An Effective Schema Mapping Model for Decentralized Network

Zhenhua Wang, Derong Shen * and Ge Yu

College of Information Science and Engineering, Shenyang, China

{wangzhenhua,shenderong,yuge}@ise.neu.edu.cn

Keywords: schema mapping; decentralized network; PDMS; uncertainty; query log.

Abstract. Schema heterogeneity among individual peers is an important issue in PDMS (Peer Data Management System). Schema mapping is a key technology to find the semantic matching relationship between schemas, and it plays an important role in PDMS. In this paper, we address the problem of schema mapping in PDMS, a typical decentralized network. Aiming at the limitations of previous methods, we propose a schema mapping model based on peer interest, schema structure information and query log, and we also considers the uncertainty of schema mapping during query propagation among peers. In our model, the peers with same interest organize a community and the schema mapping is handled in the interest community. We propose a query processing strategy in which the uncertainty of schema mapping in taken into account. In order to supplement the matching and reasoning, we mine the query log to discover the relationship between schema elements and refine the schema mapping further. Experimental results show that our method is feasible and effective for schema mapping in decentralized network.

Introduction

P2P systems have the advantages of scalability, autonomy and fault tolerance, which are widely applied in file sharing. The processed data in P2P system are usually simple, which do not contain semantics. Recently, some research efforts focus on data management with P2P paradigm. PDMS (Peer Data Management System) [1-3] emerged as a decentralized infrastructure for data sharing, which combines the decentralization and autonomy of P2P system and the rich semantics of DBMS. PDMS is a peer-to-peer network, which consists of a set of peers. Each peer in PDMS shares part of its own data, which has corresponding schema. The significant advantage of PDMS is none of single-point failure. The data are distributed over many peers, and if one peer fails, the query processing will still be continued. PDMS also has high scalability with the number of heterogeneous data sources. Another advantage of PDMS is the low cost of management as the peers create and manipulate their own database.

There are many challenges in PDMS, including high complexity, large scale, dynamic network topology and data heterogeneity. In PDMS, the peers can join or leave the network arbitrarily. There is no global schema in PDMS, and the query will be routed to many peers to obtain sufficient satisfactory results.

There is no global schema in PDMS, thus the problem of semantic heterogeneity emerged. To get enough satisfactory results, schema mapping must be handled. Schema mapping is a key technology for finding semantic relationship between different schemas, and it plays an important role in PDMS. The schema mapping is also a challenging problem. Due to the decentralization, scalability and dynamics of P2P systems, the traditional mapping methods which adopt centralized strategy are not fit for P2P, that is, it is not practical to map all the schemas to a single global schema. Therefore, in PDMS, the mapping between two different schemas is constructed directly and stored locally. The approach of schema mapping is to construct during query processing, which peer predetermine the mapping with neighbor nodes before query, and rewrite the query. When a peer issues a query, it will integrate the results from relevant peers through reformulated query into the final query results.

Aiming at the schema mapping in PDMS, we have the following observations:

Observation 1： The schemas of peers which have the same interests have high semantic similarity. The peers whose interests are the same usually belong to the same domain, so their schemas also possess high similarity. While in different domains, the peers whose interests are different, the differences of schemas are usually large.

Observation 2: Through analyzing mapping relationship of massive schemas, we find that correspondence of schema elements may influence each other during schema mapping.

Observation 3： P2P system has very large scale, and it is very difficult to find the proper peers which can contribute to the query in such a large scale P2P system. Because the peers only have local information, the query results from direct neighbors are not sufficient, and the query should be propagated in order to get better query results. When the query is propagated, there exists uncertainty, and this kind of uncertainty will be amplified during query propagation. When the query is routed between peers, the schema is transferred to get new mapping. The mismatch of attributes mapping will be amplified along with the query propagation.

P2P systems are featured with scalability and dynamics, which results in uncertainty during schema mapping. The uncertainty will be amplified during propagation among peers.

