Self Organizing Semantic Topologies in P2P Data Integration Systems

Ami Eyal, Avigdor Gal
Technion - Israel Institute of Technology
Haifa
32000 Israel
{eyalami@tx,avigal@ie}.technion.ac.il

Abstract

A semantic topology is a peer overlay network connected via semantic links, constructed using schema mappings and used for peer querying. The large-scale and dynamic environments of P2P networks dictate the use of automatic schema matching, which was shown to carry with it a degree of uncertainty. Therefore, peers prefer network topologies that improve their ability to answer queries effectively, by reducing uncertainty. We introduce a model for a peer database management system that manages the inherent uncertainty of automatic schema matching, the amplification of this uncertainty over transitive mappings, and its impact on query processing. We then briefly present the research challenges involving a dynamic topology setting where peers can change their neighbor set selection.

1 Introduction

Peer-to-Peer (P2P) systems rely on machine-to-machine ad-hoc communications to offer services to a community. P2P technologies show distinct advantages in scalability, autonomy, and robustness. Originally, P2P systems dealt with very simple data and query models: only filenames were shared and queries were composed of a single hash value or a keyword. File content was described by a simple schema (set of attributes, such as title of a song) to which all the peers in the network have to subscribe.

Peer Data Management Systems (PDMS) combine the decentralized setting and autonomy of P2P systems with the rich semantics of DBMSs. Following the data ring abstraction [3], we can conceptually envision a setting where each peer maintains a local database (*e.g.*, a DB-life like information) and a descriptive schema exposing its database to other peers. Information sharing is done by query dissemination, iterative propagation of queries among connected peers. Since peers are meant to be totally autonomous, they may use different schemata to represent their data, even if they refer to the same domain of interest [1]. Thus, in order to establish (meaningful) information exchange between peers, a required step involves identification of schema mappings for the purpose of query answering.

A grand challenge to PDMSs involves the integration of data originating from thousands of heterogeneous, dynamic and potentially unknown data sources. In a P2P setting, an assumption that all the peers rely on one global schema cannot be made [12]. This new challenge sharply contrasts with previous challenges faced in the field of data integration in terms of *scalability*, *uncertainty*, and *dynamicity*. PDMSs often involve tens or hundreds of sources with thousands or tens of thousands of semantic matches across sources [13]. Information sources come and go regularly and new sources have to be integrated on the fly with minimum overhead. Also, systems have to be resilient to node failures, including the failure of central indices or centralized servers such as mediators. At this scale, manual review of schema mappings is impractical. Instead, peers are likely to employ automated methods and return the apparent best answers. However, it has been shown in [10] that automatic matching is an uncertain process. It has also been shown that for a certain family of "well behaved" mappings termed *monotonic*, one can safely interpret a high confidence measure as a good semantic mapping. Thus, monotonic automatic matching algorithms can be trusted to associate confidence measures that reflect the accuracy of their outcome [9].

We model a PDMS as a network of peers connected by schema mapping links associated with mapping confidence measures. We assume the use of monotonic matching algorithms and take confidence measure as truly reflecting mapping accuracy. Yet, schema mappings may still be inaccurate. For example, consider the following, paraphrased from [7]. Three data sources R, S, and T, all describing personal data, reside at three different peers. Data source R describes a person by her email address, current address, permanent address, and month of hiring. Data source S describes a person by her name, email, mailing address, home address, office address, and month of job start. Finally, data source T has three attributes, namely name, email, and office address.

R = (pname,e-addr,c-addr,p-addr,hire month)

S = (name,email,m-addr,hm-addr,o-addr,start month)

T = (name,email,o-addr)

A peer, using an automatic schema matching utility may determine that c-addr from R matches with o-addr from S while the correct matching may be between c-addr and hm-addr. Therefore, the query

$\mathsf{q_R}$: SELECT c-addr FROM R

may be wrongly rewritten to be

$\mathsf{q_S}$: SELECT o-addr FROM S

Incorrect attribute matchings may propagate further in the peer network, generating imprecise mapping compositions. Therefore, we model the deterioration of accuracy over compositions. We define a PDMS semantic overlay structure, *semantic topology*, and show the influence of topology selection on the quality of queries reformulation in a PDMS. We consider a dynamic topology where a peer connects with some arbitrary peers upon joining the network and can later change its neighbor set selection. We restrict our attention in this work to topology self-organization based on schema mapping uncertainty and focus on the following two research questions: 1) Can we efficiently identify "good" topologies, those that reduce the uncertainty in the network? and 2) Can such "good" topologies self organize by self-interested peers?

