Mediation Queries Adaptation After the Removal of a Data Source

Latifa Baba-Hamed
Computer Science Department, University of Oran Es-sénia, Algeria

Abstract: *A broad variety of data is available in distinct heterogeneous sources, stored under different formats: database formats (in relational and object-oriented models), document formats (SGML/XML), browser formats (HTML), message formats, etc. The integration of such data is increasingly important for modern information systems applications such as data warehousing, data mining, and web applications. This is realized by providing a uniform view of data sources (called mediation schema or global schema) and defining a set of queries (called mediation queries or mediation mappings) which define objects of the mediation schema. One of the important problems that merit consideration is the impact of schema evolution on mediation queries. Mappings left inconsistent by a schema change have to be detected and updated. In particular, one source may be removed from the system because it provides always obsolete information or because it is unavailable. In this case it is necessary to update the inconsistent mappings. In this paper, we study the removal of a source from an integration system and show how to correctly update the mappings between the mediation schema and the distributed sources after this change, in the context of the global-as-view approach (each relation of the global schema is expressed as a view on the data source).*

Keywords: *Mediation system, schema evolution, mediation queries, relevant relation, operation graph.*

Received November 8, 2008; accepted May 17, 2009

1. Introduction

Many applications require the use of existing data stored in multiple distributed and possibly heterogeneous sources. Considering that the application needs are represented by a target schema, mappings have to be defined to express the way instances of a target schema are derived from the instances of the source schemas. Such mappings can be used in different contexts such as mediation systems [26] and data warehouses [24]. These systems are also called data integration systems since they all integrate data sources into a global representation. An integration system provides a unique access point to a set of data sources. It includes a mediator (a global system of management) and a set of local sources. The mediator supports a global schema (called also mediation schema) which is composed of local schemas completely or partially. Global queries (called also mediation queries) are posed over the global schema and treated by the mediator. Many works concerning the data integration have been developed. Some of them concern data cleaning (a state of the art on data cleaning is given in [23]). Some research approaches [5, 7, 10, 13, 16, 21, 29] have been proposed for automatically or semi-automatically generate mappings. Some approaches [4, 12, 15, 25, 28, 27] have been proposed to adapt mappings automatically when the related schemas evolve. Some works are based on data quality [1, 20] or on schema quality [18]. Some researches focus on the definition of semantic correspondences between two schemas (also called schema matching) [9, 22]. We can distinguish three kinds of approaches in data integration to define the mediation queries: the Global-As-View (GAV) approach, in which each relation of the global schema is expressed as a view on the data source (it is used in TSIMMIS [6]), the Local-As-View (LAV) approach, in which each relation in a given source is defined as view on the global schema (it is used in Information Manifold [11]), and GLAV approach which combines the benefits of LAV and GAV (it is used in Xyleme).

This paper addresses the problem of evolution in the context of the GAV approach. It studies, more precisely, the removal of a source from a mediation system using the methodology given in [3] and improved in [5] by handling the heterogeneity of data for generating mediation queries. In addition it shows how to update the mappings left inconsistent by this change. A source may be removed from the system because it provides always obsolete information or because it is unavailable.

This paper is organized as follows. Section 2 is consecrated to the related works. Section 3 presents the methodology used for generating mediation queries. Section 4 proposes an algorithm which shows how to propagate the removal of a source to the mediation level using this methodology and describes the metadata used to execute the change operations.

2. Related Works

Schema evolution is a broad research area that includes problems related to schema changes. It has been studied in different contexts and under different assumptions. In Object-Oriented DataBase Management Systems (OODBMS), Banerjee *et al*. [2] have defined a taxonomy of the changes that may occur in OODBMS and provided an implementation for each one of them. Incremental view maintenance [19] is a problem which deals with the methods for efficiently updating materialized views when the base schema data are updated. View adaptation [8, 17] is a variant of view maintenance that investigates methods of keeping the data in a materialized view up-to-date in response to changes in the view definition itself. View adaptation may be required after mapping adaptation.

