République algérienne démocratique et populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Ammar Telidji

Faculté des sciences

Département de l'informatique et mathématiques

Thème

Génération automatique des requêtes de médiation dans un contexte XML

Suivi par :

Chellama Laradj

Réalisé par :

Heraoua Wahiba

Ben Mebarek Kelthoum

2011-2012

Remerciements

C'est avec un grand plaisir qu'on réserve cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui nous ont aidés de prés ou de loin à la réalisation de ce travail.

On tient tout d'abord à exprimer nos sincères remerciements et respects à notre encadreur Mr **CHELLAMA Laradj**, maître assistant à l'université de Laghouat, pour son encouragement et les précieux conseils qu'il n'a cessé de nous prodiguer et qui nous ont permis d'améliorer certains points dans notre mémoire.

On souhaite également exprimer notre profonde gratitude à nos enseignants durant le cursus universitaire pour leurs conseils et leurs contributions.

On 'exprime aussi notre reconnaissance envers le personnel administratif du département informatique en particulier **Mr A. TAHARI**, pour les efforts qu'il a fourni afin d'assurer le bon déroulement et les bonnes conditions d'études.

Enfin, nos meilleurs et vifs remerciements s'adressent aux membres du jury pour l'honneur qu'ils nous font en acceptant d'examiner et évaluer ce modeste travail.

Dédicace

Je dédie ce modeste travail

A mes très chers parents,

Pour leur soutien permanent et inépuisable,

Que Dieu les protège.

A mes frères et soeurs.

A tous la famille Heraoua et la famille bachiri.

Et à tous ceux qui me sont chers.

A mon amie Keltoum qui m 'a donné la force de continuer…

A tous mes amis et mes collègues.

Wahiba

Dédicace

Je dédie ce modeste travail

Á ma très chère mère

Á mon père

Á mon marie Fouad

Á mes frères et ma sœur

A ma meilleure amie Wahiba

Á touts la famille Benmebarek et la famille toobi

Á toute la promotion sortante 5éme année informatique 2012

Keltoum

Résumé

Les systèmes multi-source sont de plus en plus développés. Ils sont définis comme l'intégration de plusieurs sources hétérogènes et distribuées. Parmi ces systèmes d'informations, nous distinguons les systèmes d'informations basés sur le web, les systèmes de bases de données fédérées, ou encore les systèmes de médiation. Notre étude s'inscrit principalement dans le contexte des systèmes de médiation.

Dans ce mémoire, nous avons étudié les différentes approches d'intégration des données, en particulier l'approche de médiation dans contexte XML, Il s'agit d'automatiser la génération de requêtes de médiation calculant une relation du schéma global à partir d'un ensemble de sources de données.

L'écriture manuelle des requêtes de médiation donne sans doute le résultat le plus pertinent au regard des besoins des utilisateurs. Cependant il est difficile de l'entreprendre en raison du grand nombre de sources de données qui peuvent être impliquées et du volume important de métadonnées les décrivant. La question principale est de savoir comment automatiser la génération de requêtes de médiation dans un contexte semi-structuré XML?

La phase finale de notre travail consiste en l'implémentation d'un prototype de génération automatique des requêtes.

Mots clés : XML, génération des requêtes, systèmes de médiation, métadonnées

Abstract

Multi-source systems are more developed. They are defined as the integration of multiple heterogeneous and distributed sources. Of these information systems, we distinguish between information systems web-based systems, federated databases, or mediation systems. Our study is mainly in the context of mediation systems.

In this paper, we have studied different approaches to data integration, in particular the approach of mediation in XML context, it is to automate the generation of mediation queries calculating a relationship from the global schema of a set of data sources.

The handwriting requests mediation gives probably the most relevant results to the needs of users. However, it is difficult to undertake due to the large number of data sources that may be involved and the large volume of metadata describing them. The main question is how to automate the generation of mediation queries in a semi-structured XML.

The final phase of our work is the implementation of a prototype of automatic generation of queries.

Key Words: *XML, request generation, mediation systems, metadata*

Table des matières

Liste des figures

Liste des Algorithmes

Introduction générale

1. Contexte

L'évolution constante en matière de réseaux et de bases de données ces trente dernières années poussent à la réalisation de systèmes d'intégration de données de grande envergure. Ces systèmes fournissent aux utilisateurs une vue uniforme sur un ensemble de sources de données.

La diversité des sources de données conduit à des modes de consultation qui peuvent être très différents. Ainsi, une base de données relationnelle sera interrogée par l'intermédiaire d'une requête SQL, une page Web sera consultée par une adresse Web (URL)… une telle variété de source implique diverses façon de les interroger.

De nos jours, les systèmes multi-source sont de plus en plus développés .Ils sont définis comme l'intégration de plusieurs sources hétérogènes et distribuées Parmi ces systèmes d'informations, nous distinguons les entrepôts de données, les systèmes d'informations basés sur le web, les systèmes de bases de données fédérées, ou encore les systèmes de médiation. Notre travail se focalise principalement sur les systèmes de médiation dans un contexte XML. Il s'agit d'automatiser la génération de requêtes de médiation calculant une relation du schéma global à partir d'un ensemble de sources de données.

2. Problématique

Plusieurs problèmes de conception émergent lors de l'utilisation des médiateurs. L'une des principales difficultés rencontrée dans un système de médiation est la définition du schéma global et la définition des mappings (requêtes de médiation) qui relient le schéma global aux sources de données.

L'écriture manuelle des requêtes de médiation donne sans doute le résultat le plus pertinent au regard des besoins des utilisateurs. Cependant il est difficile de l'entreprendre en

raison du grand nombre de sources de données qui peuvent être impliquées et du volume important de métadonnées les décrivant. La question principale est de savoir comment automatiser la génération de requêtes de médiation dans un contexte semi-structuré XML? L'objectif de notre travail est a face à la problématique de définition de requêtes dans un système de médiation, nous avons opté pour une approche de génération automatique qui permet de décharger l'utilisateur de l'exploration d'un volume important de métadonnées.

3. Organisation de mémoire

Notre travail à travers ce mémoire est structure en trois chapitres :

Le premier chapitre est consacré à la présentation des différentes approches d'intégration de données. Nous donnerons premièrement quelques concepts liés au domaine, ensuite les modèles de représentation des données DTDs,… et enfin quelques approches existantes.

Dans le second chapitre, nous décrivons le principe de génération des requêtes en rappelant les métadonnées utilisés ainsi que les algorithmes nécessaires pour générer automatiquement une requête dans un contexte XML

Dans le troisième chapitre, nous présentons l'outil de réalisation, ainsi que les différentes étapes de l'implémentation du prototype. L'apport personnel de notre travail a consisté en l'enrichissement des médiateurs par l'intégration des concepts XML.

Chapitre 1 : Intégration des Données Hétérogènes

1. Introduction

L'accès «transparent» aux ressources et de manière plus générale à l'information constitue un des challenges actuels majeurs de l'informatique .L'avènement du Web et des réseaux informatiques tout comme l'accroissement des données et des services produits font que les utilisateurs finaux se trouvent confrontés à des problèmes de localisation et d'accès à l'information pertinente qu'ils requièrent. L'hétérogénéité, la quantité, la dispersion et la « volatilité » des ressources constituent autant de verrous que les systèmes d'intégration doivent lever .Les systèmes d'intégration de données permettent aux utilisateurs d'accéder, à travers un schéma global unifié, à plusieurs sources de données ayant chacune un schéma local. Bien que les systèmes actuels puissent surmonter la difficulté principale d'intégration qui est l'hétérogénéité des sources (XML, HTML, fichiers plats, etc.), leur mise en œuvre pose un certain nombre de problèmes, tant en ce qui concerne la génération des liens sémantiques entre le schéma de médiation et les sources de données (requêtes de médiation) qu'en ce qui concerne l'adaptation de l'accès aux besoins des utilisateurs ou la mesure de la qualité des données obtenues. Ces problèmes sont d'autant plus cruciaux lorsque les sources sont nombreuses et hétérogènes .Tout système d'intégration doit fournir les solutions aux problèmes suivants :

- Comment fournir une vue globale intégrée des données représentées à travers différentes conceptualisations?
- Comment identifier et spécifier le mapping entre des données sémantiquement liées?
- Comment mettre à jour les données de différentes bases étant donnée une telle vue globale intégrée ? (Boussis, 2008)

2. Approches de médiation

Afin d'intégrer des données distribuées au sein d'un médiateur, on distingue deux approches [Levy 1999] pour établir la correspondance entre un schéma global et les schémas des sources de données (voir figure1). D'une part, l'approche local-as-view (LAV) ou les vues locales participent à la création de la vue globale. D'autre part, l'approche global-as-view (GAV) ou le schéma global est défini comme une vue sur les schémas locaux.

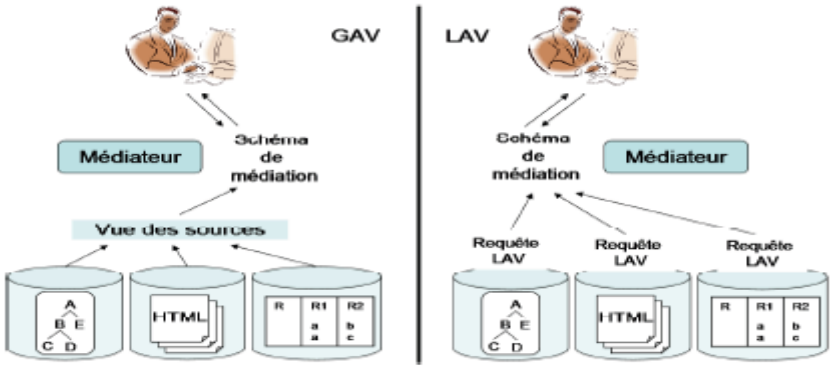


Figure 1. *Comparaison des architectures GAV et LAV*

Approche LAV. Dans cette approche, le schéma des sources est défini en fonction du schéma global de médiation. Les sources créent des requêtes décrivant leurs données en fonction du schéma de médiation. Cette approche permet d'intégrer ou de retirer facilement une source. En effet, chaque source est définie de manière indépendante des autres sources. Sa définition n'influe donc pas sur la définition du schéma global de médiation. La contrepartie de cette flexibilité est le coût de l'évaluation d'une requête sur le schéma de médiation puisqu'il est nécessaire de réécrire chacune des requêtes des sources selon un processus complexe d'inversion.

Approche GAV. Le schéma de médiation est défini en fonction des schémas sources. Le médiateur doit créer les requêtes permettant d'interroger chacune des sources pour obtenir les données du schéma de médiation. Contrairement à l'approche LAV, l'évaluation d'une requête sur le schéma de médiation consiste en une simple réécriture de la requête en fonction de chacune des sources. Par contre, cette approche demande une réécriture de la vue globale à chaque suppression, ajout ou modification d'une source.

3. Médiation de requêtes

Un médiateur présente des vues intégrées des sources de données. Lorsqu'un utilisateur interroge le médiateur, la requête est posée indépendamment de la localisation des différentes données intervenant pour calculer le résultat. Par rapport à l'interrogation d'une source unique, la médiation demande trois étapes supplémentaires :

- ***Décomposition de requêtes*** ; il s'agit d'identifier puis d'isoler les différentes parties de la requête correspondant à une localisation des données. La phase d'identification repose sur une connaissance des sources via les métadonnées permettant de localiser les sources ciblées. La phase d'isolation crée des requêtes spécifiques pour les sources à partir de ces sous-parties.
- ***Recomposition des résultats*** ; Les résultats obtenus à partir des sous-requêtes envoyées aux sources sont recomposées pour obtenir le résultat de la requête initiale. Dans ce processus, le moteur d'évaluation peut effectuer des opérations additionnelles sur les résultats dans le cas de dépendances entre des sous-requêtes (i.e., jointure, agrégat, union) ou du manque de capacité de traitement au niveau de la source (i.e., fonction de recherche de valeur, agrégat).
- ***Optimisation du plan global*** ; Le médiateur n'a pas de vision globale des coûts de traitement des sous-requêtes par les sources. L'optimisation du plan défini par le médiateur doit s'appuyer sur un modèle de coût intégrant les différentes stratégies d'évaluation proposées par les sources (i.e. stockage, indexation) ainsi que celles du médiateur. L'optimisation consiste à composer le plan en fonction de l'ensemble de ces informations.

