Approximate XQuery Reformulation Based on GLAV Mapping and Unification

Benharzallah Saber, Sahli Siham
University Mohamed Khider of Biskra
07000, Biskra
Algeria
sbharz@yahoo.fr, infodoc82@yahoo.com

ABSTRACT: *The mediation is based on an essential component called mediator. The main role of the mediator is to reformulate a user query, written in terms of global schema, in queries written in terms of sources schemas. This paper describes an algorithm for reformulation of XQuery queries. The algorithm is based on the principle of logical equivalence, simple and complex unification, to obtain a better reformulation. It takes as parameter the query XQuery, the global schema (written in XMLSchema), mappings GLAV and gives as a result a query written in terms of sources schemas. The results of implementation show the proper functioning of the algorithm.*

Keywords: Data Integration, Mediator, XML, XQuery, GLAV Approach

Received: 11 August 2011, Revised 29 September 2011, Accepted 3 October 2011

1. Introduction

Now the Web is presented as the most favored mean to disseminate information. Many companies and organizations, with any field of activity (e-commerce, education, geographical or historical applications, etc...), make this choice for disseminating information.

The diversity of information sources distributed and their heterogeneity are one of the main difficulties encountered by users of the Web today. It requires the user to respect the access methodology for each data source, this implies to know the location of the base, the description of their content, the possibilities of interrogation, the format of results, in order to receive the expected response [13]. The Mediator-based System offer interesting solutions for the integration of heterogeneous data. For these reasons, the most recent works have taken this approach include the Internet-Oriented Systems [11] [8].

The rest of this paper is organized as follows: Section 2 presents the problems studied, some solutions presented in the literature and the characteristics of our solution. Section 3 describes some concepts used in this paper. Section 4 presents the proposed architecture of our system of mediation and describes the algorithm reformulation. The programming environment and implementation are presented in Section 5. Finally a conclusion and prospects.

2. Problem Studied

The two main problems posed by the construction of a mediator are [2] - The choice of the language used to model the global schema, and the choice of the languages for modeling, according to this schema, the views on the sources to be integrated, and users requests. - And, depending on the choice and implementation of algorithms for query rewriting in terms of views in order to get all the answers to a query.

Studies have focused on the languages for modeling the global schema to represent the views of the sources to integrate and to express requests from human users or computing entities [7] [3]. Others have focused on the design and implementation of algorithms for query rewriting in terms of views on relevant data sources and, more recently, some research focuses on designing intelligent interfaces assisting the user in Query formulation [1][12].

The solution provided by this paper is characterized by the following points:

1. The use of common expressive query language XQuery to express requests from human users or computing entities.

2. The use of the model XML Schema as a common data model to model the global schema, and to represent the views of the sources to integrate.

3. Backward integration Approach and adaptation of mapping rules GLAV.

4. The algorithm reformulation is based on the principle of logical equivalence, simple and complex unification, to obtain a better reformulation.

5. The algorithm takes into account semantic conflicts, resolve them in the mappings rules Glav.

3. Definitions

We describe some concepts used in this paper.

a) Substitution: A substitution of the set of variables X = (x1, x2, ..., xn) is the finite set of the form: (x1/y1, x2/y2, ..., xn / yn) where each yi is a variable different to xi but it has the same type as xi

b) Instance: Let the substitution
θ = (x1/y1, x2/y2, ..., xn / yn) and Q a query. Considering the following queries: Q1, ..., Qi,..., Qn where: Q0 = Q and Q_i is obtained by Q_{i-1} by yi.Q_n is called the instance of Q by the substitution θ, and is denoted by Qθ.

c) logical Equivalences: Two queries Q1, Q2 are called logically equivalent if and only if they give the same results (have the same canonical form) [5].

d) Simple form: A query is in simple form if all the predicates in the *Where* clause are in conjunctive normal form. There is no imbrication in clause *For*.

e) Mapping rules: they are defined for the correspondence between the global and sources shemas.. They also intervene in the reformulation of queries.
The rules are of the form: $R_i : q_g \rightarrow q_s$
Where:

q_g: is an XQuery query relating to elements of the global schema.
q_s: is an XQuery query relating to elements of sources schemas.

