Semantic-Based Query Routing and Heterogeneous Data Integration in Peer-to-Peer Semantic Link Networks

Hai Zhuge[1], Jie Liu[1, 2], Liang Feng[1, 2] and Chao He[1, 2]

[1]China Knowledge Grid Research Group, Key Lab of Intelligent Information Processing,
Institute of Computing Technology
Chinese Academy of Sciences, Beijing, 100080, China
{zhuge@ict.ac.cn}

[2]Graduate School of the Chinese Academy of Sciences
{lj@kg.ict.ac.cn, feng_liang@kg.ict.ac.cn, hc@kg.ict.ac.cn}

Abstract. A semantic link P2P network specifies and manages semantic relationships between peers' data schemas. The proposed approach includes a tool for constructing and maintaining P2P semantic link networks, a semantic-based peer similarity measurement approach for efficient query routing, and peer schema mapping algorithms for query reformulation and heterogeneous data integration. The advantages of the proposed approach include three aspects: First, it uses semantic links to enrich relationships between peers' data schemas. Second, it considers not only node but also structure in measuring the similarity between schemas so as to efficiently and accurately forward queries to relevant peers. Finally, it deals with semantic heterogeneity, structural heterogeneity and data inconsistency to enable peers to exchange and translate heterogeneous information in single semantic image.

1 Introduction

The original motivation for most early P2P systems such as Gnutella and Napster is file sharing [23, 24]. Peer data management systems (PDMS) provide us with a flexible architecture for decentralized data sharing. Usually, a PDMS consists of a set of peers, and each peer has an associated XML schema. Heterogeneous data integration for large-scale P2P networks is a challenging issue due to the autonomous, scalable, dynamic and heterogeneous data characteristics of peers.

Heterogeneous data management in a PDMS concerns the following three key issues:

1. How to autonomously identify semantically relevant peers.
2. How to accurately and efficiently route a query requirement initiated by one peer to relevant peers so as to avoid network flooding.
3. How to integrate heterogeneous data flows returned from different peers so as to provide users and other peers with a single semantic image data usage mode [20, 21], because P2P systems do not have a global schema like traditional data integration systems [15].

Previous research on P2P computing systems and peer data management systems mainly concerns data models for P2P databases, peer clustering, peer searching and query routing algorithms, and peer schema mediation mechanism. For example, the P2P-based system PeerDB for distributed data sharing [14], the scalable P2P lookup protocol [17], the local relational model for mediating between peers in a PDMS [3], approaches for controlling the distribution of peers to cluster and form super-peer networks [10, 13], the architecture supporting data coordination between peer databases [6], approaches to automatic schema matching [16], the semantic and algorithmic issues for mapping data in P2P systems [9], the solution to achieve semantic agreement in a P2P network [1], the generic schema-matching prototype Cupid [12], query reformulation algorithms for XML-based peers [5, 7, 8], and the approach for optimizing query reformulation in a PDMS [18]. But they are not the total solutions to the above three key issues.

This paper introduces the notion of P2P semantic link network to resolve the first issue. Semantic relationships between peers' data schemas are specified through semantic links [19]. Each peer is encapsulated as a soft-device (i.e., a software service mechanism [20]) that provides services to each other and to other virtual roles according to the content of their resources and the related configuration information through XML, SOAP (Simple Object Access Protocol) messages and WSDL (Web Service Description Language). A software tool has been implemented to assist users to construct and maintain a nested P2P semantic link network.

To resolve the second issue, this paper proposes an approach for measuring semantic similarity between peers. It considers not only the semantic similarity between nodes in peers' data schemas, but also semantic similarity between structures in peer schemas. Upon receiving a query, a peer will autonomously forward the requirement to relevant peers according to the types of the semantic links as well as the similarity between nodes and between structures of peer schemas.

To resolve the third issue, this paper establishes three mappings: *semantic node mapping*, *semantic clique mapping* and *semantic path mapping* to reformulate a query on source schema over target schemas. We apply technologies of QoP (Quality of Peers) such as response time, precision and recall to manage inconsistent data in returned data flows.

2 Approach Overview

A P2P *Semantic Link Network* (P2PSLN) is a directed network, where nodes are peers or P2PSLN, and edges are typed semantic links specifying semantic relationships between peers [19]. In a P2PSLN, each peer is an active and intelligent soft-device [20], which can dynamically and intelligently establish semantic connection with each other.

The role of a peer can be a server when it provides data, information and services, a mediator when forwarding query requirements, and a client when accessing information from other peers.

As depicted in Fig. 1, each peer in a P2PSLN has two main modules: *a communication module* and *a data management module*. Peers communicate with each other

through SOAP messages. Users can query a peer through GUI (Graphical User Interface) or SSeIQL (Single Semantic Image Query Language) — an SQL-like query language designed for P2PSLN-based peer data management.