Observation 4: Even though the schemas are heterogeneous, the queries corresponding to the schemas in the same interest domain are semantically similar. If two schemas have high matching degree, then the elements in queries corresponding to the schemas also have high correspondence. The co-occurrence relationship of elements can be utilized to refine the correspondence of different schema elements.

Based on the above observations, we propose a community-oriented schema mapping model. We take into account the interests of users. The users in the same community have the same or similar interests, and the schema mapping is only processed in the same community. The query is only propagated to the peers with which have higher semantic match degree. The uncertainty during query propagation is also considered and is quantified.

The contributions of this paper are the following three aspects.

- The interests of peers are taken into account, and the accuracy of schema mapping is improved.
- The query log is utilized to refine semantic mapping.
- The uncertainty during schema mapping is taken into account, and the algorithm of query processing is given.

Related Work

The typical PDMS includes Edutella [4], Piazza [5], PeerDB [6], PIER [7], GridVine [8], etc. Edutella is a JXTA-based super peer PDMS. Edutella supports RDF-based top K queries. Edutella uses mediators to provide schema mapping. GridVine is a DHT-based Semantic Overlay Network (SON), and it uses P-Grid [9]. GridVine supports RDF-based queries, and schema mapping is on RDF schemas. GridVine supports a semantic gossiping for semant Piazza is a peer-to-peer data integration system, in which heterogeneous data can be shared in a distributed and scalable manner. It uses a decentralized schema mediation mechanism. Query processing in Piazza is based on a distributed collection of local schemas and pairwise mappings between peers. PIER is an Internet-scale query processor, and it is applied in P2P file sharing. PIER satisfies the principle of logic data independence of relational database. PIER also has a persistent storage. PeerDB uses agent to implement effective SQL-based query processing. PeerDB uses meta-data for each relation and attributes to implement decentralized schema mapping without a global schema.

There is a large amount of literature on schema mapping. These works mainly focus on the data sharing in decentralized environment. The main disadvantage of such methods is that they all require more manual intervention. In [10], the schemas of peers are classified, and the adaptive schema matching is achieved through query probing. Each peer shares pattern classification and it is selectively detected by detecting a query model for the shared adaptive pattern matching. S. Wu

proposed an adaptive multi-join solution for PDBMS in [11], which builds a Joining Search Tree and applies an approximate algorithm to find a good enough query plan. A model is introduced in [11] to manage the inherent uncertainty of automatic schema matching and its amplification during transitive mappings. Mweaver [12] is proposed as a system for sample-driven schema mapping. It automatically constructs schema mappings, in real time, from user-input sample target instances. CrowdMatcher [13] is a hybrid machine-crowd system for schema matching. In [14], a probabilistic model is developed to help to identify the most uncertain correspondences.

Schema Mapping in PDMS

A. *Related Concepts*

The data can be represented as relational tuple or other forms, we choose relational schema as our data form and mainly focus on the relational schema in this paper, the query is SQL-like. However our model is adaptable to other methods, such as Ontology-based method (RDF).

Relation Schema R={A1,…,Ak} A is attribute of relation schema.

Schema S={R1,…,Rn} is the set of relational schema. Each peer has a schema.

The schema mapping can also be represented as 3-tuple, <source schema, target schema, correspondence>, the correspondence is actually the mapping function.

Definition 1 Schema Mapping：the mapping function of schema Sx and Sy can be represented formally as：

Mapper({ei,u},{ei,v},Sx. Sy)= f

Where ei,u ∈Sx, ei,v∈Sy. {ei,u} and {ei,v} denote the set of elements of schema Sx and Sy, f is the mapping, we have：

{ei,u} {ei,v}

For any peer Pu, its neighbors are denoted as Nu, the goal of schema mapping is to find the following mapping relationship:

{ei,u} {ei,v} ( )

B. *Community-oriented Schema Mapping*

Definition 2 P2P Community: P2P community is a nonempty set of peers. P2P community has a global unique community ID which is the identifier of the community, and community description which represents the interest of the community. The community description is vector of topic or it is can be extracted from schema data. The peers in the same community usually belong to one specific domain and have similar interests.

