Semantic Query Routing Experiences in a PDMS

Federica Mandreoli*, Riccardo Martoglia*, Wilma Penzo†, and Simona Sassatelli*

*DII – University of Modena and Reggio Emilia, Italy

{fmandreoli,rmartoglia,sassatelli}@unimo.it

†DEIS – University of Bologna, Italy

{wpenzo}@deis.unibo.it

***Abstract*— Querying a PDMS means either flooding the network with messages to all peers or taking advantage of a routing mechanism to reformulate a query only on the *best* peers selected according to some given criteria. As reformulations may lead to semantic approximations, we deem that such approximations can be exploited for locating the *semantically best directions* to forward a query to.**

In this paper, we present our experiences in devising and testing a mechanism for effective query routing in a PDMS. In particular, we describe a distributed index mechanism where each peer is provided with a Semantic Routing Index (SRI) for routing queries effectively. We illustrate SRIs' structure, their use and the framework we devised for their incremental update, then we provide an extensive evaluation of their effectiveness through a set of query routing experiments on a variety of scenarios. This work is partially supported by the PRIN WISDOM and FIRB NeP4B national projects.

I. Introduction

Peer Data Management Systems (PDMSs) represent a recent evolution of P2P systems, tuning database world semantic expressiveness and P2P network flexibility. In such kind of systems, each peer is enriched with a schema that represents the peer's domain of interests, and semantic mappings are locally established between peers' schemas [1], [2], [3]. Let us consider Figure 1 as a sample scenario of a PDMS concerning data about operas. Each peer is provided with a schema, for instance XML-based, whereas bold lines denote semantic mappings between pairs of peers. In order to query a peer in a PDMS, its own schema is used for query formulation and mappings are used to reformulate the query over its immediate neighbors, then over their immediate neighbors, and so on. Thus, query answers can come from any peer in the network that is connected through a semantic path of mappings [4]. More precisely, any relevant peer may add new answers to a given query and different paths to the same peer may yield different answers [4]. In this context, query processing can either flood the network with messages to all peers or take advantage of a routing mechanism to reformulate a query only on the *best* peers selected according to some given criteria. In particular, in a semantic web perspective, a query posed over a given peer should be forwarded to the most relevant peers that offer semantically related results among its immediate neighbors first, then among their immediate neighbors, and so on. As an example, let us consider the following query, posed on the schema of Peer1: "Retrieve the main singers of the opera entitled *Aida*". The Peer1's neighbors Peer2 and Peer3 might be considered mirroring peers as to the portion of the schemas involved in the query above; as to the second step of query reformulation, Peer5 is more relevant than Peer4 and Peer6, since it deals with lyric music data, instead of written operas. For these reasons, the answers obtained from path Peer3-Peer5 fit better the query conditions than those from paths Peer2-Peer4-Peer6 and Peer2-Peer6.