4. The Proposed Architecture

4.1 General Architecture of the system

In our solution we adopt as:

◆ Q*uery reformulation approach GLAV*: There are basically two approaches to build the link between a mediated schema and sources schemas. The LAV approach (Local-As-View) and the GAV (Global-As-View) [9] [10]. We chose the Glav approach [3] [4] which presents a combination of both approaches, GlAV offer more flexibility to the updates of users or local sources.

◆ *Data Model*, the XML schema: The model of semi-structured data XML schema has been designed to easily represent irregular data from heterogeneous sources, structured or not. We also note that it is a flexible model used to represent irregular data by mixing structure and data [6]. And that's exactly the model suitable to our case, where data sources are heterogeneous, structured, semi structured or unstructured.

◆ *Query Language, XQuery:* The W3C proposed the XQuery language [14] which is more adopted for an interrogation of an XML schema. The XQuery language incorporates the benefits of XPath, XML-QL and XQL [DANG 03] this language was designed to allow to create specific requests and can be adapted to any type of XML data source, whether databases or documents. For all these reasons and also because we have chosen XML schema as a common model, we have chosen the XQuery language.

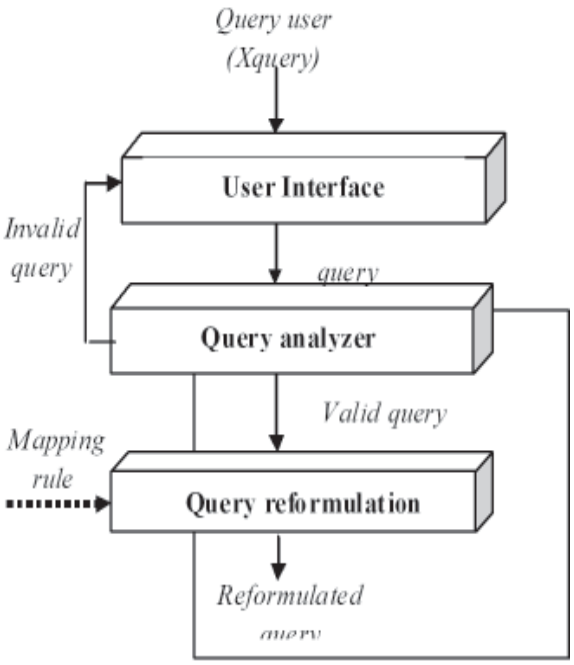


Figure 1. General Architecture of our system

Our mediator is composed of three modules (Figure 1):

1. User Interface: The interface presents the only mean that allows direct interaction between the system and the user.

2. Query analyzer: This analyzer allows a lexical analysis, syntax and semantics on the request to verify its validity.

3. Module of query reformulation: This is the module that performs a series of processing on the request user (written in terms of global schema) in order to reformulate it to a query written in terms of sources schemas.

4.1.1 Query analyzer

Can analyze the request, knowing that it is written in a restriction of the XQuery language. This grammar (Figure 2) is the heart of XQuery [15].

Q: =For x_1 in C_1, …, x_n in C_m
Where B
Return R
R: = [A_1:=R_1,…, A_k = R_k] | E | Q
E: = S | $x | E/L
$\mathbf{C_i}$: = E | Q
OU $x : est une variable
S : est la racine du schéma
L : est une étiquette
E / L : enregistrement de la projection

Figure 2. Grammar of the XQuery language restrictions

The analyzer decomposes the query user into an internal structure that can be easily manipulated by the various components of the mediator. It also checks whether the request is valid, both syntactically and in relation to data types surveyed.

4.1.2 Query reformulation Model

For each relation in the global schema we will define a view consisting of the terms of the relations of source schemas. The reformulation consists itself of two sub components [5]:

a - Simple unification

b - Complex unification

a) Simple unification

Two queries Q1 and Q2, are unifiable if Q1 is an instance of Q2 by the substitution θ, that means that : Q1 = Q2θ, we say that Q1 is logically equivalent to Q2θ.