In data integration systems, several solutions have been proposed for automatic mapping adaptation, we present them in what follows. In automed [15], schema evolution and integration are combined in one unified framework. Source schemas are integrated into a global schema by applying a sequence of primitive transformations to them. The same set of primitive transformations can be used to specify the evolution of a source schema into a new schema. The authors have shown how the transformations between the source schemas and the global schema can be used to systematically repair the global schema and the query translation pathways as source schemas evolve. They consider in particular the evolution of a source schema into a semantically equivalent, semantically contracted, or semantically expanded schema. The schemas are defined in a Hypergraph Data Model (HDM) which is a triple <Nodes, Edges, Constraints>. A query over a schema is an expression whose variables belong to Nodes∪Edges. It is expressed in a first order query language. AutoMed also considers the changes of the global schema. Adapting mappings for target schema changes is similar to the one for source changes. For the target changes, only transformation pathways will be adapted and no source schema will be changed for the target evolution.

Bouzeghoub *et al*. [4] have addressed the problem of evolution in the context of the GAV approach. They use a methodology that they have defined in [3, 5] to generate mediation queries in mediation systems based on the relational model. Given a mediation relation, a set of source schemas and a set of linguistic assertions between the mediation schema and the sources schemas, the authors have defined an algorithm which discovers the mediation queries defining this relation. The evolution process is seen as an incremental execution of this algorithm. Their solution is based on the concept of relevant relations on which propagation rules have been defined. Every evolution rule is an event-condition-action rule in which the event is a change and the action is a set of propagation primitives to execute when the conditions are satisfied. The authors have limited their study to only some changes. They have not considered the removal or the addition of a data source to a mediation system. Loscios and Salgado [14] use the approach of Bouzeghoub [4] to evolve mapping generated by Loscios [13]. Their approach is proposed in the context of mediation systems in which mediation schemas and source schemas are all expressed through an XML schema. Xue [27] proposes an incremental approach for mapping adaptation. She addresses the problem of automatic mapping generation and adaptation for XML schemas. She focuses on generating mappings from multiple source schemas and mapping can express inter-source joins. She does not assume any homogeneity between the target schema and the source schemas and she generates mappings in an abstract language and can be later translated in another more declarative language as XQuey. She adapts mappings expressed in XQuery when the target schema or the source schema evolves. In this approach, the mapping adaptation does not depend on a mapping generation process and does not require any extra metadata besides schemas and correspondences.

The study in [12] is one of the first works to introduce the problem of schema changes of Information Sources (ISs). The authors address the problem of view definition adaptation in dynamic environments; they call it view synchronization problem. To solve this problem, the authors develop the Evolvable View Environment (EVE) framework and propose an extended view definition language, called E-SQL, which is capable to define flexible views by incorporating view change preferences into the view definition. In addition, they introduce a Model for Information Source Description (MISD) which allows a large class of ISs to participate in their system dynamically, develop replacement strategies for affected view components which are designed to meet the preferences expressed by E-SQL, and provide a set of view synchronization algorithms based on those strategies. The proposed algorithms generate view definitions as output that are consistent with both the change semantics expressed by E-SQL as well as the MISD descriptions captured in an available meta knowledge base. Similarly to the approach in [4] they use the relational data model as the common data model and they also propose an extended view definition language (derived from SQL).

The approach in [4] differs from the EVE approach, because the modifications are not directly executed in the mediation query definition but in the metadata that describes the mediation query. Bouzeghoub et al. considers that a mediation query must be modified as a consequence of every kind of source schema change. The Clio project [21] have proposed an approach to generate mappings between schemas that are in either relational or XML model. Mappings are generated between one source schema and one target schema. The approach in [25] complements the above scenario. Velegrakis et al. take the mappings generated by a mapping tool or defined by a user and adapt them when schemas are changed, in order to preserve the mapping consistency. The authors consider changes not only to the structure of schemas (which may make the mapping syntactically incorrect) but also to the schema semantics (i.e. schema constraints) either in the source or in the target. They support changes not only on atomic elements, but also on more complex structures including relational tables or complex (nested) XML structures. They present a mapping adaptation algorithm that detects mappings affected by a structural or constraint change and generates all the rewritings that are consistent with the semantics of the mapped schemas. To evaluate the effectiveness and usefulness of their approach, they have implemented a prototype tool called ToMAS.

