Using Semantic Mappings for Query Routing in a PDMS Environment (Extended Abstract) *

Federica Mandreoli[1], Riccardo Martoglia[1], Wilma Penzo[2], Simona Sassatelli[1], and Paolo Tiberio[1]

[1] DII - University of Modena e Reggio Emilia, Italy
{fmandreoli, rmartoglia, sassatelli, ptiberio}@unimo.it
[2] DEIS - University of Bologna, Italy
wpenzo@deis.unibo.it

Abstract. In this paper we present the current achievement of our research activity in the WISDOM project, whose aim is the definition of intelligent techniques enabling effective and efficient information search in a distributed and decentralized PDMS scenario. We focus on the query routing problem and we define a new routing mechanism, which we call *routing by mapping*, in which the query is sent to the peers whose subnetworks best approximate the concepts required. In order to select the best subnetworks, the peer receiving the query exploits information about the semantic approximation of the query concepts, when moving towards each neighbour. This information is computed starting from the *semantic mappings* established with the peer's neighbours and it is maintained into specifically devised data structures called *Semantic Routing Indices (SRIs)*, whose update we propose specific algorithms and protocols for. The effectiveness of the achieved results has been experimentally proved through a series of exploratory tests.

1 Introduction

The huge amount of data and services available on the Web opens many possibilities for a user to answer her information need. However, without proper supporting technologies, the user can easily get lost in her struggle to find the information she requires. Usually, traditional search engines are not able to overcome this "information overloading" problem; indeed querying and accessing distributed and heterogeneous information in an effective and efficient way requires to devise a whole series of techniques in several synergic areas.

This is the stimulating scenario of the ongoing Italian Council co-funded WISDOM (Web Intelligent Search based on DOMain ontologies) project, whose aim is the definition of intelligent techniques, based on domain ontologies, to perform effective and efficient information search on the Web. The reference architecture is inspired by Peer Data Management Systems (PDMSs) [4], a recent proposal synthesizing P2P flexibility and the semantic expressiveness of database technologies. Each peer maintains its information in a OWL ontology, describing the informative contents of its underlying sources, and it is connected to its neighbours through appropriate *semantic mappings*, expressing how its own concepts are approximated by the ones available at the linked peers.

* This work is partially supported by the Italian Council co-funded project WISDOM.

In such a setting, effectively answering a query means propagating it towards the peers which are semantically best suited for answering the user needs. Flooding techniques are not adequate for both efficiency and effectiveness reasons, in that the querying peer would be overloaded with a large number of results, mostly irrelevant.

Our research activity in this project regards techniques allowing each peer to rank its own neighbors w.r.t. their ability to answer a given query effectively. In this paper we present a new routing mechanism, which we call *routing by mapping*, in which the query is sent to the peers whose subnetworks best approximate the concepts required. To this end a distributed index mechanism is adopted: each peer owns a *Semantic Routing Index (SRI)* which summarizes the ability of its subnetworks to semantically approximate the concepts of its schema. Such data structures are dynamically computed exploiting the available semantic mappings and evolve with the network topology following specifically devised algorithms and protocols.

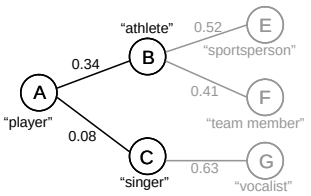
The paper is organized as follows: in Section 2 we analyze the problem of query routing and introduce our semantic approach; in Section 3 we present the SRI data structures and its management framework; Section 4 shows the results we obtained in our experimental tests and Section 5 concludes the paper.

2 Information Search in a PDMS

P2P systems offer the capability of accessing huge amounts of data, thanks to the interaction of a great number of participants. Nevertheless, this kind of systems provide very basic data management capabilities and rarely offer mechanisms to represent and exploit their semantics, with negative consequences as to the effective localization and retrieval of the data. On the other hand, PDMSs [4] represent a recent evolution of original P2P systems, synthesizing database world semantic expressiveness and P2P networks flexibility. They intend to offer a decentralized and easily extensible architecture for advanced data management, in which anytime every user can act freely on her data, while in the meantime accessing data stored by other participants.

