Bi-Level Mapping: Combining Schema and Data Level Heterogeneity in Peer Data Sharing Systems

Md. Anisur Rahman, Md. Mehedi Masud,
Iluju Kiringa, and Abdulmotaleb El Saddik

Site, University of Ottawa
800 King Edward Road, Ottawa, Canada
{mrahman,mmasud,kiringa,elsaddik}@site.uottawa.ca

Abstract. Peer data sharing systems use either schema-level or data-level mappings to resolve schema as well as data heterogeneity among data sources (peers). Schema-level mappings create structural relationships among different schemas. On the other hand, data-level mappings associate data values in two different sources. These two kinds of mappings are complementary to each other. However, existing peer database systems have been based solely on either one of these mappings. We believe that if both mappings are addressed simultaneously in a single framework, the resulting approach will enhance data sharing in a way such that we can overcome the limitations of the non-combined approaches.
In this paper, we introduce a model of a peer database management system(PDMS) which uses a mapping that combines schema-level and data-level mappings. We call this new kind of mapping *bi-level mapping*. We present the syntax and semantics of bi-level mappings. We also provide a query evaluation procedure for the PDMS that uses the bi-level mappings.

Key words: Model of Peer Data Sharing System, Schema mapping, Data mapping, Query Translation

1 Introduction

Designing a system for integrated access to distributed and heterogeneous information sources, e.g. federated database systems, distributed database systems, and peer database management systems is an important research area. The main goal of any *data integration* system [10, 7, 6, 5, 4, 3, 14] is to combine data residing at different sources, and to give users an amalgamated view of these data. In a federated database system, a global schema is defined against the data sources of all sites which gives a unique logical view to the global users for accessing the underlying data sources [2]. Users pose a query on the global schema and the query is decomposed into subqueries that are distributed to the respective sites. The distribution is based on the mappings and the query vocabularies. Generally, the mappings between the sources and the global schema are established by schema mappings. The underlying assumption is that there is no data-level heterogeneity among the sources. Several strategies are used for defining the schema mappings between the mediated and local schemas including, *global-as-view* (GAV), *local-as-view* (LAV), and *global-and-local-as-view* (GLAV) [8]. In GAV

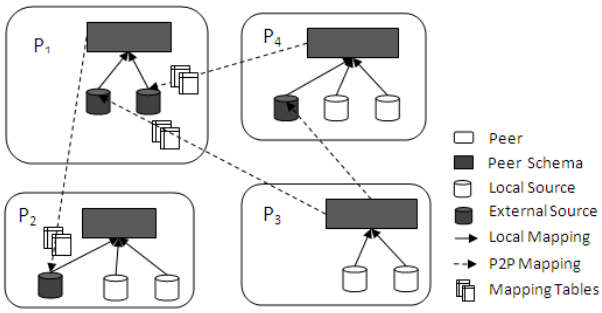


Fig. 1. General scenario of P2P system

approach, the mediated schema is described in terms of local sources. In LAV the local sources are described in terms of the mediated schema. GLAV is a combination of the two approaches where both GAV and LAV are used to integrate the mediated and local schema.

However, in a P2P system, there is no global schema, and it is not feasible to create a global schema for all the sources due to dynamic nature of peers. In addition, the heterogeneity among data sources in peers may be schema-level and data-level. A query in a P2P system is posed against the local schema and the query is translated for its neighbors based on the mappings [12]. In order to retrieve results of the query from the network, the query repetitively follows mappings through the network of peers until all relevant peers are visited [15]. When the query is executed in a peer, the partial result is returned to the query initiator. Finally, all the partial results are assembled to get an overall answer. During the execution of a query, a peer may play multiple roles. It may act as a data provider, a controller (acting like global component in a federated system), and a mediator (passing along queries without contributing to the result) [13].

A sample scenario of a peer-to-peer (P2P) system is depicted in Figure 1. As shown in figure, each peer has its own local source. The local source in a peer is designed independently during the creation of the local database of that peer. In order to provide a unique access view of local as well as remote data to the users, each peer defines a schema called *peer schema*. The result of a local query posed in a peer is produced from the local source of the peer. A peer also defines *external sources* that are used to access data from its acquainted peers. An external source is a view of a peer schema of an acquainted peer and is defined through some GLAV mappings called *P2P mappings* or *peer mappings*. Mapping tables may be used in P2P mappings for resolving data-level heterogeneity. Presence of mapping tables on the dotted lines (i.e. P2P mappings) in Figure 1 expresses the fact that, when data crosses the border between the source and destination peer, it is changed in the data-level using the mapping tables associated to the P2P mappings. Peer schema is defined in terms of the local sources and external sources by some GLAV [9, 8] mappings. These mappings are called *local mappings*.

