International Journal of Computer Applications (0975 - 8887)
Volume 66 - No. 24, March 2013

Value-Mapped Peer Data Sharing Systems and Consistent Query Answering

Mehedi Masud
Computer Science Department, Taif University
Taif, Saudi Arabia

ABSTRACT

A peer data sharing system consists of peers where a database in a peer is designed and administered autonomously. Acquaintances between peers are used to share data among peers. These acquaintances are established thorough data sharing constraints. Data between peers may be inconsistent with respect to constraints due to the change of constraints (e.g. adding, modifying, and deleting constraints between peers) and data updates in individual peer. One possible solution to resolve inconsistencies is to modify data physically in inconsistent peers through update propagation. This strategy is not practical since peers are autonomous and a peer may not have permission to modify other peers' data. Considering the possible inconsistent situations, this paper discusses a semantics and a technique for obtaining consistent answers. Consistent answers are obtained at query time by avoiding the inconsistent data.

General Terms:

Database, Peer to Peer

Keywords:

Data Interoperability, mappings, query processing, data sharing

1. INTRODUCTION

In a P2P data sharing system [1, 2], one peer acts as a source (data provider) and may act as a target (data receiver). The role depends on how the peer is related with other peers. The peers are related through schema mappings in the form of tuple-generating dependencies [3]. The mappings are equivalent to so-called global-local-as-view or GLAV mappings [4]. The ultimate goal of mappings is to share data with other peers that satisfy the mappings. Contrary to the traditional data integration settings [4, 5, 6] where a global mediated schema is required for data exchange, in a P2P data sharing, semantic relationships that exist between peers are exploited for sharing data.
Typical data integration [4] and data exchange systems [1, 2, 7] deal with *one world*, for example, a set of sources all containing information about videos or music files. Therefore, views or schema mappings are used for data integration or data exchange between data sources. Views are queries that map and restructure data between schemas. In value-mapped data sharing systems [8, 9, 10], sources may represent different worlds with different schemas and data vocabularies, and the real world entities denoted by different symbols in different sources may be semantically related [11]. A possible solution for creating semantic relationships is to create a domain relation [9] through value correspondences [12] between sources. The value-level mappings established through value correspondences map the data elements of one peer domain to the data elements of another peer domain.
Hyperion[8, 10] is the first peer data sharing system developed using the value-level mappings between peers. The mappings are created using the semantics of mapping tables [12]. Mapping tables map related data values, logically tuples, that reside in different peers. Mapping tables also bridge the differences of data vocabularies as well as impose data sharing constraints between peers. For query answering, a query is posed to a peer's instance and is applied to the peer's local database instance. After that the query is propagated to all other related peers for retrieving related data. Authors in [13, 14] proposed a query translation mechanism between peers in a value-mapped peer data sharing system. However, the consistent query answering is not focused. Data and mappings are updated in peers in course of time which may make the data in related peers inconsistent w.r.t the new data and mappings, although, data in peers are consistent w.r.t their local integrity constraints. Considering these inconsistent situations, authors in [15] presented a semantics of query answering that characterizes what are the intended and correct answers to a query posed to and answered by a value-mapped peer data sharing system. This paper is an extension of our previous [15] work where we investigated a consistent query answering mechanism and presented a semantics of obtaining consistent answers in a value-mapped peer data sharing system. This paper presents a technique to obtain consistent answers. To obtain consistent query answering we adopt a technique introduced by Arenas et al. [16]. In the previous work, we extended the semantics to be used in a peer data sharing system that is characterized by defining a set of virtual global instances called solutions for a peer where a query originates. A solution for a peer is an intended global instance that respects the mappings with its acquainted peers. After having a definition of the intended solutions for a peer, the consistent answers to a query in a peer are those that are *certain* [17] in every possible solution.
The paper is structured as follows. Section 2 presents the settings of value-mapped peer data sharing system. Section 3 explains the semantics of consistent answering of queries. Section 4 shows the process of computing consistent answers, and Section 5 discusses related work. Finally, the paper concludes with some future directions.

