

# STRUCTURED QUERY TRANSLATION IN PEER TO PEER DATABASE SHARING SYSTEMS

Mehedi Masud<sup>1\*</sup> and Fazel Keshtkar<sup>2</sup>

<sup>1</sup>*School of Information Technology and Engineering, University of Ottawa, Ottawa K1N 6N5, Canada*

Email: [mmasud@site.uottawa.ca](mailto:mmasud@site.uottawa.ca)

<sup>2</sup>*School of Information Technology and Engineering, University of Ottawa, Ottawa K1N 6N5, Canada*

Email: [akeshtka@site.uottawa.ca](mailto:akeshtka@site.uottawa.ca)

## ABSTRACT

*This paper presents a query translation mechanism between heterogeneous peers in Peer to Peer Database Sharing Systems (PDSSs). A PDSS combines a database management system with P2P functionalities. The local databases on peers are called peer databases. In a PDSS, each peer chooses its own data model and schema and maintains data independently without any global coordinator. One of the problems in such a system is translating queries between peers, taking into account both the schema and data heterogeneity. Query translation is the problem of rewriting a query posed in terms of one peer schema to a query in terms of another peer schema. This paper proposes a query translation mechanism between peers where peers are acquainted in data sharing systems through data-level mappings for sharing data.*

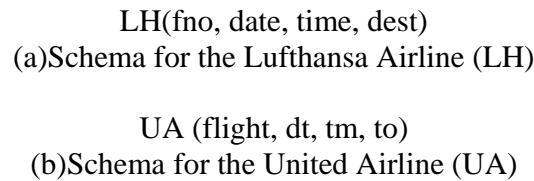
**Keywords:** P2P, Data sharing, Database, Query processing, Mappings, Heterogeneity

## 1 INTRODUCTION

In a peer to peer database sharing system, each peer manages its data autonomously and collaborates with neighboring peers in a peer to peer network by exporting part of its data. However, a peer has no global knowledge about the resources possessed by other peers in the network. A peer has only the knowledge about its neighboring peers. Existing P2P systems offer sharing of files, music, and videos in which keyword-based exact matching lookup services are used. However, the systems lack advanced database-style data management, query processing, and other functionalities. The database-style data manipulation in P2P is more robust than conventional file sharing approaches.

This paper presents a query processing mechanism, in particular translation of queries between peers, in peer database sharing systems, where peers are acquainted using data-level mappings (Kementsietsidis, Arenas, & Miller, 2003). In general, a PDSS (Serafini, Giunchiglia, Molopoulos, & Bernstein, 2003; Tatarinov, Ives, Madhavan, Halevy, Suciu, Dalvi, et al., 2003; Franconi, Kuper, Lopatenko, & Zaihrayeu, 2004; Ng, Ooi, Tan, & Zhou, 2003) offers two basic services. First, it offers to its peers the ability to share data with each other. Second, it offers the ability to peers querying each other's contents. This paper mainly focuses on the later service and investigates a query processing mechanism for collecting data from peers through translation of queries. Query processing in a PDSS is the problem of querying stored data in different peers using a single query posed in a peer w.r.t. to the peer's vocabularies and translating the query w.r.t into other peers' vocabularies. The provision of attributes, relations, and data values of each peer during this process is called by this paper the *vocabulary* of that peer.

Several previous approaches (Boyd, Kittivoravitkul, Lazanitis, McBrien, & Rizopoulos, 2004; Domenig & Dittrich, 2002; Chang & Garcia-Molina, 1999; Levy, Rajaraman, & Ordille, 1996; Haas, Kossmann, Wimmers, & Yang, 1997) to the problem of query translation across heterogeneous sources are in existence. The exchange of data between heterogeneous data sources is provided mainly through the use of inter-schema mappings (Lenzerini, 2001; HaLevy, 2001; Milo & Zohar, 1998) that specify how the schemas of the peers are related. In order to establish schema-level mappings, the peers must be willing to share at least portions of their schemas and cooperate in establishing and managing the mappings.