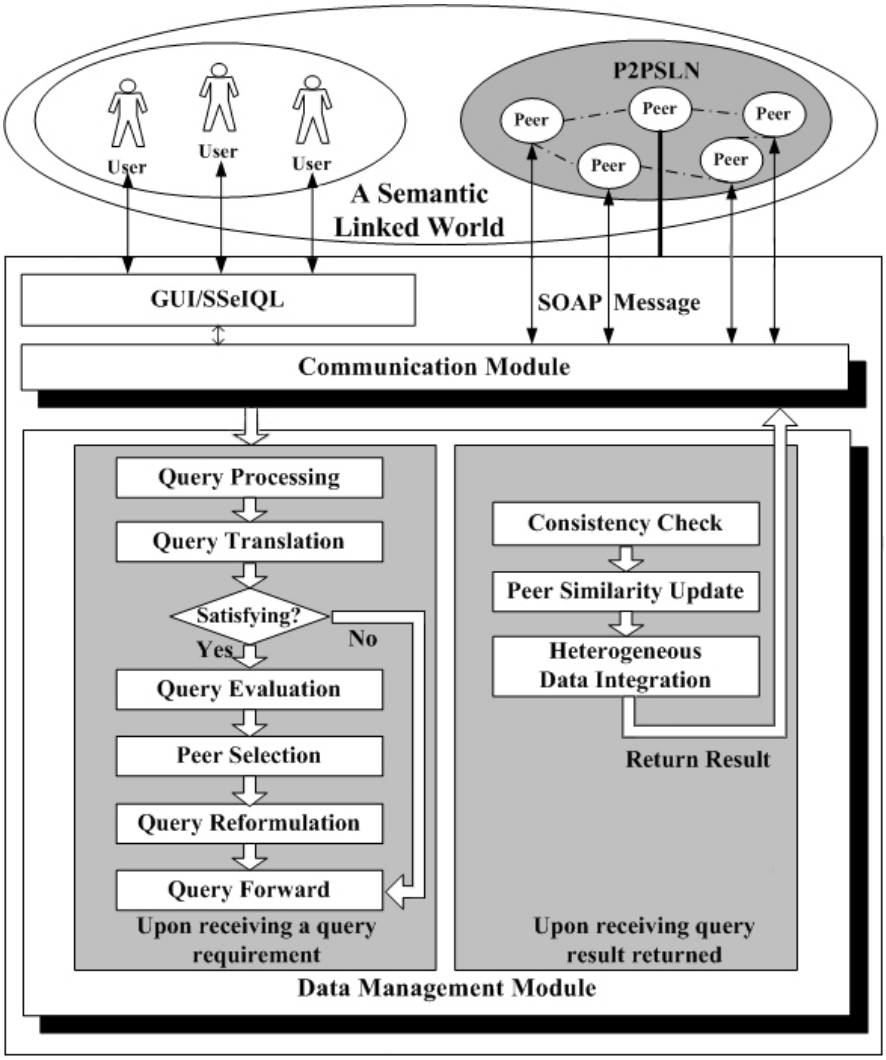

Fig. 1. An overview of a P2P semantic link network

The data management module of each peer is responsible for managing query requirements and returned query results. Upon receiving a query requirement, the data management module performs the following tasks:

1. *Query Processing* — To analyze query requirement and get query parameters.
2. *Query Translation* — To translate the query requirement against the XML schema of the current peer and check whether it can satisfy the requirement. If not, the re-

quirement will be forwarded to the successors who are likely to answer the query and to forward the query further.

3. *Query Evaluation* — To pose the query requirement on the current peer to retrieve answers.
4. *Peer Selection* — To select approximate successors according to the semantic relationship and semantic similarity between the current peer and the selected successors.
5. *Query Reformulation* — To reformulate a query on the current peer over schemas of its immediate successors.
6. *Query Forward* — To autonomously forward the query requirement to the successors highly similar to the current peer according to the routing policy and a predefined TTL (Time_to_Live) value.

Upon receiving query results returned from the successors, the data management module of the peer who initiates the query requirement will first analyze the result to detect inconsistent data in the returned data flows. For the successors who are likely to answer the query but return fewer matching results, the current peer will send SOAP messages to inquire whether there are some schema changes, and to update the schema mapping, semantic link type and similarity degree between them. Finally, the data management module will combine or join data matching query requirement in the returned data flows and provide users or peers with data from multiple sources in a uniform view.

3 P2P Semantic Link Network Model

3.1 Semantic Link

In a P2PSLN, a *semantic link* between two peers is represented as a pointer with a type (α) directed from one peer (predecessor) to another (successor). A *semantic link* can be one of the following types:

1. *Equal-to Link*, denoted as P_i—*equ*→P_j, states that the semantics of P_i is equal to that of P_j. The equal-to link is reflective, symmetric and transitive.
2. *Similar-to Link*, denoted as P_i— (*sim*, *sd*) →P_j, defines that the semantics of P_i is similar to that of P_j, and *sd* is the similarity degree between P_i and P_j.
3. *Reference Link*, denoted as P_i—*ref*→P_j, defines that the semantics of P_i refers to that of P_j.
4. *Implication Link*, denoted as P_i —*imp*→ P_j, defines that the semantics of P_i implies that of P_j. The implication link is transitive and can help the reasoning mechanism to find new semantic implication relationships.
5. *Subtype Link*, denoted as P_i —*st*→ P_j, defines that the semantics of P_j is a part of P_i. The subtype link has the transitive characteristic.
6. *Sequential Link*, denoted as P_i —*seq*→ P_j, defines that the content of P_j is the successor of the content of P_i.

7. *Empty Link*, denoted as $P_i \text{—} \varnothing \rightarrow P_j$, represents that there are no semantic relationships between P_i and P_j.
8. *Null Link* or *Unknown Link*, denoted as $P_i \text{—} N \rightarrow P_j$, represents that the semantic relationships between P_i and P_j are uncertain or unknown.

We can chain relevant semantic links to obtain uncertain semantic relations between peers according to a set of reasoning rules [19]. The heuristic rules suitable for connecting different types of semantic links in a P2PSLN are listed in Table 1, where $\alpha \in \{equ, sim, ref, imp, st, seq, \varnothing, N\}$ denotes the semantic link type between peers.