2. VALUE-MAPPED PEER DATA SHARING SYSTEM

In a data sharing system (DSS), there exits a source schema $\mathbf{S}$ and an independent target schema $\mathbf{T}$. In a value-mapped data sharing system, mapping tables are used to map data between peers. Each tuple t in a mapping table, $m(\mathbf{x}, \mathbf{y})$, is a mapping that indicates for some value of $\mathbf{x}$, the corresponding value of $\mathbf{y}$ that should be populated in the target. Intuitively, a tuple (x, y) in $m(X, Y)$ indicates a mapping that the value $x \in dom(X)$ is associated with the value $y \in dom(Y)$. Formally, a mapping

patid	name	testname	result
101	Ley	C05187	12.6 g/ml

(a) relation R_1 at P_1

ohip	pname	test	result
123NE	Ley	hemoglobin	13.7 g/ml
298GA	Kelly	hemoglobin	9.87 c/mcl
256GA	John	whitebloodcount	8.37 c/mcl

(b) relation R_2 at P_2

patid	ohip
101	123NE

(c) Mapping table mt_1

testname	test
hemoglobin	C05187

(d) Mapping table mt_2

Fig. 1. Relations at P_1 and P_2 and mapping tables

over a set of attributes U of $X \cup Y$, alternatively called a tuple *t*, in a mapping table represents that for each attribute $A \in U$, $t[A]$ is either a constant in $dom(A)$, a variable in V or an expression of the form $v - S$, where $v \in V$ and S is a finite subset of $dom(A)$ [12]. The relationship between the source and the target is called data sharing constraint, Σ_{ij}, that specifies how and what data in the peer P_i can be transformed to an instance over the peer P_j schema for sharing. The constraint uses the mapping table to determine the corresponding values which should be produced in the target instance.
A data sharing constraint Σ_{ij} between two peers P_i and P_j in a value-mapped peer data sharing system is a set of mapping that define the schema $m^S_{i,j}$ and data-level mappings $m^D_{i,j}$ between peers. The construction of constraint Σ_{ij} forms an acquaintance (*i, j*) between P_i and P_j. Therefore, the mappings in a constraint is a pair $< m^S_{i,j}, m^D_{i,j} >$, where:

- $m^S_{i,j}$ is a GLAV mapping of the form

$$\forall \vec{x}(\exists \vec{y} \varphi(\vec{x}, \vec{y}) \rightsquigarrow s(\vec{x}))$$

  where $\varphi(\vec{x}, \vec{y})$ is a conjunctive query over the peer schema of a peer P_j and $s(\vec{x})$ is the k^{th} external source of P_i.
- MT=$m^D_{i,j}$=$\{mt_1, mt_2, \ldots, mt_q\}$ is a set of mapping tables or data-level mappings. MT denotes the set of mapping tables used to map data of P_i to data of P_j.
  Combination of Schema-level and data-level mappings can be represented with the mapping assertion as follows:

$$\forall \vec{x}(\exists \vec{y} \varphi(\vec{x}, \vec{y}) \overset{MT}{\rightsquigarrow} s(\vec{x}))$$

Note that a tuple in a mapping table may contain constants or variables. The variables are used to increase expressiveness power of mapping tables. Given the presence of variables in mappings, it is necessary to introduce the notion of a valuation. A valuation ρ over a mapping table mt is a function that maps each constant value in mt to itself and each variable v of mt to the value in the intersection of the domains of the attributes where v appears [12]. Furthermore, if v appears in an expression of the form $v - S$, then $\rho(v) \notin S$. In general, if there are multiple mapping tables $M = \{mt_1, mt_2, \cdots, mt_k\}$ then the tables can be combined into a single mapping table using the $\wedge$-operator [12]. Therefore, valuation ρ on MT can be applied and is represented as $\rho(mt_1, mt_2, \cdots, mt_k)$.

DEFINITION 1 VALUATION. *[12] A valuation ρ over a mapping table mt is a function that maps each constant value in mt to itself and each variable v of mt to the value in the intersection of the domains of the attributes where v appears. Furthermore, if v appears in an expression of the form $v - S$, then $\rho(v) \notin S$.*

Since a mapping table mt from X to Y associates values from $dom(X)$ to $dom(Y)$, the set of $Y-$values with which a particular value $x \in dom(X)$ is associated can be determined by the following definition.