2 Model Definition

Let P be a set of peers, storing data in a database according to a schema S. We do not make any assumptions on the exact data model. We only require that schemata store information using attributes, where each attribute $A \in S$ may be an attribute in a relational schema, an element or an attribute in XML, and a class or a property in RDF. As a running example, we use the relational model. Section 1 provides an example of three peers, each with a database that contains a single table.

Using the open-world assumption, peers do not necessarily share the same extensions, which motivates the need to access as many peers as possible to get a complete response to a query. Depending on the data model, a query language for querying and transforming databases is used. We write $q = \{A_i | A_i \in S\}$ to denote the set of attributes involved in a query. For example, $\mathsf{q_R} = \{\text{c-addr}\}$ represents the set of R attributes that participate in $\mathsf{q_R}$. Each peer p is associated with a set of queries Q, where the frequency of issuing query q is denoted by λ.

We assume that a peer $p \in P$ can be identified by a unique identifier ID (*e.g.*, an IP address or a peer ID in a P2P network). Each peer has a basic communication mechanism that allows it to establish connection to other peers, using an access structure à la Gnutella. Thus, peers send *ping* messages with a certain Time-To-Live value and receive *pong* messages in order to learn about the network structure. Extending the Gnutella protocol, a peer also sends its schema S as part of a *pong* message.

Peers define *schema mappings* M_{ij} between a source schema S_i and a target schema S_j. Such mappings can be created manually, semi- or fully-automatically depending on the peers and the setting. A schema mapping M_{ij} allows the reformulation of a query of S_i into a new query to a target schema S_j. Schema mappings can be expressed in a variety of ways. Following [6], a schema mapping M_{ij} is given as a set of *attribute mappings* m:

$$M_{ij} = \{m\,(A \rightarrow A')\,|A \in S_i, A' \in S_j\} \tag{1}$$

where a source attribute is mapped into a target attribute.[1] For example, a mapping from R (peer p_1) to S (peer p_3, see Figure 1) is given as follows:

$$M_{13} = \{(\text{pname,name}), (\text{e-addr,email}), (\text{c-addr,o-addr}), (\text{p-addr,hm-addr})\}$$

Using schema mapping M_{ij} we can reformulate a source query q into a target query q' consisting of attributes from S_j only, using $M_{ij}(q)$. For reformulation, all attributes in q should have correspondences in S_j. A peer, receiving a reformulated query, may decide to reformulate it in turn for further dissemination. Thus, queries can be reformulated along a chain several times iteratively, *e.g.*:

$$q' = M_{T-1,T}(M_{T-2,T-1} \cdots (M_{12}(q)) \cdots) \tag{2}$$

Continuing with the example above, rewriting query $\mathsf{q_R}$ into S is represented as $\mathsf{q_S} = M_{13}\,(\mathsf{q_R}) = \{\text{o-addr}\}$, since c-addr is mapped into o-addr according to M_{12}.

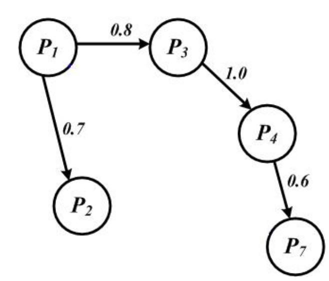

Figure 1. A semantic network graph.

A *semantic network topology* $G(P, \mathcal{M})$ is a graph, with peers as nodes and directed schema mappings as edges.

[1]Recall that attributes may be compound. Therefore, such a model is in **no way** restricted to 1 : 1 mappings.

Note that a pair of nodes can be related through opposite directed edges, whenever two peers keep mappings of one another. Figure 1 provides an illustrative example of a semantic network graph. The nodes represent peers and edges represent semantic links. For example, peers p_1, p_3 and p_4 are the peers with databases R, S, and T, respectively. Edge labels will be explained later in this paper.

Each peer p maintains a list of neighbors N. For each neighbor p_j, p_i maintains a mapping M_{ij}. Each peer has an upper bound K of the number of neighbors. In Figure 1, for example, $K_1 = 2$ and $K_4 = 1$. Our network model fits nicely with typical network models in the context of P2P networks such as power-law networks [8] and small-world networks [16] that suggest average short path lengths between peers, and limited number of neighbors distributed according to some power law. It also follows the observation in [11], stating that the the key to scalability involves minimizing the number of message passing.

We promote the notion of dynamic topologies in the context of PDMSs. New peers can be discovered by means of random *ping* messages as well as through answers to query propagation. By matching against new peers, a peer can expand or replace (if K is exceeded) neighbors, thus possibly improving its ability to obtain answers to queries.