**Figure 1.** Database schemas of peers

This paper considers an alternate setting in which cooperation between peers established through the schema mappings is neither desirable nor feasible since sources might belong to different worlds. In addition to schema heterogeneity, sources may have data-level heterogeneity. Arenas, Kantere, Kementsietsidis, Kiringa, Miller, & Mylopoulos (2003) first proposed a peer data sharing setting where acquaintances between peers are established through data-level mappings. The data-level mappings are generated through the creation of mapping tables (Kementsietsidis, Arenas, & Miller, 2003). Mapping tables allow associating seemingly unconnected and overlapping databases by associating values of the peer vocabularies. Formally, a mapping table, denoted by  $m[X, Y]$ , is a relation over the attributes  $X \subseteq U_i$  and  $Y \subseteq U_j$ , where  $U_i$  and  $U_j$  are exposed attributes from peer  $P_i$  and  $P_j$ , respectively. A tuple  $(x, y)$  in a mapping table  $m[X, Y]$  indicates a mapping where a value  $x \in \text{dom}(X)$  in  $P_i$  is associated with a value  $y \in \text{dom}(Y)$  in  $P_j$ . A set of mapping tables  $M_{ij} = \{m_1, \dots, m_k\} \subseteq M_i$  in  $P_i$  stores the mappings between data items of  $P_i$  and  $P_j$  so that  $P_i$  shares its local data with  $P_j$ . Intuitively, mapping tables are data-level associations which list pairs of corresponding values between two sources and thereby act as an interface to relate data between two peers. Two peers related through mapping tables are said to be *acquainted*. When a peer creates an acquaintance with another peer, the acquainted peer is called an *acquaintee* of that peer.

Kementsietsidis and Arenas (2004) proposed a query translation mechanism between peers using mapping tables. In terms of querying, users pose queries only with respect to their local peer schema, and then the queries are rewritten using mapping tables to a set of queries in terms of the vocabularies of the acquainted peers' schemas. This paper extends this approach, which includes the translation of projections in the queries and an efficient way of reusing computed query translation. In Particular, this paper describes some findings of the algorithm and proposes a few extensions.

The remainder of this paper is organized as follows. Section 2 describes a motivating example of a query translation scenario; Section 3 introduces the query translation strategy considering the data-level mappings between peers and analyzes the work of Kementsietsidis et al. (2004); Section 4 presents our proposed solution; Section 5 describes related work, and finally, Section 6 concludes this paper.

## 2 MOTIVATING EXAMPLE

This section presents query translation examples by considering a very common scenario of a flight reservation peer-to-peer system where peers are acquainted by data-level mappings. Figure 1 shows parts of the schemas of two partner airlines. They are called Lufthansa (LH) and United Airline (UA).

In spite of the similarities with regards to the schemas, the two peers use different data vocabularies to describe flights and destinations. Figures 2(a) and 2(b) depict the differences. Figure 2 also shows examples of mapping tables.

Assume that a user wants to retrieve information for LH flights to Los Angeles (LA) with the following query.

Q1: select \* from LH where Dest = "LA"

What if the user also wants to retrieve information for UA flights to Los Angeles? Then the following query Q2 is needed for the UA database.

Q2: select \* from UA where (to = "ORD" OR to = "ONT")

Fno	Date	Time	Dest
LH410	11/05	10:00	SF
LH541	11/15	10:30	NY
LH690	11/21	12:00	LA
LH691	11/22	12:00	LA

(a) Lufthansa airline instance (LH)

Flight	Dt	Tm	To
UA120	11/15	10:30	JFK
UA134	11/15	11:00	LAX
UA135	11/21	12:00	ONT
UA141	11/22	07:00	ORD

(b) United Airline Instance (UA)

Fno	Flight
LH541	UA120
LH690	UA135

(c) Mapping table fno2flight

Date	Dt
v	v

(d) Mapping table date2dt

Dest	To
SF	JFK
NY	LAX
LA	ONT
LA	ORD

(e) Mapping table dest2to

**Figure 2.** Database instances and mapping tables

We observe that the mapping table in Figure 2(e) provides the necessary information for mapping the value of attribute *dest* of LH to the value of attribute *to* of UA. But the question is how the translation of Q1 to Q2 can be achieved? The following section focuses on this translation.

### 3 PEER-TO-PEER QUERY

In a peer-to-peer database sharing system, when a user poses a query to its peer, she/he needs only to be aware of the local peer schema that she/he is using. In terms of query execution semantics, peer-to-peer queries are classified into two categories: (i) local query and (ii) global query. A local query is defined in terms of a local peer database schema and is executed using only the data in the local peer. On the other hand, the execution of a global query uses the peer-to-peer network to augment locally retrieved data with data that resides in other related peers. Therefore, a global query is composed of some component queries that result from the translation of the original query. The relationships between component queries are represented by a query dependency graph. The nodes in the graph represent queries, and there is an edge from a query  $q_j$  to a query  $q_k$  if  $q_k$  is the result of translation of  $q_j$ . Query  $q_k$  is said to depend on query  $q_j$ , denoted as  $\text{dep}(q_k) = q_j$ . The query dependency graph represents the translation and propagation of the queries in the network.