In general, in such kinds of architectures, a query posed at a given peer is usually answered presenting the local data, and it is then propagated through the network to retrieve further useful information possibly owned by other peers. Nevertheless, it is not desirable to forward the query regardless of the query capabilities of the destination peers. In fact, peers containing unrelated data would be unnecessarily involved, the network traffic would be uselessly multiplied and, most of all, the querying peer would be overwhelmed by irrelevant results.

To this end, query routing techniques are needed to select the best destinations, i.e. the peers able to supply the most useful information. Some works dealing with this problem [6, 8] are based on "a posteriori" learning approaches, exploiting quantitative information about the retrieved results. However, these methods do not take into account the possible presence of heterogeneous semantic knowledge about the contents of the peers in the network: In [6] all peers are assumed to share the same set of keywords they store data about, whereas in [8] each peer accepts as relevant answers which approximate the query concepts with any concept (through the use of wildcards), thus disregarding the grade of semantic similarity between them. Other approaches [2, 5] exploit such semantic information, which is computed starting from the schemas associated to the peers' contents, and provide semantic approximation strategies to determine the most promising peers to forward a given query to. However, in these works the routing mechanism is limited to the only local information provided by the neighbouring peers. In other approaches, such as in [3], the neighbouring peers inform



(a) Semantic mappings.

	sport	team	coach	player
B	0.19	0.44	0.32	0.28
C	0.02	0.08	0	0.05

(b) SRI of peer A.

Fig. 1. Example of the semantic routing by mapping mechanism.

the querying peer also about the data reachable by their own subnetwork through a summary of their subnetwork contents. This provides an interesting view of the network which extends the traditional vision limited to the only neighboring peers. However, IR style data representation and querying is assumed, and only quantitative information is used to determine the best peers to be queried. In our work we rely on the notion of summarized subnetworks as in [3], and we propose a routing mechanism, which we call *routing by mapping*, where the selection of the best answering peers is based on the semantic information about the peers' contents.

2.1 Semantic Routing by Mapping

The routing mechanism we propose relies on the *semantic mappings* (originally described in [7] for a heterogeneous centralized environment) that each peer establishes between its schema and the ones of its neighbours by performing apposite *schema matching* operations. By means of these mappings each concept of the peer schema is associated to the most similar concepts of the neighbours schemas and each of these associations is characterized by a numerical score, belonging to the interval [0,1] and quantifying the level of semantic approximation between them. In the scenario we consider, a query originating from a given peer is always expressed in terms of its reference ontology. If routing was limited to the semantic knowledge each peer has on its neighbours, every query reaching a peer would be forwarded to the neighbours having the highest scores for the required concepts, since these peers have the highest probability to produce correct results.

Example 1. Let us consider Figure 1-a, where peer A has two neighbours to which it established appropriate semantic mappings: peers B and C. We now suppose that, according to these mappings, concept "player" of peer A schema is associated to concepts "athlete" of peer B and "singer" of peer C with two scores of 0.34 and 0.08, respectively. This means that, according to our routing mechanism, a query posed to peer A and asking for concept "player" would be preferably forwarded to peer B. □

2.2 Combining Semantic Mappings Scores

A good routing mechanism can not be limited to the exploitation of the information about the neighbours alone. Indeed, in the neighbours selection, each peer should also consider the approximation capability of the peers belonging to the subnetworks routed by its neighbours (i.e. peer E, F and G in the Figure 1-a), as the query would likely be propagated to these subnetworks too. Ideally, it would be desirable for each peer to calculate a semantic mapping with each other peer of the system, so that this information could be exploited in the routing process. However, an approach of this

kind is clearly not applicable in a P2P context, due to the excessive amount of data to be stored because of the potential large number of peers.