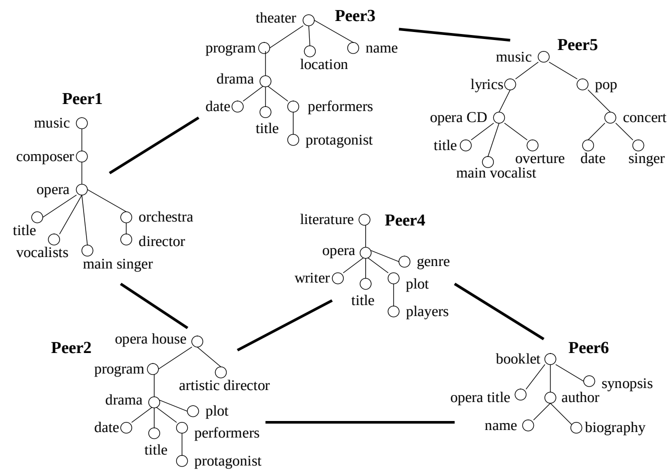


Fig. 1. Reference example

In this paper, we describe our experiences in devising and testing a mechanism for semantic query routing in a PDMS [5], [6]. Since, when a query is forwarded through a semantic path it undergoes a multi-step reformulation which may involve a chain of semantic approximations, our proposal is to exploit such approximations for selecting the direction which is *more likely* to provide the best results to a given query. Our perspective is knowledge-based, in that the routing of a query is guided by the semantic mappings between the peers. In particular, we leverage on scores which extend the semantic mappings in order to measure the semantic compatibility between neighboring peers. However, the information provided by the semantic mappings stored in a peer is not enough for deciding which is the best path. In fact, being Peer2 and Peer3 mirrors, the semantic approximation of the query would be measured as identical for both directions. Therefore, broadly speaking, some kind of information about the relevance of the whole semantic paths should be available in the network, maintained up-to-date, and easily accessible for query routing purposes. The solution we propose relies on a distributed-index mechanism called *Semantic Routing Index* (SRI) which summarizes, for each concept of its peer's schema, the semantic approximation "skills" of each subnetwork reachable from its immediate neighbors, and thus gives a hint of the relevance of the data which can be reached in each path. For instance, the Peer1's SRI will contain two entries, one for the upward subnetwork and one for the downward one. The semantic knowledge stored in a SRI is summarized on the available directions in order to maintain the size of the semantic index proportional to the number of neighbors, thus scaling well in a PDMS scenario.

In the following sections: We first introduce SRIs (Section II) by describing their structure, the framework we devised for their incremental update and their use as distributed indices for query routing, then we provide an extensive evaluation (Section III) of their effectiveness through a set of query routing experiments on a variety of scenarios. Finally, in Section IV we relate to other approaches in the literature and we discuss future extensions to our work.

II. Semantic Routing Indices

In our reference scenario each peer p_i in a set of peers $\mathcal{P}$ stores local data, modelled upon a local schema S_i that describes its semantic contents, and it is connected through semantic mapping to some other neighboring peers. A semantic mapping $M(S_i, S_j)$ can be established from a source schema S_j to a target schema S_i, and it defines how to represent S_i in terms of S_j's vocabulary. In particular, it associates each concept in S_i to a corresponding concept in S_j according to a *score*, denoting the *degree of semantic similarity* between the two concepts. A formal definition of semantic mappings can be found in [6], where the fuzzy theoretical framework of our work is presented.

In order to efficiently answer a query in such a distributed context, query routing appears as a key issue. In particular, it is fundamental to be able to forward the query towards the zones of the network which can potentially provide results semantically nearest to it.

Our idea is that each peer maintains cumulative information summarizing the semantic approximation capabilities w.r.t. its schema of the whole subnetworks routed by each of its neighbors. In particular, each peer keeps such information in a local data structure which we call *Semantic Routing Index (SRI)*. Thus, a peer p having n neighbors and m concepts in its schema stores an SRI structured as a matrix with m columns and $n+1$ rows, where the first row refers to the knowledge on the local schema of peer p. Each entry $SRI[i][j]$ of this matrix contains a score expressing how the j-th concept is semantically approximated by the subnetwork routed by i-th neighbor, i.e. by each path in the p_j's subnetwork.

A sample fragment of Peer1's SRI from the reference example is shown in Fig. 2. Notice that the scores of the first row represent the scores of the concepts of Peer1 in the self mapping, which we assume to be the identity relation. Further, the space required for storing a SRI at a peer is proportional to the number of the peer's neighbors, and thus quite modest w.r.t. the number of peers which usually join a PDMS. This makes our distributed-index mechanism scalable in a P2P context.

SRI_{Peer1}	opera	main singer	title	…
Peer1	1.0	1.0	1.0	…
Peer2	0.6	0.4	0.3	…
Peer3	0.7	0.6	0.6	…

Fig. 2. Sample SRI

A. SRI Evolution Framework

Since SRIs summarize the semantic information offered by the network, they change whenever the network itself changes. This may occur in response to either the joining/leaving of peers, or to changes in peers' schemas. We first focus our attention on the evolution of the PDMS's topology.