When a query  $q_1$  is translated into a query  $q_2$ , then the query  $q_2$  should be sound and complete. Soundness means that the translated query always gives a correct answer, and completeness means all the results are correct. Our point is that although soundness is a property that every translation must satisfy, executing queries that are incomplete may be sufficient due to the nature of the peer-to-peer network. According to Kementsietsidis et al., (2004), the sound translation of a query is defined as follows:

"Let  $q_1$  and  $q_2$  be queries over peers  $P_1$  and  $P_2$  respectively, such that  $q_1 = \sigma_E(R_1 \dots R_k)$ , where  $E$  is a conjunction of equality atoms and  $R_1, R_k$  are relations in  $P_1$ . Then  $q_2$  is a sound translation of  $q_1$  with respect to a mapping table  $m$ , denoted by  $q_1 \xrightarrow{m} q_2$ , if for every relation instance  $r_2$  of  $P_2$  and  $t_2 \in q_2(r_2)$ , there exists a valuation  $\rho$  of  $m$  and a tuple  $t \in \sigma_E(\rho(m))$  such that  $\pi_{\text{att}(q_2)}(t) = t_2$ ". According to the definition, query Q2 is a sound translation of query Q1.

#### 3.1 Algorithm for query translation

According to the algorithm (Kementsietsidis et al., 2004), a query posed in a peer is initially expressed in a query language like SQL and later transformed into some internal form. A query is expressed in S+J algebra and consists of conjunctions and disjunctions of atoms of the form  $(A=B)$  and  $(A = a)$ . During translation, a query is represented as a tableau called T-query (Kementsietsidis et al., 2004). The intuition behind this is uniformity with mapping tables and relation schemas. For example, Figure 3 depicts the representation of T-query of the following query Q.

Q: select \* from LH where Dest = "LA" OR Dest = "SF"

Fno	Date	Time	Dest
X1	Y1	Z1	LA
X2	Y2	Z2	SF

**Figure 3.** T-query of query Q

Now we briefly describe the query translation algorithm.

### Algorithm

1. Convert a query  $q$  to its corresponding T-query  $q_T = T$ .
2. Compute  $q'_T = \pi_{U'}(T \times m)$ , where  $m$  is a mapping table with attributes  $U$  and  $U'$ .
3. Output a query  $q'$  expressed by  $q'_T$ .

### Using multiple mapping tables

1. Convert  $q$  into disjunctive normal form. Then proceed by each disjunctive  $D_j$  of  $q$  in isolation.
2. For each  $D_j$ , select mapping tables for translation and compute translation using the above algorithm.

*Example:*

Q: select \* from LH where Dest = "LA" AND Date = "11/21"

The algorithm selects mapping tables *date2dt* and *dest2to* for the translation of Q and results in the following query:

Q': select \* from UA where (to = "ORD" OR to = "ONT") AND dt= "11/21"

## 3.2 Discussion

Kementsietsidis et al., (2004) presented the query translation algorithm assuming a single mapping table  $m$  that maps all the attributes of the relations in two peers. In practice, we may have mapping tables that map a subset of attributes of the two relations. The algorithm translates the projection of a query separately. It also ignores the translation of the projection when the projection includes attributes from multiple mapping tables. The algorithm also avoids the case when a query belongs to several relations, i.e. join. Finally, the paper demands that a translated query is stored in the form of a mapping table for later use in order to reduce the re-translation cost of a previously executed query. For this case, the algorithm constructs a mapping table  $T_c$  with schema  $(U, U')$ , such that for each tuple  $t \in T_c$ ,  $\pi_U(t) \in T_c$ ,  $\pi_{U'}(t) \in T_c$ , and  $\pi_{U'}(t)$  is a sound translation of  $\pi_U(t)$  with respect to a mapping table  $m [U, U']$ . Here, the first point to note is that  $m$  contains all the attributes of the relations from two peers. The second point is that the mapping table  $m$  that maps between queries may have only one disjunction. The algorithm avoids the situation when a query has multiple disjunctions and requires multiple mapping tables to translate a query. Moreover, the algorithm generates an individual mapping table for each translated query for optimization of translation costs. Maintaining each mapping table for each query is not feasible.