4. Architectures de médiation

Cette architecture se compose de trois niveaux :

Le niveau source : comporte les différentes sources de données. Les sources de données communiquent avec le médiateur à l'aide d'adaptateur (wrapper) publiant de manière homogène les données de la source. L'adaptateur est chargée de **(i)** traduire une requête exprimée dans le langage du médiateur en une requête exprimée dans le langage de la source, **(ii)** faire évaluer la requête par la source et **(iii)** renvoyer les résultats au médiateur.

Le niveau médiateur : comporte un ou plusieurs médiateurs permettant d'intégrer les données transmises par les adaptateurs des sources. Il s'occupe de faire l'interface entre une requête utilisateur et les sources évaluant la requête suivant le modèle décomposition, recomposition, optimisation citée précédemment. Il présente les données de manière homogène et centralisée à la couche supérieure, résolvant le problème d'hétérogénéité et de distribution des données.

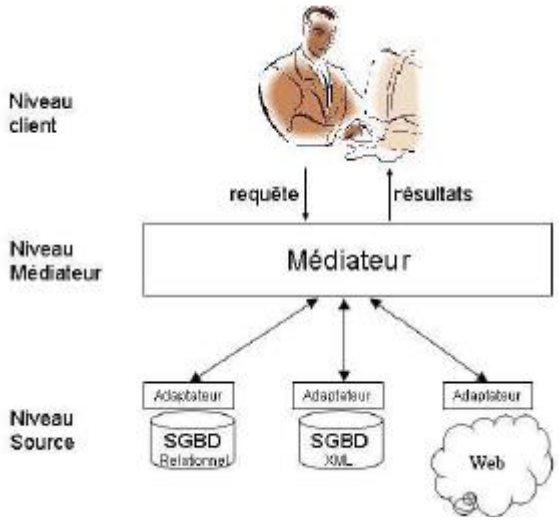


Figure 2. *Architecture de médiation*

Le niveau client : comporte les applications clientes pour accéder aux médiateurs (i.e. navigateur, interface graphique, API cliente). L'ajout d'une nouvelle source à un système de médiation entraine la définition d'un nouvel adaptateur pour la rendre accessible [Cluet et al. 1998].

5. Modèle de représentation des données hétérogènes

On trouve dans la littérature différents modèles de données sont utilisés pour représenter les données intégrées. Il existe quatre catégories de modèles : modèle relationnel, modèle orienté objet, modèle logique et modèle semi structuré.

La tendance actuelle s'oriente vers le modèle semi structuré XML [DANG 03]. XML (eXtensible Markup Language) est une recommandation du W3C. C'est un langage à balise définissant un format universel de représentation des données. Un document XML contient à la fois des données et les indications sur le rôle que jouent ces données. Ces indications permettent de déterminer la structure du document : ce sont des balises.

XML est un format largement répandu et accepté par la communauté informatique. De nombreuses sources de données sont disponibles et de nombreuses applications permettent d'exporter leurs données en XML. XML offre deux techniques pour créer une structure de document XML:

1.5.1 Les DTDs

DTD signifie Document Type Definition (ou définition de type de document), c'est la grammaire historique des documents XML. La puissance de description des DTD est faible : une DTD permet uniquement de décrire la structure d'un document XML (liste des balises et organisations

des balises), et non la typologie des données contenues (chaîne de caractère, date, entier, etc)

1.5.2 Les Schémas XML

Un schéma XML a le même rôle qu'une DTD, c'est-à-dire définir la structure d'un document XML. L'objectif des schémas XML est de proposer une solution aux inconvénients majeurs des DTD:

- Une syntaxe différente que celle de XML : on doit donc écrire un document XML dans un langage et la DTD dans un autre;
- La notion d'espace de noms n'existe pas (pour faire face aux risques de collisions de balises);
- Pas de notion de type de données;
- Pas de notion d'héritage: Les systèmes orientés objet s'appuient sur l'idée de décrire de nouveaux objets à partir d'objets existants. Ceci n'est pas possible.

Un schéma XML est composé principalement d'éléments qui sont représentés par les balises. Ces éléments sont associés à un type de données qui peut être défini comme « type simple » ou « type complexe ».

La puissance de XML Schéma réside surtout dans sa capacité à pouvoir définir des types personnels (des types complexes)[COST 02]. Ces types complexes sont créés en utilisant l'élément complexType composé généralement d'une série de déclarations d'éléments et d'attributs et de références d'éléments. Ces déclarations sont rassemblées dans des groupes modèles permettant de définir, un modèle de contenu constitué d'une séquence (ensemble ordonné), d'un choix exclusif (choice) ou d'un ensemble non-ordonné (all).

XML Schéma fournit plusieurs fonctionnalités s'appuyant sur XPath et permettant de décrire des contraintes d'unicité et des contrôles référentiels [CHAZ 01][THOM 01]. La première est une déclaration simple d'unicité et utilise l'élément "unique". La seconde déclaration, "Key" est similaire à "unique" et spécifie en plus que cette valeur unique pourra être utilisée comme une clé, ce qui lui donne deux contraintes supplémentaires: elle doit être non nul et être référençable. La déclaration, "Keyref" définit une référence à une clé.

1.5.3 Langages d'interrogation pour XML

Il y a eu de nombreuses propositions de langages de requêtes pour XML (XPath,XQL, XSLT, Quilt,…)[GIRAR 01]. Récemment, le W3C a proposé le langage XQuery[W3C 01] qui est plus adopté pour l'interrogation des schémas XML. XQuery reprend les avantages de XPath, XML-QL et XQL[DANG 03].
Ce langage a été conçu pour permettre de créer des requêtes précises tout en pouvant s'adapter à tout type de source de données XML, qu'il soit bases de données ou documents. Comme OQL, XQuery est un langage fonctionnel où chaque requête est une expression. XQuery est issue du langage de requête Quilt qui lui même empruntait de nombreuses fonctionnalités de XPath, XQL, XML-QL SQL, OQL. Il adopte de XPath et XQL la syntaxe de l'expression chemin (path expression) qui s'accorde avec les documents hiérarchiques. Il prend de SQL l'idée des séries de

clauses basées sur les mots clés qui produisent une forme pour restructurer des données (Select-From-Where dans SQL)[W3C01][BOUZ 03].
L'élément de base du langage XQuery est l'expression. Tout en XQuery est une expression. Un programme ou script XQuery n'est rien d'autre qu'une expression, avec en option des fonctions ou définitions.

Les expressions FLWR (prononcé flower) :

La forme de requête est ***FLWOR*** (For–Let–Where–Order–Return) dans des forêts :

for $<var1> in <forest1> [, $<var2> in <forest2>]…// itération
let $<varn>:= <subtree> // assignation
where <condition> // élagage
order-by $var1 // ordonnancement
return <result> // construction

• F: Collection d'arbres utilisés équivalent du FROM de SQL;
• L: Mémorisation d'arbres et affectation de variables locales;
• W: Condition (élagage) équivalent du WHERE de SQL;
• O: Ordonnancement équivalent de ORDER-BY de SQL;
• R: Sous–arbres sélectionnés, Présentation des sous–arbres équivalent du select de SQL avec une reconstruction [JACQUES 2002, DANIEL 2004]

Itère sur les arbres de la forêt forest1, forest2 et construit une nouvelle forêt formée de la séquence des arbres (accède à un arbre var1 de la forêt (séquence) forest1 et à un arbre var2 de la forêt (séquence) forest2, teste si l'arbre var1 et var2 vérifient la condition et construit un nouvel arbre result ordonné par var1)

XQuery est un langage fonctionnel, Une requête XQuery est une expression qui est évaluée dans un certain contexte. La valeur d'une expression est une séquence. Une séquence est une collection de 0 ou plusieurs items. Un item est une valeur atomique ou un noeud de l'arbre d'un document. Une valeur atomique est une instance de l'un des types atomiques de XML–Schema: (nombres, chaînes, dates…)
Un noeud est défini comme dans XPath: il est de type document, élément, attribut, texte, instruction, espace de noms ou commentaire ; Il a un nom et une valeur–chaîne [JACQUES 2002, DANIEL 2004].

Exemple

Requête: à partir du document ci-dessus donner le Titre, l'auteur et l'année de publication des livres publiés par Addison–Wesley après 1991.

```
<bib>
    <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author>W.Stevens</author>
        <publisher>Addison-Wesley</publisher>
    </book>
    <book year="1992">
        <title>Advanced Programming in the Unix environment</title>
        <author>W.Stevens</author>
        <publisher>Addison-Wesley</publisher>
    </book>
    <book year="2000">
        <title>Data on the Web</title>
        <author>Serge Abiteboul</author>
        <publisher>Morgan Kaufmann Publishers</publisher>
    </book>
    <book year="1999">
        <title>The Economics of Technology and Content for ...</title>
        <author>Darcy Gerbarg</author>
        <publisher>Kluwer Academic Publishers</publisher>
    </book>
</bib>
```

Figure 3. *Exemple de docuement XML*

La requête précédente se traduit en XQuery sous la forme du document suivant:

```
for $b in document("books.xml")//book
where
    $b/publisher = "Addison-Wesley"
    and $b/@year > 1991
return
    <livre annee="$b/@year">
        <titre>$b/title</titre>
        <auteur>$b/author</auteur>
    </livre>
```

Figure 4. *Requete en XQuery*

Le résultat de la requête sera le document suivant:

```
<livre annee="1994">
    <titre >TCP/IP Illustrated</titre>
    <auteur>W.Stevens</auteur>
</livre>
<livre annee="1992">
    <titre>Advanced Programming in the Unix environment</titre>
    <auteur>W.Stevens</auteur>
</livre>
```

Figure 5. *Résultat de la requête.*

XAlgèbre :

En algèbre relationnelle les requêtes et les évaluations de requêtes sont axées sur la notion de tuple. Un tuple est simplement un tableau de valeurs simples correspondant à des attributs. De sorte qu'un ensemble de tuples (ou relation) résultat d'un opérateur d'algèbre ne se résume que par un tableau à deux dimensions de valeurs. Les opérateurs de base de l'algèbre relationnelle sont l'union, la différence, la projection et la sélection. Avec l'algèbre relationnelle, chaque opérateur prend un ou plusieurs arguments de type relation et renvoie une relation. D'autres opérateurs (jointure, intersection) composés d'opérateurs de base sont venus enrichir cette algèbre. Des règles de simplification d'expressions permettent d'optimiser les requêtes. La simplicité de cette algèbre est à la base de l'efficacité de l'algèbre relationnelle. C'est partant de cette constatation que l'on est amené à se demander s'il n'est pas possible d'adapter les opérateurs et travaux déjà réalisés pour le traitement des requêtes XQuery.

Modèle de données de la Xalgèbre

Le principe est d'analyser finement la requête à l'aide des métadonnées afin de déterminer tous les chemins qui seront utiles au niveau de chacun des opérateurs. Il est alors possible de référencer et manipuler tous les noeuds en fonction des chemins ainsi calculés lors de la récupération des résultats par le médiateur. Nous créons ainsi à la volée des structures (nommée XTuple) lors de la construction des arbres récupérés [DANG 2003, GARDARIN 2004, GARDARIN 2005].
Un XTuple est composé d'un ensemble d'arbre A et un ensemble de références R sur A. Ces références sont appelées XAttributs. Les opérations relationnelles se font sur R alors que les parcours et recomposition se font sur A. Un ensemble de XTuples du même type forment une XRelation. Les documents XML sont remontés sous forme de flux d'évènements SAX. Les XTuples sont construits au vol sur les flux.
Les XOpérateurs (s'ils ne sont pas bloquants) traitent les XTuples au fur et à mesure.
Les XOpérateurs N–aire parallélisent les différents flux de XTuples d'entrées.

