Dynamic Configurability of a Semantic Matchmaker for Ontology-based Resource Discovery in Open Distributed Systems *

Silvana Castano, Alfio Ferrara, and Stefano Montanelli

Università degli Studi di Milano, via Comelico 39, 20135, Milano, Italy
{castano,ferrara,montanelli}@dico.unimi.it

Abstract. An important requirement for dynamic collaboration and semantic interoperability in open distributed systems is to minimize the effort needed for answering resource discovery queries by simultaneously guaranteeing the accuracy of the answers. In this paper, we propose query policies for dynamically configuring the matchmaker of a given node by taking into account the current workload of the peer as well as the requested degree of accuracy of the matching process embedded in the incoming request.

1 Introduction

We consider the semantic interoperability problem in open distributed contexts like semantic Grids and peer-based systems, where a set of independent peer nodes without prior reciprocal knowledge and no degree of relationship dynamically need to cooperate by sharing their resources (such as data, documents, services). Furthermore, due to the dynamicity and variability of collaboration and sharing requirements, we assume that no centralized authority manages a comprehensive view of the resources shared by all the nodes in the system [1]. Rather, each node is responsible of providing the knowledge description of the resources to be shared through its own ontology, thus originating a multi-ontology interoperability context. Each node implements a *semantic matchmaker* which is responsible for the evaluation of semantic affinity between an incoming query and its node ontology in order to assess whether it can provide resources matching the target. An important requirement for dynamic collaboration in such open contexts is to minimize the effort required for answering queries while simultaneously guaranteeing the quality of the answers for effective resource sharing and interoperability [2]. In this paper, we propose query policies for dynamically configuring the matchmaker of a given node by taking into account the effort that it can afford as well as the requested degree of accuracy of the answer. In

* This paper has been partially funded by "Wide-scalE, Broadband, MIddleware for Network Distributed Services (WEB-MINDS)" FIRB Project funded by the Italian Ministry of Education, University, and Research, and by NoE INTEROP, IST Project n. 508011 - 6th EU Framework Programme.

particular, in Section 2 we describe resource discovery in the HELIOS open distributed system. In Section 3, we present resource discovery queries and policies in HELIOS, while in Section 4 we describe policy selection for ontology matching configuration. In Section 5, we discuss the main applicability issues. In Section 6, we discuss related work. Finally, in Section 7, we give our concluding remarks.

2 Resource discovery in HELIOS

HELIOS (Helios EvoLving Interaction-based Ontology knowledge Sharing) is a system for ontology-based knowledge discovery and sharing in peer-based open distributed systems. In HELIOS, each peer provides a semantically rich representation of the information resources to be shared by means of a *peer ontology* which is defined according to H-MODEL [3]. H-MODEL is a language independent ontology model capable of representing the relevant features of the information resources to be shared in a Semantic Web-compatible manner, in terms of concepts, properties, and semantic relations. The HELIOS resource discovery process is based on appropriate queries, called *probe queries*, that are used to formulate knowledge requests among the peers of the system. A receiving peer uses a semantic matchmaker, called H-MATCH [4], to evaluate the semantic affinity between the target resources specified in the probe query and its peer ontology. A peer answers to an incoming probe query by sending back the matching concept descriptions in form of metadata extracted from its peer ontology. The idea behind this approach is to first discover the peers that provide knowledge about one or more resources of interest, to subsequently propagate queries to acquire data in an optimized way. A graphical representation of the HELIOS knowledge discovery process is shown in Figure 1. H-MATCH computes a *semantic affinity* value $SA(c, c')$, that is, the measure of the level of matching of two concepts c and c', by properly considering both their linguistic and contextual features. Linguistic features refer to names of concepts and their meaning. Contextual features refer to the concept context, namely the set of properties and concepts directly related to the given concept in an ontology. H-MATCH performs ontology concepts matching at different levels of depth, with four different *matching models* spanning from surface to intensive matching, with the goal of providing a wide spectrum of metrics suited for dealing with many different matching scenarios that can be encountered in comparing concept descriptions of real ontologies. The *surface matching* is defined to consider only the names of concepts. Surface matching is suited for dealing with high-level, poorly structured ontological descriptions. The *shallow matching* is defined to consider both concept names and concept properties. With this model, we want a more accurate level of matching, by taking into account not only the concept names but also information about the presence of properties and about their cardinality constraints. The *deep matching* model is defined to consider concept names and the whole context of concepts, by considering also semantic relations. Finally, the *intensive matching* model is defined to consider, in addition to the features of the deep model, also property values, for providing the highest accuracy in semantic