Yu and Popa [28] develop a tool for automatically adapting mappings generated by Clio'02 [21]. Consider three schemas S_1, S_2, and S_3, a mapping m_{12} between S_1 and S_2 and another mapping m_{23} between S_2 and S_3. This approach consists in composing m_{12} and m_{23} to get the possible mappings between S_1 and S_3. Composing two mappings m_{12} and m_{23} is performed in three steps:

a. Create a set of rules, from m_{12}, to express how the elements of S_2 are expressed using elements of S_1.
b. Use the rules to modify m_{23} by translating all references to S_2 into references to S_1 resulting in a set of mapping M_{13}.
c. Check the mappings in M_{13} to see if they are valid. For reducing the number of combinations, the authors present a method that removes all the original mappings that are not affected, and redundant mappings.

3. Approach of Mediation Query Generation

To study the removal of a source from a mediation system, we have chosen the approach given in [3, 5] for generating mediation queries. In this section, we firstly recall the main principle of this approach, and secondly we give an illustrative example.

3.1. Principle of MQG Approach

This approach has been proposed in the context of mediation systems. The mediator approach consists in defining an interface between the users who submit queries and the set of relevant sources which provide the answers. A mediation system is defined by a mediation schema (global schema) and a set of queries (mediation queries) which define objects of the mediation schema over the distributed data sources. This approach considers that both mediation and source schemas are relational schemas and mediation schemas are defined by experts independently from the sources. Mediation queries are generated for every relation of the mediation schema. The generation process follows three steps:

a. Searching for the sources that are relevant to the mediation relation.
b. Determination of candidate operations.
c. Defining mediation queries.

The step of searching for the relevant sources consists in finding all source relations that can contribute to the computation of the mediation relation. A source relation S_{ij} contributes to the computation of a mediation relation R_m if S_{ij} includes some attributes of R_m. In this case, a *mapping relation* is extracted from it; the mapping relation contains all the common attributes between the mediation relation and S_{ij}. The primary key and foreign keys of S_{ij} are added into the mapping relation. The step of candidate operation identification searches for possible joins between mapping relations. A join is possible between two mapping relations R_1 and R_2 either if (1) R_1 and R_2 are in the same source and there is an explicit referential constraint between them, or if (2) R_1 and R_2 are in different sources and the primary key of one has an equivalent attribute in the other. There is not always a candidate operation between two mapping relations following the previous rule. However, they might be joined through a third relation. Some relations that contain only primary keys and foreign keys and that have no common attribute with the mediation relation are considered by the algorithm to make possible joins between mapping relations; they are called *transition relation*. Both mapping relations and transition relations are called *relevant relations*. Relevant relations and the joins between them can be represented by a graph (called *operation graph*) in which every node is a relevant relation and every edge is a join. Over the operation graph, a mediation query is defined from a *computation path*, which is a connected, acyclic sub-graph that involves all the attributes of the mediation relation. Defining mediation queries consists in enumerating all the computation paths from the operation graph. Set-based operation such union, difference, intersection can be used over existing mediation queries to derive new mediation queries.

3.2. Illustrative Example

Consider the two mediation relations R_1, R_2 and the set of data sources S = {S_1, S_2, S_3, S_4, S_5, S_6}. The relation schemas of the mediation relations R_1, R_2 and the source relations corresponding of each source S_i in S are defined in Table 1. Primary key attributes are prefixed by # and foreign key attributes are prefixed by @. We illustrate in Table 2 the relevant relations (mapping relations and transition relations) derived from the source relations given in Table 1 for the two mediation relations R_1 and R_2.

Table 1.The schemas of relations at the mediation and source level.