Opérateurs

L'algèbre XAlgèbre comporte à la fois des opérations relationnelles pour traiter les tables de XAttributs et des opérations de navigation dans des arbres XML.
La procédure d'évaluation de chaque opérateur est décomposée en deux phases
La phase d'initialisation: cette phase analyse le(s) XRelation(s) en entrée ainsi que les paramètres associés à l'opérateur afin de déterminer quelles seront les opérations exactes à réaliser quand les XTuples arriveront.
La phase d'exécution: cette phase est réalisée lors de l'évaluation de la requête et commence lorsque les premiers XTuples arrivent en entrée. Le traitement des XTuples se fait en suivant les indications préparées par la phase d'initialisation.

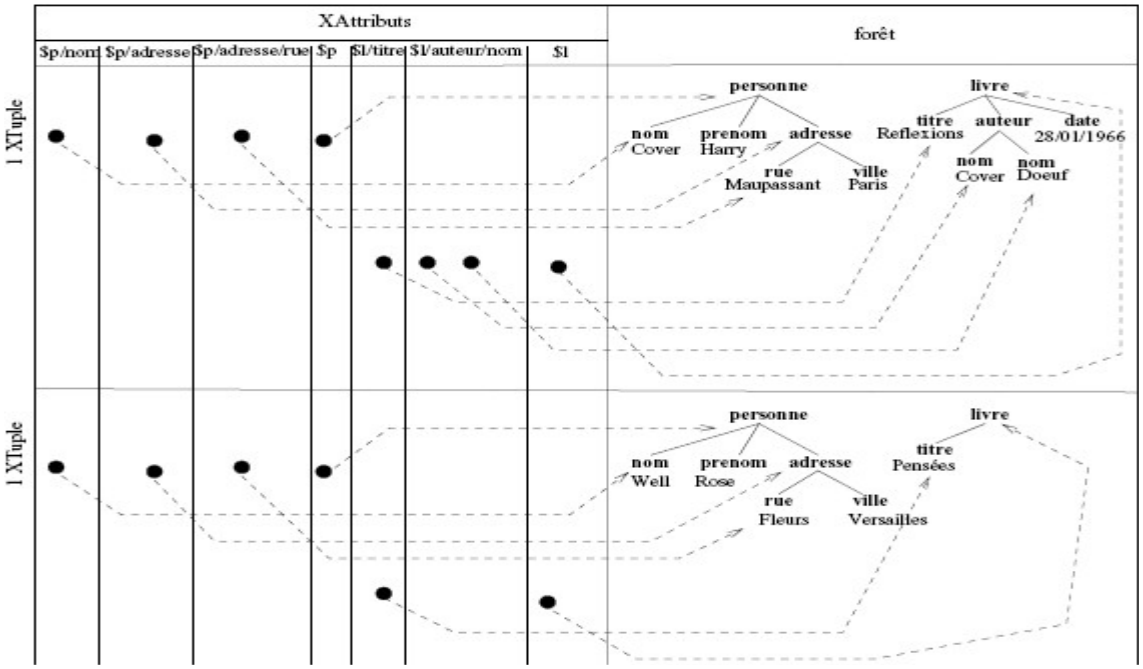


Figure 6. *Structure d'un XTuple*

6. Approches Existantes

Dans la littérature, il existe plusieurs approches traitant l'intégration des sources XML, on cite :

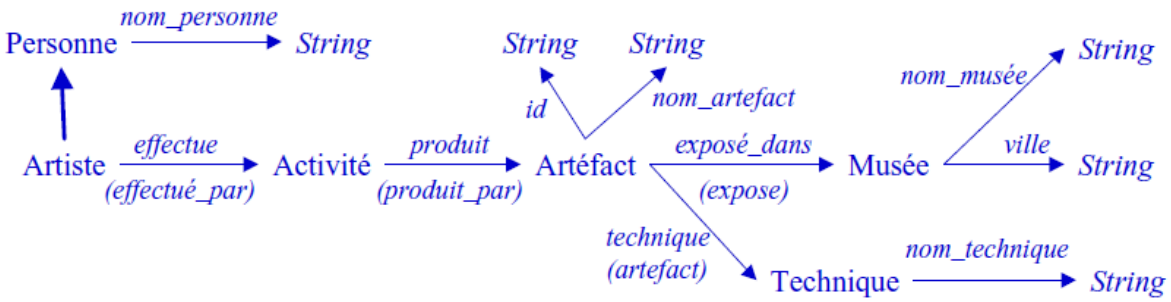
- **STYX** qui est un modèle de médiation riche Ontologie, mappings de chemins utilisant XPath , complétude par la jointure par clé.
- **Xyleme** qui traite un nombre de sources très important, performances, passage à l'échelle, calcul semi-automatique des mappings

***STYX :**

C'est un médiateur pour communauté WEB qui ce caractérise par :

- Données dans un domaine d .application bien déterminé
- Sources XML décrites par DTDs
- Modèle global décrit par une ontologie
- Mappings (chemin ontologie - chemin DTD XPath)
- Approche « local-as-view »
- Pas de matérialisation au niveau du médiateur
- Notion de clé pour les jointures entre sources
- Requêtes de type arbre dans l .ontologie traduites en requêtes de type arbre dans les sources XML

Son modèle du médiateur se base sur les Ontologie



Les sources :

Ressources XML, identifiées par une URL et décrites par une DTD.

Exemple : Œuvres de peintres http://www.peintres.com

<!ELEMENT Collection (Peintre*)>
<!ELEMENT Peintre (Peinture+, Sculpture+)>
<!ATTLIST Peintre Nom CDATA #REQUIRED>
<!ELEMENT Peinture (Technique?, Musee)>
<!ATTLIST Peinture Titre CDATA #REQUIRED>
<!ELEMENT Sculpture (Technique?, Musee)>
<!ATTLIST Sculpture Titre CDATA #REQUIRED>
<!ELEMENT Technique #PCDATA>
<!ELEMENT Musee #PCDATA>

Les mappings

Publication d'une source dans le médiateur et spécification de l'ensemble de mappings de la source

A1 : http://www.peintres.com/Collection/Peintre as u1 ® Personne
A2 : u1/@Nom as u2 ® nom_personne
A3 : u1/Peinture as u3 ® effectue.produit
A4 : u1/Sculpture as u3 ® effectue.produit
A5 : u3/Musee as u4 ® exposé_dans.nom_musée

Caractéristiques

Chemins XPath dans la source (langage puissant)
Utilisation de variables liées à des instances (fragments) XML
Factorisation de mappings (m+n mappings au lie de m*n)
Un même fragment de la source peut jouer des rôles différents

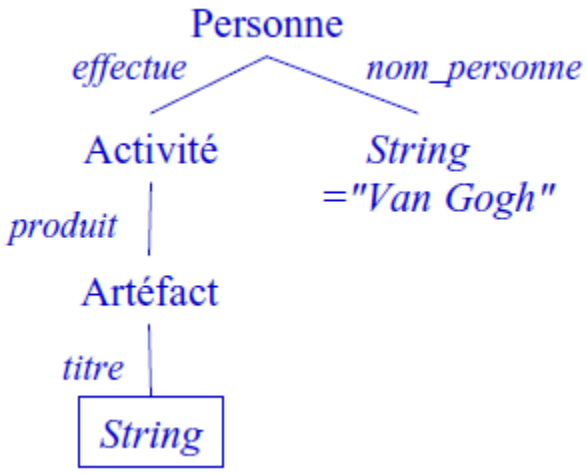
Les requêtes :

Requêtes de type arbre dans l'ontologie : Langage de type OQL

Exemple
Trouver les titres des œuvres de Van Gogh

```
Select   x3
from     Personne x1, x1.nom_personne x2,
         x1.effectue.produit.titre x3
where    x2 = "Van Gogh"
```

Remarque : des requêtes équivalentes sont possibles (ex. avec Artéfact comme racine et utilisant les rôles inverses)



Les clés :

Clés pour les concepts de l'ontologie, Clés sémantique, qui « identifient » un concept
Intuitivement : un attribut du concept (ex. nom musée, identifiant artéfact)
Plus généralement : ensemble de chemins d'attributs partant du concept
Jointure

Une instance de concept peut être éclatée entre plusieurs sources, Une requête est traduite vers toutes les sources.
Différentes sources peuvent répondre avec des fragments d'une instance
Fusion des fragments concernant une même instance de concept
Jointure basée sur l'égalité des clés du concept.

Chapitre 2 : Génération automatique des requêtes de médiation

1. Introduction

Un des principaux problèmes rencontrés dans la conception d'un système de médiation est le problème de définition de requêtes calculant une relation de médiation. En raison du grand nombre de sources de données qui peuvent être impliquées (des centaines ou des milliers) et du volume important de métadonnées les décrivant (description des schémas des sources et du schéma global, assertions de correspondance linguistique, assertions intra-source, . . . etc.), il est difficile d'envisager une écriture manuelle des requêtes de médiation. La question principale est de savoir comment automatiser la génération de requêtes de médiation ?

En réponse à cette problématique, nous adoptons l'approche proposée par [SOUKANE 05] en vue de la génération automatique de requêtes de médiation, pour le contexte XML.Les schémas de médiation sont supposés, déjà définis, ainsi que l'ensemble de métadonnées.

On se place dans une approche GAV (Global As View) où chaque objet du schéma global est défini par une requête sur les sources de données.

On peut résumer le processus de génération automatique de requêtes de médiation, calculant une relation $\mathbf{R_m}$ du schéma de médiation, par les étapes suivantes :

1. Identification des relations sources pertinentes pour la définition d'une requêtes de médiation **Q** du schéma de médiation, et génération des relations de mapping $\mathbf{T_i}$ qui sont obtenues par la projection des relations sources sur leurs attributs communs avec la relation $\mathbf{R_m}$.

2. Identification des opérations relationnelles possibles entre les relations de mapping $\mathbf{T_i}$ en fonction de leur schéma et de leurs clés, et génération du graphe d'opérations.

3. Recherche des chemins de calcul à partir du graphe d'opération pour calculer la relation de médiation $\mathbf{R_m}$.

4. Génération de requêtes de médiation déduites à partir des chemins de calcul de la relation de médiation $\mathbf{R_m}$.

Dans un premier temps, on suppose que nous sommes dans un environnement «semi hétérogène», les conflits sémantiques liés à l'hétérogénéité des données sont supposés résolus.

Ensuite, dans le cas d'un environnement hétérogène, la notion de type étendu, pour chaque attribut source, est introduite.

2. Méta données utilisées

Avoir une bonne connaissance du schéma global et de chaque schéma local est nécessaire dans la définition de requêtes de médiation pour répondre, au mieux, aux besoins des utilisateurs.

Nous présentons l'ensemble de métadonnées exploitées par le processus de génération de requêtes de médiation. Certaines de ces connaissances sont prédéfinies par le concepteur du système de médiation telles que : la description des schémas de relations, les clés des relations, les dépendances fonctionnelles, les contraintes référentielles entre relations, et d'autres sont ajoutées dans la base de connaissances au fur et à mesure de leur découverte automatique, au cours du processus de génération de requêtes de médiation telles que : les correspondances linguistiques entre les concepts des sources et les concepts du schéma de médiation, et les correspondances linguistiques entre les relations de sources différentes.

La base de connaissances notée A est constituée de trois catégories de métadonnées à savoir :

2.2.1 Métadonnées au niveau des sources

Les métadonnées définies au niveau des sources décrivent le schéma de chaque source de données, l'ensemble des relations sources appartenant à chaque schéma source, les clés des relations, les dépendances fonctionnelles éventuelles, les attributs de chaque relation, les assertions intra-source et inter-source entre les relations.

2.2.2 Métadonnées au niveau de la médiation

Les métadonnées définies au niveau de la médiation caractérisent le schéma de médiation, l'ensemble des relations de médiation appartenant à ce schéma de médiation, les clés des relations, les dépendances fonctionnelles éventuelles, et les attributs de chaque relation.

