Toward an Effective and Efficient Query Processing in the NeP4B Project*

C. Gennaro[1], F. Mandreoli[2], R. Martoglia[2], M. Mordacchini[1], S. Orlando[3], W. Penzo[4], S. Sassatelli[2], P. Tiberio[2]

Abstract. In this paper we present our main current research activity in the Italian co-funded FIRB Project NeP4B (Networked Peers for Business). In particular, we provide an overview of our P2P query routing approach which combines semantics and multimedia aspects in order to make query processing effective and efficient.

Motivation

Information and communication technologies (ICTs) over the Web have become a strategic asset in the global economic context. The Web fosters the vision of an Internet-based global marketplace where automatic cooperation and competition are allowed and enhanced. This is the stimulating scenario of the ongoing Italian Council co-funded NeP4B (Networked Peers for Business) Project whose aim is to develop an advanced technological infrastructure for small and medium enterprises (SMEs) to allow them to search for partners, exchange data and negotiate without limitations and constraints.

According to the recent proposal of Peer Data Management Systems (PDMSs) [1, 2], the project infrastructure is based on independent and interoperable semantic peers who behave as nodes of a virtual peer-to-peer (P2P) network for data and service sharing. In this context, a semantic peer can be a single SME, as well as a mediator representing groups of companies, and consists of a set of data sources (e.g. data repositories, catalogues) placed at the P2P network disposal through an OWL ontology. These data sources include multimedia objects, such as the descriptions/presentations of the products/services extracted from the companies' Web sites. This information is represented by means of appropriate multimedia at-

* This work is partially supported by the Italian co-funded FIRB Project NeP4B (Networked Peers for Business). `http://dbgroup.unimo.it/nep4b`.

[1] ISTI - CNR, Area della Ricerca di Pisa, Italy, {firstname.lastname}@isti.cnr.it
[2] DII - Univ. of Modena and Reggio Emilia, Italy, {firstname.lastname}@unimo.it
[3] DSI - Ca' Foscari University of Venice, Italy, orlando@dsi.unive.it
[4] DEIS - University of Bologna, Italy, wilma.penzo@unibo.it

tributes in the peers' ontologies (e.g. `image` in Peer1's ontology of Figure 1) that are exploited in the searching process by using a SPARQL-like language properly extended to support similarity predicates. As an example, let us consider the query in Figure 1 which asks Peer1 for the Italian products similar to the represented one. As can be seen, the clause `WHERE` is extended with the operator `LIKE` indicating the referenced image.

Each peer is connected to its neighbors through semantic mappings, appropriately extended with scores expressing their strength, which are exploited for query processing purposes: In order to query a peer, its own ontology is used for query formulation and semantic mappings are used to reformulate the query over its immediate neighbors, then over their immediate neighbors, and so on. For instance, in Figure 1 the concepts `product`, `origin` and `image` of the sample query must be reformulated in `item`, `provenance` and `photo` when the query is forwarded to Peer2. As to the computation of the semantic mappings and the associated scores, in the project an effective approach which exploits the semantics and the structure of the available schemas and which descends from the one proposed in [3] is employed.

In such a distributed scenario, where query answers can come from any peer in the network which is connected through a semantic path of mappings [2], a key challenge is *query routing*, i.e. the capability of selecting a small subset of relevant peers to forward a query to. Flooding-based techniques are indeed not adequate for both efficiency and effectiveness reasons: Not only they overload the network (forwarded messages and computational effort required to solve queries), but also overwhelm the querying peer with a large number of results, mostly irrelevant.

As part of the NeP4B project, we leverage our distinct experiences on semantic [4,5] and multimedia [6] query routing and propose to combine the approaches we presented in past works in order to design an innovative mechanism which exploits the two main aspects characterizing the querying process in such a context: The semantics of the concepts in the peers' ontologies and the multimedia contents in the peers' repositories. More precisely, since the reformulation process may lead to some semantic approximation, we pursue *effectiveness* by selecting, for each query, the peers which are semantically best suited for answering it. Further, since the execution of multimedia similarity queries is inherently costly (they typically require the application of complex distance functions) we also pursue *efficiency* by limiting their forwarding to the network's zones where potentially matching objects could be found, while pruning the others.