SRIs evolution is managed in an incremental fashion as follows. As a base case, the SRI of an isolated peer p having schema S is made of the single row $[1, \ldots, 1]$, i.e., it contains the membership grades of the concepts in S in the self mapping. This row expresses the semantic approximation offered by the subnetwork rooted in p, yet made of the only peer p.

A simplification of the process of Peer1's SRI update when Peer1 connects to Peer2 is shown in Fig. 3 (P1 and P2 in the figure). When a peer connects to another peer, each one *aggregates* its own SRI SRI by rows, according to an appropriate aggregation function g. The result of this aggregation operation and the schema S are then sent to the other peer. After a peer, say p_i, receives such knowledge from the other peer, say p_j, a semantic mapping $M(S_i, S_j)$ is established between S_i and S_j. Then, p_i extends its SRI SRI_i with a new row for p_j. The membership grades of this newly created row are obtained in two steps: 1) $M(S_i, S_j)$ is *composed*, according to an appropriate composition function, with the aggregated SRI provided by p_j to obtain the extension of the semantic paths originating from p_j (represented by the aggregated SRI) with the connection between p_i and p_j; 2) the so obtained result is then aggregated with $M(S_i, S_j)$ to include the semantic path connecting p_i with p_j.

Aggregation and composition are operations for which a fuzzy interpretation is given in [6]. Possible choices for composition are the algebraic product or the minimum. Several options are also possible for aggregation, for instance functions such as the maximum, the algebraic sum or the travel function, which is inspired to a function commonly used in travel demand applications when modelling the aggregation of several alternatives [6]. To verify the effectiveness of different alternatives, we performed several exploratory tests, whose results are given in Section III.

Afterwards, both peers p_i and p_j need to inform their own other neighbors that a change occurred in the network and thus they have to update their SRIs accordingly. To this end, each peer, say p_i, sends to each other neighbor p_{i_k} an aggregate of its SRI, excluding the p_{i_k}'s row. When p_{i_k} receives such

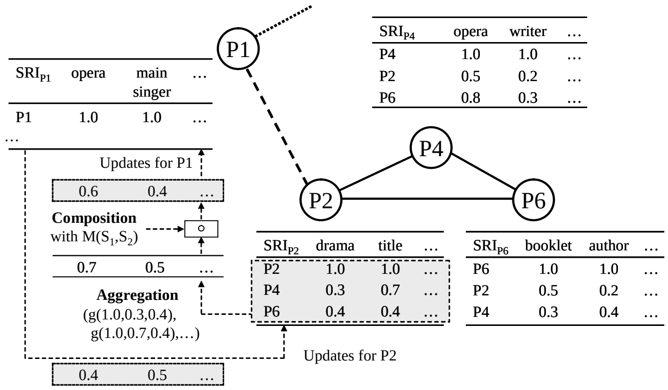


Fig. 3. SRI evolution

aggregated information, it updates the i-th row of its SRI by recomputing the membership values as discussed above.

Disconnections are treated in a similar way as connections. When a node disconnects from the network, each of its neighbors must delete the row of the disconnected peer from its own SRI and then inform the remaining neighbors that a change on its own subnetwork has occurred by sending new aggregates of its SRI to them. A similar procedure applies in case of modifications of the semantic knowledge maintained at each peer, for instance when a new concept is added to the peer's schema. When many changes occur in the PDMS, a careful policy of updates propagation may be adopted. For instance, when changes has a little impact on its SRI, a peer may also decide not to notify the network. This would reduce the amount of exchanged messages as well as the computational costs due to SRI manipulation.