## 4 PROPOSED SOLUTION

In this paper we propose some extensions of the algorithm (Kementsietsidis et al. 2004) by relaxing some of its assumptions. The first extension of the algorithm is to compute a complete translation of a query considering a mapping table that maps a subset of the attributes from the relations of two peers. For this, we consider a mapping table such as  $m[X, Y]$ , where  $X \subseteq U$  and  $Y \subseteq U'$  and where  $U$  and  $U'$  are non-empty sets of attributes exposed by two acquainted peers. For example, Figure 2(c) shows a mapping table from a set of attributes  $X = \{fno\}$  to a set of attributes  $Y = \{flight\}$ . Before describing the proposed solution, we first illustrate the problem of the algorithm (Kementsietsidis et al. 2004) using an example. Note that in order to translate a query, the algorithm converts the source query  $q$  to its corresponding T-query,  $q_T = T$ . Next, it considers a mapping table  $m$ , which maps all the attributes  $U$  and  $U'$  of two acquainted peers and computes  $q'_T = \pi_{U'}(T \times m)$ . Finally, the algorithm outputs the query  $q'$  represented by  $q'_T$ .

All these steps assume a single mapping table that contains all the attributes of two acquainted peers. However, there may be a situation where a mapping table contains only few attributes of two relations, for example, consider Figure 2(e). If we consider the query Q1 to translate into query Q2, then the selected mapping table is dest2to as shown in Figure 2(e). The mapping table does not contain all the attributes of relations of peers LH and UA.

Fno	Date	Time	Dest	To
X1	Y1	Z1	LA	LAX
X2	Y2	Z2	LA	ONT

**Figure 4.** Result of  $(T \times m)$

The proposed solution considers a mapping table that may not contain all the attributes. Let a mapping table  $m[X, Y]$  exist, where  $X \subseteq U$  and  $Y \subseteq U'$ . The algorithm first computes  $T \times m$  and then computes the projection as  $T' = \pi_Y(T \times m)$ . Finally, the algorithm computes  $T' \cup U'$ . The extra attributes may have any value. The example is given as follows.

Consider the query Q1. The result of  $T \times m$  for query Q1 is shown in Figure 4. From Figure 4, the result of  $T'$  is as follows:

To
LAX
ONT

The result of the computation of  $T' \cup U'$  is shown in Figure 5.

Fno	Date	Time	To
X2	Y2	Z2	LAX
X4	Y4	Z4	ONT

**Figure 5.**  $T' = T' \cup U'$

Observe that Figure 5 looks like a T-query. We can now convert the T-query into its equivalent query  $q'$  which is actually the query Q2.

#### 4.1 Dealing with projection

In this section, we give a solution to handling projection using mapping tables using the following assumption.

- We assume that there exists an identity-mapping table if there is no data mapping involved between two exposed attributes. The example of an identity mapping is shown in Figure 6.

An identity mapping table uses variables to represent the identity function, i.e., each value of the first database is mapped to a same value in the second database. Consider Figure 5. Here, every valuation of the variables gives a *date* value that can be mapped to a value in *Dt*.

Date	Dt
v	v

**Figure 6.** Identity mapping table

When projection attributes exist in a query, the translation is performed as follows.

1. Separate each attribute from the projection and look for an identity mapping table  $m_i[X, Y]$ .
2. If a mapping is found, then compute  $\pi_Y(m_i)$
3. After the query is translated, replace the attributes in the projection with the corresponding attributes in  $\pi_Y(m_i)$ .