We adapt the algorithm defined in [5] which allow to verify the unification of two OQL queries. The unification is simply divided into three main stages

- The unification of collections (C_i).
- The unification of predicates.
- The unification of projections (return).

If all goes well, the unification succeeds and returns the substitution θ. The substitution θ is calculated iteratively and we obtain $a\theta$ such that: $Q_1 = Q_2\theta$.

b) Complex unification

If two queries Q_1 and Q_2, are not unifiable by the simple unification, it is possible to verify that if it exist a query Q_3 logically equivalent to Q_2 and it content $Q_1\theta$ as a sub query .

We say that Q_3 is written in terms de $Q_1\theta$ which unify with Q_1 by the substitution θ, and Q_3 is the reformulation of Q_2 by using Q_1. We adapt the algorithm defined in [5] which allow to verify from two queries Q_1 and Q_2, if it's possible to reformulate Q_2 in a query Q_3 containing $Q_1\theta$ such as a sub query.

The complex unification is divided into three main stages:

1. The unification of collections.
2. The unification of predicates.
3. Construction the new query Q_3 and the substitution θ.

The next figure explains the structure of the component of reformulation.

We describe in our solution the decomposition process of a query Q written in global schema into a recomposition query and sub-queries. Each sub-query q_i is written in source schema S_{s_i} .

Our process of reformulation will be in four stages: transform the query Q into a more simple form to process, reformulation, identification of sources involved in the execution of the request and the generation of sub-queries.

1. The transformation of the request is to write it in the canonical form or approximate to the canonical form.

2. The reformulation query Q (figure algorithm reformulation): our algorithm consists to reformulate a query Q (using mapping rules M) into a query logically equivalent to Q and written in terms (s) $q_{g_i}\theta$. This stage prepare to the stage of identification of information sources participant to the execution of query Q . There are three cases :

The case which exist a rule , r_i: $q_{g_i} \rightarrow q_{s_i}$ *as:* $Q = q_{g_i}\theta$,

The case where Q is in terms of $q_{g_i}\theta$, and

Algorithm Reformulation

input: Q (written in XQuery), M // where

M = $\{r_1, r_2, ..., r_i, ... r_n\}$ and $r_i : q_{g_i} \rightarrow q_{s_i}$

output : S

{

S = $\{Q\}$, R = $\{Q\}$ /* Q is in simple form */

while ($R \neq \phi$) do {

$\lambda = \phi$;

For each $q \in R$ *and* $r_i \in M$ {

Verify **Unification Simple**(q, q_{g_i}),

if successful with the substitution θ **then** : replace q by $q_{g_i}\theta$ (the

Replacement is done by the header ($q_{g_i}\theta$)

Else : Verify **UnificationComplexe**(q_{g_i}, q)

If successful with the substitution θ and the query q' then

- replace q by q' (as q' contains $q_{g_i}\theta$ like sub query).

– Add the result to λ

$R = \lambda - S$; $S = S \cup R$

}

If exist in S a query of the form $q_{g_i}\theta$ or in the form:

For x_j in $q_{g_i}\theta_i$,..., x_t in $q_{g_{tt}}\theta_t$

Where p_h ***and*** p_j ***and*** ... p_y

Return *proj*

So keep this request and eliminate the others and out of the loop.

}

For each $r_i \in M$ {If q_{g_i} appear in S then replace q_{g_i} by q_{s_i} in S }

Return S

}

The case where there doesn't exist a reformulation of the query because of the lack of mapping rules.In this case the algorithm gives failure.

We propose that the mapping rules follow an order of priority to ensure proper reformulation and also allow to take into account the constraints on the sources that are defined in the mapping rules. So the algorithm should avoid shorts reformulations.