Level	Relations	Schemas
Mediation	R1	R1(#K, A, B, C)
	R2	R2(#K', D, E, F)
S1	S11	S11(#K, A, @X, R')
	S12	S12(#X, B, @Y, T)
S2	S21	S21(#Y, C, @W, U)
	S22	S22(#K', D, E, @P)
S3	S31	S31(#X, C, V)
	S32	S32(#R, V, O, @W)
S4	S41	S41(#X, C, J)
	S42	S42(#Z, C, L)
	S43	S43(#K', D, E, L, @P)
S5	S51	S51(#W, @Z, F)
	S52	S52(#P, N, @R)
S6	S61	S61(#K1, A1, B1, @D1)
	S62	S62(#D1, C1, E1)

Table 2.The relevant relations of R_1 and R_2.

Source relations	Relevant relations of R1	Relevant relations of R2
S11(#K, A, @X, R')	T11(#K, A, @X)	
S12(#X, B, @Y, T)	T12(#X, B, @Y)	
S21(#Y, C, @W, U)	T21(#Y, C, @W)	
S22(#K', D, E, @P)		T22(#K', D, E, @P)
S31(#X, C, V)	T31(#X, C)	
S32(#R, V, O, @W)		T32(#R, @W)
S41(#X, C, J)	T41(#X, C)	
S42(#Z, C, L)	T42(#Z, C)	
S43(#K', D, E, L, @P)		T43(#K', D, E, @P)
S51(#W, @Z, F)	T51(#W, @Z)	T51(#W, F)
S52(#P, N, @R)		T52(#P, @R)
S61(#K1, A1, B1, @D1)		
S62(#D1, C1, E1)		

T_{21} (#Y, C, @W) and T_{42} (#Z, C) (in the second column of the table of Table 2) are two examples of mapping relations corresponding to the mediation relation R1 and derived from the source relations S_{21} (#Y, C, @W, U) and S_{42} (#Z, C, L), respectively. We can see that there is no direct join (i.e., there is not an explicit referential constraint between them and the primary key of one has not an equivalent attribute in the other) between them. One possible way to join T_{21} and T_{42} is to use the source relation S_{51}(#W, @Z, F) (note that none of W, Z and F is in the mediation relation R1): a join between T_{21} and S_{51} with the predicate over the attribute W and a join between S_{51} and T_{42} with the predicate over the attribute Z. A transition relation is generated from S_{51}; it contains only primary keys and foreign keys: T_{51}(#W, @Z). The join is possible between T_{11} and T_{12} because there is a referential constraint from T_{11} to T_{12} through the attribute X. The join is possible between T_{11} and T_{31} because the attribute X exists in both T_{11} and T_{31} and X is defined to be a key in T_{31}. We proceed in the same way to find all the relational operations which permit to combine each pair of relevant relations. This step leads to the determination of the operation graph of a given mediation relation R_m denoted by G_{Rm}. Figures 1 and 2 present the operation graphs G_{R1} and G_{R2} corresponding to the mediation relations R_1 and R_2, respectively.

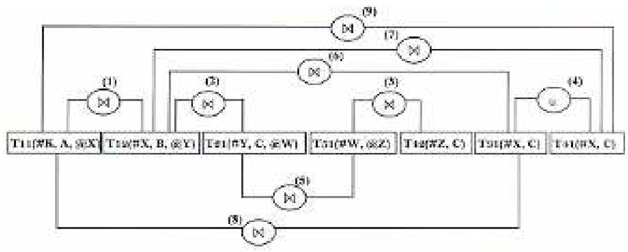

Figure 1. The operation graph G_{R1}.

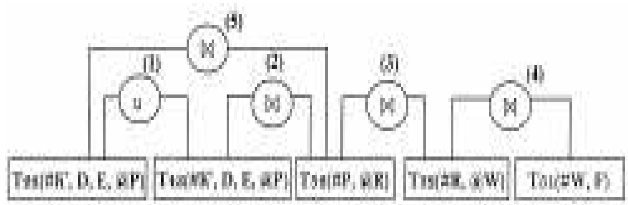

Figure 2. The operation graph G_{R2}.