There are some advantages of using external sources instead of making direct link between the peer schemas. Firstly, it can tackle the dynamic nature of a P2P system with ease. Whenever a peer finds an acquaintance peer for the first time, it creates some

external sources and links it both to its own peer schema and with the peer schema of the acquainted peer. Thus the external source can be treated by the target peer in the same way as local sources. When the source peer of an external source becomes unavailable in the network, it simply becomes an empty relation. Once an external source is created, any time the corresponding source peer becomes available it gets activated. Secondly, treating external sources as local sources, peer schema can be defined in a straight forward and autonomous way.

Since peers are fully autonomous to design their database and store data with their own format, heterogeneity among the peers may come in two forms: (i) schema-level and (ii) data-level. Notice that by creating the mappings through GLAV, the schema-level heterogeneity can be resolved. Authors in [16] proposed such a scheme for creating mappings between peers that resolve schema-level heterogeneity. However, the GLAV approach is not sufficient to resolve the data-level heterogeneity among peers. On the other hand, mapping table [11], used for resolving data-level heterogeneity between peers, is not sufficient for resolving schema-level heterogeneity.

1.1 Motivating Example: Need For Bi-level Mappings

Consider two peers P_1 and P_2 in a P2P system as shown in Figure 2. Assume that peers store employee information to be shared with each other. P_1 has a local source *Empl_List(Id, Name, Position, salary)*. The attributes *Id*, *Name*, *Position*, and *Salary* represent identification number, name, position, and the salary of an employee, respectively. Similarly, P_2 stores its employee information by the local sources *Employee(Id, Name, Jid)* and *Job_Desc(Jid, Job_Description)*. Attributes *Jid* and *Job_Description* represent the job identification number and the title of the job, respectively.

Now, assume that P_1 has an external source *E(Name, Position)* that illustrates that peer P_1 is interested in only the names and positions of employees stored in P_2. In order to give a unique access view to the data stored in P_1 and P_2, peer P_1 defines a peer schema $PS1$ which contains a single view N_P*(Name, Position)* for its users. Similarly, peer P_2 creates its peer schema $PS2$ which contains two views *P2E(Name, Jid)* and *P2J(Jid, Job_Description)*. This schema is designed considering its local source and other external sources from other peers (not mentioned in the figure).

From the Figure 2, we also notice that P_1 has a mapping table mt that maps the data vocabularies of the attribute *Job_Description* in source *Job_Desc* of P_2 with the attribute *Position* in source *Empl_List*. This mapping table is created since the two peers store job information using two different vocabularies. External source E in P_1 is defined as a view on the relations of the peer schema of P_2. In the following, we illustrate different situations that may occur when a query is posed to a peer. The examples show the need for the bi-level mappings that this paper advocates for P2P systems.

Example 1 (Considering only schema mappings). Assume that the mapping table mt is absent in peer P_2. In this case, peer P_1 has only the schema-level mappings with P_2. Suppose the query

$$q_1 : \pi_{Name}(\sigma_{Position='CEO'}(N\_P))$$

is posed at P_1 through its peer schema. Considering the mappings between N_P and $Empl\_List$, q_1 is translated for the local source at P_1 as

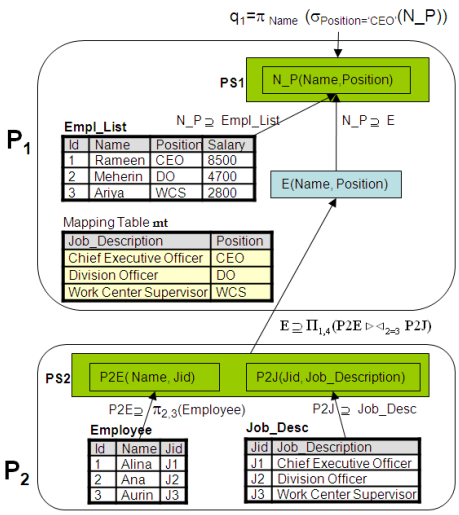


Fig. 2. Motivating example

$$q_1^1 : \pi_{Name}(\sigma_{Position='CEO'}(Empl\_List)).$$

Moreover, using the mappings between N_P and E, q_1 is translated for the external source at P_1 as

$$q_1^{1\prime} : \pi_{Name}(\sigma_{Position='CEO'}(E)).$$

Based on the mappings between E and the peer schema at P_2, query $q_1^{1\prime}$ is translated as

$$q_1^2 : \pi_{Name}(\sigma_{Job\_Description='CEO'}(P2E \bowtie P2J))$$

which is finally translated according to the local vocabulary of P_2 as

$$q_1^{2\prime} : \pi_{Name}(\sigma_{Job\_Description='CEO'}(Employee \bowtie Job\_Desc))$$