B. Exploiting SRIs for Query Routing

When a peer p needs to forward a query q, it accesses its own SRI for determining the neighboring peers which are most semantically related to the concepts in q. For the sake of clarity, we start simple, and we assume the query q refers to a single concept C. The choice of the semantically best neighboring peers is done by evaluating the column corresponding to C in the local SRI. In particular, the highest values in this column make the corresponding neighbors to be the selected peers. For instance, given the query `main singer` on Peer1 whose SRI is shown in Fig. 2, Peer3 will be preferred to Peer2. In the general case of a complex query involving more concepts, the choice of the best neighbors is given by applying scoring rules which, for each neighboring peer p_i, combine the corresponding grades in the SRI for all the corresponding concepts in q. As to how these values can be effectively combined, see [6]. Once the best neighboring peer p_i is found, the corresponding semantic mapping $M(S_i, S_j)$ is exploited to unfold the query q in q'. q' is then routed towards the subnetwork, where, starting from p_i which in turn evaluates q' and returns its local results to p, the process possibly reiterates.

III. SRI IN ACTION

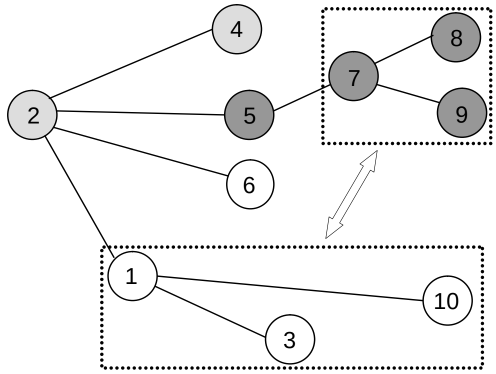
In this section we discuss a selection of experiments we performed to test the effectiveness of SRIs for query routing.

A. Experimental Setting

For our experiments we used a simulation framework able to reproduce the main conditions characterizing a PDMS environment. In particular, we employed SimJava 2.0, a discrete, event-based, general purpose simulator, which allowed us to evaluate the impact of exploiting SRIs for query routing. Through this framework we modelled scenarios corresponding to networks of semantic peers, each with its own schema describing a particular reality. We chose peers belonging to different semantic categories, where the schemas of the peers in the same category describe the same topic from different points of view. As in [4], the schemas are derived from real world-data sets, collected from many different available web sites, such as the DBLP Computer Society Bibliography and the ACM SIGMOD Record and enlarged with new schemas created by introducing structural and terminological variations on the original ones. Then, we distributed these schemas in the network in a clustered way, i.e. the schemas belonging to the same semantic category are more likely to belong to peers connected through semantic mappings. This reflects realistic scenarios where nodes with semantically similar content are often clustered together. As to the topology of the semantic network of mappings connecting the peers, we tested our techniques on different alternatives: We started with tree networks and then explored more realistic and uncontrolled ones generated with the BRITE topology generator tool [7]. The mean size of our networks was in the order of some hundreds of nodes. As we will see, this is sufficient for our effectiveness testing purposes. In particular, in these last cases, we also exploited the Barabasi option, which exploits mathematical distributions in order to generate topologies similar to actual networks. In such complex networks the problem of cycles can not be ignored. Indeed, in order to avoid the presence of cyclic paths in the SRI updates propagation, when a peer connects to the network a cycle detection mechanism based on global unique identifiers, as in [8], is adopted.

B. SRI Evaluation

In order to evaluate the effectiveness of our approach, we started with an initial testing phase executed on a simple tree network. In Figure 4-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). We considered two alternative scenarios: the original one as shown in Figure 4-a, and the one obtained by swapping the peers included in the dotted regions. In Figure 4-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



(a) Experimental scenario

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) Effectiveness of different functions on Peer5's SRI

Fig. 4. Initial testing phase results: effectiveness on a simple network

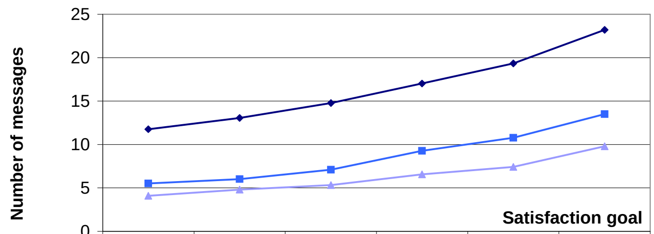
are reported, corresponding to the results obtained applying different mathematical functions implementing aggregation and composition operations. In particular, the possible tested alternatives for aggregation and composition are: a) maximum and product; b) maximum and minimum; c) algebraic sum and product; d) algebraic sum and minimum; e) travel function and product; f) travel function and minimum. 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 4. 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 behavior, but we observed that the travel function more clearly discriminates the "good" subnetworks. Thus, in the following tests we will refer to the product and travel functions.