Over the operation graph, a mediation query is defined from a *computation path*. Examples of computation paths in the operation graph given in Figure 2 are C_1= (1, 2, 3, 4), C_2= (5, 3, 4) and C_3= (2, 3, 4). We can define, for example, the corresponding mediation queries from C_1 and C_2. They are, respectively:

$E_1= (T_{22} \cup T_{43}) \bowtie T_{52} \bowtie T_{32} \bowtie T_{51}$

and $E_2= T_{22} \bowtie T_{52} \bowtie T_{32} \bowtie T_{51}$.

4. The Removal of A Source from A Mediation System

In mediation systems, the evolution problem is mainly related to changes raised at the data source level: adding or removing a relation schema, an attribute, a constraint or a data source. The mediation schema itself is supposed to be not subject to intensive changes. In this section we show how to remove a data source from a mediation system in the context of the GAV approach. This work is considered as the continuation of the approach [3, 5], presented in Section 3 to propose a design methodology to generate mediation queries based on the relational model. It assumes that the mediation query generation system maintains all intermediate results of the generation process that are operation graphs (relevant relations and candidate operations) and the computation paths. It assumes also that there is no equivalent source for the removed source, so it deletes the mediation relation which can not be computed. For this change (i.e. the deletion of a source), our algorithm updates step-by-step the affected parts of each intermediate result. The mappings left inconsistent by this change must be updated in order to provide correct answers to users' queries. Not all the source removals affect the mapping. For example, a mapping will not be affected by the removal of a source that is not related to the mapping (it is the case of the removal of the source S_6 which does not affect the mappings corresponding to R_1 and R_2 because both source relations S_{61} and S_{62} do not contribute to the computation of the mediation relations $R_1$and R_2). In this section, we first describe the metabase over which we perform our change and then present the general algorithm of the source removal.

4.1. The Metabase Description

Among the set of meta-data used by the MQG process, some of them are predefined by the mediation system designer such that: the description of the relations schemas, the relations keys, the functional dependencies, the referential constraints, and others are added to the knowledge base during the process such that: the linguistic correspondences between the sources concepts and the mediation schema concepts, and the linguistic correspondences between the different sources relations. We use these metadata in our evolution system, so we try to describe it in what follows. We can distinguish three levels in our metabase: local (which describes the sources), global (which describes the mediation schema) and intermediate (which describes the relevant relations, the semantic correspondences, and the operations graphs). The metabase is composed by a set of tables (relations). In the local level we have defined four relations with the following schemas:

Source (id-source, source-name)
Source-relation (id-src-rel, rel-src-name, id-source)
Src-attrib (id-src-att, att-src-name, id-src-rel, att-type)
Ref-Constraint (id-cons, id-src-att1, id-src-att2)

The table *Source* includes all the sources of the system. The table *Source-relation* regroups all the relations of the sources. The table *Src-attrib* keeps all the attributes of the sources. The table *Ref-Constraint* saves all the reference constraints of sources. At the global level, we define two tables:

Mediation-relation (id-med-rel, rel-med-name)
Med-att (id-med-att, att-med-name, id-med-rel, att-type)

The table *Mediation-relation* regroups all the relations of the mediation schema. The table *Med-att* includes all the attributes of the mediation schema. At the intermediate level, we distinguish five tables:

M-Relation (id-rel-map, id-src-rel, id-med-rel)
T-Relation (id-rel-trans, id-rel-map1, id-rel-map2,id-src-rel)
Correspond-S-M (id-corresp-S-M, id-src-att, id-med-att)
Correspond-S-S (id-corresp-S-S, id-src-att1, id-src-att2)
Operation (id-Op, type-op, rel1, rel2, id-med-rel)

The tables *M-Relation* and *T-Relation* regroup, respectively, the mapping relations and the transition relations derived from all the sources. The tables *Correspond-S-M* and *Correspond-S-S* include respectively the linguistic correspondences between the sources concepts and the mediation schema concepts, and the linguistic correspondences between the different sources relations. The table *Operation* provides information concerning the operations graphs.