Notice that the final result of the query q_1 is { **Rameen**} which is returned only from the local source at P_1. If we consider 'CEO' and 'Chief Executive Officer' to be semantically equivalent then q_1 should extract 'Alina' from P_2. Due to absence of data-level mappings, the query can not produce this result. Now assume that the mapping table mt exists. Hence, $q_1^{1\prime}$ is translated for P_2 as

$$q_1^{2\prime} : \pi_{Name}(\sigma_{Position='CEO'}(Employee \bowtie Job\_Desc \bowtie mt)).$$

In this case, we get more results for the query q_1 and the complete answer to this query becomes { **Rameen, Alina**}

Example 2 (Considering only data mappings). In Figure 2 the external source E of P_1 is defined in terms of $P2E$ and $P2J$ of peer P_2. $Projection$ and $Join$ operators are used in that definition. Mapping tables are not expressive enough to express $Projection$ or $Join$. Schema mappings are needed for such association between two sources.

So, a mapping is necessary that is capable of dealing with both the syntactical (schema-level) and the semantic (data-level) heterogeneities at the same time.

In this paper,we present a new, generalized kind of mappings that combines both the schema-level as well as data-level mappings. We call these mappings *bi-level* mappings. We present semantics of the bi-level mappings and we give a model of a PDMS which allows bi-level mappings as a mean of bridging the heterogeneity gap between peers. If a user of a peer wants data from the local as well as from other peers in the network, she needs to pose the query on her local peer schema. We explain the query evaluation procedure by showing how the underlying bi-level mappings can be used to rewrite the query for execution in the local database and in remote ones.

The paper is organized as follows. Section 2 defines a model for a PDMS. In Section 3, we give the evaluation procedure of PDMS queries. Finally, in Section 4, we conclude with some future directions.

2 Model of a PDMS

A P2P system Π is defined by a pair $< \mathcal{P}, \mathcal{M} >$, where $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ is a set of peers and $\mathcal{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_n\}$ is a set of peer mappings. A set of mappings $M_i^j \subseteq \mathcal{M}_i$ in P_i defines the mappings between peer P_i and P_j. The construction of mappings M_i^j forms an acquaintance (*i, j*) between P_i and P_j.

Suppose a P2P system $\Pi =< \mathcal{P}, \mathcal{M} >$. Formally, a peer $P_i \in \mathcal{P}$ is a tuple, where $P_i = (PS_i, R_i, L_i, \mathcal{M}_i)$. Where,

- PS_i is the peer schema through which data in a peer is exposed to the external world.
- R_i is the set of sources comprised of local and external sources. We call it peer source or simply source.
- L_i is the set of GLAV local mappings which define the mappings between R_i and PS_i. Each local mapping, called *mapping assertion* (aka *tuple generating dependency*), in L_i has the form

$$\forall \boldsymbol{x}(\exists \boldsymbol{y}\varphi(\boldsymbol{x}, \boldsymbol{y}) \rightsquigarrow \exists \boldsymbol{z}\psi(\boldsymbol{x}, \boldsymbol{z}))$$

  where $\varphi(\boldsymbol{x}, \boldsymbol{y})$ and $\psi(\boldsymbol{x}, \boldsymbol{z})$ are conjunctive queries over the relations in R_i and PS_i respectively.
- $\mathcal{M}_i$ is a set of mappings, called bi-level mapping or *peer mappings*, that define the schema and data-level mappings between peers. Each mapping $m \in \mathcal{M}_i$ is a pair $< m_{j,k}^S, m_{j,k}^D >$, where:
  - $m_{j,k}^S$ is a GLAV mapping (practically GAV, since $s(\boldsymbol{x})$ is always a single relation) of the form
$$\forall \boldsymbol{x}(\exists \boldsymbol{y}\varphi(\boldsymbol{x}, \boldsymbol{y}) \rightsquigarrow s(\boldsymbol{x}))$$
    where $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is a conjunctive query over the peer schema of a peer P_j and $s(\boldsymbol{x})$ is the k^{th} external source of P_i.
  - $m_{j,k}^D$=MT=$\{mt_1, mt_2, \ldots, mt_q\} \subseteq MT_j^i$ is a set of mapping tables. MT_j^i denotes the set of mapping tables used to map data of P_j to data of P_i.

  m can alternatively be represented with the mapping assertion as follows:

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

We assumed that a curator with expertise in different domains is responsible for generating the mapping tables and the peer administrator maintains them in a peer. Schema mappings between two peers are initially created by the corresponding peer administrators when they agree to share data. Once created, the mappings are activated or deactivated depending on the presence of the corresponding peers in the network. Generating the mappings automatically is another research area and we did not address about this issue in this paper.