101	Ley	C05187	12.6 g/ml

(a) Answer from P_1

123NE	Ley	hemoglobin	13.7 g/ml
298GA	Kelly	hemoglobin	9.87 c/mcl

(b) Answer from P_2

Fig. 2. Answers to the query q_1

DEFINITION 2 $Y-$VALUES. *Let mt be a mapping table from X to Y. $Y-$values, denoted as $Y_{mt}(x)$, with which a particular value $x \in dom(X)$ is associated as follows:*
$Y_{mt}(x) = \{y | \exists t \in mt$ and there exists valuation ρ over mt such that $\rho(t[X]) = x$ and $\rho(t[Y]) = y\}$

The intuition behind this association of tuples is that a tuple $t_1 \in r_1$ such that $t_1[X] = x$ is associated with respect to the mapping table $mt(X, Y)$, only the tuples $t_2 \in r_2$ for which $t_2[Y] \in Y_m(x)$. Therefore, the mapping table mt can be considered as a condition to filter relation r_{12} that is subset of r_{12} contains only the tuples that mt associates the tuples of relations r_1 and r_2.
A mapping (x, y) in mt indicates that a tuple $t \in r_i$ such that $t[X] = x$ is associated with a tuple $t' \in r_j$ such that $t'[Y] = y$. Considering the existence of mapping tables between peers, the definition of the data sharing constraints between peers is defined as follows.

EXAMPLE 1. *Consider Figure 1 where two peers P_1 and P_2 with relational schemas $R_1(patid, name, testname, result)$, $R_2(ohip, pname, test, result)$. Assume that P_1 is connected to peer P_2 by the data sharing constraint Σ_{12} which includes schema-level mapping*
$m^S_{2,1} = \forall xyzw(R_2(x, y, z, w) \rightsquigarrow R_1(x, y, z, w))$ and data-level mappings $m^D_{2,1} = \{mt_1, mt_2\}$.
The first mapping table expresses that a tuple t' in R_2 is related to a particular tuple t in R_1 w.r.t mt_1 such that $t[X'] \in \pi_{X'}(rho(mt_1))$ and $t'[X] \in Y_{mt_1}(t[X'])$. Figure 1 shows that patient with ohip='123NE' at P_2 is related to the patient with patid='101' at P_1 w.r.t mt_1.

3. CONSISTENT QUERY ANSWERING

A query in a peer data sharing system retrieves answers from the peer where the query is initiated and form peer's acquaintees. The consistent answers from acquaintees depends on the data sharing constraints. In the following the notion of consistent query answering is explained.

EXAMPLE 2. *Consider a query q_1 at peer P_1 below:*
"select patid, name, testname, result from R_1 where testname = C05187"
In usual case, the answers from P_1 and P_2 to the query q_1 is shown in Figure 2. The answer is not consistent w.r.t mapping tables $\{mt_1, mt_2\}$ since the second tuple in the answer of P_2 does not satisfy mt_1. That is the patient with ohip value "298GA" in the answer at P_2 is not mapped to any patient at P_1 although the tuple satisfies the second mapping table mt_2. The expected consistent answer to the query is shown in Figure 3.

Authors in [16] shows a technique to obtain consistent answers using repair semantics in inconsistent data databases during data integration. The formal definition of repair of a database D is the following.

101	Ley	C05187	12.6 g/ml

(a) Answer from P_1

123NE	Ley	hemoglobin	13.7 g/ml

(b) Answer from P_2

Fig. 3. Consistent answers for the query q_1

DEFINITION 3 REPAIR [16]. *Let D be a database with integrity constraints IC. A database D' is a repair of D with respect to IC if:*

—*$D' \models IC$, and*

—*there is no repair database D'' such that $D'' \models IC$ and $\triangle(D, D'') \subset \triangle(D, D')$, where $\triangle(D, D') = (D - D') \cup (D' - D)$ is the symmetric differences (also called distance) between two databases D and D'.*

This repair concept is originally developed in the area of consistent query answering in an inconsistent database. The repair based on data sharing constraint can produce multiple solutions.

DEFINITION 4 CONSISTENT ANSWERS [16]. *Let D be database and IC be a set of integrity constraints. Let q be a query over D. A tuple t is a consistent answer to q w.r.t IC if t is an answer to query q in every repair D' of D w.r.t IC.*

In a peer data sharing system, the repair of data at P_i with the data of its acquaintee P_j creates a solution instance, called *peer solution*, that satisfies the data sharing constraint between P_i and P_j. Authors in [15] proposed the semantics of consistent answers in peer data sharing systems. However, no mechanism is proposed to find consistent answers. This paper shows a technique how to achieve consistent answers using query translation. The semantics of consistent query answering is briefly describe below.