- Un schéma de médiation est constitué d'un ensemble de relations de médiation.

- L'ensemble d'assertions définies sur une relation de médiation est composé, essentiellement, de dépendances fonctionnelles qui relient l'attribut clé aux attributs non clés.

2.2.3 Métadonnées entre la médiation et les sources

Les métadonnées entre la médiation et les sources sont des correspondances linguistiques reliant un attribut d'une relation de médiation à un attribut d'une relation source. En d'autres termes, un attribut A d'une relation de médiation Rm est relié par une correspondant

3. Recherche des relations de mapping

2.3.1 Recherche des mapping étendus

La première étape de la génération de requêtes de médiation consiste à identifier les relations sources pertinentes au calcul de la relation de médiation $\mathbf{R_m}$, et à générer des relations de mapping $\mathbf{T_i}$ qui sont obtenues par la projection des relations sources sur leur attributs communs avec la relation $\mathbf{R_m}$.

Pour une relation de médiation donnée $\mathbf{R_m}$, la recherche des relations de mapping s'effectue en considérant, successivement, les relations de chaque source de données. Pour chaque relation source $\mathbf{R_i}$, chaque attribut **B** de $\mathbf{R_i}$ est comparé aux attributs de la relation de médiation en se basant sur les correspondances linguistiques définies entre un attribut d'une relation source et un attribut d'une relation de médiation.

Lorsque l'ensemble des attributs communs noté **E**, entre la relation de médiation et la relation source, est différent de l'ensemble vide, les clés primaires et étrangères sont recherchées

en se basant sur les dépendances fonctionnelles, les contraintes référentielles et sur les assertions inter source.

Les relations obtenues sont alors appelées relations de mapping étendu. L'ensemble des relations de mapping étendu associées à une relation de médiation sur l'ensemble des sources **S** est noté **Me.** L'algorithme suivant illustre le principe de la recherche des relations de mapping étendu.

```
Recherche de MappingEtendu (Rm (X), S, Me)
Entrée : Rm (X): la relation de médiation
         S : l'ensemble de sources de données
Sortie : Me : l'ensemble de relations de mapping étendu

1.  Me : = ∅
2.  Pour chaque source S dans S
3.   Pour chaque relation source R dans S
4.      Tester l'équivalence entre chaque attribut de R et les attributs de Rm
        et ajout des attributs équivalents dans l'ensemble E
5.      RechercheCléP (R, K) // recherche de la clé primaire K de R dans A
6.      RechercheCléE (R, B) // recherche des clés étrangères de R dans A
7.      Ajout des clés déterminées dans E
8.      Me = Me ∪{Ti = Π E R (Y)}// Création d'une relation de mapping étendu Ti
9.   FinPour
10. FinPour
Fin Recherche de MappingEtendu
```

Algorithme 1 : Algorithme de recherche de mapping étendu

2.3.2 Recherche des mapping de transition

Une fois les relations de mapping étendu $\mathbf{T_i}$ générées, le but est de trouver des opérations relationnelles pour les combiner, lorsqu' il n'y pas d'attributs communs entre deux relations de mapping étendu $\mathbf{T_i}$ **et** $\mathbf{T_j}$, aucun opérateur relationnel ne peut leur être appliqué. Cela revient donc à chercher en plus des relations de mapping étendu, une ou plusieurs relations dans les sources pouvant être utilisés pour combiner les deux relations de mapping. Nous désignons ces relations par des relations de transition.

La recherche des relations de mapping de transition pour une relation de médiation particulière revient à chercher, pour chaque paire de relations de mapping étendu **(Ti, Tj)** entre

lesquelles aucune assertion n'est définie, s'il existe, parmi les assertions définies dans la base de métaconnaissances ***A*** une séquence d'assertions permettant de lier la relation $\mathbf{R_i}$ (respectivement $\mathbf{R_j}$) à d'autres relations sources hormis les relations contributives. $\mathbf{R_i}$ **et** $\mathbf{R_j}$ sont les relations sources qui ont conduit à dériver $\mathbf{T_i}$ **et** $\mathbf{T_j}$**.**

La recherche des séquences d'assertions est effectuée par une procédure qui est appelée par l'algorithme de recherche des transitions. Cette procédure prend en entrée les relations source **Ri et Rj** et la base de connaissances **A.** Elle cherche, pour toute assertion **'a'** contenue dans la base **A**, s'il existe un lien entre la relation d'origine **Ri** avec d'autres relations. S'il existe, la procédure ***concatène*** l'assertion a identifiée à la séquence d'assertions courante notée ***SeqCourante***, elle continue à chercher des assertions jusqu'à ce qu'elle identifie une séquence d'assertions pertinente permettant de lier la relation origine **Ri** à la relation cible **Rj.** ***RechercheSéquence*** est une procédure complexe et récursive, elle cherche toutes les séquences possibles entre ***Ri et Rj.***

Une fois les séquences d'assertions calculées, l'algorithme de recherche de transitions prend en entrée les séquences d'assertions pertinentes contenues dans l'ensemble ***SéqTrouvée,*** il déduit, pour chaque séquence pertinente, la (les) relation(s) de transition. Ces relations sont obtenues par la projection des relations sources intermédiaires sur leurs attributs clés.

L'ensemble des relations de mapping de transition associées à une relation de médiation sur l'ensemble des sources **S** est noté **Mt.**

Ci-dessous est donné le principe de l'algorithme de recherche de mapping de transition et le principe de la procédure de recherche de séquence.

```
Recherche de MappingTransition (Me, SeqTrouvée, Mt)
Entrée : SeqTrouvée : l'ensemble de séquences d'assertions
         Me: l'ensemble de relations de mapping étendus
Sortie : Mt: l'ensemble de relations de mapping de transition

1. Mt : = ∅
2. Pour  chaque paire de relation de mapping étendu (Ti,Tj) dans Me tel que
     Ti ∩ Tj = ∅
3.   RechercheSéquence (Ri, Rj, A, SeqCourante, SeqTrouvée)
     //Rechercher toutes les séquences d'assertions dans A qui lient Ri et Rj
4.     Mt : = Mt ∪{Π B, B' (R'(Y))} // Création d'une relation de transition qui
   est une projection de la relation pertinente R' sur les attributs B et B'
5. FinPour
Fin Recherche de MappingTransition
```

Algorithme 2 : Algorithme de recherche de mapping de transition

```
RechercheSéquence (Ri, Rj, A, SeqCourante, SeqTrouvée)
Entrée : Ri : la relation source
         Rj : la relation cible
         A : l'ensemble d'assertions
         SeqCourante : la séquence d'assertions courante
Sortie : SeqTrouvée : l'ensemble de séquences d'assertions trouvées

1. Tantque Ri ≠ Rj
2.  Pour toute assertion a dans A
3.     RechercheSequence (Ri, Rj, SeqCourante ‖ a)
       // Procédure récursive qui cherche toutes les séquences  d'assertions
       dans A entre Ri et Rj
4.  FinPour
5. FinTantque
6.   SeqTrouvée : = SeqTrouvée  ∪ SeqCourante
     // Création d'une séquence d'assertions pertinente
Fin RechercheSéquence
```

Algorithme 3 : procédure de recherche de séquence d'assertions

4. Recherche du graphe d'opération

Étant donné l'ensemble de relations de mapping étendu **Me**, et l'ensemble de relations de mapping de transition **Mt** générés par la recherche des relations de mapping, le but de cette étape est de trouver les opérations relationnelles susceptibles de combiner chaque paire de relations en tenant compte des métadonnées.

La recherche de ces opérations est guidée par des règles d'intégration spécifiées sur les

connaissances. Ces règles d'intégration permettent, pour une relation de médiation particulière, de déterminer l'ensemble des jointures candidates, et ce pour chaque paire de relations de mapping. Une opération de jointure déterminée entre deux relations de mapping peut combiner soit une relation de mapping étendu avec une autre relation de mapping étendu, soit une relation de mapping étendu avec une relation de transition, ou encore une relation de transition avec une autre relation de transition. L'ensemble de ces opérateurs est représenté par un graphe d'opérations noté **GRM** où chaque noeud correspond à une relation de mapping, et chaque arc entre deux noeuds correspond à une jointure candidate déterminée à l'aide d'une règle d'intégration.

Règle 1 : Si les deux relations appartiennent à la même source et que leurs schémas ne sont pas disjoints, et où l'une des relations référence l'autre, alors l'opération candidate est une jointure naturelle déterminée par la règle suivante :

$$\text{Si} \quad \underline{T_i} \cap \underline{T_j} \neq \varnothing \text{ et } T_i.B \subseteq T_j.B$$

$$\text{Alors } T_i.B \bowtie T_j.B$$

Règle 2 : Si les deux relations n'appartiennent pas à la même source de données, on ne pourra pas disposer de contraintes référentielles auquel cas on utilisera la correspondance linguistique entre deux attributs, où l'un des deux attributs est clé dans l'une ou l'autre des relations. Dans ce cas, la règle d'intégration est la suivante :

$$\text{Si} \quad \underline{T_i} \cap \underline{T_j} \neq \varnothing \text{ et } T_i.B \cong T_j.B'$$

$$\text{Alors } T_i.B \bowtie T_j.B'$$

Compte tenu de ces règles d'intégration, nous présentons ci-dessous le principe de

l'algorithme de recherche du graphe d'opération.

```
Recherche de GrapheOpérations (Me, Mt, J)
Entrée : Me : l'ensemble de relations de mapping étendus
         Mt : l'ensemble de relations de mapping de transition
Sortie : J : l'ensemble de Jointures

1. J : = ∅
2. Pour  chaque paire (Ti,Tj) dans Me et Mt  tel que Ti ∩ Tj ≠ ∅
3.     Si Ti et Tj appartiennent à la même source
4.     Alors J := J ∪ {j = Ti.B  ⋈  Tj.B}
       // Création d'une opération de jointure entre Ti et Tj sur le
      critère de jointure Ti.B = Tj.B
5.    Sinon J := J ∪ {j = Ti.B  ⋈  Tj.B'}
6.    FinSi
7. FinPour
Fin Recherche de GrapheOpérations
```

Algorithme 4 : Algorithme de recherche du graphe d'opérations

5. Recherche des chemins de calcul

Les règles d'intégration permettent, pour une relation de médiation et pour l'ensemble de relations de mapping associées, de déterminer l'ensemble des opérations de jointures candidates, et ce pour chaque paire de relations de mapping. L'ensemble de ces opérations est représenté dans le graphe d'opérations décrit précédemment, où chaque nœud correspond à une relation de mapping, et chaque arc entre deux nœuds correspond à un opérateur candidat déterminé à l'aide d'une règle d'intégration.

La génération de requêtes de médiation se fait en recherchant des chemins de calcul dans le graphe d'opérations **GRM**.

Un chemin de calcul **GRM** associé à la relation de médiation **Rm** est un sous-graphe connexe et acyclique du graphe **GRM** où chaque attribut de la relation de médiation est équivalent à un attribut figurant dans le sous-graphe. En d'autres termes, chaque attribut de la relation de médiation figure dans au moins une des relations de mapping du sous graphe. Il peut arriver que tous les attributs de **Rm** figurent tous dans une seule relation de mapping. Dans ce cas, le chemin de calcul est constitué d'un seul nœud représentant une relation de mapping.