4.2. The Removal Algorithm

This algorithm (as shown Figure 3) specifies the modifications that are performed in the local level and that must be propagated to the mediation level. The removal of a source S_i consists in the deletion of all its source relations. The deletion of a source relation S_{ij} leads to the deletion of all its constraints and all its attributes. The removal of an attribute implies the removal of all the linguistic correspondences involving it. To reflect the deletion of the local relation S_{ij}, the corresponding relevant relation T_{ij} (i.e. mapping and transition relations) must be removed from the operation graph, along with all the operations involving T_{ij}. Our algorithm includes some modules which we describe in what follows. The module *Update-corresp-s-s* deletes the linguistic correspondences between two sources to which the attribute B belongs, from the correspondences set *corresp-s-s*. It updates the table *Correspond-S-S*. The role of the module *Update-corresp-s-m* is to delete the linguistic correspondences between the sources concepts and the mediation concepts to which the attribute B belongs, from the correspondences set *corresp-s-m*. It updates the table *Correspond-S-M*. The module *Update-ref-constraint* removes, from the reference constraints set *ref-cons*, the constraints to which the attribute B belongs. It needs the table *Ref-Constraint*.

```
Remove-source (Si, S)
Si: is the source to be removed
S: the set of the sources
Sij: the schema of the relation source Sij
 If Si ∈ S then
   For each source relation Sij
      For each mediation relation Rm
        For each attribute B of Sij
          Update-corresp-s-s (corresp-s-s, B);
          Update-corresp-s-m (corresp-s-m, B);
          Update-ref-constraint (ref-cons, B);
          Sij = Sij - {B}; // remove B from Sij
        EndFor
        Update-relevant-rel (Sij, M, TM, OP);
      EndFor
    Si = Si - Sij; // remove Sij from the source Si
   EndFor
   For each affected mediation relation Rm
     Search-computation-path (GRm, CP);
      // GRm is composed by M, TM and OP
     // CP is the set of computation paths of Rm
     If CP ≠ ∅ then
        Generate-query (CP, Q);
        // Q is the set of queries to compute Rm
     EndIf
```

EndFor
S= S-{S_i}; // remove the source S_i from S
EndIf
End Remove-source

Figure 3. The source removal algorithm.

The module *Update-relevant-rel* which is specified in Figure 4, updates the set of relevant relations (mapping relations and transition relations) associated with relation R_m in the mediation schema and updates the operations graph G_{Rm}. It uses the tables *M-Relation*, *T-Relation* and *Operation*. Using G_{Rm} (i.e., the table *Operation*), the module *Search-computation-path* searches the computation paths set *CP* corresponding to R_m. It may occur that no computation path can be found after the propagation (i.e., CP=∅); in this case, R_m becomes not computable. The module *Generate-query* generates the queries set Q corresponding to the set CP only when CP≠ ∅. The deletion of B, S_{ij} and S_i are performed respectively in the tables *Src-attrib*, *Source-relation* and *Source*.

Update-relevant-rel (S_{ij}, M, TM, OP)
M: the set of mapping relations in R_m
TM: the set of transition relations in R_m
OP: the set of relational operations in G_{Rm}
S_{ij}: a relation source in the source S_i
T_{ij}: a relevant relation
If $T_{ij} \in M$ such that $\underline{T_{ij}} \subseteq \underline{S_{ij}}$
Then M=M-{T_{ij}};
Else
If $T_{ij} \in TM$ such that $\underline{T_{ij}} \subseteq \underline{S_{ij}}$
Then TM=TM-{T_{ij}};
EndIf
EndIf
For each operation op involving T_{ij}
OP=OP-{op};
EndFor
End Update-relevant-rel

Figure 4. The relevant relations removal algorithm.