Table 1. Resasoning rules for P2P semantic link networks

No.	Rules
Rule 1	$P_i \text{—} equ \rightarrow P_i$
Rule 2	$P_i \text{—} equ \rightarrow P_j \Rightarrow P_j \text{—} equ \rightarrow P_i$
Rule 3	$P_i \text{—} equ \rightarrow P_j$, $P_j \text{—} equ \rightarrow P_k \Rightarrow P_i \text{—} equ \rightarrow P_k$
Rule 4	$P_i \text{—} equ \rightarrow P_j$, $P_j \text{—} \alpha \rightarrow P_k \Rightarrow P_i \text{—} \alpha \rightarrow P_k$
Rule 5	$P_i \text{—} imp \rightarrow P_j$, $P_j \text{—} imp \rightarrow P_k \Rightarrow P_i \text{—} imp \rightarrow P_k$
Rule 6	$P_i \text{—} st \rightarrow P_j$, $P_j \text{—} st \rightarrow P_k \Rightarrow P_i \text{—} st \rightarrow P_k$
Rule 7	$P_i \text{—} imp \rightarrow P_j$, $P_j \text{—} st \rightarrow P_k \Rightarrow P_i \text{—} imp \rightarrow P_k$
Rule 8	$P_i \text{—} imp \rightarrow P_j$, $P_j \text{—} ref \rightarrow P_k \Rightarrow P_i \text{—} ref \rightarrow P_k$
Rule 9	$P_i \text{—} st \rightarrow P_j$, $P_j \text{—} imp \rightarrow P_k \Rightarrow P_i \text{—} imp \rightarrow P_k$
Rule 10	$P_i \text{—} st \rightarrow P_j$, $P_j \text{—} ref \rightarrow P_k \Rightarrow P_i \text{—} ref \rightarrow P_k$
Rule 11	$P_i \text{—} N \rightarrow P_j$, $P_j \text{—} \alpha \rightarrow P_k \Rightarrow P_i \text{—} N \rightarrow P_k$
Rule 12	$P_i \text{—} \varnothing \rightarrow P_j$, $P_j \text{—} \alpha \rightarrow P_k \Rightarrow P_i \text{—} N \rightarrow P_k$

3.2 Operations on P2P Semantic Link Networks

A P2PSLN supports three types of operations: peer join, peer departure and peer stabilization.

1. *Peer Join*. When a peer P_i joins a P2PSLN, it will first identify the semantic relationship between itself and a peer P_j in the network, and then take P_j as its immediate successor by calling function 'P_i.*Join* (*P2PSLN*, P_j, α)', where α denotes the semantic relationship between P_i and P_j. The detail of each function is listed in Table 2. The semantic relationships between P_i and other peers in the current P2PSLN could be derived according to rules shown in Table 1. To find other successors, P_i will ask each successor P_k of P_j by calling function 'P_i. *FindSuccessors* (*P2PSLN,* P_j, α, P_k, β)'. If $P_i \text{—} \alpha \rightarrow P_j$, $P_j \text{—} \beta \rightarrow P_k \Rightarrow P_i \text{—} \gamma \rightarrow P_k$ satisfies the reasoning rules in Table 1, then P_i makes P_k as its successor, and calls function 'P_i.*FindSuccessors* (*P2PSLN,* P_k, γ, P_m, δ)' iteratively. After establishing the semantic relationships between P_i and its successors, P_i calls function 'P_i. *SchemaInquiry* (*P2PSLN*, P_j)' to acquire the XML schemas of each successor P_j. The process to measure similarity degree between peers with *Similar-to* link type will be illustrated in Section 5.

2. *Peer Departure*. When a peer P_i leaves a P2PSLN, it may notify its predecessors and successors before its departure. In turn, predecessor P_j will remove P_i from its successor list, delete the semantic links between P_j and P_i, and add each successor P_k of P_i as its own successor provided that: (1) $P_k \notin P_j$' successor list, and (2) there is a semantic relationship between P_j and P_k. Similarly, successor P_k will remove P_i from its predecessor list, delete the corresponding semantic links, and add each predecessor P_j of P_i as its own predecessor if: (1) $P_j \notin P_k$'s predecessor list, and (2) there is a semantic relationship between P_k and P_j.
3. *Peer Stabilization*. To ensure the up-to-date semantic links between peers, each peer P_i in a P2PSLN runs function 'P_i.*Stabilization* (*P2PSLN*, P_j)' periodically in the background and updates semantic link types, predecessor pointers and successor pointers accordingly. If P_j (i.e., the predecessor or the successor of P_i) exists in the network, it will notify P_i its existence and schema change information. If P_j does not exist in the current P2PSLN, P_i will remove P_j from its predecessor/successor list and modify its neighbor index accordingly. When the XML schema of a peer changes, it will autonomously notify its predecessors and successors the new schema through SOAP messages.

Table 2. Operations on P2P semantic link networks

ID	Operation	Function
1	P_i.*Join* (*P2PSLN*, P_j, α)	To take P_j as the successor of P_i and specify the semantic link type between P_i and P_j as α in a P2PSLN.
2	P_i.*FindSuccessors* (*P2PSLN*, P_j, α, P_k, β)	To deduce semantic relationships between P_i and P_k in a P2PSLN provided that P_i—α→P_j and P_j—β→P_k hold.
3	P_i.*SchemaInquiry* (*P2PSLN*, P_j)	To acquire XML schema of P_j in a certain P2PSLN.
4	P_i. *Departure* (*P2PSLN*)	To leave a P2PSLN.
5	P_i. *Stabilization* (*P2PSLN*, P_j)	To ask for the existence and schema change from P_j in a P2PSLN.

3.3 P2P Semantic Link Network Definition Tool

There are two kinds of basic elements in a nested P2PSLN: nodes and semantic links. A node can be either a peer or a P2PSLN (i.e. a component), while a semantic link denotes the semantic relationship and similarity degree between two peer schemas. We have developed a tool to assist users to construct and maintain a P2PSLN. A graphical interface of the definition tool is shown in Fig.2. Users can define a P2PSLN by clicking the operation buttons arranged at the top portion and drawing on the screen. The scalable and nested node hierarchy of the current P2PSLN is arranged

on the left column. The description for each peer (i.e., *PeerID*, *Peer Name*, *Peer IP*, *Peer Description*) and each semantic link (i.e., *Predecessor*, *Successor*, *Semantic Relationship*, *Similarity Degree*) is listed at the bottom.