L'algorithme de recherche des chemins de calcul est un processus récursif complexe qui consiste à chercher dans le graphe de jointures **GRM** tous les chemins possibles permettant de calculer la relation de médiation **Rm.**

Il prend, en entrée, le graphe d'opérations et la relation de médiation. Il teste tout d'abord pour une jointure donnée reliant deux relations de mapping si tous les attributs de la relation de médiation figurent dans les deux relations, si oui un chemin de calcul est déjà identifié, sinon il ajoute la jointure **J** au chemin de jointures courant ***ChemCourant*** et il continue à chercher dans le graphe une jointure ayant un lien avec le chemin de jointures courant jusqu'à ce qu'il trouve un chemin de calcul pertinent où tous les attributs de **Rm** figurent. Un lien entre une jointure donnée et un chemin courant est établi si l'extrémité droite ou gauche de la jointure est égale à L'une des extrémités du ***chemcourant.***

Le processus de recherche des chemins de calcul est réitéré jusqu'à ce tous les chemins possibles soient identifiés.

```
RechercheChemin (XCourant, X, GRM , ChemCourant, ChemTrouvé)
Entrée : XCourant : l'ensemble des attributs du chemin courant
         X : l'ensemble des attributs de la relation Rm.
         GRM  : le graphe d'opérations
         ChemCourant : le chemin de jointures courant
Sortie : ChemTrouvé : l'ensemble de chemins de jointures trouvées

1. TantQue le chemin courant ne contient pas tous les attributs de Rm
2.    Pour toute jointure j ∈ GRM
3.      RechercheChemin (XCourant, X, ChemCourant|j)
        // Procédure récursive qui cherche toutes les chemins de jointures
        dans GRM  qui calculent Rm
4.    FinPour
5. FinTantQue
6.      ChemTrouvé : = ChemTrouvé  ∪ ChemCourant
        //création d'un chemin de calcul
FinRechercheChemin
```

Algorithme 5 : Algorithme de recherche des chemins de calcul

6. Prise en compte de l'hétérogénéité

Lors du processus de génération de requêtes de médiation, deux types de conflits liés à l'hétérogénéité des sources sont distingués à savoir :

Les conflits sémantiques liés au schéma,

L'utilisation d'une terminologie différente pour désigner deux concepts identiques entraîne la présence de conflits sémantiques liés au schéma. Par exemple, dans une relation de médiation : **produit**(Num-produit, désignation, prix) et dans la relation **produit** (Num-produit, désignation, prix-produit), les attributs prix et prix-produit ont deux terminologies différentes mais une sémantique identique.

Les conflits sémantiques liés aux données.

La provenance de données de diverses origines, leur saisie à des moments distincts par des personnes différentes qui n'ont pas la même perception du réel, et qui utilisent des conventions différentes entraîne ce type de conflits. Par exemple, différence d'unité de mesure, de précision, d'échelle, de format de date, etc.

2.6.1 Les métadonnées utilisées

En plus des métadonnées décrites dans la base de connaissances initiale, il existe :

- Un dictionnaire linguistique automatique pour détecter et résoudre automatiquement les conflits liés au schéma,

- Un type étendu d'un attribut pour détecter les conflits sémantiques liés aux données,

- Une librairie de fonctions de transformations pour transformer les données hétérogènes et garantir leur conformité mutuelle et leur conformité par rapport au schéma global.

2.6.2 Exploitation des métadonnées

L'approche présentée ici consiste à revisiter chaque étape principale de l'algorithme de génération de requêtes de médiation afin de détecter et de résoudre les conflits liés à l'hétérogénéité des sources au cours du processus de génération de requêtes de médiation.

On ajoute trois procédures : ***Compare, CheckType et Search*** qui exploitent les métadonnées précédentes à savoir le dictionnaire linguistique, le type étendu, et la librairie de fonctions de transformations pour l'identification et la résolution des conflits liés à l'hétérogénéité des données. Ces procédures sont appelées au cours de la génération de requêtes de médiation.

Nous présentons le principe et la spécification de chaque procédure.

1. La procédure Compare

Elle prend comme paramètres d'entrée un attribut A de la relation de médiation Rm et un attribut B d'une relation source R. Elle teste tout d'abord la correspondance linguistique des attributs A et B en utilisant le dictionnaire linguistique. Si A et B sont sémantiquement équivalents($A \cong B$), elle appelle la procédure **CheckType** (algorithme 7) sinon elle retourne faux.

```
Compare (A, B, CF)
Entrée : A : un attribut de la relation Rm
         B : un attribut de la relation source
Sortie : CF : ensemble de fonctions de transformations
Retour : un booléen

1. CF = ∅
2.     Si A ≅ B
3.     Alors return CheckType (TE (Rm.A), TE(R, B))
4.     Sinon return (faux)
5.     FinSi
Fin Compare
```

Algorithme 6 : La procédure Compare

2. La procédure CheckType

Elle est appelée pour détecter les conflits sémantiques liés aux données entre les attributs A et B. Elle exploite le type étendu des attributs. Elle prend comme paramètres d'entrée le type étendu de l'attribut A et le type étendu de l'attribut B.

```
    ChekType (TE(Rm.A) , TE(R.B),CF )
    Entrée : TE(Rm.A) : le type étendu de A
             TE(R.B)  : le type étendu de B
    Sortie : CF : l'ensemble de fonction de transformation
    Retour : un booléan CT

     CT = vrai
 1   pour chaque élément ei de TE(Rm,A)  telleque CT==vrai
 2      pour chaque élément ej de TE(R,B)
 3         si ei!=ej alors
 4            CT=faux
 5         Fin Si
 6      FinPour
 7   FinPour
 8   si CT =faux
 9      appeler Search()
10   sinon
11      return (CT)
12   Fin Si
13  Fin CheckType
```

Algorithme 7 : La procédure CheckType

3. La procédure Search

Elle prend comme paramètres d'entrée l'élément ei à traiter (ex : unité), sa valeur en entrée (ex : francs) et sa valeur en sortie (ex : euros), elle exploite la librairie de fonctions et cherche une fonction de transformation f qui a une valeur de paramètre d'entrée (valeur_in) égale à la valeur de l'élément ej et qui a une valeur de paramètre de sortie (valeur_out) égale à la valeur de l'élément ei. Si cette fonction existe, la procédure Search retourne la fonction f, la procédure CheckType retourne vrai et l'ensemble de fonctions CF, Compare retourne vrai et l'ensemble de fonctions de transformation CF, sinon CheckType et Compare retournent faux.

```
Search (e_i, TE_(R.B) [e_j], TE_(Rm.A)[e_i], Lib)
   Entrée :   e_i : l'élément du type étendu à traiter
              TE_(R.B) [e_j] : la valeur de l'élément e_j
              TE_(Rm.A) [e_i] : la valeur de l'élément e_i
              Lib : la librairie de fonctions
   Retour :   la fonction de transformation

   1.  Pour chaque fonction f ∈ Lib
   2.    Si (TE_(R.B) [e_j] = f.valeur_in) and TE_Rm.A) [e_i] = f.valeur_out)
   3.    alors return(f)
   4.    Finsi
   5.  FinPour
FinSearch
```

Algorithme 8 : La procédure Search

Chapitre 3 : Implémentation

1. Introduction

L'ensemble des algorithmes spécifiés dans le chapitre II ont été implémentés et composés pour former un outil de génération des requêtes de médiation dans le contexte XML.

L'implémentation du notre prototype de génération automatique des requêtes de médiation a été réalisée en Java (jdk-7u5), sous l'IDE NetBeans 7.1 et la méta-base est stockée sous Microsoft Access.

2. Architecture générale

Notre prototype de génération automatique des requêtes de médiation comporte, essentiellement, une Meta-Base et sept modules comme le montre la figure (7) :

- Interface graphique utilisateur.
- Interface graphique administrateur.
- Recherche des relations de mapping.
- Recherche des relations de transitions.
- Recherche des opérations de jointures.
- Recherche des chemins de calcul.
- Génération de requêtes de médiation.

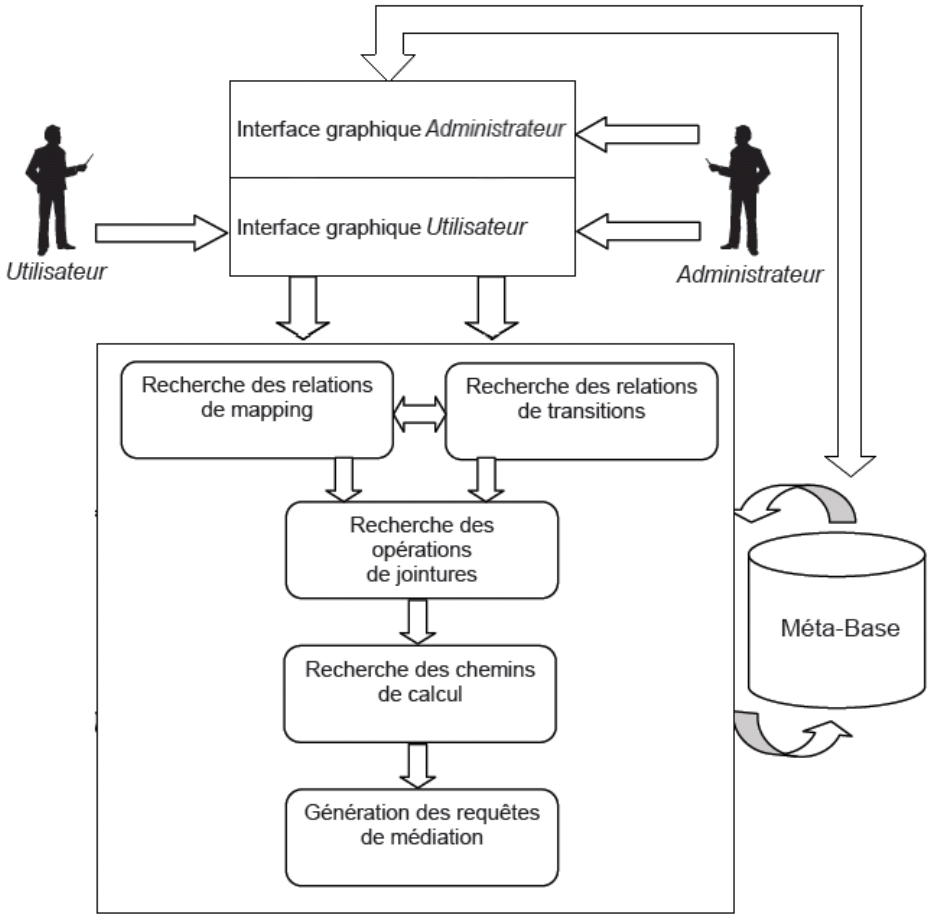


Figure 7. *Architecture du prototype de GARM*

L'interface graphique administrateur permet de mettre à jour (ajout, suppression et modification) toutes les métadonnées utilisées pour la génération des requêtes de médiation.

L'interface graphique utilisateur permet d'interagir avec les cinq autres modules, et de montrer au fur et à mesure les résultats de chaque module du processus de génération des requêtes.

Tous les modules communiquent avec la méta-base, les résultats produits sont également stockés dans cette dernière.

3. Description de la méta-base

La méta-base exploitée par le processus de génération des requêtes est constituée de :

3.3.1 Métadonnées au niveau de la médiation

La description du schéma global (Schéma_Médiation) comporte les schémas des relations de médiation, les clés des relations. Chaque relation (Relation_Médiation) est constituée d'un ensemble d'attributs de médiation (Attribut_Médiation), et pour chaque attribut d'une relation son type étendu (Element_Médiation).

3.3.2 Métadonnées au niveau des sources :

La description du schéma local (Schéma_Source)à chaque sources de données comporte les schémas des relations sources, les clés des relations. Chaque relation (Relation_Source) est constituée d'un ensemble d'attributs source (Attribut_Source), et pour chaque attribut d'une relation, son type étendu (Element_Source).

3.3.3 Métadonnées au niveau intermédiaire :

Ce niveau décrit les assertions intra-source (Contrainte_Ref) définies entre deux attributs d'une même source, les assertions intersource (Corresp Ling (S-S)) définies entre deux attributs de sources différentes, les correspondances linguistiques (Corresp Ling (S-M)) définies entre un attribut médiation et un attribut source, et les fonctions de transformations (Fonction) qui transforment la valeur d'un élément du type étendu d'un attribut source en une autre valeur.