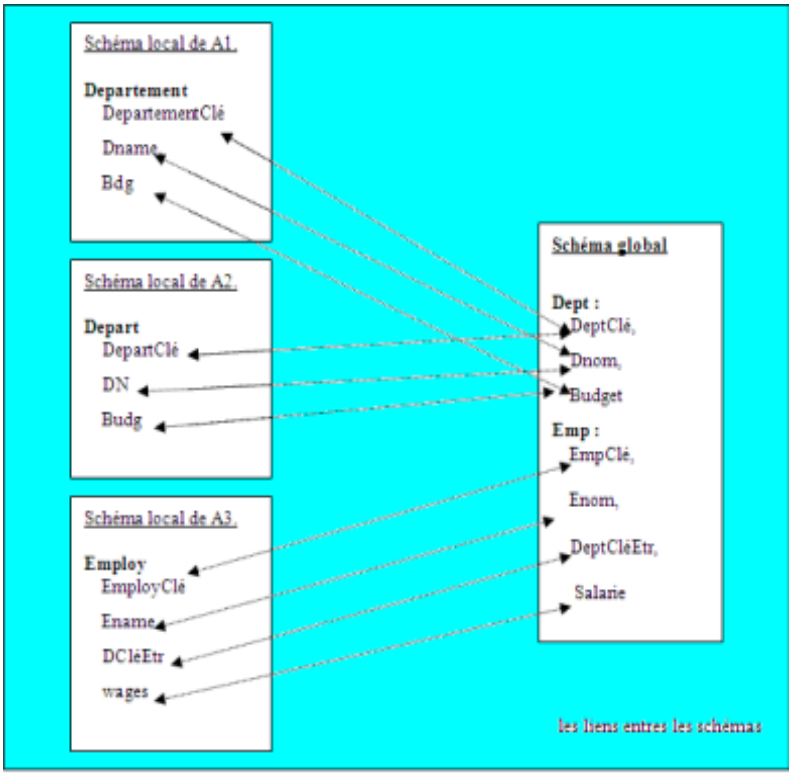
5. Programming Environment and Results of Implementation

We have implemented our prototype using the environment C++ Builder. Among the different categories of applications of mediation systems include applications of information retrieval on the Web, those of decision support online, more generally, knowledge management in the broad sense [13]. We present the following case study to demonstrate the operation of the algorithm.

We have the global schema and source schemas S1, S2 et S3 represent databases «Département, Employeurs».

Global Schema:
Dept(DeptClé, Dnom, Budget) ;
Emp(EmpClé, Enom, DeptCléEtr, Salarie) ;
The Global Schema is written in XML Schema and interrogated by XQuery.
Local schema of the agent A1:
Departement(DepartementClé, Dname, Bdg) ;
Local schema of the agent A2 :
Depart(DepartClé, DN, Budg) ;
Local schema of the agent A3 :
Employ(EmployClé, Ename, DCléEtr, wages);

The following figure shows the relation between different schemas.

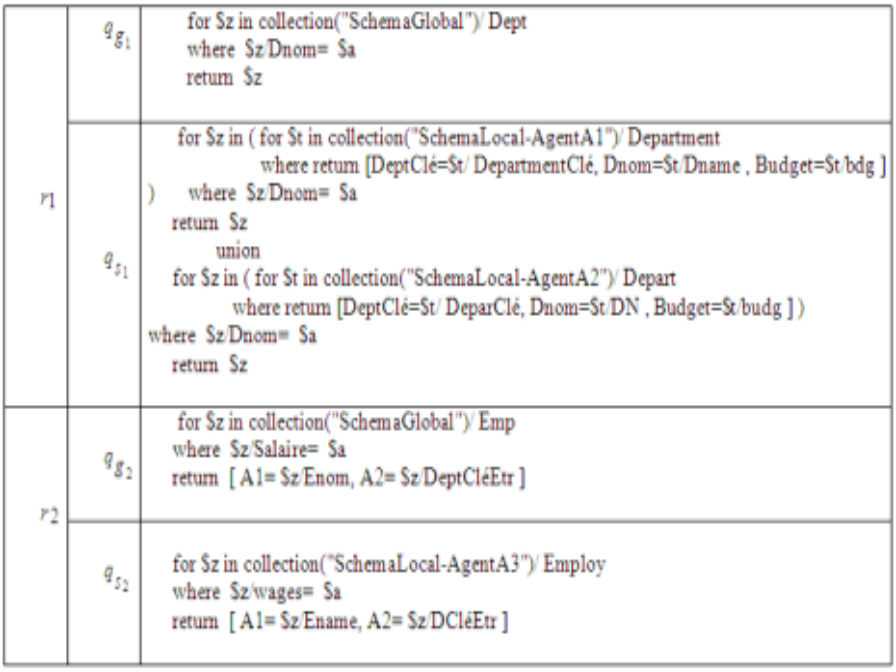


and the subset of rules mapping GLAV consisting of two rules :