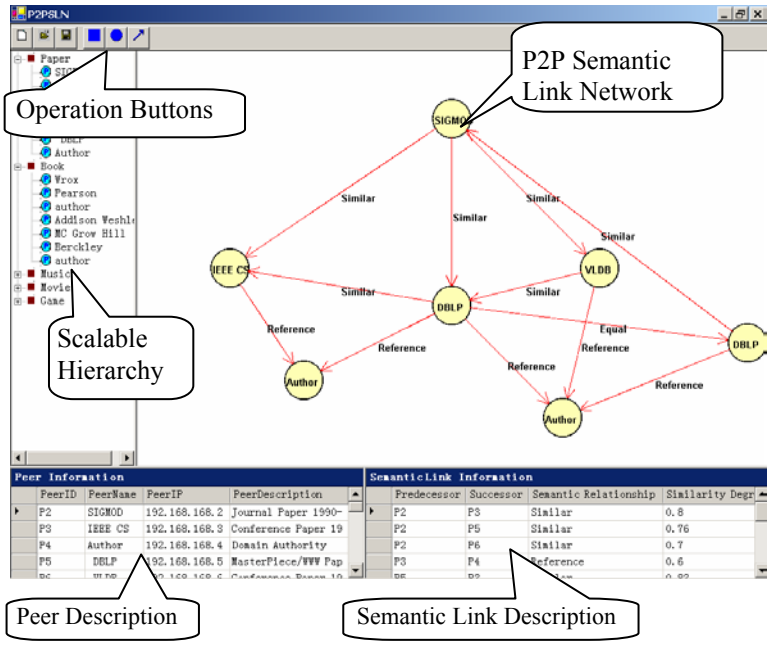

Fig. 2. An interface of the proposed P2PSLN definition tool

4 Peer Schema Mapping

Peer schema mapping is to resolve the issue of the semantic inconsistency between source schemas and target schemas. Upon receiving peer schemas through SOAP messages, a peer will traverse the schemas recursively in depth-first order and extract node and path information from the target, then carry out three types of mappings: semantic node mapping, semantic clique mapping and semantic path mapping.

4.1 Semantic Node Mapping

Semantic Node Mapping is to resolve the semantic inconsistency between nodes by mapping nodes in the source schema into nodes in target schemas. A peer encapsulates a global dictionary that defines a set of semantically related terms (synonymy, abbreviations, etc.) and the similarity degree between terms. After acquiring the target peer schemas, the source peer will automatically build mapping and similarity degree between nodes according to the definition in the global dictionary. The nodes in

source schemas and the mapping nodes in target schemas are called *Semantic Nodes* and *Semantic Mapping Nodes* when semantic links have been established between the source and the target. We also provide tools to enable users to manually modify the semantic node mappings automatically generated by the system, and to keep the new mappings by using a local dictionary.

4.2 Semantic Clique Mapping

A semantic clique represents the semantic structure such as the parent-child relationship and ancestor-descendant relationship between a set of closely related semantic nodes. The semantics of a node in a semantic clique is constrained by semantics of all nodes on the path from the root to it.

Semantic Clique Mapping is to identify semantic cliques (sub-trees that cover a set of closely related semantic nodes) and map each semantic clique in a source schema into the target schemas, where the mapping images are called *Semantic Mapping Cliques*. Semantic mapping nodes in a semantic mapping clique hold the semantic structure that the semantic nodes in a semantic clique hold.

To find the semantic cliques, we first divide all the semantic nodes in a source schema into a set of closely related sets, i.e., the semantic node set. The following algorithm is to identify the semantic clique corresponding to each semantic node set.

```
Algorithm SemanticCliqueRecognition (T1, T2, SN)

/* Given a set of closely related semantic mapping nodes, to
find semantic cliques in sub-tree rooted at T1 and semantic
mapping cliques in sub-tree rooted at T2 */

Input: T1, T2, SN={SN1,…, SNn} /* SN={SN1,…, SNn} is a set
of closely related semantic nodes, SN1 is a semantic node;*/

Output: SC={SC1, …, SCk}, SMC={SMC1,…, SMCk} /*Semantic
clique set in a source schema and Semantic mapping clique set
in target schemas*/

Begin
IF (T1= =Null)
THEN Return True;
R1=T1.FirstChild; Temp=True;
WHILE (R1 ! = NULL)
R2= Semantic- Mapping-Node (T2, R1); /* To find semantic
                                  mapping node of R1 in T2 */
IF (R2= =Null)
THEN Return False;
ELSE
 Temp=Temp And SemanticCliqueRecognition (R1, R2, SN);
 IF Temp== False
 THEN Return False;
 ELSE
  Add R1 To SC; /* add R1 to semantic clique set*/
```

```
    Add R2 To SMC; /*add R2 to semantic mapping clique set*/
    R1=T1.NextChild;
  END IF;
END IF;
END While;
Return Temp;
End
```

The Maximum Semantic Clique is the semantic clique that is not semantically included by any other semantic clique. *The Minimum Common Sub-tree* denoted as *MCS* (SC_1, …, SC_p, SN_1, …, SN_q) is the sub-tree that covers all the semantic cliques (SC_1,…, SC_p) and all the identified semantic nodes (SN_1,…, SN_q) not belonging to any semantic clique in a source schema. The root of the minimum common sub-tree is called the *Nearest Common Predecessor* of the involved semantic cliques and semantic nodes. Algorithms to find minimum common sub-tree are introduced in [11].

Besides the semantic cliques automatically identified by algorithm *Semantic-CliqueRecognition*, we have developed a tool to assist users to define semantic cliques that are required under certain circumstances. User interface to define semantic cliques and semantic mapping cliques is depicted in Fig. 3. The left-and-middle portion displays the source schema hierarchy and the corresponding graphical representation, while the right-and-middle portion corresponds to the target schema. The black nodes in the source schema form a user-defined semantic clique, while black nodes in the target schema are the corresponding semantic mapping cliques.