3 Schema Matching in PDMS

The query reformulation mechanism in [15, 4] assumes semantically correct schema mappings. As PDMSs target large scale, decentralized, and heterogeneous environments, it is not always possible to create correct mappings between schemata. Given the vibrant activity in the area of (semi) automatic schemata alignment, we can expect some (or even most) of the mappings to be generated automatically in large-scale settings, with all the associated issues in terms of quality. In this section we define an estimation measure for mapping quality, namely *matching accuracy*, extended to the setting of PDMS where chained transitive mappings are used, in the form of *accuracy preservation*. We refer the interested reader to the vast literature on schema matching (*e.g.*, [5, 10, 13]), although we note that another layer of complexity is added to schema matching in a P2P setting, since peers do not even know which peers are worthwhile matching against.

Based on the discussion above, a peer p_i assigns a measure of accuracy ($\mu_{ij}(q)$) with the rewriting of a query q to fit the schema of peer p_j. We extend this measure to a set of queries from p_i to p_j, using query frequencies as weights:

$$\mu_{ij}(Q) = \frac{1}{|Q|} \sum_{q \in Q} \lambda\left(q\right) \cdot \mu_{ij}(q) \tag{3}$$

Note that this measure is directional, from p_i to p_j. This setting is reasonable, since the peers do not necessarily share the same set of attributes and therefore not all queries from p_i to p_j will also make sense in the other direction.

When a query is posed over a schema of a peer, the network will utilize data from any peer that is transitively connected by schema matchings. Recall that automatic matching between two schemata may involve a degree of uncertainty. For transitively chained matchings, uncertainty degree is amplified due to a composition of translations each of which uncertainty affects the accuracy of the following translations, resulting in a *matching accuracy decay*.

Query reformulation preservation measure, α, for a chain of matchings (see Eq. 2) is a function of the query matching accuracies of neighbors in the chain. Natural candidates for α are triangular norms (*i.e.*, minimum, product) extended to multiple number of arguments using their associativity property. In this work, query reformulation chain preservation is calculated using product over the accuracy measurements of the transitive query translations:

$$\begin{aligned} \alpha_C\left(q\right) &= \alpha\left(M_{T-1,T}(M_{T-2,T-1}\cdots(M_{12}(q))\cdots)\right) \\ &= \prod_{i=2}^{T} \mu_{i-1,i}\left(q_i\right) \end{aligned} \tag{4}$$

where C is a chain $(T, T-1, ..., 1)$, for each i, $q_i = M_{i-1,i}\left(q_{i-1}\right)$, and $q_1 = q$. Our empirical results show that nice properties are maintained even with such a simple composition. Finally, preservation over a set of queries in a given chain is calculated as a weighted average:

$$\alpha_C\left(Q\right) = \frac{1}{|Q|} \sum_{q \in Q} \lambda\left(q\right) \cdot \alpha_C\left(q\right) \tag{5}$$

Accuracies may be calculated on the fly and passed along reformulation chains to incrementally calculate accuracy preservation, as shown in Example 1.

Example 1 *Figure 1 provides a semantic network graph for query $\mathsf{q_R}$, issued by p_1. p_1 calculates the translation of $\mathsf{q_R}$ into $\mathsf{q_S}$ using its matchings to p_3 and propagates the translated query along with its calculated preservation measure:*

$$\alpha_{13}\left(\mathsf{q_R}\right) = \mu_{13}\left(\mathsf{q_R}\right) = 0.8$$

p_3 translates $\mathsf{q_S}$ to p_4, calculates the accumulated preservation using Eq. 4 and propagates the result to p_4:

$$\alpha_{134}\left(\mathsf{q_R}\right) = \alpha_{13}\left(\mathsf{q_R}\right) \cdot \mu_{34}\left(\mathsf{q_S}\right) = 0.8 \cdot 1.0 = 0.8$$

4 Self Organizing Semantic Topologies

Example 1 suggests that direct matching between peers may be more accurate than transitive mappings. Hence, peers strive to "improve" their queries reformulation quality by using direct mappings. As a goal, we would like to observe a topology evolution that yields network clustering, based on interest similarity. However, peers have limited resources to devote to neighbor list maintenance, so acquiring new neighbors may be at the cost of existing ones.

Semantic topology can be evaluated from two different points of view. First, in the decentralized setting of PDMS, a peer does not obtain knowledge about other peer mappings nor can it enforce other peers to create mapping links [14]. Under these restrictions, peers choose to couple according to their best private knowledge. Alternatively, a wider point of view of a collaborative network of peers will aim at global welfare, representing the mutual interest of peers to maximize the overall query span and accuracy.