Each peer only exchanges information with its neighbors. The peer sends message with a value of Time-To-Live (TTL), and obtains network structure from the responses.

If the peers p1 and p2 belong to the same community, the assumption is that the domains that p1 and p2 are interested in are similar. If p1 belongs to the domain of bookstore, p2 belongs to the domain of auto, obviously the mostly elements of these two peers can not match each other, it is no meaningful to schema mapping.

The construction of community is as follows: All the peers belong to one default community. When a peer joins the P2P network, it can join one or more communities according to its interests, and it also can create a new community. The peers among one community have the same or similar interest, and we assume that these peers belong to the same domain. The schema mapping is only processed between peers belonging to the same community.

The peer exchanges schema information with its semantic acquaintance to adapt the situation of the generation of new schema or the change of schema.

During probing and response, the schema information is as part of message. Through the schema information, each peer can construct schema mapping with its neighbors, and the peer choose k neighbors with the maximum schema mapping degree.

The number of peers in one community may be very large, and for one peer, it is unpractical to store all the information of other peers. The peer only maintains semantic links to the peers which

have higher semantic similarity in its community in order to improve the accuracy of schema mapping. The peer who is linked is called **semantic acquaintance**.

Definition 3 Semantic acquaintance: The peers which belong to the same community with the same peer and have higher semantic match degree are called the semantic acquaintance of peers. The peer maintains the semantic links which lies between itself and its semantic acquaintances.

The computation of peer semantic similarity is as follows. The frequency of query aiming to relation is as weight. For example, peer p1 includes relation R1, R2 and R3, and peer p2 includes relation R4 and R5. The match degree between R1 and R4 is 1, and the match degree between R3 and R5 is 0.8. Then the semantic similarity between p1 and p2 is 0.9 by default. If the query frequency of R1 is 6, the query frequency of R3 is 4, and then the semantic similarity between p1 and p2 is adjusted to 1*0.6+0.8*0.4=0.92.

The maintenance strategy of semantic acquaintance is as follows: Each peers maintains K (a tunable parameter) semantic acquaintances at most, the key is which peers should be selected as semantic acquaintances. The ideal situation is that the K semantic acquaintances have highest semantic relevance. When a peer just joins the network, it does not know which peers are relevant. Therefore, he can only get multiple neighbors through random probing, and selects the neighbors according matching degree in descendent order as semantic acquaintances. To get better semantic acquaintances and avoid high cost resulting from frequent probing, we only send probing message periodically to get new peers with higher semantic relevance, and executes the replace algorithm. During the query processing, the list of semantic list is refined.

C. *Refining Schema Mapping Using Query Log*

We use frequent pattern of data mining to identify common co-occurrence elements. The query attribute set and the query condition set are taken into consideration. FP-tree is employed to mine frequent item sets.

For different modes, the history query logs of users are mined to find frequent item sets. The query corresponding table is constructed according to the query logs, in which each query has an ID and the element (attributes) corresponds to the column. We get statistical frequency of elements appearing in the query, including the frequency of "select" and "where" conditions, and we generate frequent pattern tree.

First, establish the associated rule among query attributes in different schemas according to the historical query log, and then build the mapping rules of elements (attributes) based on the associated relationship. Utilizing the query log analysis, we can refine the mapping rules between elements further. We adopt top-down matching order in order to avoid matching errors. It is obvious that more query log more the guidance on schema mapping.

D. *Uncertainty in Schema Mapping*

Using schema mapping, we can rewrite a query q of source schema S to a new query q' of target schema. Each attribute in q has corresponding attribute in target schema. The peer which receives the query can determine whether to rewrite the query and propagate the query further. Therefore, a query can be rewrite along a chain continually, and a query chain is formed.

Definition 4 Schema Matching Graph(SMG): SMG=(V, E), where V is the finite nonempty set of vertices of the graph, V={S1,S2,…,Sn}, E is the set of edges of the graph, E={SMxij| (Si,Sj)ÎV SMxij } is the finite set of the matching matrixes, SMxij is the similar matching matrix of schema Si and Sj. The annotation on the edge is the matching degree of those two schemas.