Either a user query Q:
Q = for $x in collection (“SchemaGlobal”)/Dept, $y in collection(“SchemaGlobal”)/Emp
where $x/DeptClé= $y/DeptCléEtr and $x/Dnom= “département 1” and
$y/Salaire= “20000,00 DA”
return [Nom = $y/Enom]

To decompose Q into sub-queries, we apply the following steps:
simplification of the application: Q is in a simple form.
Reformulation: we apply algorithm reformulation. The query is reformulated using q_{g_1}:

Q = for $x0 in q_{g_1} θ, $y in collection (“SchemaGlobal”)/Emp
where $x0/DeptClé= $y/DeptCléEtr and $y/Salaire = “20000,00 DA”
return [Nom = $y/Enom]



where: θ = {$z / $x , $a/" *department*1"}

Then the query is reformulated using q_{g_2}

Q = for $x 0 in $q_{g_1}\theta$, $x 01 in $q_{g_2}\theta$'
where $x 0 / DeptClé = $x 01/A2
return [Nom = $x 01 / A1]

where : θ = {$z / $x , $a / "20000,00 *DA*"}

Thus the query is reformulated. Q is written in terms of $q_{g_i}\theta_i$

The identification of sources involved in the execution of the request Q :

- q_{g_1} corresponds q_{s_1} (so the sources A1 and A2 are involved in the execution of Q) and
- q_{g_2} corresponds q_{s_2} (so the source A3 participate in the execution of Q).

The mediator has the recomposition query Q.

The source A1 translated its sub-query in the local language associated with its local source.The subquery of A1 is (after application of the substitution θ = {$z / $x , $a/ "*department*1"}) :

for $x in (for $t in collection (" SchemaLocal-AgentA1 ")/ Department
where return [DeptClé = $t/ DepartmentClé, Dnom=$t / Dname , Budget = $t/bdg])
where $x/Dnom= "département1"
return $x.

The sub-query to the source A2 is:

for $x in (for $t in collection(" SchemaLocal-AgentA2 ")/ Depart
where return [DeptClé = $t/ DeparClé, Dnom=$t / DN , Budget=$t / budg])
where $x/Dnom= " département1 "
return $x .

The sub-query to the source is A3:

for $x in collection(" SchemaLocal-AgentA3 ")/ Employ
where $x / wages= " 20000,00*DA* "
return [A1= $x / Ename, A2= $x / DCléEtr]

Figure 5. The result of compilation

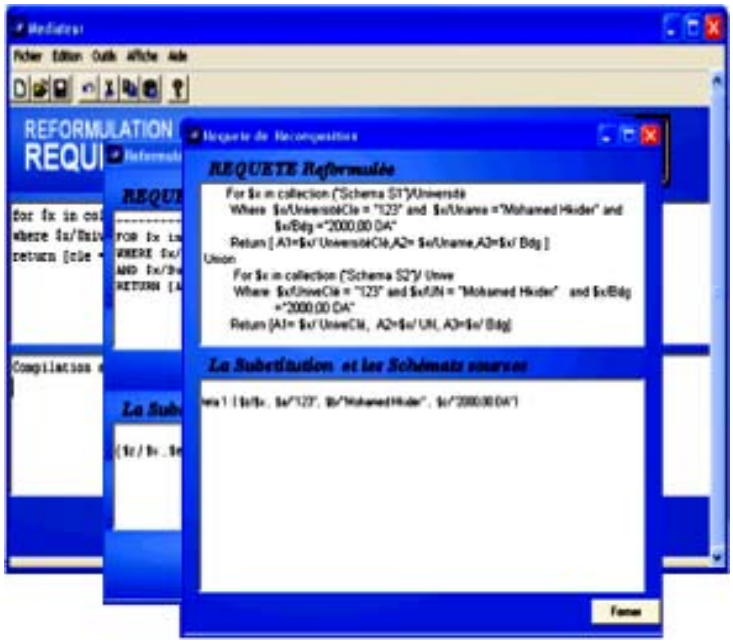


Figure 6. The result of reformulation

Figures 5 and 6 present the results of our system: The Figure 5 presents the compilation of the application user who's been passed successfully and the figure6 presents the query reformulated in terms of source schemas.