The semantics of $\overset{MT}{\rightsquigarrow}$ is described in the following section.

2.1 Semantics of local mappings

For each peer P_i, we introduce a first order logic (FOL) theory F_i, called peer theory, where alphabet contains all the relation symbols in a peer schema PS_i and the relations in R_i. The axioms of F_i include all the constraints of PS_i and one logical formula representing each local mapping in L_i. For a local mapping of the form $\forall \boldsymbol{x}(\exists \boldsymbol{y}\varphi(\boldsymbol{x},\boldsymbol{y}) \rightsquigarrow \exists \boldsymbol{z}\psi(\boldsymbol{x},\boldsymbol{z}))$ in L_i, a formula of the form $\forall \boldsymbol{x}(\exists \boldsymbol{y}\varphi(\boldsymbol{x},\boldsymbol{y}) \subseteq \exists \boldsymbol{z}\psi(\boldsymbol{x},\boldsymbol{z}))$ is added to F_i. F_i does not consider peer mappings. Thus, modeling a peer as a GLAV integration system becomes equivalent to modeling a FOL theory F_i (ignoring the peer mappings in $\mathcal{M}_i$).

2.2 Semantics of peer mappings

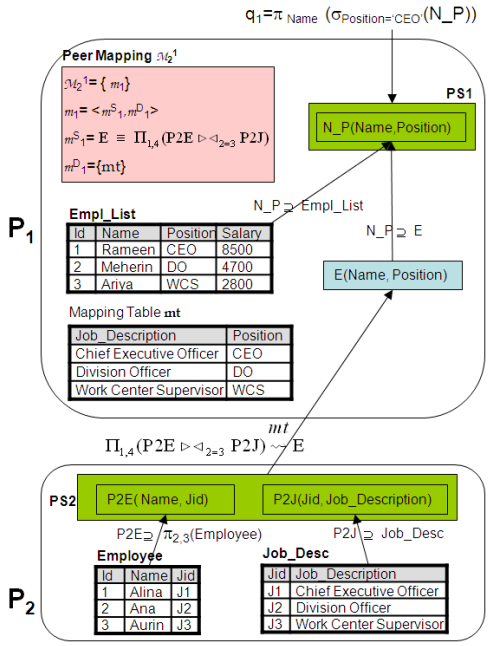
Similar to the local mappings, the semantics of peer mappings can also be given in terms of FOL. However, to incorporate the mapping tables, we will use notations and definitions provided below.

Definition 1. Given a tuple t and a set of attributes U, $t[U]$ denotes the values of tuple t corresponding to the attributes in U.

Definition 2 (Mapping Table). Assume that U_i and U_j are non-empty set of attributes in two peers P_i and P_j respectively. A mapping table $mt[\boldsymbol{P},\boldsymbol{Q}]$ is a finite relation over the attributes $\boldsymbol{P} \subseteq U_i$ and $\boldsymbol{Q} \subseteq U_j$. A tuple $t = (\boldsymbol{a},\ \boldsymbol{b})$ in the mapping table indicates that the value $\boldsymbol{a} \in dom(\boldsymbol{P})$ is associated with the value $\boldsymbol{b} \in dom(\boldsymbol{Q})$. Variables can be used to simplify the expression of value associations. Consider a mapping table $m(L,R)$ where both the domain of L and R are same, say D. A tuple $(v, D-v)$ in m, where v is a variable, can be used to denote that any value of L can be mapped to any value of R except to itself.

Definition 3 (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 appear.

Definition 4 ($Map(mt,\boldsymbol{p})$). Let $mt[\boldsymbol{P},\boldsymbol{Q}]$ be a mapping table from $\boldsymbol{P}$ to $\boldsymbol{Q}$ and $\boldsymbol{p}$ is an element of the domain of $\boldsymbol{P}$. $Map(mt,\boldsymbol{p})$ returns a set of values $\boldsymbol{\Phi}$. $\boldsymbol{q} \in \boldsymbol{\Phi}$ if for



(a) Bi-Level mappings for P2P mappings

E(Name, Position)

Name	Position
Alina	CEO
Ana	DO
Aurin	WCS

N_P(Name,Position)

Name	Position
Rameen	CEO
Meherin	DO
Ariya	WCS
Alina	CEO
Ana	DO
Aurin	WCS

(b) Extension of E and N_P

Fig. 3. Bi-Level Mapping Example

some $t \in mt$ there exists a valuation ρ such that $\rho(t[\boldsymbol{P}]) = \boldsymbol{p}$ and $\rho(t[\boldsymbol{Q}]) = \boldsymbol{q}$. If the mapping for the value $\boldsymbol{p}$ is not defined in mt and type of $\boldsymbol{P}$ and $\boldsymbol{Q}$ matches then $\boldsymbol{\Phi} = \{\boldsymbol{p}\}$. If neither $\boldsymbol{p}$ is mapped to any value in mt, nor type of $\boldsymbol{P}$ and $\boldsymbol{Q}$ matches, then $\boldsymbol{\Phi} = Null$.