DEFINITION 5 PEER SOLUTION. *Let $\mathcal{P}$ be a data sharing system. Consider a peer P_i in $\mathcal{P}$ with a database D_i, and D' is a database instance on schema $\bigcup_{P_j \in \mathcal{N}(P_i)} R_j$. $\mathcal{N}(P_i)$ is the set of acquaintees of P_i. Let D'' is a repair on database instance $D_i \cup D'$ w.r.t $\bigcup_{P_j \in \mathcal{N}(P_i)} \Sigma_{ij}$. D'' is a peer solution for P_i if: (a) $D'' \models \bigcup_{P_j \in \mathcal{N}(P_i)} \Sigma_{ij}$ and (b) there is no instance D''' that satisfies (a) and such that $D''' \subset D''$.*

DEFINITION 6 CONSISTENT ANSWER. *Given a query q to a peer P_i, a ground tuple t is in consistent answer iff t is an answer to q in every possible peer solution r w.r.t Σ_{ij}.*

EXAMPLE 3. *Considering the setting in Example 1 and the query q_1 in Example 2. There are two solutions using repairs for making databases in P_1 and P_2 consistent w.r.t the mapping tables mt_1 and mt_2. The solutions are shown in Figure 4 and 5.*

It seems that in order to obtain consistent answers according to the solution concept, the peers need to physically change data, but this is not applicable in a peer to peer context. Actually, the notion of solution is used as an auxiliary notion to characterize the semantically correct answers from P_i's point of view. Ideally, P_i should be able to obtain consistent answers just by querying the already available local instance and instances of acquaintees which may be inconsistent w.r.t data sharing constraints. Given the solutions, according to the semantics of consistent answers the consistent answer is shown in Figure 6. The tuples exist in both the solutions.

patid	name	testname	result
101	Ley	C05187	12.6 g/ml
102	Kelly	C05187	9.87 c/mcl

(a) Updated relation R_1 at P_1

ohip	pname	test	result
123NE	Ley	hemoglobin	13.7 g/ml
298GA	Kelly	hemoglobin	9.87 c/mcl
256GA	John	whitebloodcount	8.37 c/mcl

(b) Instance at P2

patid	ohip
101	123NE
102	298GA

(c) Updated Mapping table mt_1

testname	test
hemoglobin	C05187

(d) Mapping table mt_2

Fig. 4. Solution 1

patid	name	testname	result
101	Ley	C05187	12.6 g/ml

(a) Instance at P_1

ohip	pname	test	result
123NE	Ley	hemoglobin	13.7 g/ml
256GA	John	whitebloodcount	8.37 c/mcl

(b) Updated instance at P_2

patid	ohip
101	123NE

(c) Mapping table mt_1

testname	test
hemoglobin	C05187

(d) Mapping table mt_2

Fig. 5. Solution 2

101	Ley	C05187	12.6 g/ml

(a) Answer from P1

123NE	Ley	hemoglobin	13.7 g/ml

(b) Answer from P2

Fig. 6. Consistent answers for the query q_1

There are mechanisms [7, 18, 19] for computing consistent answers in peer data exchange settings which only consider schema-level mappings that avoid or minimize the physical generation of repairs. This paper shows how to achieve consistent answers of queries in a value-mapped data sharing system where peers are related with value-level constraints. In particular, we propose an approach such that, given a query q at a peer, translate the query in such a way that returns answers which are consistent w.r.t to the constraint.

4. COMPUTING CONSISTENT ANSWERS

We consider the gossiping mechanism for query execution [13]. When a query is posed to a peer it is executed in the local database of the peer and is forwarded to the acquaintances of the current peer. Whenever a peer gets a query forwarded by another peer, it executes and forwards the query to it's acquaintances causing in turn the further propagation of the query. This process continues until all the reachable peers have been processed or a fixed number of propagations of the initial query has occurred.
We assume that each query is defined w.r.t. the schema of a single peer (called *initial peer*). The initial peer executes the query in a straight forward fashion and also propagates it to its acquainted peers. At the time of propagation, the query is transformed to get compatible with the peer schema of the acquaintance peer. The