Instead, in our approach, each peer creates and maintains cumulative information summarizing the approximation capabilities of the whole subnetworks routed by each of its neighbours. These summarized information is calculated by each peer by appropriately combining the semantic mappings scores towards its neighbours with the summarized information each neighbour has about its own subnetwork. Being such information computed in the same manner, we obtain that the knowledge about mappings is propagated throughout the whole system and each peer can learn about all other peers without being directly connected or interacting with them. Further, in order to avoid the presence of cyclic paths in the updates propagation, when a peer connects to the network a cycle detection mechanism based on global unique identifiers, as in [3], may be adopted. To obtain the cumulative information we apply two different types of operations, named *aggregation* and *composition*, to the original mapping scores. Before introducing in detail the data structures we devised for conveniently maintaining this summarized information, let us show by means of an example of use of these operations.

Example 2. Consider Figure 1-a. Peer A computes its semantic score towards B by *composing* the similarity score between "player" and "athlete" (i.e. 0.34) with a score obtained from peer B indicating how well concept "athlete" can be approximated in the subnetwork including peer E and F. This last score is computed by peer B by *aggregating* the scores characterizing its mappings for concept "athlete" towards neighbours E and F. Specifically, these mappings involve concepts "sportsperson" (peer E) and "team member" (peer F). Peer B sends the aggregated result to A, which compose it with its own score for the mapping "player"-"athlete", in order to obtain a final score expressing how well the concept "player" can be semantically approximated by the subnetwork routed at peer B. □

As to the choice of the composition and aggregation operators, in order to verify the effectiveness of different alternatives, we performed several exploratory tests, whose results are presented in section 4.

3 Semantic Routing Indices

To maintain the information about mapping scores, each peer owns a specially devised data structure called *Semantic Routing Index (SRI)*. The index is represented by a matrix where the rows are associated to the peer's neighbours, while the columns refer to the concepts of its schema. An example of such data structures is represented in Figure 1-b for peer A, whose schema is supposed to include only four concepts: "sport", "team", "coach" and "player".

Our idea is that each cell of the matrix stores a score representing how the concept associated to a given column is semantically approximated by the subnetwork routed by the neighbour associated to a given row. For example, the number 0.28 in the cell corresponding to the last column and the second row, means that concept "player" of peer A can be approximated through the subnetwork routed by peer B with a similarity score of 0.28.

The scores into the indices are computed in an incremental way, on the basis of the peer connections to the system, and following its evolution.

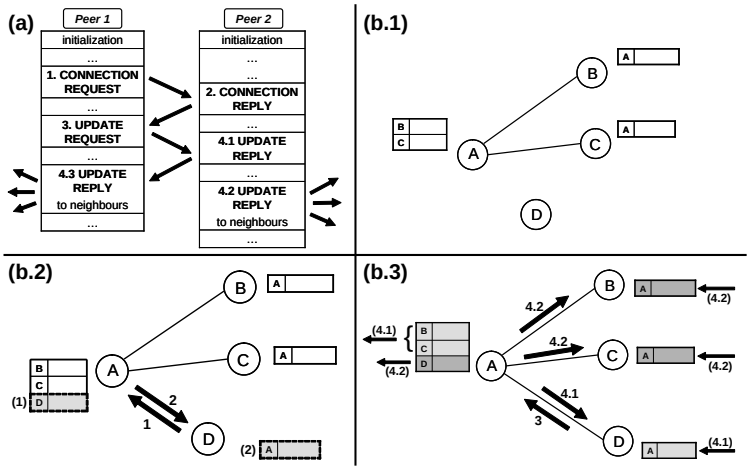


Fig. 2. P2P Protocol (a) and sample scenario (b)

3.1 A Framework for SRI Evolution

In our framework an entity (a peer) is identified by a unique alphanumeric id and owns, besides its actual data, a schema (`MySchema`) which the data comply to, a Semantic Routing Index (`SRI`), and a list (`MappingList`) containing all the mappings between the current peer and its neighbours.