Definition 5 (*Augmentation function* τ)**.** *Let* $\boldsymbol{x}$ *be a tuple whose schema contains the attributes* $\boldsymbol{P}$*. An augmentation function* $\tau(\boldsymbol{x}, \boldsymbol{P}, \boldsymbol{Q}, q)$ *returns a tuple* $\boldsymbol{x'}$*, where* $\boldsymbol{x'}$ *is exactly like* $\boldsymbol{x}$ *except the schema of* $\boldsymbol{x'}$ *has the attributes* $\boldsymbol{Q}$ *in place of* $\boldsymbol{P}$ *and* $\boldsymbol{x'}[\boldsymbol{Q}] = q$*.*

Definition 6. *Let* $mt[\boldsymbol{P}, \boldsymbol{Q}]$ *be a mapping table and* $\boldsymbol{x}$ *be a tuple,* $mt[\boldsymbol{x}]$ *denotes a set of tuples obtained by replacing the values of* $\boldsymbol{P}$ *attributes of* $\boldsymbol{x}$ *by the corresponding mapped values of* $\boldsymbol{Q}$ *in* mt*. Formally,*

$$mt[\boldsymbol{x}] \equiv \{\tau(\boldsymbol{x}, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{q}) \mid \boldsymbol{q} \in Map(mt, \boldsymbol{x}[\boldsymbol{P}])\}$$

Let $MT = \{mt_1[\boldsymbol{P_1}, \boldsymbol{Q_1}], mt_2[\boldsymbol{P_2}, \boldsymbol{Q_2}], \ldots, mt_n[\boldsymbol{P_n}, \boldsymbol{Q_n}]\}$ *be a set of mapping tables, where for any pairs of mapping tables* $(mt_i[\boldsymbol{P_i}, \boldsymbol{Q_i}], mt_j[\boldsymbol{P_j}, \boldsymbol{Q_j}])$*,* $mt_i \neq mt_j \implies \boldsymbol{P_i} \neq \boldsymbol{P_j}$*, then* $MT[\boldsymbol{x}]$ *denotes a set of tuples resulted from transformation of* $\boldsymbol{x}$ *by all the member mapping tables of* MT*. Formally,*

$$MT[\boldsymbol{x}] \equiv \{\tau(\ldots\tau(\tau(\boldsymbol{x}, \boldsymbol{P_1}, \boldsymbol{Q_1}, \boldsymbol{q_1}), \boldsymbol{P_2}, \boldsymbol{Q_2}, \boldsymbol{q_2})\ldots, \boldsymbol{P_n}, \boldsymbol{Q_n}, \boldsymbol{q_n}) \mid \boldsymbol{q_1} \in Map(mt_1, \boldsymbol{x}[\boldsymbol{P_1}]) \wedge \quad \ldots \wedge \boldsymbol{q_n} \in Map(mt_n, \boldsymbol{x}[\boldsymbol{P_n}])\}$$

We have overloaded [ ] in many of our definitions; its meaning, however, will be clearly understood from the context.

Definition 7. *Let $\boldsymbol{x}$ be a tuple. $Schema(\boldsymbol{x})$ returns the schema of that tuple, i.e. it returns the set of attributes whose values constituted the tuple.* Schema() *can be overloaded by providing its parameter as a* relation/view. *In this case, it would return the schema of the relation.*

Now we define the semantics of peer mappings. We already mentioned that a mapping assertion of a peer mapping is of the form:

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

Let us assume that the above assertion defines an external source s of peer P_i in terms of the peer schema of peer P_j. We say that an interpretation of the schema of P_i and P_j satisfies the assertion if that interpretation satisfies the following FOL formula

$$\forall \boldsymbol{x}\forall \boldsymbol{z}(\exists \boldsymbol{y}(\varphi(\boldsymbol{x},\boldsymbol{y}) \wedge \boldsymbol{z} \in MT[\boldsymbol{x}]) \equiv s(\boldsymbol{z}))$$

We can interpret a mapping as a definition of how the data of the external source would be instantiated by the data of other peers. The formula also tells us that before instantiating the external source, data is converted using the corresponding mapping tables of MT. However, if there is no data-level heterogeneity, no mapping table is needed. In that case, an empty mapping table ϕ is used in the assertion. In that case, a peer mapping is represented as $\forall \boldsymbol{x}(\exists \boldsymbol{y}\varphi(\boldsymbol{x},\boldsymbol{y}) \overset{\phi}{\rightsquigarrow} s(\boldsymbol{x}))$ which satisfies the FOL formula $\forall \boldsymbol{x}(\exists \boldsymbol{y}\varphi(\boldsymbol{x},\boldsymbol{y}) \equiv s(\boldsymbol{x}))$.