local execution of the query and it's transformation and propagation to other peers goes on parallel. The transformation of the query is unfolding the definition of the external sources defined in the peer mappings between two peers. Each peer in the propagation path gets the answer from it's descendant, merges the answer with it's own answer, and then back propagates to it's ancestor. When an answer is back propagated to a peer, it is transformed according to the peer schema of the recipient peer, so that it can be merged with the answer of the local answer. Merging two answers is done by simply taking the inner union of them. Since the answers may need some semantic translation, instead of directly returning them to the initial peer, they are returned along the reverse way of query propagation.
There are two phases to return consistent answers to a query. First is the query translation and query propagation phase and second is the solution building phase. The query is translated in such a way that the translated query returns consistent answers from acquaintees. In the first phase, the initial peer executes the query in a straight forward fashion and propagates the query to its acquainted peers after translation w.r.t the vocabularies of the acquainted peers. The query translation is required since users submit queries w.r.t the schema of the local peer. In translation, the query is defined into a compatible form for the schema and data vocabularies of acquaintees. When an acquaintee receives the query it also performs the same task, i.e., local execution, translation, and propagation. The local execution of the query and its translation and propagation to other peers goes on parallel.
When a query is placed over the schema of a peer, the query has to be translated according to the schema of the acquainted peer's schema. The translated query is then passed to the acquainted peer to get the answer back to the peer where the query was originally posted. Tableaux representation [20] can help this query translation process.

EXAMPLE 4. *Consider the Example 1. The query q_1 and the data sharing constraint Σ_{21} are defined as follows:*

$q_1 : \pi_{patid,name,testname,result}(\sigma_{testname='C05187'}(R_1))$
$\Sigma_{21} : \forall xyzw(\exists(ywR_2(x',y,z',w) \wedge mt_1(x',x) \wedge mt_2(z',z)) \rightsquigarrow R_1(x,y,z,w))$

The query q_1, posted against the peer schema of P_1, results in a global query $q_{P_1} = \{q_1^1, q_1^2\}$ where:

$q_1^1 : \pi_{patid,name,testname,result}(\sigma_{testname='C05187'}(R_1))$
$q_1^2 : \pi_{ohip,pname,test,result}(\sigma_{test='homoglobin'}(R_2 \bowtie mt_1 \bowtie mt_2))$

The queries q_1^1, q_1^2 are executed on the local instances of the peers P_1 and P_2, respectively to produce the answers $A_{1,1}^1$ and $A_{1,2}^2$ as follows:

101	Ley	C05187	12.6 g/ml

(a) $A_{1,1}^1$

123NE	Ley	hemoglobin	13.7 g/ml

(b) $A_{1,2}^2$

Fig. 7. Local answers of q_1^1 and q_1^2

Notice that the answer is consistent with respect to the mappings. Below, we show how query q_1^2 is generated using tableaux.

4.1 Query Translation for consistent answer by Tableaux

Figure 8 shows step by step how the query translation is done. First, query q_1 is represented to a tableaux T_{q_1} [20] as in Figure 8(b). The constraint Σ_{21} is converted to a $T_{\Sigma_{21}}$ as shown in Figure 8(c) and 8(d). Figure 8(e) depicts $T_{\Sigma_{21}}$ with an empty summary added to it. Each column of $T_{\Sigma_{21}}$ is compared to the corresponding column of T_{R_1} and whenever an mismatch is found, $T_{\Sigma_{21}}$ is changed accordingly. Summary of $T_{\Sigma_{21}}$ is also changed according to the summary of T_{q_1}. Final $T_{q_1^2}$, after modifying summary and rows, is shown in Figure 8(f). $T_{q_1^2}$ is then converted to the query q_1^2 as shown in Figure 8(g).

$q_1 \equiv \{x,y,z,p \mid \forall x,y,z,p(R_1(x,y,z,\text{‘}C05187'))\}$

(a) Query q_1

	patid	name	testname	result
	a_3	b_3	c_3	d_3
R_1	a_3	b_3	c_3	$C05187$

(b) Tableaux T_{q_1} for q_1

	ohip	pname	test	result	patid	testname
R_2	a_1	b_1	c_1	d_1		
m_1	a_1				a_2	
m_2			c_1			c_2

(c) Tableaux $T_{\Sigma_{21}}$ for LHS(Σ_{21})

	ohip	pname	test	result
R_2	a_1	b_1	c_1	d_1

(d) Tableaux $T_{\Sigma_{21}}$ for RHS(Σ_{21})

	ohip	pname	test	result	patid	testname
R_2	a_1	b_1	c_1	d_1		
m_1	a_1				a_2	
m_2			c_1			c_2