On Query Routing

In the context of the NeP4B Project a query posed at a peer usually contains predicates involving the concepts of the peer's ontology and multimedia similarity constraints. Thus, both the semantics and the multimedia features of the retrieved data are fundamental: An image that, according to some given multimedia fea-

tures, is very similar to the required one is not a relevant result if the two represented concepts are completely semantically unrelated (e.g. a church and a hotel with similar shapes). Thus, in order to provide an effective and efficient query processing both the aspects need to be considered.

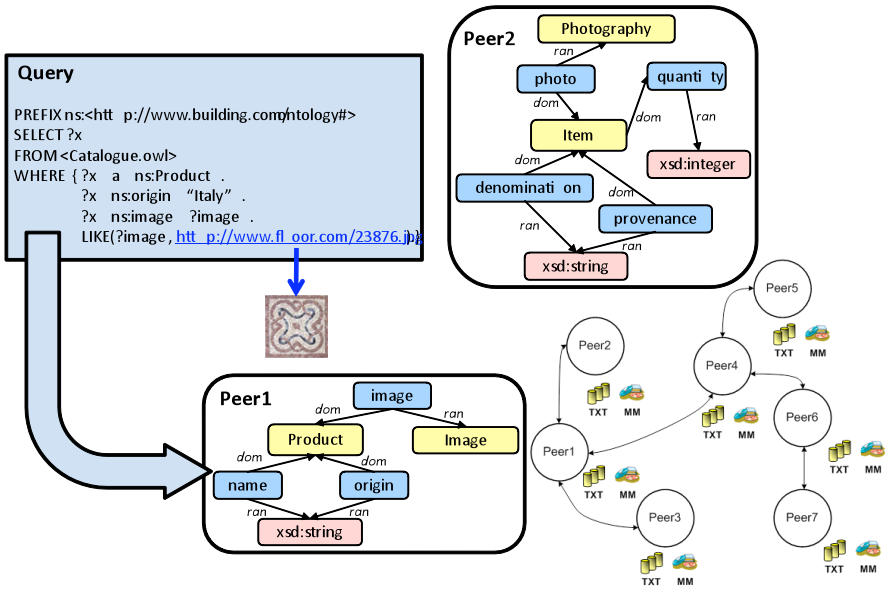


Fig. 1. Reference scenario

In [4] an effective semantic query routing approach for PDMSs is presented. In the work, each peer maintains cumulative information summarizing the semantic approximation capabilities, w.r.t. its ontological schema, of the whole subnetworks rooted at each of its neighbors. Such information is kept in a local data structure called *Semantic Routing Index (SRI)*. In particular, a peer p having n neighbors and m concepts in its ontology 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 in [0,1] expressing how the j-th concept is semantically approximated by the subnetwork rooted at the i-th neighbor, i.e. by each semantic path of mappings originated at the i-th neighbor. A sample fragment of Peer1'SRI is represented in Figure 2, where, for instance, the score 0.34 in the Peer4 row and the `Product` column is the outcome of the aggregation of the scores associated to the paths Peer4, Peer4-Peer5, Peer4-Peer6 and Peer4-Peer6-Peer7. Notice that, since SRIs summarize the semantic information offered by the network, they need to change whenever the network itself changes. SRIs construction and evolution is thus managed in an incremental fashion by exploiting the specifically devised process presented in [4].

When a peer needs to forward a query, it accesses its own SRI for determining the neighboring peers which are most semantically related to the query's concepts.

For instance, considering the concept `Product` of the query in Figure 1, the most promising subnetwork would be the one rooted at Peer 2 (score 0.73 in Figure 2). More precisely, if the query involves more concepts, the choice of the best neighbors is given by applying scoring rules which, for each neighboring peer, combine the corresponding SRI grades of all the query's concepts [5]. As a result, each neighbor is associated a score in [0,1] reflecting the semantic relevance of its subnetwork w.r.t. the query. These scores allow the forwarding peer to compute a ranking that can be exploited in order to implement different semantic routing policies [5].

SRI_{Peer1}	Product	name	origin	...
Peer1	1.0	1.0	1.0	...
Peer2	0.73	0.88	0.75	...
Peer3	0.69	0.48	0.30	...
Peer4	0.34	0.21	0.22	...