C. Query routing evaluation

In the second phase of our testing, we considered larger and more complex network topologies and we simulated the querying process by instantiating different queries on randomly selected peers and concepts and propagating them until a stop condition is reached. We considered two alternatives: stopping the querying process when a given number of peers (*messages*) has been queried and measuring the quality of the results (*satisfaction*) or, in a dual way, stopping when a given satisfaction is obtained and measuring the required number of messages. Satisfaction is a specifically introduced quantity that grows proportionally to the goodness of the results returned by each queried peer. Each contribution is computed by composing the semantic mappings scores of the traversed peers (the same composition function exploited for SRI construction is applied). The search strategy employed is the depth-first search (DFS): Each peer receiving a query produces an answer on its local schema, then selects its most promising neighbor among the unvisited ones and forwards the query to it. In this section, we compare our neighbor selection mechanism based on SRIs (*SRI*) with a mechanism where the selection of the next neighbor to be visited is based on the semantic mapping values (*SemMap*) and with a baseline corresponding to a random strategy (*Random*). Notice that all the results we present are computed as a mean on several hundreds query executions.

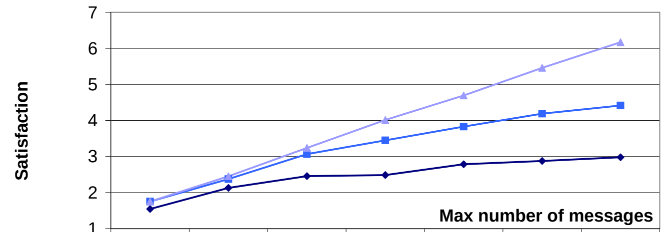
We started by studying the behaviour of the different routing strategies when we gradually vary the stop conditions in tree networks. Figure 5-a shows the trend of the number of required messages for a given satisfaction goal, while, from a dual perspective, Figure 5-b shows the trend of the obtained satisfaction at a given message limit. From both points of view, the Random strategy is outdistanced by the other two. Further, the difference between the SemMap and SRI performance appears closer at the initial part of the graphs but becomes increasingly more significant at growing stop conditions. This means that SRIs are indeed able to discriminate better subnetworks to explore and consequently increase the satisfaction at each step in a more substantial way. Nevertheless, tree topologies may not be considered completely realistic for a PDMS setting and may facilitate the SRI routing process, because in such kind of topologies different subnetworks are not overlapped and it is consequently probably easier to identify the better ones.

For these reasons, we deepened our tests by introducing realistic network topologies, also involving redundant and cyclic paths. The results referring to these situations are shown in Figure 6, which presents the graphs of Figure 5 for the new environments. Even though the distance between SRIs and SemMap is slightly reduced due to the complications introduced, we can see that, despite the new unfavourable scenarios, the trend of these graphs are roughly similar to the previous ones. Specifically, even in this case, we can observe that the SRI curves are clearly separated from the others, reflecting that the SRIs' ability to identify the best subnetworks



	2.5	3	3.5	4	4.5	5
Random	11.76	13.06	14.78	17.02	19.34	23.21
SemMap	5.51	6.02	7.11	9.28	10.78	13.51
SRI	4.09	4.81	5.32	6.57	7.43	9.79

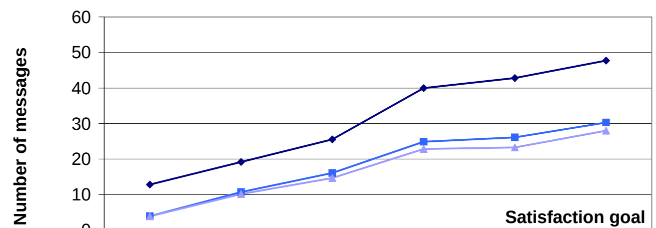
(a) Required messages trend for a given satisfaction goal