The removal of the source S_1 from the system considered in the example presented in section 3.2, affects only the operation graph G_{R1} presented in Figure 1, G_{R2} remains unchanged. This change leads to the deletion of the relevant relations T_{11} and T_{12} from G_{R1} and all the operations involving them. Figure 5 shows the graph G_{R1} after the removal operation. It results that no computation path can be found after the propagation which implies that the mediation relation R_1 becomes no computable.

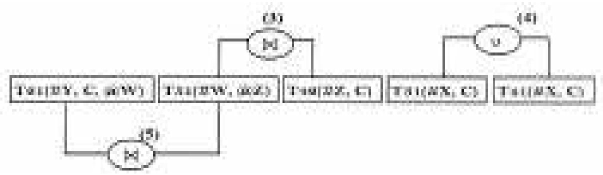

Figure 5.The graph G_{R1} after the deletion of the source S1.

5. Conclusions

In this paper, we have presented the source removal operation from a mediation system in the context of the GAV approach and showed the impact of this change on the mediation level. To perform this operation, we have followed a well-defined methodology which is based on the concept of the operations graph which contains relevant relations, candidate operations and computation paths. The algorithm that we have developed here consists in updating only parts of this operations graph which are dependent of the source removal operation, instead of redefining the whole mediation schema. The re-definition of the mapping when the schemas evolve is a time consuming process, especially at runtime. The main reason of adapting the original mappings to the new schemas is to avoid regenerating mappings every time changes occur in the schemas. Our algorithm is efficient for the contexts in which the removed source contains few relations, few attributes and few constraints and also in the contexts in which the removal operation does not occur frequently in the system. The time to adapt mappings for one single change is in most cases shorter than from scratch generation. When the sequence of changes is long, the total time for all changes of the sequence may be longer than the time to directly generate mappings for the new schemas. As future work, we shall study the removal of a source which has an equivalent source in the mediation system.

References

[1] Akoka J., Berti-Equille L., Boucelma O., Bouzeghoub M., Comyn-Wattiau I., Cosquer M., Goasdoué-Thion V., Kedad Z., Nugier S., Peralta V., and Sisaid-Cherfi S., "A Framework for Quality Evaluation in Data Integration Systems," *in Proceedings of the 9th International Conference on Enterprise Information Systems (ICEIS'2007),* Funchal, 2007.

[2] Banerjee J., Kim W., Kim H., and Korth H., "Semantics and Implementation of Schema Evolution in Object-Oriented Databases," *ACM SIGMOD Record*, pp. 311-322, 1987.

[3] Bouzeghoub M. and Kedad Z., "Discovery View Expressions from a Multi-Source Information System," *in Proceedings of the Fourth IFCIS International Conference on Cooperative Information Systems (COOPIS'99)*, Scotland, pp.57-68, 1999.

[4] Bouzeghoub M., Farias Lóscio B., Kedad Z., and Salgado A., "Managing the Evolution of Mediation Queries," *in Proceedings of the International Conference on CoopIS'2003*, pp. 22-37, 2003.

[5] Bouzeghoub M., Kedad Z., and Soukane A., "Improving Mediation Query Generation Using Constraints and Metadata," *Bases de Données Avancées (BDA*), pp. 385-405, 2004.

[6] Chawathe S., Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y., Ullman J., and Widom J., "TSIMMIS Project: Integration of Heterogeneous Information Sources," *in Proceedings of IPSI Conference*, Japan, 1994.

[7] Fletcher G. and Wyss C., "Data Mapping as Search", *Extending Database Technology EDTB, Advances in Database Technology-EDTB 2006,* pp. 95-111, 2006.

[8] Gupta A., Mumick I., and Ross K., "Adapting Materialized Views After Redefinition," *in Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pp. 211-222, 1995.

[9] He B. and Chang K., "Automatic Complex Schema Matching Across Web Query Interfaces: A Correlation Mining Approach," *in Proceedings of ACM Transactions on Database Systems*, 2006.