Example 3. Let us modify the peer mapping of Figure 2 by a bi-level mapping assertion m_1 which is expressed as follows.

$$\pi_{1,4}(P2E \bowtie_{2=3} P2J) \overset{\{\mathrm{mt}\}}{\rightsquigarrow} E$$

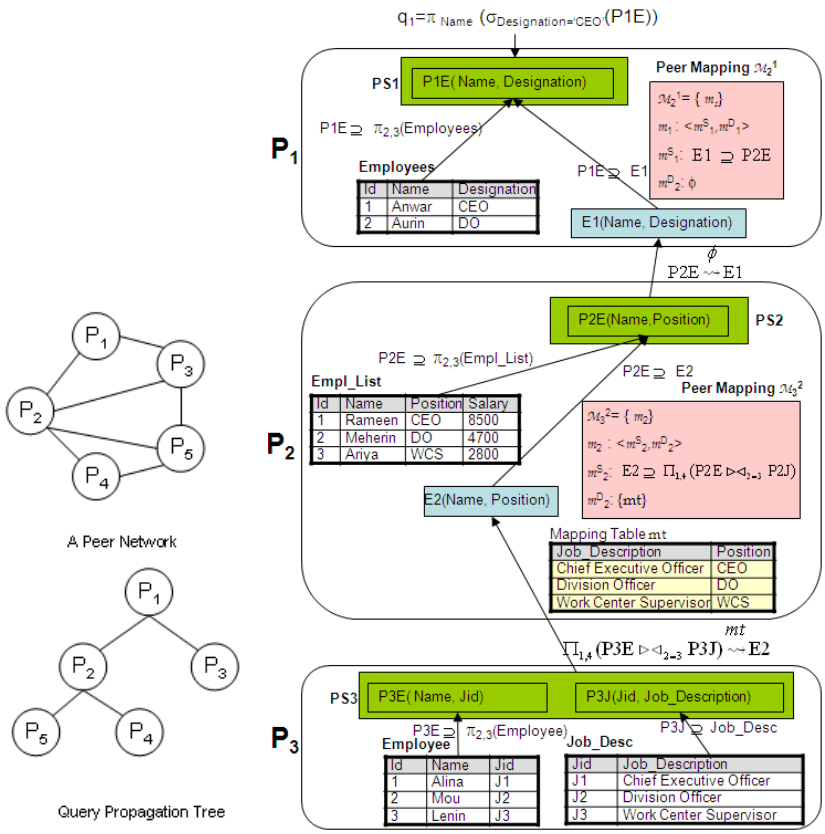
The new scenario is shown in Figure 3(a). To satisfy the assertion, the following formula has to be satisfied.

$$\forall rtt'(\exists s(P2E(r,s) \wedge P2J(s,t) \wedge (r,t') \in mt[(r,t)]) \equiv E(r,t'))$$

Given the source database in Figure 3(a), for satisfying the above formula, the extension of the intensional source $E(Name, Position)$ has to be as shown in Figure 3(b). Figure 3(b) also shows the ultimate extension of the intentional relation $N\_P(Name, Position)$ in the peer schema of peer P_1. Consequently, in response to the query $q_1 : \pi_{Name}(\sigma_{Position='CEO'}(N\_P))$ to peer P_1, the PDMS will return the result {**Rameen, Alina**}.

2.3 Semantics of a P2P system

We give the semantics of a P2P system Π in terms of a set of models that satisfy the local and peer mappings of Π. Let a source database $\mathcal{D}$ for Π be a disjoint union of a set of local databases in each peer P_i of Π. Given a source database $\mathcal{D}$ for Π, the set of models of Π relative to $\mathcal{D}$ is:



(a) Peer network and query propagation tree

(b) Merging example

Fig. 4. Query propagation tree and Merging example

$$sem^{\mathcal{D}}(\Pi) = \{\mathcal{I} | \mathcal{I} \text{ is a finite model of all peer theories } F_i \text{ relative to } \mathcal{D}, \text{ and } \mathcal{I} \text{ satisfies all peer mappings}\}$$