For any $u \in V, v \in V$, if $(u,v) \in E$, then $(v,u) \in E$. For any $u \in V$, its neighbors are denoted as :$Nu=\{v \mid (u,v) \in E\}$.

Schema Matching Graph is a directional weighted graph, the peers are the vertexes of graph, the schema mappings are the edges of graph, and the schema mapping degree is the weight of edge. In PDMS, each peer maintains a list of neighbors. The link between peers are called semantic link.

Schema Matching Graph is constructed as follows. The peers exchange schema information with their neighbors, and handle schema mapping based on the above algorithm. During the query

propagation, the uncertainty will be amplified.References are cited in the text just by square brackets [1]. (If square brackets are not available, slashes may be used instead, e.g. /2/.) Two or more references at a time may be put in one set of brackets [3,4]. The references are to be numbered in the order in which they are cited in the text and are to be listed at the end of the contribution under a heading References, see our example below.

Query Propagation

The query can be represented as Q(queryid, pid, TTL, similarity, threashold, querybody), where Queryid is the identification of query and it is used to distinguish different queries. Pid is the identification of the initial peer which issues the query and the query results should be returned to the initial peer according to pid. TTL (Time-To-Live) is the maximum hop that the query can be forwarded. similarity denotes the current match degree. threashold is the minimum match degree, if the uncertainty is less than this value, the query propagation is stopped. Querybody denotes the actual query, which can be represented as triple: <R, A, C>, where R denotes relation, A denotes attributes, and C denotes conditions.

Definition 5 Query Reformulation: q'=QR(q), where q is the source query, and q' is the target query which has been reformulated. QR is the reformulation function.

The query is propagated among multiple peers, and is reformulated multiple times. Actually a chain is constructed during query propagation, which is defined as Query Chain. Query Chain is denoted as QC(q)=QRsm(……(QRsm(QRsm(q)))), where q is the initial query.

Definition 6 Schema Match Degree (SMD): The attributes are assigned weights according to the number of matched attributes and similarity of attributes, the average value is computed.

Definition 7 Peer Semantic Similarity: The schema match degree is considered, the query frequency responding to schema is as weight, and the average is computed.

The user issues a query, and the query is only propagated in the community which the peers belongs to. The steps of query processing are as follows:

(1) The user issues a query at the initial peer p_{ini}

(2) p_{ini} executes local query

(3) p_{ini} gets the community IDs which it belongs to.

(4) p_{ini} selects semantic acquaintance in the community and reformulates the query according to information of schema mapping

(5) p_{ini} sends the reformulated query to semantic acquaintance.

(6) The semantic acquaintances execute the query and return the results to p_{ini}, and also can propagate the query further.

(7) p_{ini} merges the query results and returns the results to the user.

The algorithm of query processing is shown in Alg. 1.

The query processing is as follows: the user issues a query at the initial peer p_{ini}, p_{ini} firstly executes the query locally. p_{ini} gets the community IDs which it belongs to. Then p_{ini} reformulates the query according to its knowledge about the information of its neighbors in its community. After query reformulation, the initial peer forwards the reformulated query to its neighbors with which the schema similarity is over the specified threshold. When the neighbors receive the query request, they process the query locally and return the query results to the initial peer. The peer then reformulates the query based on schema mapping, TTL=TTL-1, if the value of TTL is zero, the query propagation is stopped. If the value of TTL is above zero, recompute the value of similarity = similarity *schema mapping degree and select the peers whose similarity is over threshold and forward the query to those peers. The query is propagated until the TTL is zero or there doe not exist pees whose schema mapping degree are over threshold. When p_{ini} receives the returned query results from other peers, it merges the query results and returns the results to the user. p_{ini} evaluates the peers which have query results, and maintains the list of Semantic Neighbors.