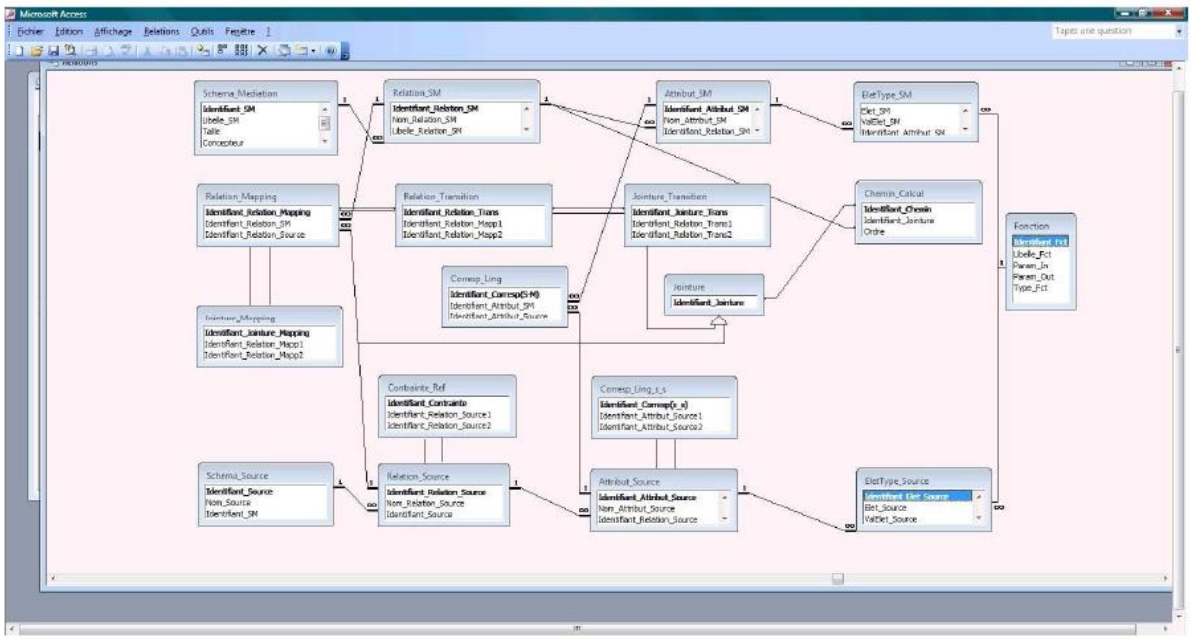


Figure 8. *La description du Méta-Base*

4. La description des différents modules

3.4.1 La recherche des relations de mapping étendu

Ce module permet d'identifier les relations sources pertinentes au calcul de la relation de médiation, et de générer, en sortie, les relations de mapping étendu. Les fonctionnalités de ce module se trouvent dans la classe MappingEtendu.

Il récupère les méta-données relatives à un schéma de médiation, a partir de l'accès via JDBC à la Méta-Base.

Il appelle la classe Compare qui a pour objectif de mettre en conformité les données hétérogènes par rapport au schéma global, cette classe prend en paramètres d'entrée un attribut de médiation et un attribut source, et pour chaque attribut son type étendu, elle compare tout d'abord l'équivalence linguistique entre les deux attributs, et ensuite elle compare terme à terme chaque élément du type étendu de l'attribut de médiation avec chaque élément du type étendu de l'attribut source, elle accède à la table Fonction de la méta-base par la classe Connexion Metabase, elle retourne en paramètre de sortie un ensemble de fonctions de transformations CF.

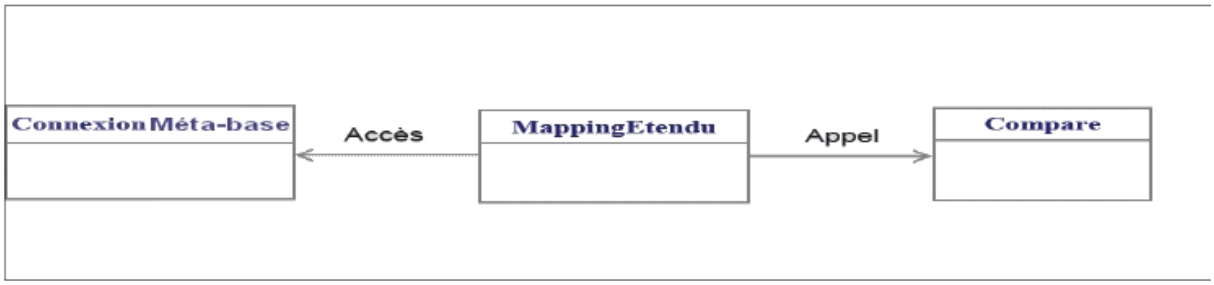


Figure 9. *Diagramme La recherche des relations de mapping étendu*

3.4.2 La recherche des relations de transition

Ce module identifie des relations sources intermédiaires pour établir des liens entre les relations de mapping étendu. Les fonctionnalités de ce module se trouvent dans la classe MappingTransition. Il prend en paramètre d'entrée l'ensemble de relations de mapping étendu généré par la classe MappingEtendu. Il appelle la classe RechercheSeq qui a pour objectif de calculer la séquence d'assertions entre chaque paire de relations de mapping étendu.

La classe MappingTransition accède, via un accès JDBC, à la méta-base des tables Contrainte_Ref et Corresp Ling (S-S), elle retourne en sortie un ensemble de séquences d'assertions pertinentes entre chaque relation de mapping étendu.

Lorsque deux relations de mapping étendu n'appartiennent pas à la même source, la classe MappingTransition appelle la classe Compare pour gérer les conflits liés à l'hétérogénéité des données qui génère en sortie un ensemble de fonctions de transformations CF.

Le diagramme de classe suivant montre les fonctionnalités de ce module :

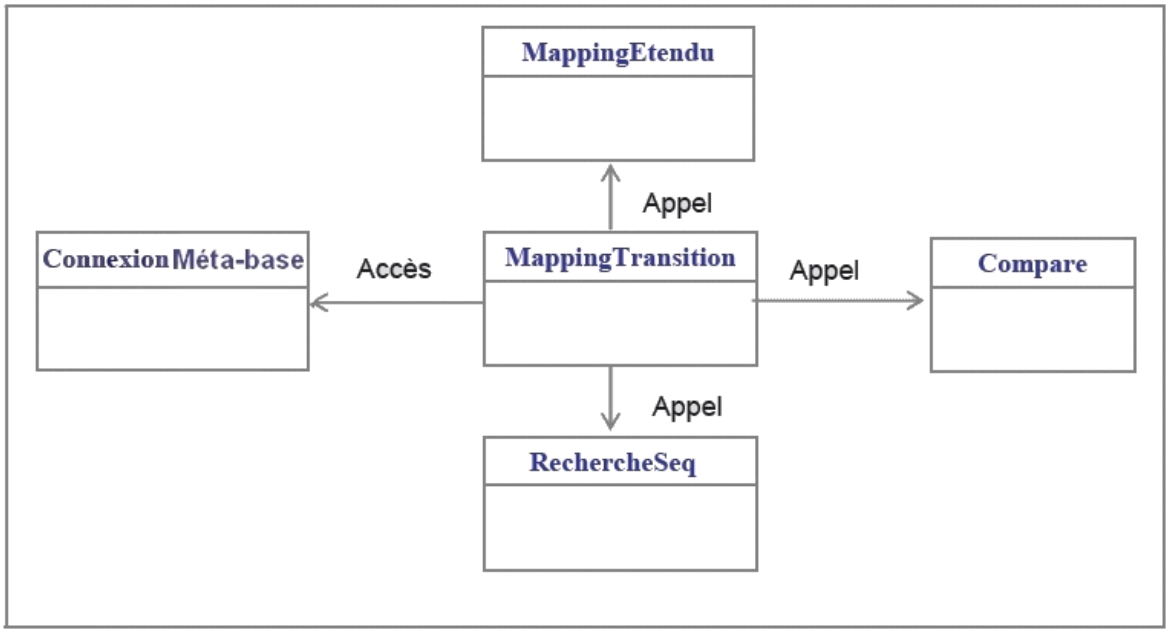


Figure 10. *Diagramme de la recherche des relations de transition*

3.4.3 La recherche des opérations de jointures

Ce module identifie les jointures candidates pour combiner les relations de mapping. Il génère en sortie un graphe d'opérations de jointures. Les fonctionnalités de ce module se trouvent dans la classe GrapheOperations.

Il prend comme paramètres d'entrée l'ensemble de relations de mapping étendu Me et l'ensemble de relations de mapping de transition Mt, il récupère via un accès JDBC les tables Contrainte_Ref et Corresp Ling (S-S).

Cette classe appelle aussi la classe Compare pour détecter et résoudre les conflits liés à l'hétérogénéité des données qui génère un ensemble de fonctions de transformations.

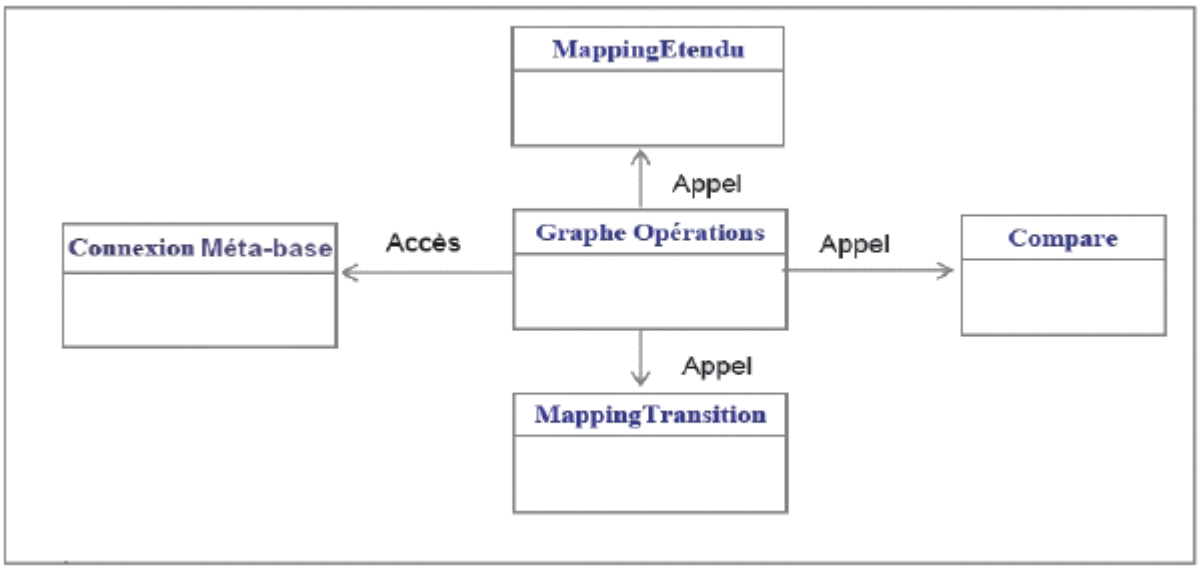


Figure 11. *Diagramme 3 La recherche des opérations de jointure*

3.4.4 La recherche des chemins de calcul

Ce module a pour objectif d'identifier les chemins de jointure au calcul d'une relation de médiation. Il génère en sortie un ensemble de chemins de calcul. Les fonctionnalités de ce module se trouvent dans la classe RechercheChemin. Il prend comme paramètres d'entrée la relation de médiation et le graphe d'opérations généré par la classe GrapheOperations.