Given a query q of arity k posed to a peer P_i of Π, and a source database $\mathcal{D}$, the certain answers to q relative to $\mathcal{D}$ are

$$ans(q, \Pi, \mathcal{D}) = \{\boldsymbol{t} | \boldsymbol{t} \in q^{\mathcal{I}}, \text{for every } \mathcal{I} \in sem^{\mathcal{D}}(\Pi)\}$$

3 Query Evaluation

We adopt the gossiping mechanism for query execution [1]. 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. Considering how a query propagates through the peers of a peer network, the peer network can be converted to a shortest path spanning tree. We call this tree a *propagation tree*. Figure 4(a) shows a peer network and the corresponding query propagation tree with respect to peer P_1 where the query originates. A single peer is visited only once per execution of a query, i.e. in the propagation tree of the query a peer can appear at most once. Note that the propagation tree is constructed dynamically and depends on how the peers are acquainted to one another. When a peer forwards the query to one of its acquaintance, the former is said to be the ancestor of the later in the tree.

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 parallelly. 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 query result from it's descendant, merges the result with it's own result, and then back propagates to it's ancestor. When a result 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 result of the local result. Merging two results is done by simply taking the inner union of them. Since the results may need some semantic translation, instead of directly returning them to the initial peer, they are returned along the reverse way of query propagation. We borrowed the concept of *local query* and *global query* from [12]. A local query is executed using the data in the local peer. On the other hand, a global query uses the peer network to get the amalgamated result of the locally retrieved data (i.e. the result of the local queries). We now formalize the above notions.

Consider a P2P system $\Pi = (\mathcal{P}, \mathcal{M})$ with $\mathcal{P} = \{P_1, P_2, ..., P_n\}$. Assume I_i be an instance of P_i. Let $TranQ(q, M_i^j)$ be a function that translates the query q of peer P_i to the vocabulary (both data and schema) of peer schema of P_j according to the peer mappings M_i^j between P_i and P_j. Now a global query q_{P_i} with respect to a specific query q_i posed on the peer P_i is defined as a set of queries $\{q_i^1, q_i^2, \ldots, q_i^n, \}$ where q_i^j is defined as follows:

$$q_i^j = \begin{cases} q_i & \text{If } i = j \\ TranQ(q_i, M_i^j) & \text{If } i \neq j \text{ and there exists a mapping } M_i^j \text{ between peers } P_j \text{ and } P_i \\ TranQ(q_i^k, M_k^j) & \text{If } i \neq j \text{ and } P_i \text{ is indirectly mapped to } P_j \text{ through some intermediate peers and } P_k \text{ be the immediate predecessor of } P_j \text{ in the propagation path} \end{cases}$$

Let $TranV(V, M_i^j)$ be a function that translates the view V of the peer P_i to the vocabulary of the peer P_j using the peer mapping M_i^j. Moreover, let $q_i^j(I_j)$ denotes

the view resulted from application of query q_i^j to the local instance I_j of peer P_j. Semantics of q_i^j is given above. The result of the global query $Result(q_{P_i})$ is defined as $Result(q_{P_i}) = \cup_{j=1}^{n} I_{i,j}^i$. Where $I_{i,j}^k$ is defined as follows:

$$I_{i,j}^k = \begin{cases} q_i^j(I_j) & \text{If } j = k \\ TranV(q_i^j(I_j), M_j^i) & \text{If } i = k \neq j \text{ and there exists a mapping } M_j^i \text{ between peers } P_j \text{ and } P_i \\ TranV(I_{i,j}^l, M_l^k) & \text{If } i \neq j \neq k \text{ and } P_j \text{ is indirectly mapped to } P_i \text{ through some intermediate peers and } P_l \text{ be the immediate predecessor of } P_i \text{ in the propagation path} \end{cases}$$

Since all the local views are ultimately transformed according to the schema of the initial peer P_i, an inner union can be taken on the translated results. Our approach differs from the approach of [12] where outer union is taken among the local views of peers which provides far less meaningful information. Obviously, computation time can be saved if two or more mappings can be composed together to generate a direct mapping between two peers. We will address this issue in a separate work.