(e) Empty summary added to $T_{\Sigma_{21}}$ as query translation stage

	ohip	pname	test	result	patid	testname
	a_1	b_1	c_1	d_1		
R_2	a_1	b_1	c_1	d_1		
m_1	a_1				a_2	
m_2			c_1			$C05187$

(f) Final $T_{q_1^2}$ after modifying Summary and Rows

$$q_1^2 \equiv \{x',y',z',w' \mid \forall x',y',z',w'(R_2(x',y',z',w')\wedge mt_1(x',x) \wedge mt_2(z',\text{‘}C05187'\text{)}\}$$

(g) Query q_1^2 obtained from $T_{q_1^2}$

$$q_1^2 : \pi_{ohip,pname,test,result}(\sigma_{test='C05187'}(R_2, mt_1, mt_2))$$

(h) Final Query q_1^2 to P_2

Fig. 8. Query translation process of q_1

4.2 Solution building phase

The solution building phase starts at terminate peers and ends at initial peer. In the solution building phase, each peer in a query propagation path receives consistent answers from it's acquaintees where the query is propagated. After receiving consistent answers from all acquaintees, a peer builds its solution, called *peer solution* that satisfies the data sharing constraints of the peer's acquaintees.
After building the solution, consistent answers are produced and the result is propagated to the peer that has forwarded the query. This back propagation of consistent answers continues until the

Algorithm QueryTranslation
Input: A query q and constraint Σ_{ij}
from P_i to P_j with the mapping $\forall\vec{x}(\exists\vec{y}\varphi(\vec{x},\vec{y}) \overset{MT}{\rightsquigarrow} e(\vec{x}))$
Output: A translated query q' on P_j
begin
$T_q \leftarrow ConvertQueryToTableau(q)$ [20]
$T_{\Sigma_{ij}}$ = Convert Σ_{ij} to Tableau using algorithm BLM2MAT [21]
LHS($T_{\Sigma_{ij}}$)= left part of $T_{\Sigma_{ij}}$
RHS($T_{\Sigma_{ij}}$)= right part of $T_{\Sigma_{ij}}$
$T_{q'} \leftarrow LHS(T_{\Sigma_{ij}})$
$T_e \leftarrow LHS(T_{\Sigma_{ij}})$
add an empty summary to $T_{q'}$
for each column $C \in T_q.Columns$ do
temp=$T_e.Rows[R_2][C]$
for each row $R \in T_{q'}.Tags$ do
for each column $Col \in T_{q'}.Columns$ do
if $T_{q'}[R][Col] = temp$ then
if $IsNotBlank(T_q[Summary][C])$ then
if $IsConstant(T_q[e][C])$ then
$T_{q'}[R][Col] \leftarrow T_q[e][C]$
$T_{q'}[Summary][Col] \leftarrow T_q[e][C]$
else
$T_{q'}[Summary][Col] \leftarrow temp$
else
if $IsConstant(T_q[e][C])$ then
$T_{q'}[R][Col] = T_q[e][C]$
else
$T_{q'}[R][Col] = FreshNonDistinguished()$
$q' \leftarrow$ *ConvertTableauToQuery*$(T_{q'})$ [20]
return q'
end

Fig. 9. Algorithm for query translation

initial peer receives results from the acquaintees where the query is initially forwarded. When the initial peer receives data from the acquaintees it builds its own solution instances and returns the consistent answers to the user who initiated the query.

5. RELATED WORK

Authors in [7, 19] presented a semantics for obtaining consistent answers in peer data exchange systems. The semantics is based repair [16] semantics that is proposed to obtain consistent answers in inconsistent databases. The proposed work also goes in this direction. This paper considers value-mapped peer data sharing system where peers are related with schema-level and data-level constraints. Authors in [13] proposed a query translation algorithm considering that the peers are related with value-level constraints. However, the authors do not consider the case of consistent query answering.
The systems [8, 22, 23], proposed query processing using appropriate propagation techniques. However, obtaining consistent answers to queries is missing in the presence of the situations where peers may be inconsistent w.r.t mappings.

6. CONCLUSION

This paper proposed an approach for obtaining consistent answers of queries in a value-mapped peer data sharing system. Each peer in the system returns consistent answers at query time. The inconsistency results when data in peers do not satisfy mappings in mapping tables or when mapping are changed. Mainly, each peer translates a query in such a way that returns consistent answers from the acquainted peers. The proposed system assumes that peers are related with value-level constraints and acquaintance graph is acyclic. However, our future goal is to analyze the approach in the presence of cycles in mappings. Moreover, we like to implement and investigate our approach in a large peer data sharing system.