The peers interact on the basis of the protocol articulating the communication in four types of messages, whose sequence is depicted in Figure 2-a: (1) *CONNECTION_REQUEST*, to request the creation of a new connection and information on the schema of the receiver; (2) *CONNECTION_REPLY*, to complete the connection replying with the requested schema; (3) *UPDATE_REQUEST*, sent by a peer that has received a CONNECTION_REPLY to request information on the newly accessible subnetwork; (4.1-4.3) *UPDATE_REPLY*, sent in reply to propagate the SRI update information for the new connection.

This protocol is implemented by Algorithm 3.1, which each peer executes during its lifetime, starting from the moment it connects to the P2P network. The algorithm is composed by three main phases:

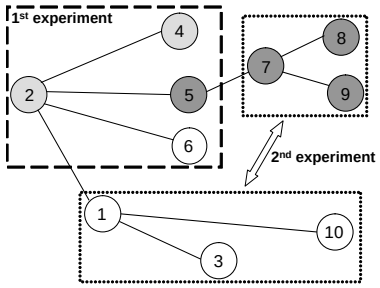
1. *Initialization* (lines 1-2), in which the peer executes the *schemaMatching()* operation on its schema (`MySchema`), calculating the values which will be implicitly used in all the subsequent *schemaMatching()* operations, in order to make the scores comparable and normalized;
2. *Index creation* (lines 3-4), in which the peer selects its neighbours and sends them (by means of the *send()* function) a CONNECTION_REQUEST message containing its schema (`MySchema`);
3. *Index update* (lines 5-27), in which the peer waits indefinitely for incoming messages and, depending on their type, performs different operations to maintain the routing indices. These operations include the aggregation (*aggregateExcept()*) and composition (*compose()*) introduced in the previous sections. Further, for each message received, another one is generated according to the protocol.

Example 3. Let us consider the scenario in Figure 2-b.1, where peer A is connected to peers B and C, and peer D is going to connect to peer A. The table depicted besides each peer represents its SRI. In the following, we denote the messages with the numbers

Algorithm 3.1 P2P Algorithm

```
1: MappingList [] ← null; { // Initialization Phase}
2: schemaMatching(MySchema,MySchema);
3: for all selected neighbours do { // Index Creation Phase}
4:    send(neighbour,CONNECTION_REQUEST,MySchema);
5: while true do { // Index Update Phase}
6:    wait for msg;
7:    if msg == CONNECTION_REQUEST then
8:       (SRI [sender], MappingList [sender]) ← schemaMatching(MySchema,senderSchema);
9:       send(sender,CONNECTION_REPLY,MySchema);
10:   else if msg == CONNECTION_REPLY then
11:      (SRI [sender], MappingList [sender]) ← schemaMatching(MySchema,senderSchema);
12:      aggr ← aggregateExcept(sender);
13:      send(sender,CONNECTION_REPLY,aggr);
14:   else if msg== UPDATE_REQUEST then
15:      SRI [sender] ← compose (MappingList [sender],aggr);
16:      aggr ← aggregateExcept(sender);
17:      send(sender,UPDATE_REPLY,aggr);
18:      for all neighbours do
19:         if neighbour ≠ sender then
20:            aggr ← aggregateExcept(neighbour);
21:            send(neighbour,UPDATE_REPLY,aggr);
22:   else if msg == UPDATE_REPLY then
23:      SRI [sender] ← compose(MappingList [sender],aggr);
24:      for all neighbours do
25:         if neighbour ≠ sender then
26:            aggr ← aggregateExcept(neighbour);
27:            send(neighbour,UPDATE_REPLY,aggr);
```