	1	2	3	4	5	6	7
Random	1.5432	2.1322	2.4567	2.4876	2.7852	2.8754	2.9811
SemMap	1.7529	2.3756	3.0668	3.4495	3.8308	4.1891	4.4143
SRI	1.7529	2.4559	3.2389	4.0119	4.6949	5.4579	6.1709

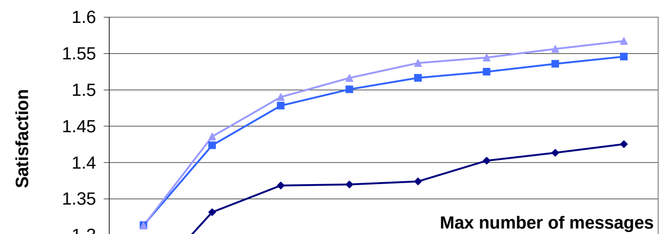
(b) Obtained satisfaction trend at a given message limit

Fig. 5. Tree networks results



	1.1	1.2	1.3	1.5	1.6	1.7
Random	12.84	19.17	25.57	39.99	42.8	47.73
SemMap	3.93	10.74	16.09	24.89	26.13	30.27
SRI	3.91	10.18	14.7	22.84	23.29	28

(a) Required messages trend for a given satisfaction goal



	1	2	3	4	5	6	7	8
Random	1.2281	1.3317	1.3685	1.3699	1.3741	1.4026	1.4135	1.4254
SemMap	1.3139	1.4238	1.4783	1.5008	1.5165	1.5249	1.5358	1.5458
SRI	1.3133	1.4359	1.4902	1.5164	1.5369	1.5445	1.5564	1.5673

(b) Obtained satisfaction trend at a given message limit

Fig. 6. Real networks results

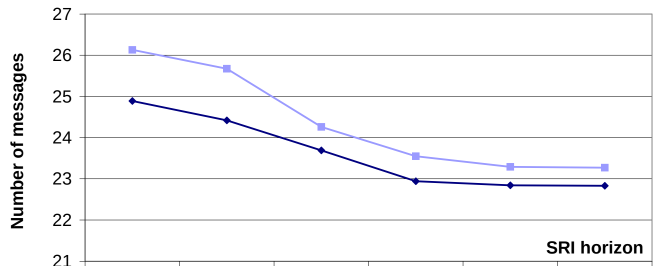
to explore facilitates a faster retrieval of the highest ranked results.

The previous tests show that by exploiting SRI query routing performance is improved. However, what is the optimal trade-off between such improvements in routing effectiveness and the cost of SRIs' updates? In order to give an answer we performed another typology of experiments (Figure 7), aiming to verify how the performances of the SRI routing mechanism are affected when we vary the update horizon, i.e. how far (number of hops) an update on a peer schema propagates in the network, limiting the portion of the nodes the SRI scores summarize. The curves represent the messages trend for growing values of the horizon, starting from the baseline 1 which corresponds to the SimMap case, and for two different satisfaction goals. Observing the graph, it is clear that increasing the horizon allows us to perform a better routing mechanism, because it relies on more precise information. In particular, from our tests, horizon extensions up to 8 clearly provide significant benefits, then the results appear to stabilize (see Figure 7). Notice that the use of an horizon, limiting the updates propagation for the SRIs scores, clearly introduces a kind of approximation on the information stored, but it is also useful in limiting the SRI maintenance costs. Therefore, due to the above considerations, since higher horizons lead to larger propagation costs, we consequently estimate that an horizon value of 8 represents a good trade-off (this is also the value at which all tests in this section have been performed).