6. Conclusion

A mediation system is a powerful mean allowing an easy access to various information collected from data sources can be quite disparate. It must integrate diverse data in order to provide to the user a centralized and uniform view of data by hiding the features specific to their location, access method and formats. We presented in this paper an algorithm reformulation of Xquery

queries for mediation systems using GLAV mappins and unification. The implementation of the prototype mediation system, illustrates the operation of the algorithm. As a prospect our mediation system will be improved by taking into account the following points: The use of all possibilities of Xquery language, Building a module that lets to add or remove rules mapping, building adapters to resolve structural conflicts of heterogeneous sources (XML schema model, relational model ...).

References

[1] Charlet, J., Laublet, P., Reynaud, C. (2003). Rapport final «Web sémantique», Action spécifique 32 CNRS / STIC.

[2] Rousset, C., Bidault, Froidevaux, C., Gagliardi, H., Goasdoué, P., Reynaud, R., et safar, B. (2002). « Construction de Médiateurs pour intégré des sources d'information multiples et hétérogènes le projet PICSEL » Université Paris_Sud, France.

[3] Reynaud, C., Safar , B., (2008). «Construction automatique d'adaptateurs guidée par une ontologie pour l'intégration de sources et de données XML» . Univ. Paris-Sud, juin.

[4] Djema, L., Boumghar, F., Debiane, J. (2007). « L'imagerie Médicale Dans une Base De Données Distribuée Multimédia Sous Oracle 9i». Dépt. Informatique Université Tizi-ouzou , Algérie.SETIT 2007 4th International Conference: Sciences of Electronic, Technologies of Information and Telecommunications March, 25-29.

[5] Daniela Florescu : Thèse doctorat, (1996). «Espaces de Recherche pour l'Optimisation de Requêtes Objet » l'Université de Paris VI.

[6] Goerges Gardain. (2008). «XML Des bases de données aux services Web» Paris.

[7] François Goasdoue F., Lattes, S V. , Rousset, M.-CH *International Journal of Cooperative Information Systems*: « The use of the Carin language and algorithms for Integration Information: the PICSEL system »

[8] Elazami, I., Doukkali, D., Cherkaoui, O. (2007). «Approche à base de Patterns pour la Médiation entre les Systèmes d'Information Hospitaliers». Département de Mathématiques et Informatique Faculté des Sciences Dhar EL Mahraz Fès, Maroc. FMP de Fès, le 02.

[9] Maurizio Lenzerini.(2001). « Data integration is harder than you thought », Dipartimento di Informatica e Sistemistica Universit'a di Roma "La Sapienza", CoopIS 2001_Trento, Italy.

[10] Alon Levy, Y. (2002). « logic-based techniques in data integration » , Département of computer Science and Engineering University of Washington.

[11] Mr. Moussa LO : Thèse DOCTORAT. (2002). «Dataweb bases sur XML modélisationet recherche d'information pertinentes » L'université de pau et des pays de l'adour.

[12] Maiz, N., Boussaid, O., et Bentayeb, F. (2006). «Un système de médiation basé sur les ontologies» .Laboratoire ERIC Université Lumière Lyon 2. 17 Janvier 2006 Lille, France.

[13] Hacid, M, S., Reynaud, C. (1918). Thèse doctorat«L'intégration de sources de données » l'Université Claude Bernard Lyon 1.

[14] XQuery 1.0. (2007). An XML Query Language, W3C working Draft.

[15] Cong Yu, Lucian Popa, (2004). Constraint based XML Query Rewriting for data integration, SICMOD 2004, June, Paris, France.