Algorithm 1 *Q(queryid, pid, TTL, similarity, threashold, querybody)*
Input; TTL //hop
MIN_MATCH: maximum match degree
Output: the query results

```
1     D=computeMatchDegree(S1, S2)
2     If d<threshold then
3        Return
4     Else
5        Execute the query locally
6        Return the query results
7     End if
8     TTL = TTL-1
9     If TTL <=0 then
10       return
11    else
12      q'=reformulate(q)
13     Get Neighbor List
14     For each node in neighbor list
15      Call queryProcessing(q', TTL,
      matchdegree*d,  threshold)
16     End if
```

Here is an example of query processing, where the value of TTL is 3 and the value of MIN_MATCH is 0.5. Fig. 1 shows the tree structure generating during query processing.

Loop 1: TTL=3，Sim(S1, S2)=0.8, Sim(S1, S3)=0.5, Sim(S1, S4)=0.6, so S2, S3 and S4 are returned. S2 forwards the query to S6 and S7. S4 forwards the query to S8 and S9.

Loop 2: TTL=2, Sim(S1, S6)=Sim(S1, S2)*Sim(S2, S6) = 0.8*0.7=0.56, S6 meets the requirement and is selected. S6 forwards the query to S10 and S11.

Loop 3: TTL=1, Sim(S1,S10)=Sim(S1,S6)*Sim(S6,S10) =0.56*0.5=0.28

Sim(S1,S11)=Sim(S1,S6)*Sim(S6,S11)=0.56*0.9=0.504, so S11 is chosen.

The peers which satisfy the request are S2, S3, S4, S6 and S11.

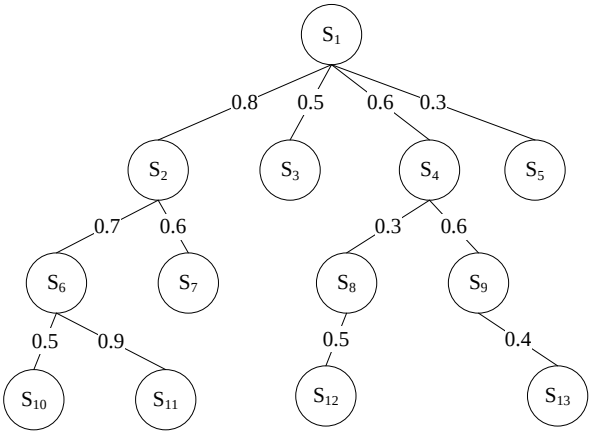

Fig. 1 Query propagation example

Experiments

In this section, we provide an evaluation of the proposed model. We conducted a series of experiments measuring the performance of the model.

A. *Experiment Environment*

We implemented a simulator of relatively large P2P network, and each peer is implemented as a distinct thread running within the same Java Virtual Machine.

We conducted the experiments on P2P network with different sizes. For each P2P network, we generated schema data, contents of the peers' databases and the peers' acquaintances. All the peers had an equal average number of acquaintances.

JXTA protocols use advertisement to describe and publish the resources of peers. The advertisement is described in the form of XML document, and it is a language-independent metadata structure. JXTA protocols provide Peer Group Advertisement, which describe the resources and the corresponding properties, such as name, ID, specification, etc.

Based on JXTA platform, the peer randomly sends probing advertisement message in the form of XML periodically in the community. The peers which receive the probing message send response. We extend JXTA protocol, the schema of peer is as part of response message.

In JXTA, there is the concept of Peer Group. Peer Group is the set of multiple peers, and it provides a series of services. The peers who have the same interests can construct a peer group. Each peer group has a unique identifier, PeerGroup ID. A peer can join multiple peer groups. JXTA provides the method to create and manage Peer Group. Before creating a Peer Group, a Group Advertisement should be created and published to the network in order to notify other peers finding this advertisement through JXTA discovery service.

We extend the Advertisement, and put the schema metadata and topic information into the Advertisement.

To implement our processing mechanism, we need to extend some services in JXTA. We use custom group service to create our user-defined JXTA Group.

B. *Experiment Results*

We evaluate our method from two aspects. First, we study the effectiveness of our schema mapping method for matching two schemas. Second, we evaluate the performance of schema mapping and query processing in P2P network.