Finally, the last type of test we present explores a possible enhancement on the routing process, involving a mechanism of pruning on the selection of the paths to follow: We introduce a threshold for the selection of neighbors to which propagate the queries, preventing the exploration of those paths characterized by small mapping and/or SRI scores and consequently leading to uninteresting results. The graph shows the results for both SemMap and SRI routing strategy, expressing the number of messages necessary to reach a given satisfaction goal when we apply three different pruning thresholds. Notice that when we use a zero threshold, and consequently apply no pruning mechanism, we obtain the same results presented earlier in the section. As can be seen, increasing the threshold leads to significant savings in the number of messages for both strategies. However, performing pruning on the basis of SRIs appears to perform better in every situation, showing a small but consistent number of saved messages w.r.t SemMap.

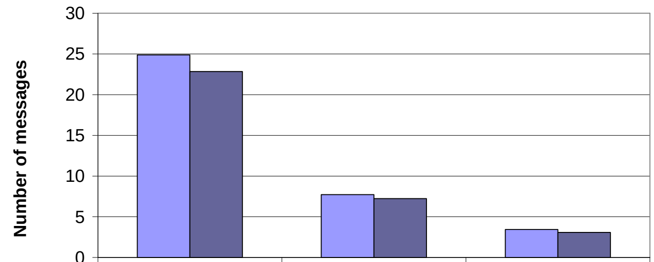
IV. Related Work and Concluding Remarks

As envisioned by the Semantic Web, the need of complementing the Web with more semantics has spurred much efforts towards a rich representation of data. To this end, knowledge representation languages (e.g., XML, RDF, and



	SemMap	SRI (h=2)	SRI (h=4)	SRI (h=6)	SRI (h=8)	SRI (h=10)
sat 1.5	24.89	24.42	23.69	22.94	22.84	22.83
sat 1.6	26.13	25.67	24.26	23.55	23.29	23.27

(a) ...SRI horizon



	Pruning off	Pruning med	Pruning high
SemMap	24.89	7.73	3.45
SRI	22.84	7.2315	3.09

(b) ...pruning

Fig. 7. Additional results for real networks: Impact on required messages of...

OWL) has flourished in recent years. In this view, peer data management systems (PDMSs) have been introduced as a solution to the problem of large-scale sharing of semantically rich data [1]. Indeed, a key challenge when querying a large set of peers is query routing, i.e., the capability of selecting a small subset of relevant peers to forward a query to. Much research work in the P2P area has focused on this issue [9], [10], [8], [11], [12], [13], [14], [15], [16]. Some of these works discuss id/keyword-based search of documents [9], [11], [14], some assume a common vocabulary/ontology is shared by peers in the network [8], [12], some address scalability of query routing by means of a properly tailored super-peer topology for the network [15], or by adapting their own semantic topology according to the observation of query answering [16].

Most of these proposals are based on IR-style and machine-learning techniques [8], [11], [13], [14], [16]. Basically, they utilize measures that rely on keyword statistics, on the probability of keywords to appear into documents, on the number of documents that can be found along a path of peers, on caching/learning-from the number of results returned for a query. Then, all of them (but [8]) provide routing techniques which either assume distributed indices which are indeed conceptually global [9], [14], or support completely decentralized search algorithms which, nevertheless, exploit information about neighboring peers only. More precisely, the only work [8] proposes a routing mechanism which does not limit the peer's capability of selecting peers to the information available at a 1-hop horizon, rather it extends this view by using summaries of subnetworks' content to provide a *direction* to send a query to.

Nevertheless, querying a PDMS is different than querying a P2P system, primarily because of the presence of heterogeneous schemas at the peers. On the other hand, the novelty in a PDMS lies in its ability to exploit the transitive relationships among such schemas for query answering [10], [1]. In this scenario, our work aims to support query routing in a PDMS, and it appears to be the first having this purpose. The main differences between our proposal and the P2P techniques discussed above are: 1) We do not assume any global characterization of documents in the network; 2) We move in a PDMS scenario, then assuming the presence of schemas describing the content of peers' data, as well as pairwise semantic relationships between the peers' schemas; 3) We make a schema-based rather than a key(word)-based search; 4) inspired to [8], we rely on fully distributed semantic indices, called SRIs, which summarize the *semantics* (rather than the number of documents as in [8]) that can be retrieved following a given direction in the network; 5) in order to cope with schema heterogeneity, we rank (subnetwork of) peers according to the *semantic similarity* occurring between concepts in the peers' schemas. The experiments we conducted on a simulation environment led to very encouraging results.