[10] Kedad Z. and Xue X., "Mapping Generation for XML Data Sources: A General Framework," *in Proceedings of Web Information Retrieval and Integration*, Japan, pp. 164-172, 2005.

[11] Kirk T., Levy A., Sagiv Y., and Srivastava D., "The Information Manifold," *in Proceedings of AAAI 95 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, pp. 85-91, 1995.

[12] Lee A., Nica A., and Rundensteiner E., "The EVE Approach: View Synchronization in Dynamic Distributed Environments," *IEEE TKDE*, vol. 14, no. 5, pp. 931-954, 2002.

[13] Loscios B., "Managing the Evolution of XML-Based Mediation Queries," *PHD thesis, University of Versailles (France) and Universidade Federal de Pernambuco (Brésil)*, 2003.

[14] Loscios B. and Salgado A., "Evolution of XML-Based Mediation Queries in a Data Integration System", *in Proceedings of ER Workshops,* Shanghai, China, pp. 402-414, 2004.

[15] McBien P. and Poulovassilis A., "Schema Evolution in Heterogeneous Database Architectures, A Schema Transformation Approach," *in Proceedings of CAiSE'02*, Toronto, pp. 484-499, 2002.

[16] Miller R., Hernandez M., Haas L., "Schema Mapping as Query Discovery," *in Proceedings of the 26th International Conference on VLDB'00*, Cairo, Egypt, pp. 77-88, 2000.

[17] Mohania M. and Dong G., "Algorithms for Adapting Materialized Views in Data Warehouses," *in Proceedings of CODAS*, pp. 309-316, 1996.

[18] Batista M. and Salgado A., "Minimality Quality Criterion Evaluation for Integrated Schemas," *ICDM*, 2007.

[19] Mumick I., Quass D., and Mumick B., "Maintenance of Data Cubes and Summary Tables in a Warehouse," *in Proceedings of SIGMOD*, pp. 100-111, 1997.

[20] Peralta V., " Data Quality Evaluation in Data Integration Systems ," *thèse de Doctorat, Université de Versailles (France) and Universidad de la República (Uruguay)*, Versailles, France, 2006.

[21] Popa L., Velegrakis Y., Miller R., Hernandez M., and Fagin R., "Translating Web Data", *Proc. of the 28th Int. Conf. on VLDB'02*, Hong Kong, China, pp. 598-609, 2002.

[22] Shvaiko P. and Euzenat J., "A Survey of Schema-based Matching Approaches", *in Proceedings of Journal Data Semantics IV*, pp. 146-171, 2005.

[23] Soukane A., "Génération Automatique Des Requêtes De Médiation Dans Un Environnement Hétérogène," Thèse De Docteur De L'université De Versailles Saint-Quentin En Yvelines, 2005.

[24] Theodoratos D. and Sellis T., "Designing Data Warehouses," *Data Knowledge Engineering*, vol. 31 no. 3, pp. 279-301, 1999.

[25] Velegrakis Y., Miller R., and Popa L., "Mapping Adaptation Under Evolving Schemas," *in Proceedings of the 29th VLDB Conference,* Berlin, 2003.

[26] Wiederhold G., "Mediators in the Architecture of Future Information Systems," *IEEE Computer*, vol. 25, no. 3, pp. 38-49, 1992.

[27] Xue X., "Automatic Mapping Generation and Adaptation for XML Data Sources," *Thèse De Docteur De L'université De Versailles Saint-Quentin En Yvelines*, 2006.

[28] Yu C. and Popa L., "Semantic Adaptation of Schema Mapping When Schemas Evolve," *in Proceedings of the 31st International Conference on Very Large Data Bases*, Trondheim, Norway, 2005.

[29] Zamboulis L., "XML Data Integration by Graph Restructuring," *in Proceedings of British National Conferences on Databases*, pp. 57-71, 2004.

Latifa Baba-Hamed received her PhD thesis in computer sciences from the University of Oran, Algeria, in 2007. She is a member of Database Systems Group at USTO. Her main research interests include object-oriented, temporal, active databases and integration of data.