7. REFERENCES

[1] R. Fagin, P. Kolaitis, R. J. Miller, and L.Popa, "Data exchange: Semantics and query answering", In Theoretical Computer Science, 2005.

[2] A. Fuxman, P. Kolaitis, R. Miller, and W. Tan, "Peer data exchange", In PODS, 2005.

[3] C. Beeri and M. Y. Vardi, "A Proof Procedure for Data Dependencies", In Journal of the ACM, 31(4):718-741, 1984.

[4] M. Lenzerini, "Data Integration: A Theoretical Prespective", In PODS, 2001.

[5] J. D. Ullman, "Information Integration Using Logical Views", In Proc. of the Int'l Conf. on Database Theory (ICDT), pages 19-40, 1997.

[6] P. Ziegler and K. R. Dittrich, "Three Decades of Data Integration - All Problems Solved?", In Proc. of the World Computer Congress (WCC), pages 3-12, 2004.

[7] L. Bertossi and L. Bravo, "The Semantics of Consistency and Trust in Peer Data Exchange Systems", In Proc. International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR), 2007.

[8] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R.J. Miller, and J. Mylopoulos, "The Hyperion Project: From Data Integration to Data Coordination", In SIGMOD RECORD, 2003.

[9] L. Serafini, F. Giunchiglia, J. Molopoulos, and P. Bernstein, "Local Relational Model:A Logocal Formalization of Database Coordination", Technical Report, Informatica e Telecomunicazioni, University of Trento, 2003.

[10] P. Rodriguez-Gianolli, M. Garzetti, L. Jiang, A. Kementsietsidis, I. Kiringa, M. Masud, R. Miller, and J. Mylopoulos, "Data Sharing in the Hyperion Peer Database System", In VLDB, 2005.

[11] P. Bernstein, F. Giunchiglia, A. Kementsietsidis, and J. Mylopulos, "Data management for peer-to-peer computing: A vision" In WebDB, 2002.

[12] A. Kementsietsidis, M. Arenas, and R.J. Miller, "Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues", In SIGMOD, 2003.

[13] A. Kementsietsidis and M. Arenas, "Data Sharing Through Query Translation in Autonomous Sources", In VLDB, pages 468-479, 2004.

[14] M. Masud, "Query translation between sources in peer-to-peer database sharing systems", In Intl. Journal of Knowledge Engineering and Data Mining, Vol 1(2), pp.115-146, 2010.

[15] M. Masud and S. Aljahdali, "Relational Peer Data Sharing Settings and Consistent Query Answers" In Intl. Journal of Computer Science and Engineering (IJCSE), Vol 2(9), 2010.

[16] M. Arenas, L. Bertossi, and J. Chomicki, "Consistent Query Ansers in Inconsistent Databases", In PODC, 1999

[17] L. Bertossi and J. Chomicki, "Query Answering in Inconsistent Databases", In Proc. Logics for Emerging Applications of Databases, 2003.

[18] L. Bertossi and L. Bravo, "Information Sharing Agents in a Peer Data Exchange System", In Proc. First International Conference on Data Management in Grid and P2P Systems (Globe), 2008

[19] L. Bertossi and L. Bravo, "Query Answering in Peer-to-Peer Data Exchange Systems", In Proc. International (EDBT) Workshop on Peer-to-Peer Computing and DataBases (P2P&DB), 2004

[20] A. Rahman, M. Masud, I. Kiringa, and A. El Saddik, "A Peer Data Sharing System Combining Schema and Data Level Mappings", Int. Journal of Semantic Computing, Vol. 3(1): 105-129, 2009.

[21] A. Rahman, M. Masud, I. Kiringa, and A. El Saddik, "Tableaux-Based Optimization of Schema Mappings for Data Integration", Journal of Intelligent Information System, Vol 38(2), pp. 533-554, 2012.

[22] I. Tatarinov, Z. Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, and X. Dong, "The piazza peer data management project", In ICDE, 2003.

[23] W. S. Ng, B. C. Ooi, K. L. Tan, and A. Y. Zhou, "PeerDB:A p2p-based system for distributed data sharing", In Data Engineering, 2003.