Fig. 2. Peer1's SRI

As to the multimedia contents, *MRoute* [6] is a P2P routing index mechanism for efficient similarity search in metric spaces. To this end, each peer builds, for each of its objects O and for each considered multimedia feature F_i, different *feature indices*, in order to allow both multi- and single-feature queries. Each index exploits a *reference object* R_j^{Fi}, i.e. an object that is used to determine the position of other objects in a metric space. More precisely, it is a k binary vector $(b_0,...,b_{k-1})$ which originates from a uniform partition of the distance between the object and the reference point $d(O.F_i, R_j^{Fi})$ into k intervals $[a_0,a_1),\ldots, [a_{k-1},a_k]$. The vector contains one bit $b_s=1$ in correspondence with the interval $[a_s,a_{s+1})$ in which $d(O.\ F_i, R_j^{Fi})$ falls, 0s in all the other entries (e.g. see Figure 3).

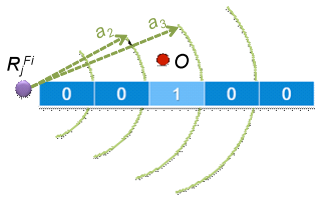


Fig. 3. Feature index of object O for the feature F_i

Then, considering the reference object R_j^{Fi}, each peer maintains a *global index* as the sum of the local indices associated with it (shown with dark background in the example of Figure 4). Such an index shows how the peer's objects are distributed in the given intervals. Thus, it can be regarded as a histogram of a peer's objects feature distribution. Moreover, each peer also maintains *Multimedia Routing Indices (MRIs)* for each of its neighbors. Each MRI represents the aggregated description of the resources available in the subnetwork rooted at each neighbor and is built by summing up the global indices of the peers in the subnetwork (shown in

light background in Figure 4 for Peer1). More details on the process of construction and evolution of the MRIs can be found in [6].

When a query is issued to the network, the query object (i.e. the `LIKE` argument) is mapped into the same metric space, thus giving rise to as many bit vectors as the number of reference objects. Each peer that receives the query forwards it to the neighbors whose indices intersect the query ones. Further, since MRIs can be viewed as histograms, they allow peers to estimate the number of potentially matching objects in the neighbor's subnetwork. In particular, each neighbor is assigned a score in [0,1] reflecting that estimation and a ranking on the most promising directions can be computed. For example, going back to Figure 4 and supposing the bit vector of the query w.r.t. reference object R_j^{Fi} is (0,0,1,0,0), the most promising neighbor for Peer1 would be Peer4.

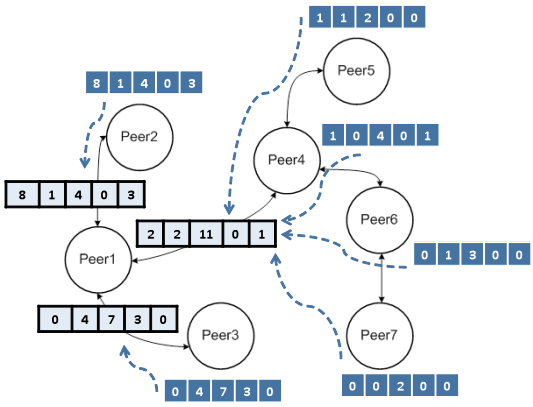


Fig. 4. Creation of Peer1's MRIs

Leveraging our experience on SRIs and MRoute, our final objective in the NeP4B Project is the development of an advanced routing mechanism that allows each peer to rank its own neighbors w.r.t. their ability to answer a given query both effectively (i.e. minimizing the information loss due to its reformulation along semantic mappings) and efficiently (i.e. minimizing the network load due to the exploration of useless subnetworks). At each query reformulation step, such a routing mechanism works by exploiting and properly combining the neighbor rankings computed by the two approaches. Indeed, when a peer p receives a query, both the SRI and MRoute approaches associate each p's neighbor a score in [0,1] quantifying the semantic relevance and the amount of potential matching objects in its subnetwork, respectively. These scores are homogeneous (i.e. graded in [0,1]) and can be combined by means of a meaningful aggregation function in order to obtain a unique ranking. In [7] it is stated that optimal aggregation algorithms can work only with monotone aggregation function. Typical examples of these functions are the min and mean functions (or the sum, in the case we are not interested in having a combined grade in the interval [0,1]).