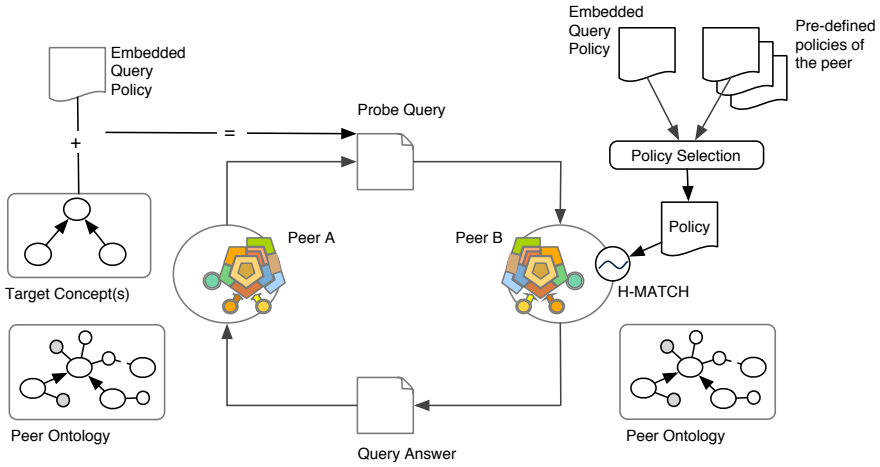

Fig. 1. Knowledge discovery process in HELIOS

affinity evaluation. The semantic affinity $SA(c, c')$ is evaluated as follows:

$$SA(c, c') = W_{LA} \cdot LA(c, c') + (1 - W_{LA}) \cdot CA(c, c') \quad (1)$$

where W_{LA} is a weight expressing the relevance of the linguistic affinity in the semantic affinity evaluation. A threshold Thr specifies the minimum semantic affinity value required to consider c and c' as matching concepts. For a more detailed description of H-MATCH and of the different matching models see [3]. The H-MATCH algorithm is exploited as a matchmaker tool by the peers for processing incoming requests over the ontology. An important requirement for peer-based collaboration and interoperability is related to the capability of a peer of balancing the effort requested for processing an incoming query with its actual workload, in order to satisfy a great number of incoming queries at the best. In HELIOS, this problem can be addressed by dynamically configuring the H-MATCH parameters, that is W_{LA} and Thr, based on the contents of a probe query by simultaneously taking into account the processing capabilities of the answering peer. To this end, we introduce the notion of *query policy* and a mechanism for policy selection based on i) an accuracy factor for the answer and ii) a cost factor for query processing.

3 Query policies

A policy in HELIOS specifies a parameter setting for the H-MATCH matchmaker configuration and it is defined as follows.

Definition 1. *A policy P is a 4-tuple in the form $< Q_{Type}, M_{Model}, Thr, W_{LA} >$, where:*

- Q_{Type} = simple | conjunctive | disjunctive*: it denotes the query type associated with the policy. In particular, simple queries are composed by a single target concept. Conjunctive queries and disjunctive queries are composed by more than one target concept. Conjunctive queries are satisfied by answers that contain at least a matching concept for each concept in the query target. Disjunctive queries are satisfied by answers that contain a matching concept for at least one concept of the query target.*
- M_{Model} = surface | shallow | deep | intensive*: it specifies the matching model to be used for configuring* H-MATCH*.*
- $Thr \in (0, 1]$*: it denotes the matching threshold value to be used to configure* H-MATCH*.*
- $W_{LA} \in [0, 1]$*; it denotes the linguistic affinity weight to be used for configuring* H-MATCH*.*

In HELIOS, policies are associated both with probe queries and with the peers of the network. A probe query in HELIOS is defined as follows.

Definition 2. *A probe query Q is a pair of the form $< T_Q, P_Q >$, where T_Q denotes the target of Q and P_Q denotes the policy embedded in Q.*

The probe query target specifies a set of concepts describing the resources that a peer is going to discover over the network, while the embedded policy denotes the parameters to be used for configuring H-MATCH at the destination for processing Q. An example of probe queries with their embedded policies is shown in Figure 2, together with the corresponding H-MODEL graphical representation of their target concepts. The clause Find contains the description of the target concept(s). Each target concept c is characterized by an optional set of properties and semantic relations, represented by the With_Property clause and by the With_Relation clause, respectively. Properties and/or relations can be specified to constrain the semantics of a target concept.