Example: If we want to find all the flight numbers and their dates of departure from LH peer where the destination is LA, then the query will be as follows:

*q: select fno, date from LH where Dest="LA"*

The algorithm first searches the mapping tables, which contain attributes *fno* and *date*. Therefore, there are two mapping tables 2(c) and 2(d) that are found and shown in Figure 2. Therefore, *fno* is replaced with  $\pi_{\text{flight}}(\text{fno}, \text{flight})$ , and similarly *date* is replaced with *dt*. The remaining translation of the query is the same as discussed before in Section 3.1. Hence, the equivalent translated query is as follows:

*Q': select flight, dt from UA where dest = "ORD" or dest = "ONT"*

## 4.2 Reusing computed translation

Kementsietsidis et al. (2004) also introduced a concept to store translated queries in the form of a mapping table for the associations of two queries (original query and its translated query) in order to reduce re-translation cost. The algorithm creates an individual mapping table for each translation of a disjunctive query. We believe that maintaining each mapping table for each query is not feasible. In this paper, we introduce a global mapping table  $T_{\text{global}}$  for all the translated queries. The table includes all the attributes  $U$  and  $U'$  in two peers. Each row of the table represents the association of an original query and its translated query. The left part of a row represents the T-query of an original query, and the right side represents the T'-query of the translated query. The algorithm to use the translation of query is given below.

1. Convert a query into a T-query.
2. Search for the entry in the left part of  $T_{\text{global}}$  that matches the record in the T-query. If  $\pi_x(T_{\text{global}}) = \text{T-query}$ , where  $X$  are the attributes of T-query, then compute T'-query =  $\pi_y(T_{\text{global}})$  for all  $\pi_x(T_{\text{global}})$ .
3. Translate the T'-query into its equivalent query

Example: Consider a query  $Q$ : *select fno, date from LH where Dest = "LA" and Date = "11/21"*. The resulting T-query for the query is shown in Figure 7.

Fno	Date	Dest
X	11/21	LA

**Figure 7.** T-query

Assume that the query is executed previously and the corresponding entry in the  $T_{\text{global}}$  is as follows:

Fno	Date	Time	Dest	Flight	Dt	Tm	To
X1	11/20	Y1	LA	X2	11/21	Y2	ORD
X3	11/20	Y1	LA	X4	11/21	Y3	ONT

**Figure 8.**  $T_{\text{global}}$

We observe that the entry in Figure 7 exists in Figure 8. Therefore, after computing  $\pi_{\text{flight}, \text{dt}, \text{to}}(T_{\text{global}})$ , the following T'-query is obtained:

Flight	Dt	To
X2	11/21	ORD
X4	11/21	ONT

**Figure 9.** T'-query

Therefore, the translated query from the T'-query is

*Q': select flight, dt from UA where (to="ORD" OR to="ONT") and date = "11/21"*

### 4.3 Discussion

In this paper, we consider a data sharing system where peers are related with data-level mappings by using mapping tables. However, there may be a situation where differences in the syntax of two records in two peers may influence the interpretation of the semantics of the records. For example, a name of a person may be represented in two peers in two different formats, e.g. name is represented as lastname, firstname in one peer and as firstname, lastname in another peer. Another example could be the representation of dates. For example, date in UA peer is represented as mm/dd syntax and LH peer uses the syntax dd/mm. In order to deal with these cases, one can use a mapping table by storing the corresponding data associations. One can also use an identity mapping table with an appropriate valuation function that converts from mm/dd format to dd/mm format. Masud, Kiringa, and Kementsietsidis (2005) address this issue and propose four rules (merge, separation, data association, and data conversion rules) that consider the syntactic and data-level heterogeneity.

## 5 RELATED WORK

Extensive efforts on peer-to-peer database sharing systems concerning issues such as data integration, placement, mappings, and query formulation have been made. Gribble, Halevy, Ives, Rodrig, and Suciu (2001) address challenging issues in P2P systems and propose initial ideas for data placement and query answering. Arenas et al. (2003) propose the architecture of a P2P Multidatabase system and address issues such as the mapping of data. The Piazza (Tatarinov, Ives, Madhavan, Halevy, Suciu, Dalvi, & Dong, 2003) system provides a solution for query answering in a peer-to-peer environment, where the associations between peers are expressed either in global-as-view (GAV) or local-as-view (LAV) mappings (Lenzerini, 2001). If the mappings define the sources in terms of the global schema then the approach is LAV, and if the global schema is defined in terms of sources then this is GAV. GAV suffers from the necessity of changing the global schema (in design time) if there is a change in a local schema. Chang and Garcia-Molina (1999) also propose an idea that uses syntactic rules to translate queries between heterogeneous databases.