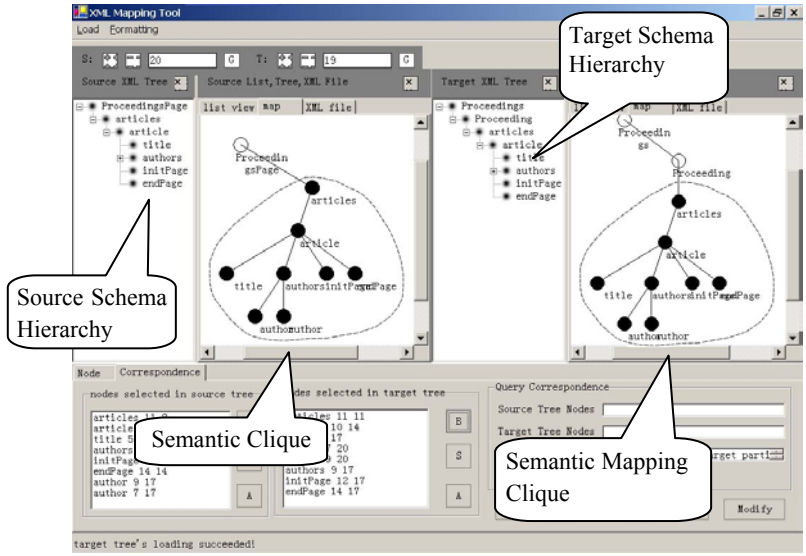

Fig. 3. User interface to define semantic cliques and semantic mapping cliques

Fig. 4 depicts the schema trees of SIGMOD proceedings and VLDB proceedings. The identified semantic nodes and semantic mapping nodes are the circles in the same color. The semantic clique, the maximum semantic clique and the minimum common sub-tree are denoted by the dashed close curves as described in the legend.

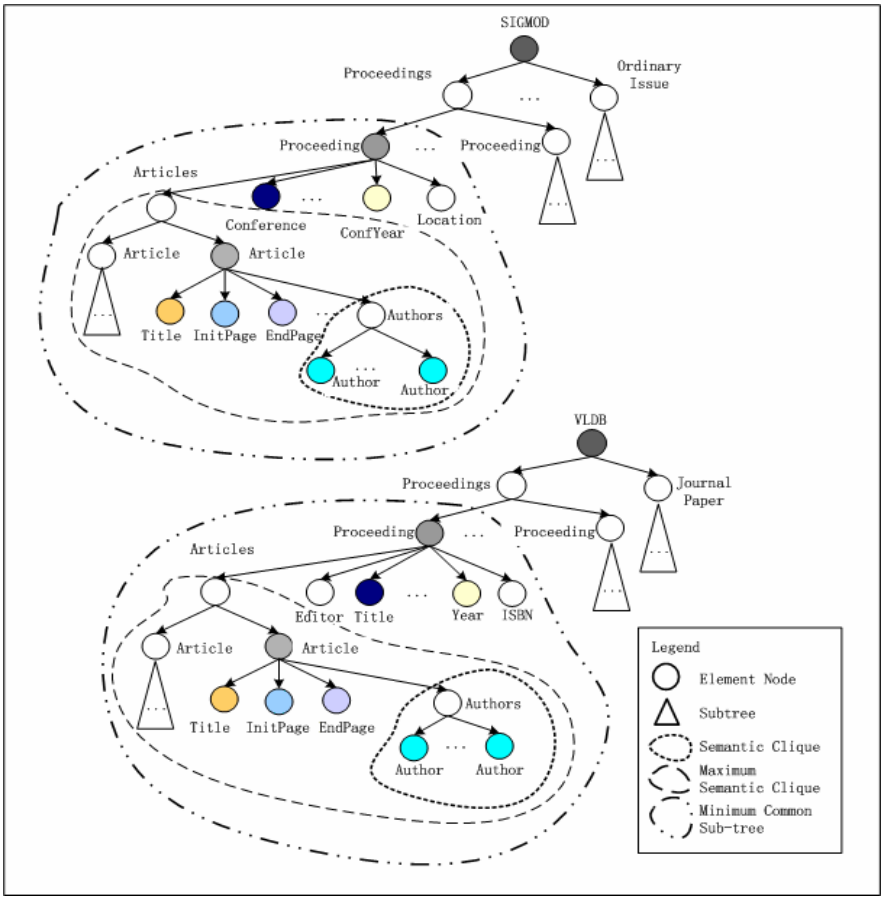

Fig. 4. XML trees conforming to the schemas of proceedings of ACM SIGMOD and VLDB

4.3 Semantic Path Mapping

Semantic Path Mapping is to map each semantic path from the root to the semantic nodes in the source schema into the paths in target schemas (i.e., semantic mapping paths). Let *Semantic-Path* (N_i) be the path from *Root* (N_i) to semantic node N_i in a source schema, and *Semantic-Mapping-Path* (N_i) be the mapping path of *Semantic-Path* (N_i) in target schemas. The process of semantic path mapping can be described as follows:

```
Algorithm SemanticMappingPath (Pi,Ni)

Input: Pi /*Schema of Pi*/;
       Ni /*Semantic Node in Pi*/;

Output: Semantic-Mapping-Path (Ni);

Step 1: For each node on Semantic-Path (Ni)
        Find semantic mapping nodes in target schemas
        according to node mapping definition in
        global dictionary and local dictionary;

Step 2: Connect semantic mapping nodes in target
        schemas to form an identified path;

Step 3: IF the identified path matches a path SMPath
        in target schemas
        THEN Return (SMPath);
        ELSE
          Extend the identified path by replacing par
          ent-child relations with ancestor-
          descendant relations between adjacent
          nodes;
          IF SMPath in target schema contains the ex-
          tended identified path;
          THEN Return (SMPath);
        END IF.
```

Based on the idea illustrated above, Table 3-5 respectively show the semantic node mapping, the semantic clique mapping and the semantic path mapping corresponding to schemas in Fig. 4. Table 3 is generated according to the definition in the global dictionary and local dictionary. Table 4 is generated based on the algorithm *SemanticCliqueRecognition* (Section 4.2). Table 5 is formed according to the algorithm *SemanticMappingPath* illustrated above.