The main challenge of self organizing semantic topologies is the lack of a global topology. Therefore, research should focus on mechanisms for dynamic self organization of topologies, where self-learning peers cooperatively establish semantic interoperability [2]. In our model, peers are equipped with three useful abilities for this task. First, upon propagating a query, a peer can calculate the reformulation accuracy and preservation and forward the latter along with the rewritten query. Secondly, upon receiving query results, it can analyze the feedback and update its semantic view of the network. Finally, it can periodically use this information to adjust its mappings. We partition the task of self organization into the following two subtasks:

Semantic Acquaintance: Peers can meet other peers using one of two means: (1) random connection through ping messages as part of the underlying network protocol or (2) connecting with peers that produce results to queries issued by the peer. A new peer randomly connects to a set of arbitrary neighbors. Later on, queries issued by the peer would be reformulated and passed along to semantically connected peers. This way, neighbors are chosen according to their semantic similarity to the selecting peer, reflecting their ability to translate its queries. Schema matching is the basic operation used to assess similarity in our model. Therefore, semantic acquaintance involves the selection of "good" candidates for schema matching among the discovered peers.

Semantic Replacement: With incomplete network knowledge, newly acquainted peers are not necessarily the best candidates for matching. Therefore, once new acquaintances arrive, a replacement policy will decide on whether a new comer should replace an existing peer. A replacement policy maintains a neighbor list by providing a decision policy for acceptance of new peers, given that each peer is limited to a finite number of neighbors.

References

[1] K. Aberer. Special topic section on peer to peer data management: Guest editor's introduction. *SIGMOD Record*, 32(3):21–22, 2003.

[2] K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. Start making sense: The chatty web approach for global semantic agreements. *Journal of Web Semantics*, 1(1), 2003.

[3] S. Abiteboul and N. Polyzotis. The data ring: Community content sharing. In *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research*, pages 154–163, Asilomar, CA, USA, Jan. 2007.

[4] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. Miller, and J. Mylopoulos. The hyperion project: From data integration to data coordination. *SIGMOD Record*, 32(3), 2003.

[5] H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proceedings of the 2nd Int. Workshop on Web Databases (German Informatics Society), 2002.*, 2002.

[6] C. Domshlak, A. Gal, and H. Roitman. Rank aggregation for automatic schema matching. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(4):538–553, 2007.

[7] X. Dong, A. Halevy, and C. Yu. Data integration with uncertainty. In *Proceedings of the International conference on Very Large Data Bases (VLDB)*, pages 687–698, 2007.

[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, 1999.

[9] A. Gal. Interpreting similarity measures: Bridging the gap between schema matching and data integration. In *Proceedings of the 24th International Conference on Data Engineering Workshops*, pages 278–285, Cancún, México, Apr. 2008.

[10] A. Gal, A. Anaby-Tavor, A. Trombetta, and D. Montesi. A framework for modeling and evaluating automatic semantic reconciliation. *VLDB Journal*, 14(1):50–67, 2005.

[11] L. Galanis, Y. Wang, S. Jeffery, and D. DeWitt. Locating data sources in large distributed systems. In *Proceedings of the International conference on Very Large Data Bases (VLDB)*, pages 874–885, 2003.

[12] F. Giunchiglia and I. Zaihrayeu. Making peer databases interact - a vision for an architecture supporting data coordination. In M. Klusch, S. Ossowski, and O. Shehory, editors, *Cooperative Information Agents VI, 6th International Workshop, CIA 2002, Madrid, Spain, September 18-20, 2002, Proceedings*, volume 2446 of *Lecture Notes in Computer Science*, pages 18–35. Springer, 2002.

[13] Y. Lee, M. Sayyadian, A. Doan, and A. Rosenthal. eTuner: tuning schema matching software using synthetic scenarios. *VLDB Journal*, 16(1):97–122, 2007.

[14] T. Moscibroda, S. Schmid, and R. Wattenhofer. On the topologies formed by selfish peers. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 133–142, New York, NY, USA, 2006. ACM.

[15] I. Tatarinov, Z. Ives, J. amd, A. Halevy, D. Suciu, N. Dalvi, X. Dong, Y. Kadiyaska, G. Miklau, and P. Mork. The piazza peer data management project, 2003.

[16] D. Watts and S. Strogatz. Collective dynamics of 'small world' networks. *Nature 393*, 1998.