As a final consideration, it should be noted that the focus of this paper was not on schema matching effectiveness, even if the approach for query routing we propose is heavily dependent on the used matches. Effective approaches such as [17], which exploits the semantics and the structure of the available schemas, or [18], which also exploit the information given by surrounding nodes, could be adopted for schema matching.

In the future, SRIs could be integrated in a more general framework together with other approaches such as [8], [14] which are orthogonal to ours, and which cover complementary aspects such as knowledge on quantitative information, as well as on novelty of results, so as to blend different dimensions a peer can be queried on. Then, as also stated in [16], the *best* peer has been understood as a peer that has the most knowledge. Other aspects one might include in the evaluation of peers are properties like latency, costs, etc.

References

[1] A. Halevy, Z. Ives, J. Madhavan, P. Mork, D. Suciu, and I. Tatarinov, "The Piazza Peer Data Management System," *IEEE TKDE*, vol. 16, no. 7, pp. 787–798, July 2004.

[2] W. Nejdl, B. Wolf, S. Staab, and J. Tane, "EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network." in *Proc. of WWW Intl. Workshop on the Semantic Web*, 2002.

[3] M. Arenas, V. Kantere, A. Kementsietsidis, I. Kiringa, R. Miller, and J. Mylopoulos, "The hyperion project: from data integration to data coordination," *SIGMOD Record*, vol. 32, no. 3, pp. 53–58, 2003.

[4] I. Tatarinov and A. Halevy, "Efficient Query Reformulation in Peer Data Management Systems," in *Proc. of SIGMOD*, 2004.

[5] F. Mandreoli, R. Martoglia, P. Tiberio, S. Sassatelli, and W. Penzo, "Using Semantic Mappings for Query Routing in a PDMS Environment," in *Proc. of SEBD*, 2006.
[6] F. Mandreoli, R. Martoglia, S. Sassatelli, and W. Penzo, "SRI: Exploiting Semantic Information for Effective Query Routing in a PDMS," in *Proc. of WIDM*, 2006.
[7] "BRITE," `http://www.cs.bu.edu/brite/`.
[8] A. Crespo and H. Garcia-Molina, "Routing Indices for Peer-to-Peer Systems," in *Proc. of ICDCS*, 2002.
[9] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications," in *Proc. of SIGCOMM*, 2001.
[10] K. Aberer, P. Cudré-Mauroux, and M. Hauswirth, "The Chatty Web: Emergent Semantics Through Gossiping," in *Proc. of WWW*, 2003.
[11] B. Cooper, "Using Information Retrieval Techniques to Route Queries in an InfoBeacons Network," in *Proc. of DBISP2P*, 2004.
[12] P. Haase, R. Siebes, , and F. van Harmelen, "Peer Selection in Peer-to-Peer Networks with Semantic Topologies," in *Proc. of ICNSW*, 2004.
[13] G. Koloniari and E. Pitoura, "Content-Based Routing of Path Queries in Peer-to-Peer Systems," in *Proc. of EDBT*, 2004.
[14] S. Michel, M. Bender, P. Triantafillou, and G. Weikum, "IQN Routing: Integrating Quality and Novelty in P2P Querying and Ranking," in *Proc. of EDBT*, 2006.
[15] W. Nejdl et al., "Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-to-Peer Networks," in *Proc. of WWW*, 2003.
[16] 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.
[17] F. Mandreoli, R. Martoglia, and P. Tiberio, "Approximate Query Answering for a Heterogeneous XML Document Base," in *Proc. of WISE*, 2004.
[18] J. Madhavan, P. A. Bernstein, A. Doan, and A. Y. Halevy, "Corpus-based schema matching," in *Proc. of ICDE*, 2005.