La classe RechercheChemin énumère tous les chemins de jointures possibles depuis le graphe de jointures et génère en sortie un ensemble de chemins. Le diagramme de classe suivant montre les fonctionnalités de ce module :

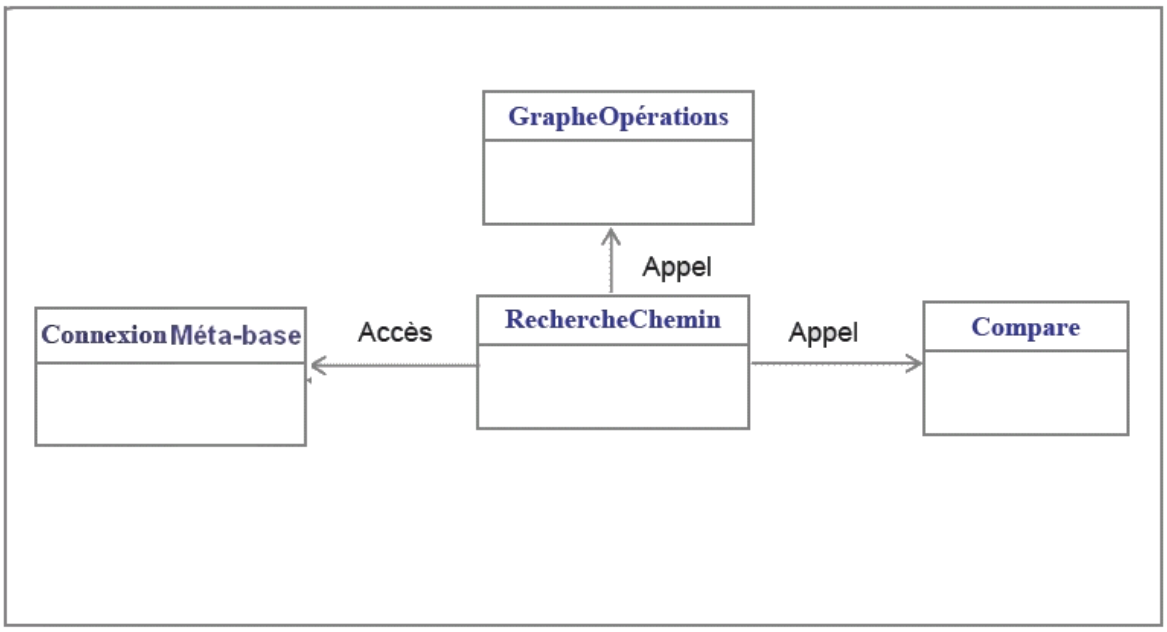


Figure 12. *Diagramme la recherche des chemins de calcul*

3.4.5 La recherche des requêtes de médiation

Ce module a pour objectif de générer les requêtes de médiation relatives à une relation de médiation particulière. Il génère en sortie des requêtes SQL. Les fonctionnalités de ce module se trouvent aussi dans la classe RechercheChemin. Cette classe déduit des requêtes SQL à partir des chemins de calcul.

Le diagramme de classe suivant montre les fonctionnalités de ce module :

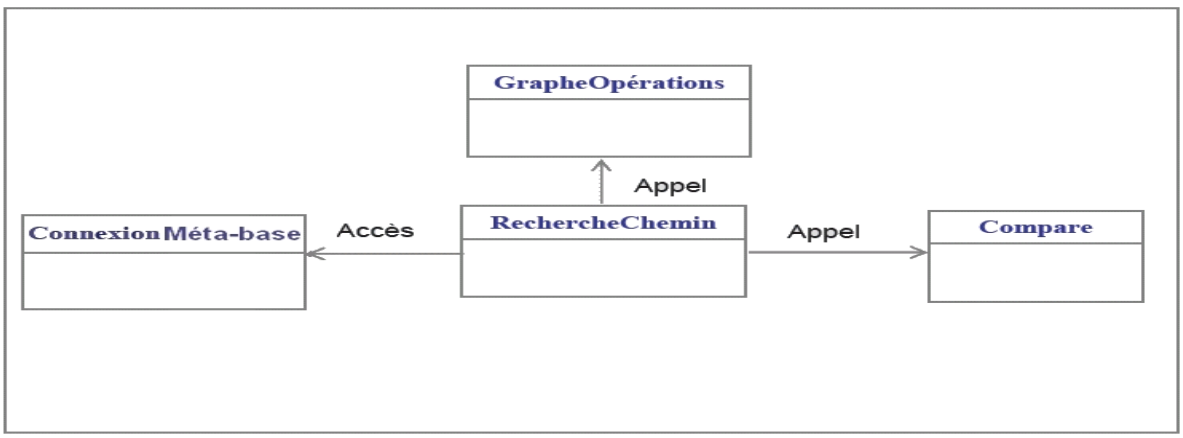


Figure 13. *Diagramme La génération des requêtes de médiation*

5. Scénario de fonctionnement

Le prototype implémenté génère les requêtes de médiation, qui à partir des schémas sources et de médiation, produit un ensemble de requêtes potentiel calculant cette relation. Pour ce faire, l'administrateur doit configurer d'abord la Méta-Base, pour que l'utilisateur puisse générer les requêtes de médiation.

3.5.1 Administrateur

La fenêtre suivante regroupe les tâches effectuées par l'administrateur.

Figure 14. *Interface Administrateur*

1 : Ce bouton permet d'accéder à une autre fenêtre pour configurer la connexion aux bases de données (médiation et sources de données), comme le montre la figure (15).

2 : Ce bouton permet d'ouvrir une autre fenêtre pour ajouter, supprimer ou modifier une relation de médiation, comme le montre la figure 3.5.

3 : Ce bouton affiche une fenêtre où l'administrateur procède à la configuration de la méta-Base, comme le montre la figure 3.6.

4 : Ce bouton permet d'afficher une fenêtre qui contient des informations sur la session administrateur qu'il peut modifier, comme le montre la figure 3.7.

3.5.1.1 Configuration des connexions aux bases de données

L'administrateur configure les connexions en spécifiant leur pilote, adresse URL, login, et leur mot de passe, pour le schéma de médiation et pour les sources de données.

Figure 15. *Configuration de la méta-base*

3.5.1.2 Fenêtre de création d'une relation de médiation

Figure 16. *Création d'une relation de médiation*

3.5.1.3 La configuration de la méta-base

Cette fenêtre permet de configurer la Méta-Base en suivant les étapes :

1 : permet de charger automatique les informations sur le schéma médiation via la Méta-Base.

2 : charge automatiquement les informations sur les sources.

3 : permet d'ouvrir la fenêtre de configuration du type étendu d'un attribut.

4 : pour ouvrir la fenêtre de configuration des contraintes linguistiques inter-sources.

5 : pour ouvrir la fenêtre de configuration des contraintes linguistique source-médiation.

6 : pour ouvrir la fenêtre de configuration des contraintes référentielles.

7 : pour ouvrir la fenêtre de configuration de la liste des fonctions.

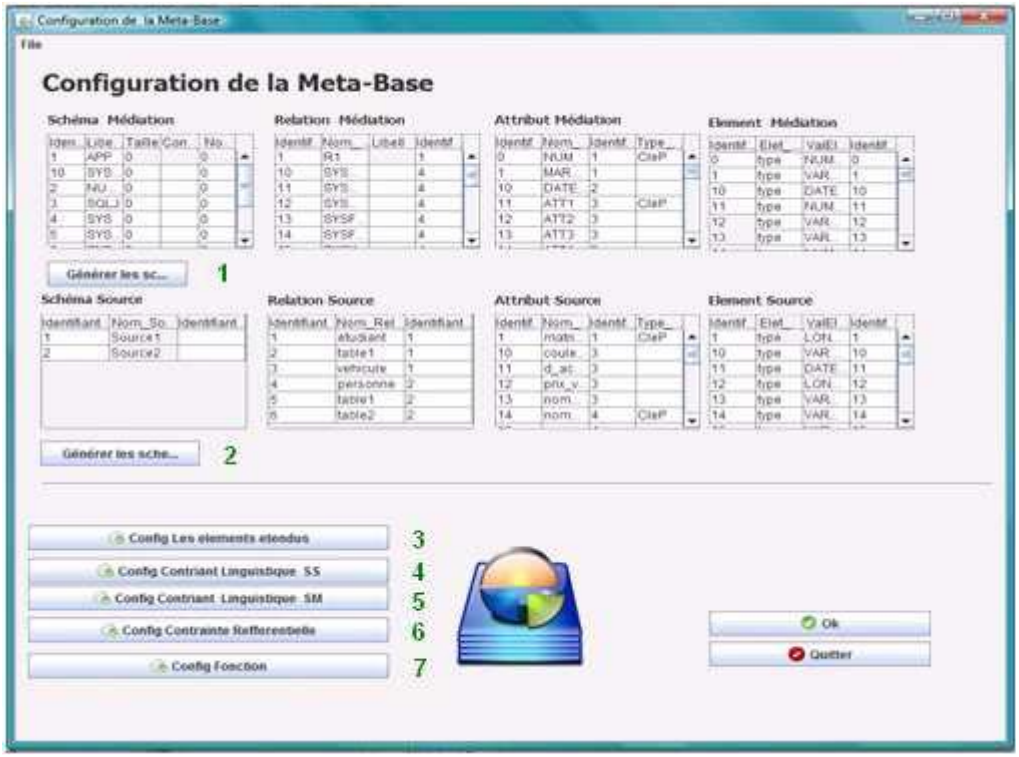


Figure 17. *Configuration de la méta-base*

3.5.1.4 Fenêtre de gestion des Comptes

Figure 18. *Gestion des comptes*

3.5.2 Partie Utilisateur

La fenêtre principale de l'utilisateur de notre prototype se présente comme suit :

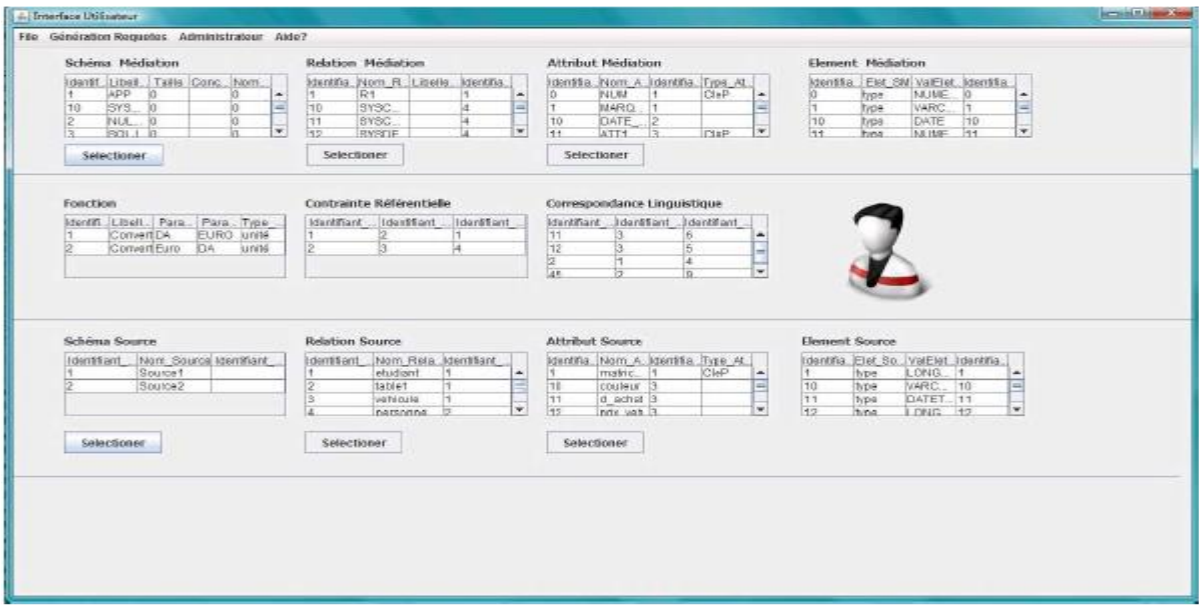


Figure 19. *Fenêtre utilisateur*

L'utilisateur peut consulter la base de méta connaissances pour demander la génération des requêtes de médiation selon les étapes suivantes :

3.5.2.1 Fenêtre de génération des relations de mapping étendu :

L'utilisateur peut choisir le schéma et la relation de médiation comme le montre la figure :