As an example of how the aggregation process works, let us go back to the sample query in Figure 1 and suppose Peer1 obtains the scores in Figure 5. The rankings computed by SRI and MRoute are thus Peer2-Peer3-Peer4 and Peer4-Peer3-Peer2, respectively. An example of straightforward aggregation function is the standard fuzzy conjunction *min(score1,score2)*. Thus, by using it, we compute the following final ranking: Peer3-Peer4-Peer2. As a result, the most promising subnetwork will be the one rooted at neighbor Peer3.

Notice that, in the computation, irrelevant subnetworks (i.e. subnetworks with a score of 0) can be safely pruned. The obtained ranking reflects the foreseen subnetworks ability in solving the received query both at schema (SRI-based information) and at multimedia (MRoute-based information) level and can thus be exploited in order to implement clever routing strategies like the ones proposed in [5].

	SRI_{Peer1}	$MRoute_{Peer1}$	*min( )*
Peer2	0.65	0.38	0.38
Peer3	0.51	0.53	0.51
Peer4	0.48	0.76	0.48

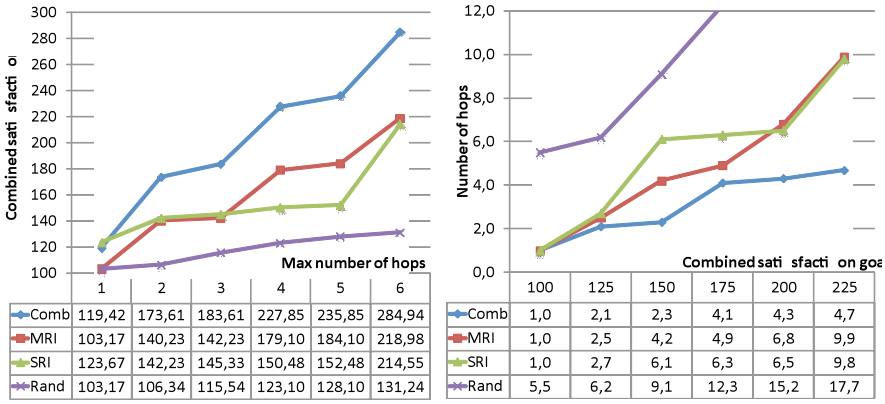
Fig. 5. Peer1's scores for the sample query

Experiments

In this section we present an initial set of experiments we performed in order to evaluate our combined query routing approach. Notice that, since we are currently in the initial phase of our testing, the considered scenarios are not particularly complex; in the future we will enrich them with more complicated and larger ones. For our experiments, we exploited our simulation environments for putting into action the SRI [5] and MRoute [6] approaches. Through these environments we modelled scenarios corresponding to networks of semantic peers, each with its own schema, consisting of a small number of concepts, and a repository of multimedia objects. We chose peers belonging to different semantic categories, where the peers in the same category have schemas describing the same topic from different points of view and own multimedia data related to that topic. The schemas are distributed in a clustered way: This reflects realistic scenarios where nodes with semantically similar contents are often connected through semantic mappings. As to the multimedia contents, we use 1300 images taken from the Corel Photo CDs and characterized by two MPEG-7 standard features: scalable color and edge histogram. We tested our techniques on different alternatives network topologies, randomly generated with the BRITE tool[5], whose mean size was in the order of few dozens of nodes. In order to evaluate the performance of our techniques we simulated the querying process by instantiating different queries on

[5] `http://www.cs.bu.edu/brite/`

randomly selected peers and propagating them until a stopping condition is reached: We evaluated the effectiveness improvement by measuring the quality of the results (*combined satisfaction*) when a given number of *hops* has been performed or, in a dual way, the efficiency improvement by measuring the number of hops required to reach a given combined satisfaction goal. Combined satisfaction is a specifically introduced quantity which grows proportionally to the goodness of the results returned by each queried peer: Each contribution is computed by combining the semantic mapping scores of the traversed peers (satisfaction measure [3]) and the multimedia similarity scores of the retrieved objects. The search strategy employed is the depth first search (DFS). In our experiments we compare our neighbor selection mechanism based on a combination of SRIs and MRoute (*Comb*) with the two mechanisms which only exploit the SRI (*SRI*) and MRI (*MRI*) values and with a baseline corresponding to a random strategy (*Rand*). The employed aggregation function is the mean. Notice that all the results we present are computed as a mean on some query executions.