Table 3. Semantic node mapping between the schema of SIGMOD proceedings and schema of VLDB proceedings

Source	Semantic Node	Target	Semantic Mapping Node	Similarity Degree (SD)
SIGMOD	Title	VLDB	Title	1
SIGMOD	Author	VLDB	Author	1
SIGMOD	Conference	VLDB	Title	0.5
SIGMOD	ConfYear	VLDB	Year	0.9
SIGMOD	SIGMOD	VLDB	VLDB	0.9
SIGMOD	…	VLDB	…	…

Table 4. Semantic clique mapping between the schema of SIGMOD proceedings and the schema of VLDB proceedings

Source	Semantic Clique	Target	Semantic Mapping Clique	SD
SIGMOD	Authors (Author,…, Author)	VLDB	Authors (Author,…, Author)	1
SIGMOD	Articles (Title, InitPage, EndPage, Authors (Author,…, Author))	VLDB	Articles (Title, InitPage, EndPage, Authors (Author,…, Author))	1
SIGMOD	…	VLDB		…

Table 5. Semantic path mapping between the schema of SIGMOD proceedings and the schema of VLDB proceedings

Source	Semantic Path	Target	Semantic Mapping Path	SD
SIGMOD	SIGMOD/Proceedings/Proceeding /Articles/Article/ Title	VLDB	VLDB/Proceedings/Proceeding/Articles/ Article/Title	1
SIGMOD	SIGMOD/Proceedings/Proceeding/Conference	VLDB	VLDB/Proceedings/Proceeding/Title	0.7
SIGMOD	…	VLDB		…

5 Semantic-based Peer Similarity Measurements and Query Routing

An effective query should forward queries only to relevant peers whose schemas are likely to match the queries. So it is necessary to have an effective similarity measurement for qualifying semantic relativity between peer schemas. We characterize the similarity degree between a set of peers according to the node similarity and structure similarity. The similarity between semantic nodes focuses on obtaining the semantic interoperability among peers, and can be measured by the methods of cycle analysis and functional dependency analysis as proposed in [1]. The similarity between semantic structures is to capture the semantic structure such as the parent-child relationship between closely related semantic nodes in a maximum semantic clique or a minimum common sub-tree. A peer determines the destination to forward a query according to the similarity between semantic nodes and between semantic structures. To define the similarity between semantic structures, we introduce the following notions:

- *Peer* (N_i) denotes the semantic mapping node for semantic node N_i.
- *Length* (N_i, N_j) denotes the number of nodes on the path from N_i to N_j.
- *MaxSC* (N_i) denotes the maximum semantic clique that semantic node N_i belongs to.
- *MinCS* (N_i) denotes the minimum common sub-tree that N_i belongs to.
- *Semantic-Node-SD* (N_i, N_j) denotes the similarity degree between N_i and N_j.

The algorithm to measure the structure similarity between the semantic node N_i in the source schema and its semantic mapping node N_j in the target schema is as follows:

Input: N_i, N_j
/* N_i is a Semantic Leaf Node, and N_j=Peer (N_i) */
Output: Semantic-Structure-SD (N_i, N_j) /*Semantic structure similarity between N_i and N_j */
Step 1: IF N_i belongs to one of the Maximum Semantic-Cliques
THEN T=MaxSC (N_i)
ELSE T= MinCS (N_i)
END IF
Step 2: Root (N_i)= T
IF Length (N_i, T)=1
THEN Semantic-Structure-SD (N_i, N_j)= Semantic-Node-SD (N_i, N_j)
ELSE
NodeSet={N_i, …, Root (N_i)} /* Nodes on path from N_i to Root (N_i) */
$\overrightarrow{FV} = (fv_{N_i},...,fv_{Root(N_i)})$ /* semantic structure similarity feature vector */

$$fv_{N_k} = \begin{cases} 0, Peer(N_k) \notin Semantic-Mapping-Path(N_i) \\ Semantic-Node-SD(N_k, Peer(N_k)), \quad Otherwise \end{cases} \tag{1}$$

$\overrightarrow{W} = (W_{N_i},...,W_{Root(N_i)})$ /* weight vector to denote node importance for node on path from N_i to Root (N_i)*/

$$W_{N_k} = \begin{cases} 1/2, N_k = N_i \\ (1/2)^k, k = length(N_i, N_k), and\ N_k \neq Root(N_i) \\ 1 - \sum_{l=1}^{n-1} W_{N_l} = (1/2)^{n-1}, n = length(N_i, Root(N_i)), \quad N_k = Root(N_i) \end{cases} \tag{2}$$

$$\text{Semantic-Structure-SD } (N_i, N_j) = \frac{\overrightarrow{W} \cdot \overrightarrow{FV}}{\left|\overrightarrow{W}\right|\left|\overrightarrow{FV}\right|} \tag{3}$$

, where $\overrightarrow{W} \cdot \overrightarrow{FV} = W_{N_i} fv_{N_i} + ... + W_{Root(N_i)} fv_{Root(N_i)}$, and

$$\left|\overrightarrow{X}\right| = \left\|\overrightarrow{X}\right\|_2 = \sqrt{x_1^2 + ... + x_k^2}$$

END IF

Let $SN=\{N_1,\ldots,N_m\}$ be semantic node set of source schema, $\overrightarrow{\text{Semantic-Structure-SD}} = (SR_{N_1},...,SR_{N_m})$ be the feature vector for the se-

mantic-structure similarity of each semantic node calculated according to formula (3), and $\overrightarrow{W} = (W_{N_1},...,W_{N_m})$ be the user-defined weight vector representing the importance of each semantic node. The semantic structure similarity between two peer schemas is defined as follows:

$$\textit{Semantic-Structure-SD}\ (P_i, P_j) = \frac{\overrightarrow{\mathrm{W}} \cdot \overrightarrow{\mathrm{Semantic-Structure-SD}}}{\left|\overrightarrow{\mathrm{W}}\right| \left|\overrightarrow{\mathrm{Semantic-Structure-SD}}\right|} \quad (4)$$

6 Query Reformulation and Heterogeneous Data Integration

Upon receiving a query requirement, a peer will identify a set of relevant peers according to semantic relationships and similarity degree between peers to answer the query. We distinguish query requirements as follows:

1. A query that could be answered by separate peers.
2. A query that should be answered by joining data on multiple peers.