specified in Figure 2-a. According to the protocol, the involved peers perform the following operations (see Figures 2-b.2 and 2-b.3): (i) peers D and A exchange their schemas by sending messages 1 and 2, so that they can calculate the schema matching between them and extend their indices with an additional row (highlighted in gray in the figure) corresponding to the new neighbour (lines 3-11 of Algorithm 3.1); (ii) D sends message 3 to A, requesting information about the subnetwork routed by A (lines 12-13); (iii) A aggregates all the rows of its routing index, except that corresponding to the requesting peer, and sends the result to D (message 4.1); then D composes this information with its mapping associated to A and stores the result in its routing index (lines 14-17); (iv) A sends a message 4.2 to each member of its original subnetwork, containing the row of its routing index corresponding to the subnetwork newly reachable through D (in this case peer D alone) (lines 18-21); (v) each peer that receives a message 4.2 from A (peers B and C in our case) updates the row of its routing index corresponding to A, composing its mapping associated to A with the received information (lines 22-23). Finally, notice that if peer D had had an existing subnetwork of connected peers, it would have also had to send its aggregated information to A through message 3, and eventually send its original neighbours the aggregated information it received from A through message 4.3. □



(a) Experimental scenario

Peer2	Peer4		Peer5		Peer6	
sport	sport	0.1965	storage	0.0193		
team	club	0.1202	track	0.0356	article	0.0585
coach	trainer	0.3858	signboard	0.0765	journal	0.0606
player	athlete	0.1721	singer	0.0834	author	0.0962

Peer5 >7	Max-Prod	Max-Min	Sum-Prod	Sum-Min	Tr-Prod	Tr-Min
tracklist	0.0251	0.2689	0.0420	0.2689	0.0093	0.1685
track	0.0080	0.1325	0.0122	0.1325	0.0025	0.1326
singer	0.0647	0.3750	0.1042	0.3750	0.0253	0.3188
albumTitle	0.0686	0.3848	0.1080	0.3848	0.0269	0.2333

Peer5 >1	Max-Prod	Max-Min	Sum-Prod	Sum-Min	Tr-Prod	Tr-Min
tracklist	0.0079	0.1907	0.0141	0.1982	0.0026	0.0945
track	0.0000	0.0530	0.0000	0.0530	0.0002	0.0444
singer	0.0409	0.2153	0.0642	0.2153	0.0161	0.2153
albumTitle	0.0023	0.0127	0.0027	0.0127	0.0002	0.0127

Gr Ratio	Max-Prod	Max-Min	Sum-Prod	Sum-Min	Tr-Prod	Tr-Min
tracklist	3.18	1.41	2.98	1.36	3.51	1.78
track	n/a	2.50	n/a	2.50	14.77	2.99
singer	1.58	1.74	1.62	1.74	1.58	1.48
albumTitle	29.83	30.30	40.00	30.30	174.44	18.31

(b) Results of 1^{st} (top) and 2^{nd} (bottom) experiment

Fig. 3. Effectiveness test results

4 Experimental Evaluation

For our experiments we used SimJava 2.0, a discrete, event-based, general purpose simulator, which allows us to verify the behaviour of our algorithms without using a real P2P system. The scenario we modelled through this simulator corresponds to a network of peers, each one with its own schema, different from the others and describing a particular reality. Further, in order to deepen the tests at different levels of semantic heterogeneity, we considered peers belonging to a small set of categories, where the schemas of the peers in the same category describe the same reality from different points of view. In Figure 3-a a portion of this network is depicted, where peers belonging to the same category are identified by the same color. In particular, peers in the figure belong to three different categories: sport (peers 2 and 4), music (5, 7, 8 and 9) and publications (1, 3, 6 and 10). Notice that, since we currently are only in the initial phase of our testing, the considered network scenarios are not particularly complex. In the future, we will enrich them with more complicated network topologies and consider a larger number of peers. In our experiments we evaluated the performances of our techniques mainly in terms of effectiveness. We performed experiments about 1) comparability of mapping scores and 2) the usefulness of semantic routing indices. Due to the lack of space, we now present only a small selection of results for each type of these tests.