4 Policy selection for resource discovery

Each HELIOS peer is configured with a set of pre-defined policies for processing incoming probe queries. When a probe query is received, the answering peer compares the embedded query policy against its pre-defined policies. If the embedded query policy is compatible with the pre-defined policies of the answering peer, the embedded policy is used to configure H-MATCH. Otherwise, the answering peer choses its pre-defined policy that best fits the incoming probe query. The process of policy selection is shown in Figure 3.

4.1 Cost and accuracy factors

For policy selection, a policy P is associated with a *cost factor* CF^P and with an *accuracy factor* AF^P.

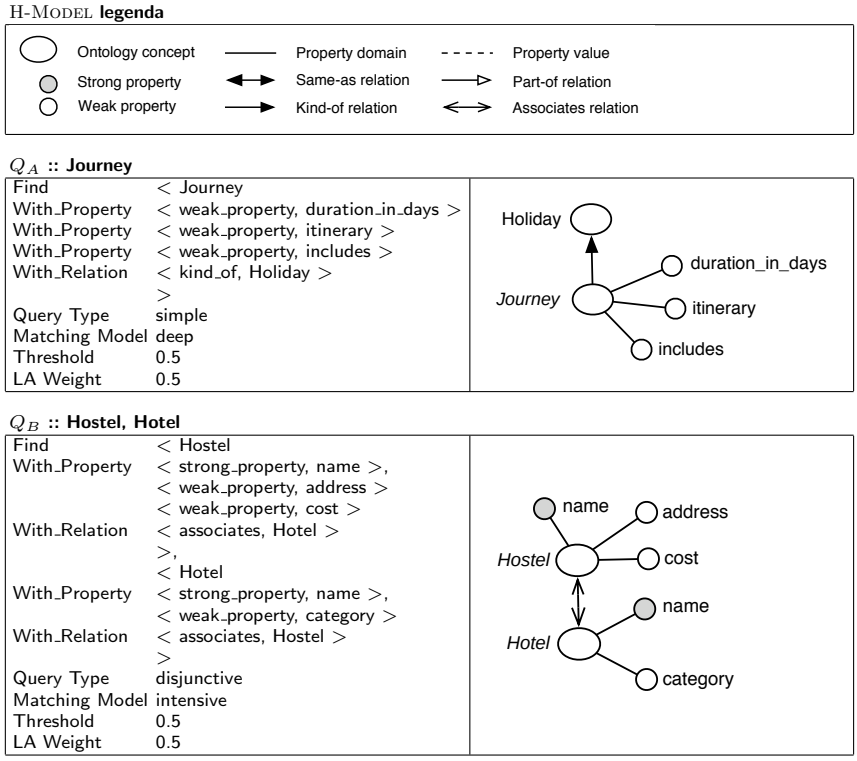

Fig. 2. Example of probe query composition

Cost factor. The cost factor of a policy P, denoted by CF^P, provides a measure of the computational cost of probe query processing using P, and depends basically on the number of H-MATCH executions and on the complexity of each H-MATCH execution, respectively. The number of H-MATCH executions is basically affected by the query type and by the threshold value, while the complexity of a single H-MATCH execution depends on the matching model. A formal definition of the cost factor is given in [5].

Accuracy factor. The accuracy factor of a policy P, denoted by AF^P, provides a measure of the accuracy of the query results obtained using P in terms of *precision* and *recall* [6]. Precision is defined as the number of relevant concepts effectively retrieved over the total number of retrieved concepts. Recall is defined as the number of relevant concepts effectively retrieved over the total number of relevant concepts. The accuracy factor is affected by the matching model, by the threshold, and by the W_{LA} weight. In particular, we have determined experimentally five accuracy classes of policies. Each class has associated an accuracy factor AF from 1 to 5. The higher the AF, the higher the answer accuracy provided by the policy. Given a policy P, the accuracy factor AF^P