Query reformulation is to reformulate a peer's query over its immediate successors, then over the successors' immediate successors, and so on. Whenever the forwarded query requirement reaches a peer that stores the matching data, the query will be posed on that peer. The semantic node mapping, semantic clique mapping and semantic path mapping in Table 3-5 are used for reformulating a query over target schemas.

Within a predefined timeout, the peer initiating a query will analyze data flows returned. To solve the problem of data inconsistency, we take into account the QoP, the user-perceived qualities such as the number of returned results, response time, traffic overhead, precision and recall etc. The data returned by peers with higher QoP is considered more reliable to solve the problem of data inconsistency. Finally, the peer initiating the query will combine or join relevant data according to the pre-defined data flow and then provide users and peers with a single semantic image.

7 Experiments and Discussion

To illustrate and evaluate the proposed approach, we simulate a small but realistic P2PSLN application. The simulation environment consists of 50 peers. Each peer randomly selects a group of peers as its neighbors, and the average degree is equal to 6. The metadata of 50,000 papers collected from DBLP XML databases [4] and ACM SIGMOD XML records [2] is distributed over all peers under a uniform distribution. XML document size of each peer varies from 275K to 14, 207K. It is assumed that each peer has the same bandwidth and process ability. Twenty randomly generated queries are randomly submitted to twenty peers to test the performance of the P2PSLN with the following two types of routing mechanisms: (1) the *Breadth First Search* (BFS), each peer broadcasts query requirements to all the neighbors; and, (2) the *Random Walk Search* (RW), each peer forward the received query requests to a number of randomly selected neighbor. Our evaluation metrics are the *recall* rate (i.e.,

the fraction of the relevant data which has been retrieved), and the *bandwidth consumption* (i.e., the number of messages per query).

In the first experiment we measure the recall rate of three routing mechanisms when the TTL field of the request message is set to 5. Fig. 5 represents recall rate of the three routing mechanisms. On average, the recall rate of BFS, P2PSLN, and RW is 0.58, 0.43, and 0.28 respectively. The BFS routing policy achieves the highest recall rate. This is because the BFS broadcasts query requirements to all its neighbors and it is sure to get the most of the relevant data. The P2PSLN forwards query requirement according to the semantic relationship and the similarity degree, so it is possible to get the higher recall than the RW within a predefined TTL value.

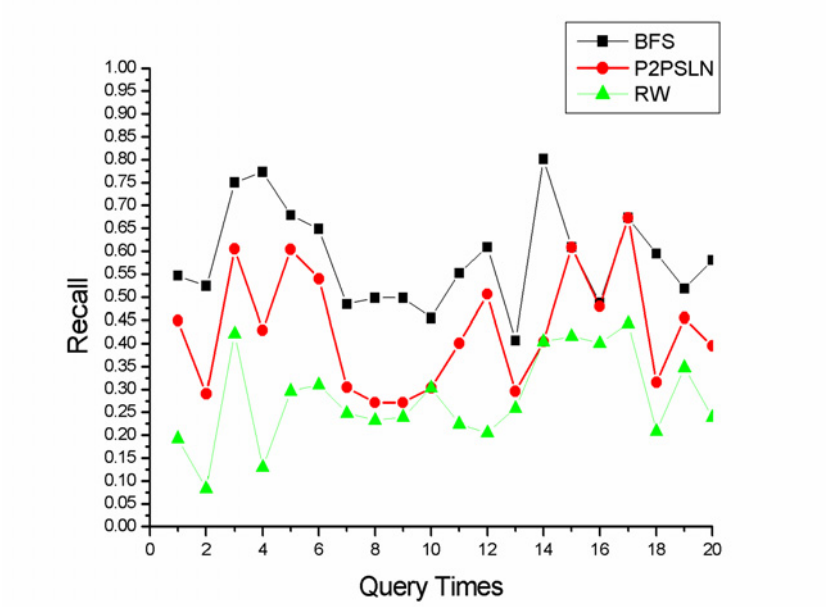

Fig. 5. Recall rate for 20 queries in BFS, P2PSLN, and RW routing policies (TTL=5)

In the second experiment we measure the number of messages that the three search mechanisms generate to process a query requirement within a predefined TTL. Fig. 6 shows that the number of messages generated by BFS is the most (25 on average). The number of messages generated by P2PSLN and RW is 9 and 13 on average respectively. We are able to reduce the number of messages by 2/3 in P2PSLN when compared to the BFS policy. In the P2PSLN search mechanism, each peer in the query path determines the neighbors according to the semantic relationship between them and then sends the query request to 3 neighbors with the highest similarity degree. Therefore, the number of messages to be forwarded can be reduced obviously.

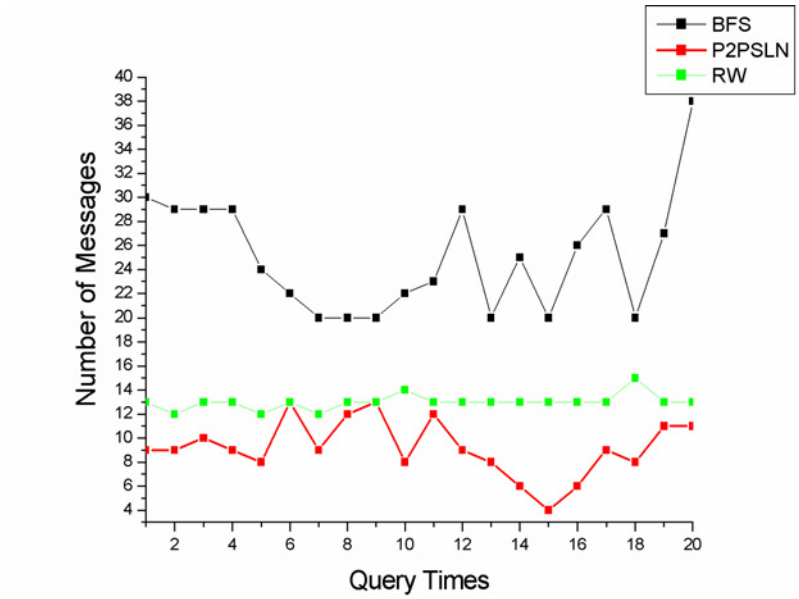

Fig. 6. Number of messages generated by 20 queries in BFS, P2PSLN, and RW routing policies (TTL=5)