Example 4. Consider the example in Figure 4(b). A query q_1, posted against the peer schema of P_1, is as follows

$$q_1 : \pi_{Name}(\sigma_{Designation='CEO'}(P1E))$$

results in a global query $q_{P_1} = \{q_1^1, q_1^2, q_1^3\}$ where:

$$q_1^1 : \pi_{Name}(\sigma_{Designation='CEO'}(P1E))$$
$$q_1^2 : \pi_{Name}(\sigma_{Position='CEO'}(P2E))$$
$$q_1^3 : \pi_{Name}(\sigma_{Position='CEO'}(P3E \bowtie P3J \bowtie mt))$$

The queries q_1^1,q_1^2, q_1^3 are executed on the local instances of the peers P_1, P_2, and P_3, respectively to produce the results $I_{1,1}^1$,$I_{1,2}^2$ and $I_{1,3}^3$ as follows:

Name
Anwar

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

Name
Rameen

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

Name
Alina

(c) $I_{1,3}^3$

Fig. 5. Local results of q_1^1,q_1^2 and q_1^3

Peer P_3 converts $I_{1,3}^3$ to $I_{1,3}^2$ using the mapping M_3^2 and sends it to peer P_2. P_2 converts $I_{1,2}^2$ and $I_{1,3}^2$ to $I_{1,2}^1$ and $I_{1,3}^1$, respectively using the mapping M_2^1 and sends it to peer P_1. Note that mapping M_i^j is the inverse of mapping M_j^i. In Figure 4(b), all the mappings are not shown due to lack of space. The attribute `Name` is common in all the peers and so the converted views remains the same as the original views. P_1 combines the views $I_{1,1}^1$, $I_{1,2}^1$, and $I_{1,3}^1$ to produce the final result as {**Anwar**, **Rameen**, **Alina**}.

4 Conclusion

We have considered the problem of data sharing among heterogeneous data sources. The heterogeneity may come from differences in the schemas and/or from presenting the same (or semantically related) data with different vocabulary. Schema-level mapping can resolve the former while data-level mapping can be used for the later type of heterogeneity. In P2P systems, practically in most of the cases, both type of heterogeneity appears simultaneously. We present a peer mapping between two peers, called bi-level mapping, which can address both schema-level and data-level mappings at the same time. We give the semantics of bi-level mappings. Then we define a P2P system as a model that satisfies the local mappings as well as the bi-level mappings.

We did not take into account the local constraints that may not be satisfied by the external sources and consequently make the model inconsistent. Presence of cycles during evaluation of queries is another important issue that we did not address. Future research includes resolving these issues and finding algorithms for composition of the bi-level mappings.

References

1. Cuenca-Acuna, F.M., Peery, C., Martin, R. P., Nguyen,T.D.: PlanetP: using gossiping to build content addressable peer-to-peer information sharing communities. In: HPDC(2003)
2. Sheth, A.P., Larson, J.A.: Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Comput. Surv. 22,3,183-236(1990)
3. Miller, R.J., Hernndez, M., Haas, L.M., Yan, L., Howard, C.T., Fagin R., Popa, L.: The Clio Project: Managing Heterogeneity. SIGMOD Record(2001)
4. Boyd M., Kittivoravitkul, S., Lazanitis, C., McBrien, P.J., Rizopoulos, N.: AutoMed: A BAV Data Integration System for Heterogeneous Data Sources. In: CAiSE (2004)
5. Maluf, D.A., Ashish,A.: Information Integration Using Logical Views. In: SIGMOD (2005)
6. Ullman, J.D.: Information Integration Using Logical Views. In: ICDT(1997)
7. Chawathe, S., Garcia-Molina, H., Hammer,J., Ireland, K., Papakonstantinou, Y., Ullman, J. and Widom, J.: The TSIMMIS Project: Integration of Heterogeneous Information Sources. In: IPSJ (1994)
8. Lenzerini, M.: Data Integration: A Theoretical Prespective. In: PODS(2001)
9. Calvanese, D., Giacomo, G.D., Lenzerini, M., Rosati, R.: Logical Foundations of Peer-To-Peer Data Integration. In: PODS(2004)
10. Arenas,M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R.J., Mylopoulos, J.: The Hyperion Project: From Data Integration to Data Coordination. In: SIGMOD RECORD (2003)
11. Kemensietsidis, A., Arenas,M.: Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues. In: ACM SIGMOD (2003)
12. Kemensietsidis, A., Arenas,M., Miller, R.J.: Data sharing through query translation in autonomous sources. In: VLDB (2004)
13. Heese, R., Naumann, F., Roth, A.: Self-Extending Peer Data Management. In: BTW (2005)
14. Levy, A.Y., Rajaraman A., Ordille, J.J.: Querying Heterogeneous Information Sources Using Source Descriptions. In: VLDB (1996)
15. Halevy, A.: Answering queries using views: A survey. VLDB Journal, 10, 270-294 (2001)
16. Halevy, A.Y., Ives Z.G., Madhavan, J., Mork, P., Suciu, D., Tatarinov, I.: The Piazza Peer Data Management System. IEEE T.K.D.E. 16, 787–798 (2004)