For the first type of experiments we consider the part of the network in Figure 3-a surrounded by the broken line, including peers 2, 4, 5 and 6. The top of Figure 3-b shows the mapping scores of peer 2, and the concepts these scores refer to. As can be seen, the matching algorithm correctly maps each peer 2 concept to the corresponding peer 4 concept. Also for peer 5 and 6, whose schemas belong to different categories, associations are built between concepts considered the most similar for their semantics and positions, however in this case the mapping scores are very low. Nevertheless, mapping scores comparability is demonstrated because, for each peer 2 term, the mapping with the highest score is towards peer 4; this reflects the fact that peer 4, belonging to the same peer 2 category, can semantically approximate peer 2 concepts in a better way than peer 5 and 6 do.

For the second type of experiments we considered two alternative scenarios: the original one as shown in Figure 3-b, and the one obtained by swapping the peers included in the dotted regions. In the bottom part of Figure 3-b the first two tables show how the scores in peer 5 routing index change when its subnetwork, originally including three peers of the same peer 5 category, is replaced by a subnetwork of peers belonging to a different category. In these tables, for each concept, six different scores are reported, corresponding to the results obtained applying different mathematical functions implementing aggregation and composition operations. In particular, the possible tested alternatives for aggregation and composition are: a) maximum and product; b) maximum and minimum; c) algebraic sum and product; d) algebraic sum and minimum; e) travel function and product; f) travel function and minimum. The travel function is inspired to a function commonly used in travel demand applications when modelling the aggregation of several alternatives [1]. In this type of tests, the key parameter for effectiveness evaluation is the growth ratio, i.e. the measure of how bigger are the scores of the original scenario w.r.t. the alternative one; we show these values in the last table of Figure 3-b. As expected, the scores of the original scenario are significantly higher (growth ratio greater than 1), reflecting that peer 5 concepts are semantically approximated in a better way by the subnetwork of peers belonging to the same peer 5 category. As to the use of the composition function, all the combinations involving product show a higher growth ratio (for example for "albumTitle" we have 40 with algebraic sum and almost 175 with travel function). As to the use of the aggregation function, all the possibilities show a satisfying behaviour, but we observed that the travel function more clearly discriminates the "good" subnetworks.

5 Conclusions

In this paper we presented our research activity in the WISDOM project on distributed query evaluation and, in particular, our new semantic routing by mapping mechanism, the associated SRI structures and their update algorithms. In the future, we will strengthen the proposed approach by including it in a theoretical framework, investigating the semantics of the involved operations, and we will perform tests on larger and more complex network topologies.

References

1. M. Ben-Akiva and S. R. Lerman. *Discrete Choice Analyses: Theory and Application to Travel Demand*. The MIT Press, 1985.
2. A. Castano, S. Ferrara, S. Montanelli, E. Pagani, and G. Rossi. Ontology-addressable contents in P2P networks. In *Proc. of the SemPGRID WWW Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID)*, 2003.
3. A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proc. of ICDCS*, 2002.
4. S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suciu. What Can Databases do for Peer-to-Peer? In *Proc. of WebDB*, 2001.
5. P. Haase, R. Siebes, and F. van Harmelen. Peer Selection in Peer-to-Peer Networks with Semantic Topologies. In *Proc. of ICSNW*, 2004.
6. S. Joseph. NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks. In *Int. Workshop on Peer-to-Peer Computing*, 2002.
7. F. Mandreoli, R. Martoglia, and P. Tiberio. Approximate Query Answering for a Heterogeneous XML Document Base. In *Proc. of WISE*, 2004.
8. C. Tempich, S. Staab, and A. Wranik. REMINDIN': Semantic Query Routing in Peer-to-Peer Networks Based on Social Metaphors. In *Proc. of WWW*, 2004.