Experimental results show that the P2PSLN is more effective and efficient in query routing than the BFS and RW routing policy in general. The major differences between the proposed approach and the previous work are as follows:

1. The P2PSLN specifies semantic relationships between peer schemas. Each peer is encapsulated as an active and intelligent soft-device, which could autonomously identify semantic relationships and dynamically interact with each other.
2. The semantic-based peer similarity measurement for efficient query routing provides a way to measure the similarity between a set of closely related nodes in peer schemas. We propose the semantic clique to denote the semantic structure between closely related semantic nodes.
3. The semantic node mapping, semantic clique mapping and semantic path mapping resolve the issues of semantic heterogeneity and structural heterogeneity between source schemas and target schemas. The data inconsistency issue in the returned data flows is resolved based on the quality of involved peers.

8 Conclusions

To resolve the issues of heterogeneous data integration in peer data management, this paper proposes a P2P semantic link network, a semantic-based peer similarity measurement for query routing, and a peer schema mapping approach for query reformulation. Results from theoretical analysis and simulations show that the proposed approach is effective. Contributions include three aspects: 1) propose the notions of P2P

semantic link network and provide with a tool for constructing and maintaining a nested P2PSLN; 2) incorporate the semantic node similarity and the semantic structure similarity to measure the similarity between peers so as to improve the effectiveness and efficiency of query routing; and 3) provide users and peers with data obtained from multiple peers in single semantic image. Experiments show that the proposed approach is a promising approach for peer data management.

The proposed approach has been integrated into the China E-Science Knowledge Grid Environment IMAGINE (Integrated Multidisciplinary Autonomous Global Innovation Networking Environment), , which aims at providing access to distributed resources (i.e., information, knowledge and services) and speeding up the processes of knowledge generation, propagation, fusion and management in cooperative research [21, 22].

Ongoing work focuses on incorporating user-defined integrity constraints and query reformulation optimization into the proposed approach.

ACKNOWLEDGMENTS

The research work was supported by the National Grand Fundamental Research 973 Program of China and the National Science Foundation. We thank all team members of China Knowledge Grid Research Group (http://kg.ict.ac.cn, http://www.knowledgegrid.net) for their diligent work and cooperation.

References

1. K. Aberer, P. Cudre-Mauroux, and M. Hauswirth. The Chatty Web: Emergent Semantics through Gossiping. WWW 2003, Budapest, Hungary, May 2003
2. ACM SIGMOD Xml Version. http://www.acm.org/sigmod/record/xml/SigmodRecord/SigmodRecord.xml
3. P. Bernstein et al. Data Management for Peer-to-Peer Computing: A Vision. In ACM SIGMOD WebDB Workshop 2002, Madison, Wisconsin, June 2002
4. DBLP XML Database. http://dblp.uni-trier.de/xml/
5. A. Deutsch and V. Tannen. MARS: A System for Publishing XML from Mixed and Redundant Storage. Proceedings of the 29th VLDB Conference, Berlin, Germany, September 2003
6. F. Giunchiglia and I. Zaihrayeu. Making Peer Databases Interact — A Vision for an Architecture Supporting Data Coordination. In Proc. of the Conference on Information Agents (CIA 2002), Madrid, Spain, September 2002
7. A. Halevy et al. Schema Mediation in Peer Data Management Systems. In Proc. of ICDE 2003, Bangalore, India, March 2003

8. A. Halevy et al. Piazza: Data Management Infrastructure for Semantic Web Applications. In Proc. of the Intl. WWW Conf. 2003, Budapest, Hungary, May 2003
9. A. Kementsietsidis, M. Arenas, and R. Miller. Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues. In Proc. of the ACM SIGMOD International Conference on Management of Data 2003, San Diego, California, June 2003
10. A. Loser et al. Semantic Overlay Clusters within Supper-Peer Networks. International Workshop on Databases, Information Systems, and P2P Computing, Berlin, Germany, September 2003
11. S.Y.Lu. A Tree-Matching Algorithm Based on Node Splitting and Merging. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 6 (2) (1984) 249-256
12. J. Madhavan, P. Bernstein, and E. Rahm. Generic Schema Matching with Cupid. Proceedings of the 27th VLDB Conference, Roma, Italy, September 2001
13. W. Nejdl et al. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. WWW2003, Budapest, Hungary, May 2003
14. W.S. Ng et al. PeerDB: A P2P-Based System for Distributed Data Sharing. In Intl. Conf. on Data Engineering (ICDE) 2003, Bangalore, India, March 2003
15. B.Ooi, Y. Shu, and K. Tan. DB-Enabled Peers for Managing Distributed Data. 5th Asia-Pacific Web Conference, APWeb2003, Xian, China, April 2003
16. E. Rahm and P. Bernstein. A Survey of Approaches to Automatic Schema Matching. VLDB Journal 10(4) (2001) 334-350
17. I. Stoica et al. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. IEEE/ACM Transactions on Networking 11 (2003) 17-32
18. I. Tatarinov and A. Halevy. Efficient Query Reformulation in Peer-Data Management Systems. ACM SIGMOD 2004, Paris, France, June 2004
19. H. Zhuge. Active E-Document Framework ADF: Model and Tool. Information and Management 41 (1) (2003) 87-97
20. H. Zhuge. Clustering Soft-Devices in Semantic Grid. IEEE Computing in Science and Engineering 4 (6) (2002) 60-63
21. H. Zhuge. China's E-Science Knowledge Grid Environment. IEEE Intelligent Systems 19 (1) (2004) 13-17
22. H.Zhuge, Future Interconnection Environment — Dream, Principle, Challenge and Practice, Keynote at The 5th International Conference on Web-Age Information Management, Dalian, China, July, 2004. http://www.knowledgegrid.net
23. Gnutella website. http://www.gnutella.com
24. Napster website. http://www.napster.com