	1	2	3	4	5	6
Comb	119,42	173,61	183,61	227,85	235,85	284,94
MRI	103,17	140,23	142,23	179,10	184,10	218,98
SRI	123,67	142,23	145,33	150,48	152,48	214,55
Rand	103,17	106,34	115,54	123,10	128,10	131,24

	100	125	150	175	200	225
Comb	1,0	2,1	2,3	4,1	4,3	4,7
MRI	1,0	2,5	4,2	4,9	6,8	9,9
SRI	1,0	2,7	6,1	6,3	6,5	9,8
Rand	5,5	6,2	9,1	12,3	15,2	17,7

Fig. 6. Obtained combined satisfaction for a given number of hops (left) and mean number of hops for a combined satisfaction goal (right).

Figure 6 shows the trend of the obtained combined satisfaction when we gradually vary the stopping condition on hops (left) and the dual situation (right) where the number of hops required to reach a given satisfaction goal is measured. As we expected, both the *SRI* and the *MRI* strategies outperform the *Rand* one, but, as we can see, the winner is the *Comb* mechanism. In particular, the difference between *SRI* - *MRI* and *Comb* performance appears closer in the initial part of the graphs but becomes increasingly more significant at growing stop conditions. This means that *Comb* is indeed able to discriminate better subnetworks to explore and consequently increases the combined satisfaction and decreases the number of hops in a more substantial way. As an example of this behavior, when we executed a query involving the concept `Monument` and a similarity constraint on an image of the

Pisa tower, we observed that the *Rand* strategy worked by randomly selecting peers which were completed unrelated with the image and the concept required. On the other hand, the *SRI* strategy proceeded by firstly selecting some peers which have the concept `Monument` (and thus a very high SRI's score) but no image similar to the Pisa tower. Further, the *MRI* approach preferred some peers which store the images of some chimneys (whose multimedia features were very similar to the Pisa tower's ones) even if they were associated to the concept `Factory`. Only the *Comb* strategy was able to identify the best peers, i.e. the peers where the images of the Pisa tower are associated to concepts similar to the required one.

Conclusions

In this paper we presented our idea of query routing for the NeP4B Project which combines two strategies in order to answer queries both effectively and efficiently. The initial set of experiments we performed shows promising results. In the future we will deepen the testing of our techniques by using larger and more complex scenarios.

References

1. Arenas, M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R. J. and Mylopoulos, J. (2003). The Hyperion Project: From Data Integration to Data Coordination. *SIGMOD Record,* 32(3): 53-58.

2. Halevy, Y. A., Ives, Z., Madhavan, J., Mork, P., Suciu, D. and Tatarinov, I. (2004). The Piazza Peer Data Management System. *IEEE Transactions on Knowledge and Data Engineering (TKDE),* 16(7): 787-798.

3. Mandreoli F., Martoglia R., and Tiberio P. (2004). Approximate Query Answering for a Heterogeneous XML Document Base. In Proc. of the *5th International Conference on Web Information Systems Engineering (WISE)*: 337-351.

4. Mandreoli, F., Martoglia, R., Penzo, W. and Sassatelli, S. (2006). SRI: Exploiting Semantic Information for Effective Query Routing in a PDMS. In *Proc. of the 8th ACM CIKM International Workshop on Web Information and Data Management (WIDM)*: 19-26.

5. Mandreoli, F., Martoglia, R., Penzo, W., Sassatelli, S. and Villani, G. (2007). SRI@work: Efficient and Effective Routing Strategies in a PDMS. In *Proc. of the 8th International Conference on Web Information Systems Engineering (WISE)*: 285-297.

6. Gennaro, C., Mordacchini, M., Orlando, S. and Rabitti, F. (2007). MRoute: A Peer-to-Peer Routing Index for Similarity Search in Metric Spaces. In *Proc. of the 5th International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P).*

7. Fagin, R., Lotem, A. and Naor, M. (2003). Optimal Aggregation Algorithms for Middleware. *Journal of Computer and System Sciences,* 66: 47-58.