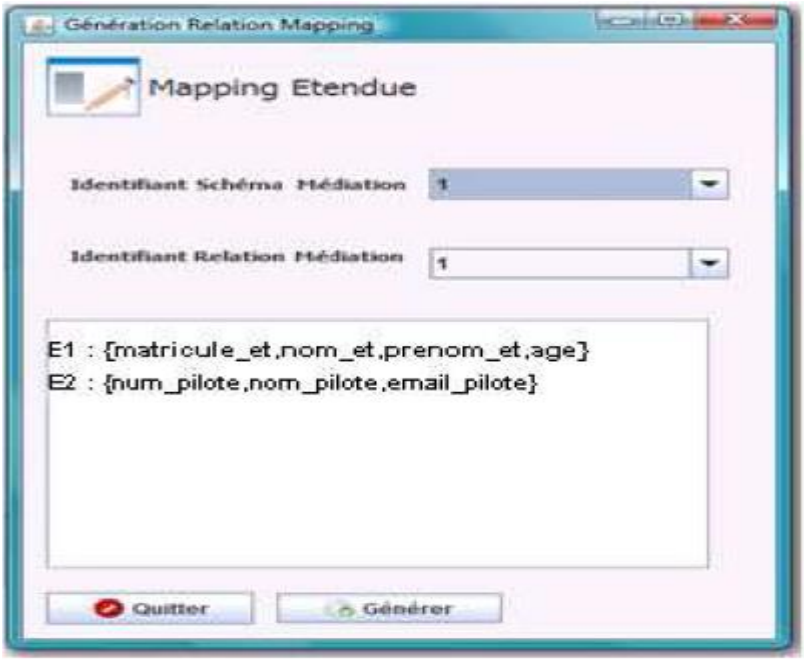


Figure 20. *Fenêtre de mapping*

3.5.2.2 Fenêtre de génération des relations de mapping transition

Figure 21. *Fenêtre de mapping de transition*

3.5.2.3 Fenêtre de génération des opérations de jointure

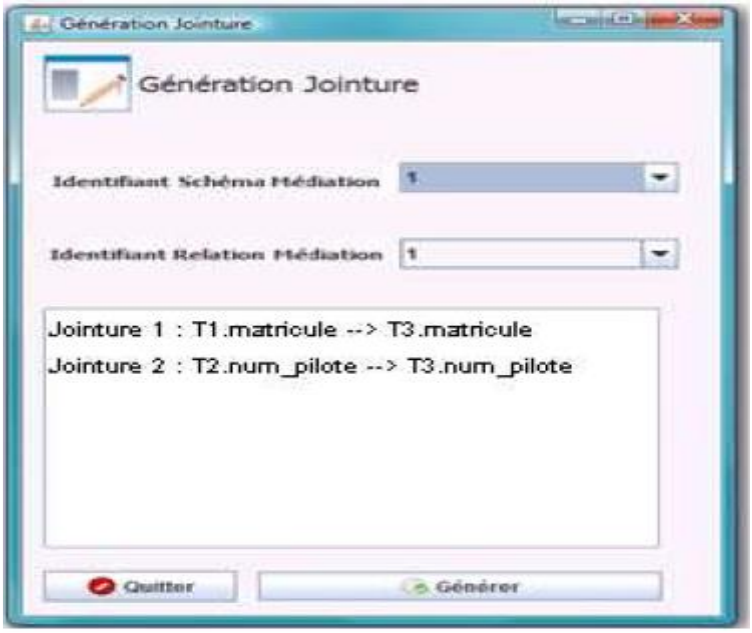


Figure 22. *Fenêtre de génération des opérations de jointure*

3.5.2.4 Fenêtre Génération des chemins de calcul

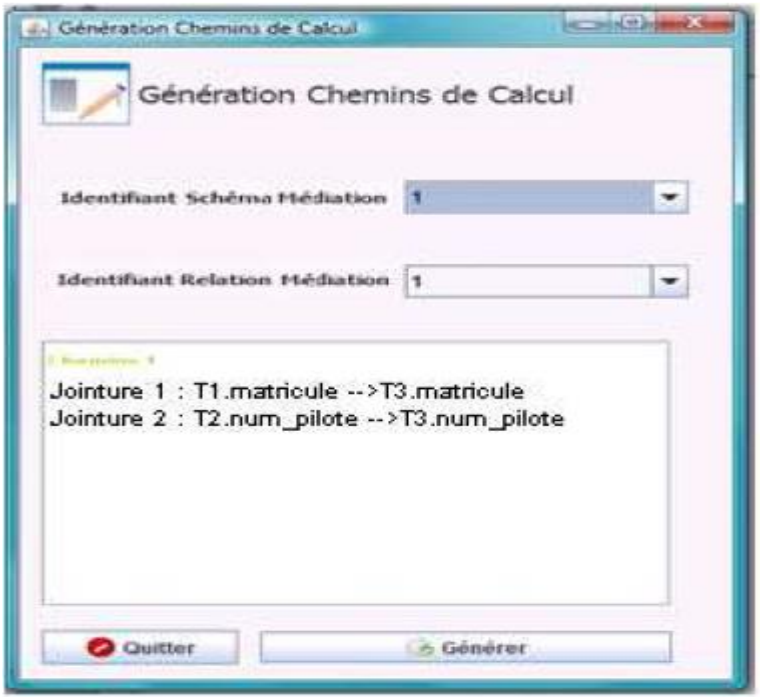


Figure 23. *Fenêtre de génération des chemins de calcul*

3.5.2.5 Fenêtre Génération des requêtes de médiation

Une fois les chemins de calcul générés, le but est de déduire une requête de médiation qui calcule la relation de médiation. Chaque chemin de calcul peut dériver une requête XQuery, comme le montre la figure suivante :

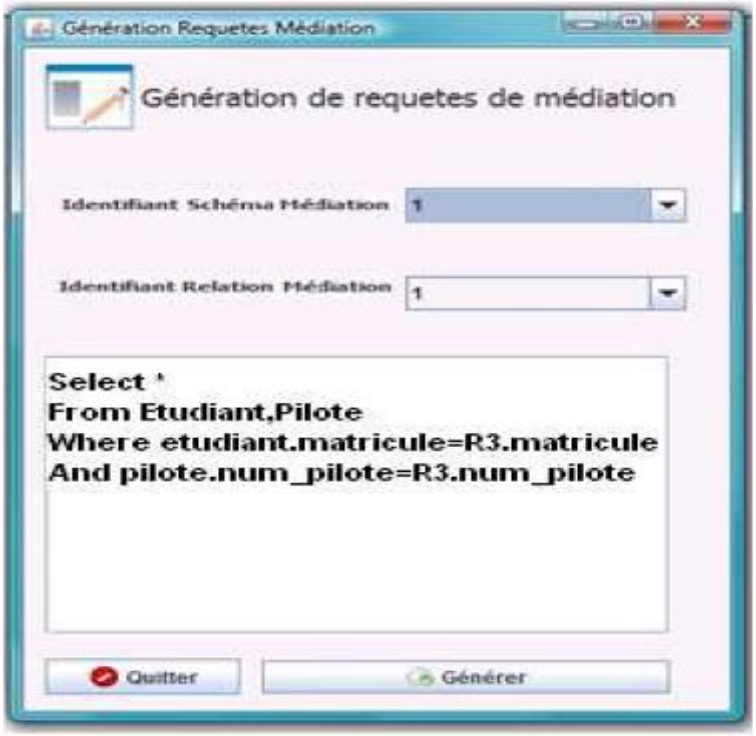


Figure 24. *Fenêtre de génération des requêtes de médiation*

6. Conclusion

Le processus de génération de requêtes de médiation pour le contexte XML tente de trouver toutes les requêtes de médiation possibles au calcul d'une relation du schéma de médiation. Ceci rend le processus de génération automatique ardu, complexe et difficile à maîtriser. Nous avons vu, à travers ce chapitre, qu'il est possible de générer, dans des temps raisonnables, des requêtes de médiation, malgré la difficulté de la tâche.

Pour arriver à ces résultats, nous sommes confrontés avec la complexité et l'indispensabilité des API manipulant les métas donnés en java. Le prototype réalisé a été implémenté en Java utilisant NetBeans IDE7.1.

Cette petite expérience nous a montré la faisabilité de l'automatisation des différentes phases de génération des requêtes de médiation, dans un contexte XML.

Conclusion générale et perspectives

Le problème de définition des requêtes de médiation est un problème complexe en raison de la grande diversité des sources hétérogènes et distribuées qui peuvent intervenir dans un système de médiation et du grand volume de métadonnées qui les décrivent, mais aussi en raison des conflits liés à l'hétérogénéité des données qui peuvent exister entre deux sources de données.

Face à cette problématique, nous avons réalisé, pour le contexte XML, un outil de génération automatique des requêtes de médiation.

Etant donné le schéma d'une relation de médiation, de dépendances fonctionnelles définies sur cette relation, d'assertions de correspondance linguistique existant entre les sources et le schéma de médiation, et d'assertions intra-source et inter-source reliant les relations sources entre elles, engendre la necessité de concevoir un outil qui permettera de produire un ensemble de requêtes potentiel calculant cette relation, tenant en compte des conflits liés à l'hétérogénéité des sources, et s'appuyant sur un ensemble de connaissances regroupées dans une base de méta-connaissances.

La conception du prototype réalisé est modulaire, ce qui lui permet d'être amélioré par de nouvelles fonctionnalités dont :

- Adaptation du système de génération de requêtes de médiation au contexte Objet

- Implémenter un gestionnaire de coûts et de statistiques pour tester les performances de l'outil en vue de son passage à l'échelle.

- Intégrer des ontologies locales au niveau des sources et une ontologie de domaine au niveau du médiateur pour faciliter l'ajout de nouvelles sources . . . etc.

Bibliographie

[BOUZ 03] Bouzahzah Mounira, Mémoire Magister « Gestion de la sémantique dans l'intégration de données hétérogènes. Approche basée sur les logiques de descriptions » , Université Mentouri de Constantine, 2003.

[BOUSSIS 08] Amel Boussis , Intégration de sources de données à base ontologique dans un environnement P2P, Thèse de magistère. L'institut national d'informatique 2008.

[DANG 03] Tuyêt Trâm DANG NGOC "Fédération de données semi–structurées avec XML" Thèse de Doctorat Université de Versailles Saint-Quentin -en-Yvelines, 2003.

[GIRAR 01] R. GIRARDI, "An analysis of the contributions of the agent paradigm for the development of complex systems", In (SCI 2001) and (ISAS 2001), Orlando, Florida. 2001.

[DANIEL 2004] Jacques Le Maitre "XQuery, le langage d'interrogation de données XML", 2002. Daniel K. Schneider - Vivian Synteta " Introduction à XQuery", 2004.

[SOUKANE 05] Assia soukane, « génération automatique des requêtes de médiation dans unenvironnement hétérogène », Thèse de doctorat,université de versailles saint-quentin yvelynes Décembre 2005

[W3C 01] XQuery 1.0 : An XML Query Language, 07 June 2001. W3C Working Draft, http://www.w3.org/TR/2001/WD-xquery-20010607

Glossaire

XML [eXtensible Markup Language**]**

XML est un langage de balises analysable, destiné à une diffusion à grande échelle sur le Web, lisible par l'homme, flexible et adaptable.

HTML [Hyper Text Markup Language**]**

C'est un standard web reconnu par tous les navigateurs, permettant la mise en forme d'un texte.

GAV [Global-as-View**]**

C'est une approche de médiation descendante où chaque objet du schéma global est défini par une requête sur les sources.

LAV [Local as View**]**

C'est approche de médiation ascendante où chaque objet d'une source de données est défini par une requête sur le schéma global.

API [Application Programming Interface]

Interface de programmation d'applications, contenant un ensemble de fonction courantes de bas niveau, bien documentées, permettant de programmer des applications de « Haut Niveau ».

JDK [Java Development Kit]

Environnement de développement de Sun permettant de produire du code Java

Et servant de référence.

Métadonnées [*metadata*]

Les métadonnées sont des données structurées ,standardisées qui décrivent le contenu de document que souhaite partager des utilisateurs .ce sont des données qui renseigne sur la nature de certains autre données et qui permet ainsi leur utilisation pertinente.

Mapping

Correspondance entre le schéma global et schéma source