X. and Rundensteiner E.A. (2002.XAT). *XML Algebra for the Rainbow System*. Technical report *WPI-CS-TR-02-24*.

Notes de bas de page

1 "a formal, explicit specification of a shared conceptualisation."

Pour citer cet article

Laïla BENHLIMA et Dalila CHIADMI. «Vers l'interopérabilité des systèmes d'information hétérogènes». e-TI - la revue électronique des technologies d'information, Numéro 3, 27 décembre 2006, http://www.revue-eti.netdocument.php?id=1166.