## 6 CONCLUSION

In this paper, we describe an approach to data sharing between autonomous relational data sources where mappings between sources are established through the data-level mappings. There are different approaches to data sharing using schema mappings but they impose constraints on the schemas of sources. The approach that is proposed by Kementsietsidis et al. (2004) considers mappings that impose no such constraint. This paper addresses some findings of the query translation algorithm proposed by Kementsietsidis et al. (2004) and minimizes problems in those findings by proposing some extensions. Moreover, in this paper, we include some additions to the algorithm that deal with projection and reusing translated queries. In the future, we would like to propose an approach to query translation that combines schema-level and data-level mappings and implement the algorithm in a large peer database sharing system.

## 7 REFERENCES

- Arenas, M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R. J., & Mylopoulos, J. (2003) The Hyperion Project: From Data Integration to Data Coordination. *ACM SIGMOD RECORD* 32(3), 53-58.
- Boyd, M., Kittivoravikul, S., Lazanitis, C., McBrien, P., & Rizopoulos, N. (2004) AutoMed: A BAV Data Integration System for Heterogeneous Data Sources. *Proc. Int'l Conf on Advance Information Systems Engineering (CAiSE)*, Riga, Latvia.
- Chang, C. C. K., & Garcia-Molina, H. (1999) Mind your vocabulary: Query mapping across heterogeneous information source. *Proc. Int'l Conf. on the Management of Data (ACMSIGMOD)*, Philadelphia, Pennsylvania, USA
- Domenig, R., & Dittrich, K.R. (2002) Query Explorativeness for Integrated Search in Heterogeneous Data Sources. *Proc. Int'l Conf on Advance Information Systems Engineering (CAiSE)*, Toronto, Canada.

- Franconi, E., Kuper, G., Lopatenko, A., & Zaihrayeu, I. (2004) The coDB Robust Peer-to-Peer Database System. *Proc. Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGrid)*, New York, USA.
- Gribble, S., Halevy, A., Ives, Z., Rodrig, M., & Suciu, D. (2001) What can databases Do for Peer-to Peer? *Proc. Web and Databases( WebDB)*, Santa Barbara, California, USA.
- Haas, L., Kossmann, D., Wimmers, E.L., & Yang, J. (1997) Optimizing queries across diverse data sources. *Proc. Int'l Conf. on Very Large Data Bases (VLDB)*, Athens, Greece.
- HaLevy, A. Y. (2001) Answering Queries Using Views: A Survey. *VLDB Journal* 10(4), 270-294.
- Kementsietsidis, A., & Arenas, M. (2004) Data Sharing Through Query Translation in Autonomous Sources. *Proc. Int'l Conf. on Very Large Data Bases (VLDB)*, Toronto, Canada.
- Kementsietsidis, A., Arenas, M., & Miller, R. J. (2003) Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues. *Proc. Int'l Conf. on the Management of Data (ACMSIGMOD)*, California, USA.
- Lenzerini, M. (2001) Data Integration: A Theoretical Perspective. *Proc. of the ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, Santa Barbara, California, USA.
- Levy, A., Rajaraman, A., & Ordille, J.J. (1996) Querying heterogeneous information sources using source descriptions. *Proc. Int'l Conf. on Very Large Data Bases (VLDB)*, Bombay, India.
- Masud, M., Kiringa, I., & Kementsietsidis, A. (2005) Don't Mind Your Vocabulary: Data Sharing Across Heterogeneous Peers. *Proc. of Int'l Conf. on Cooperative Information Systems (CooPIS)*, Agia Napa, Cyprus.
- Milo, T. & Zohar, S. (1998) Using Schema Matching to Simplify Heterogeneous Data Translation. *Proc. Int'l Conf. on Very Large Data Bases (VLDB)*, New York City, USA.
- Ng, W. S., Ooi, B. C., Tan, K. L., & Zhou, A. Y. (2003) PeerDB: A p2p-based system for distributed data sharing. *Proc. Int'l Conf. on Data Engineering (ICDE)*, Bangalore, India.
- Serafini, L., Giunchiglia, F., Molopoulos, J., & Bernstein, P. (2003) *Local Relational Model : A Logical Formalization of Database Coordination*. Technical Report, Informatica e Telecomunicazioni, University of Trento, Trento, Italy.
- Tatarinov, I., Ives, Z., Madhavan, J., Halevy, A., Suciu, D., Dalvi, N., & Dong, X. (2003) The Piazza Peer Data Management System. *IEEE Transactions on Knowledge and Data Engineering (TKDE)* 16(7), 787-798.

(Article history: Received 25 January 2009, Accepted 18 May 2009, Available online 30 September 2009)