We first create three communities, including bookstore, automobile, computer, etc. Each peer shares its schema and joins one community according to its interests. We study the effectiveness of query processing with the created schema mappings. We evaluate the performance of query processing in PDMS on quality of schema mapping and effectiveness of query processing.

Fig. 2 shows the experimental results of precision and Fig. 3 shows the experimental results of recall.

From Fig. 2 and Fig. 3, we find that our method (PSM, Peer Schema Mapping) is more effective than the naïve approach. The experiment results show that our method has higher precision and recall. Our experiments show that our method achieves high accuracy for schema mapping and query processing in decentralized network.

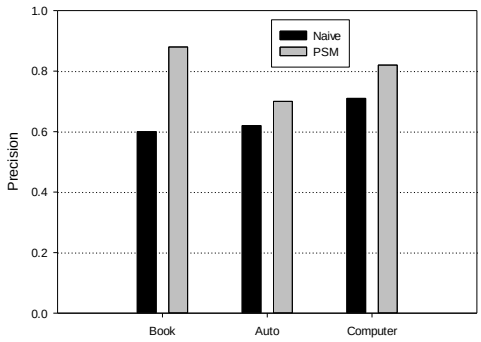

Fig. 2 Precision

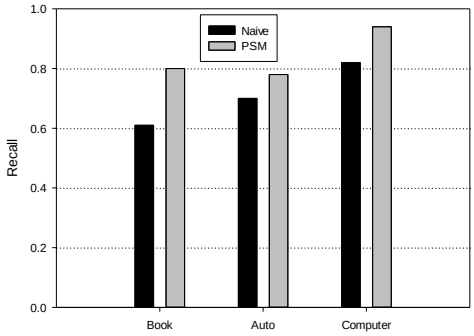

Fig. 3 Recall

Conclusions

In this paper, we propose an effective schema mapping model based on schema structure and known knowledge, and the model is suitable for PDMS. The model considers the uncertainty during query propagation, and refines the schema mapping through neutral network influence procedure. The experiments on real datasets show the high performance and accuracy of our model. In the future, we will introduce domain knowledge to PDMS, and improve the accuracy of schema mapping further.

Acknowledgment

This work is supported by the Fundamental Research Funds of the Central Universities (N130304002).

References

[1] A. Bonifati, P. K. Chrysanthis, A. M. Ouksel, K. Sattler, Distributed databases and peer-to-peer databases: past and present, SIGMOD Rec., 37(2008), p. 5--11.
[2] A. Eyal, A. Gal, Self Organizing Semantic Topologies in P2P Data Integration Systems, *ICDE*(2009), p. 1159-1162.
[3] A. Halevy, A. Rajaraman, J. Ordille, Data Integration: The Teenage Years, *VLDB*(2006), p. 9-16.
[4] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palm ́er, T. Risch, EDUTELLA: a P2P Networking Infrastructure Based on RDF, *Proc. of WWW*(2002), p. 604-615.
[5] I. Tatarinov, Z. Ives, J. Madhavan, A. Halevy, D. Suciu, N. Dalvi, X. L. Dong, Y. Kadiyska, G. Miklau, P. Mork, The Piazza peer data management project, SIGMOD Rec., 32(2003), p. 47--52.
[6] B. C. Ooi, K. Tan, A. Zhou, C. H. Goh, Y. Li, C. Y. Liau, B. Ling, W. S. Ng, Y. Shu, X. Wang, M. Zhang, PeerDB: peering into personal databases, 2003, p. 659--659.
[7] R. Huebsch, B. N. Chun, J. M. Hellerstein, B. T. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, A. R. Yumerefendi, The Architecture of PIER: an Internet-Scale Query Processor, 2005, p. 28-43.
[8] P. Cudr e Mauroux, S. Agarwal, A. Budura, P. Haghani, K. Aberer, Self-organizing schema mappings in the GridVine peer data management system , 2007, p. 1334--1337.
[9] K. Aberer, P. Cudr E Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, R. Schmidt, P-Grid: a self-organizing structured P2P system, SIGMOD Rec., 32(2003), p. 29--33.