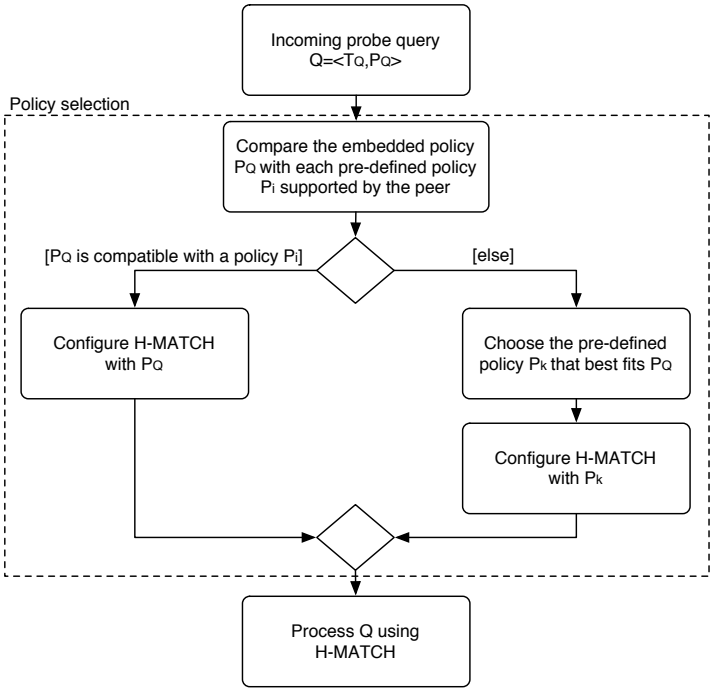

Fig. 3. The process of policy selection

associated with P is determined by means of a heuristics that associates a value of accuracy with each combination of matching model, threshold, and W_{LA} weight on an experimental basis. A formal definition of the accuracy factor is given in [5].

We have defined three different pre-defined policies (see Figure 4); each peer of the system is equipped with one or more pre-defined policies depending on its computational capabilities. The first pre-defined policy $P_{LC/LA}$ is the low

$$P_{LC/LA} =< \mathsf{surface}, \mathsf{simple}, 0.9, 0.8 >$$
$$P_{MC/MA} =< \mathsf{shallow}, \mathsf{conjunctive}, 0.7, 0.7 >$$
$$P_{HC/HA} =< \mathsf{intensive}, \mathsf{disjunctive}, 0.4, 0.6 >$$

Fig. 4. The three pre-defined policies

cost/low accuracy policy. It can be adopted by a peer that is overloaded by a high number of incoming requests. This policy provides a low level of answer accuracy but a high level of efficiency in probe query processing. The second pre-defined

policy $P_{MC/MA}$ represents the medium cost/medium accuracy policy. The third pre-defined policy $P_{HC/HA}$ represents the high cost/high accuracy policy. It has the highest level of accuracy but requires a higher computational effort to the answering peer.

4.2 Policy selection

When a probe query is received, the answering peer has to determine whether the embedded query policy is compatible with its pre-defined policies. To this end, a *compatibility* threshold CT is set by the peer, representing the minimum level of compatibility required to consider an embedded policy to fit a pre-defined policy. The compatibility between P_Q and a pre-defined policy P_i is evaluated by taking into account i) the compatibility between the cost factors of P_Q and P_i and ii) the compatibility between the accuracy factors of P_Q and P_i. A comprehensive policy selection factor $PSF(P_Q, P_i)$ is evaluated by taking into account the cost and accuracy factors of P_Q and P_i, respectively. If P_Q is compatible with P_i (i.e., $PSF(P_Q, P_i) \geq CT$), P_Q is selected for processing the incoming probe query Q; otherwise the pre-defined policy P_i with the highest $PSF(P_Q, P_i)$ is selected for processing the incoming probe query Q. In order to evaluate the $PSF(P_Q, P_i)$ factor, we define the coefficients $A_{Cost}(P_Q, P_i)$, Cost-based Applicability, and $A_{Accuracy}(P_Q, P_i)$, Accuracy-based Applicability, defined as follows:

$$A_{Cost}(P_Q, P_i) = min(1, \frac{CF^{P_i}}{CF^{P_Q}}) \tag{2}$$

$$A_{Accuracy}(P_Q, P_i) = min(1, \frac{AF^{P_i}}{AF^{P_Q}}) \tag{3}$$

where CF^{P_i} and CF^{P_Q} denote the cost factor of P_i and P_Q, respectively, and AF^{P_i} and AF^{P_Q} denote the accuracy factor of P_i and P_Q, respectively. Based on the cost-based and accuracy-based applicability coefficients, we provide the following definition of policy selection factor for an embedded policy P_Q and a pre-defined policy P_i.

Definition 3. *Given an embedded policy P_Q and a pre-defined policy P_i, the policy selection factor $PSF(P_Q, P_i)$ between P_Q and P_i is defined as:*

$$PSF(P_Q, P_i) = \frac{A_{Cost}(P_Q, P_i) + A_{Accuracy}(P_Q, P_i)}{2} \tag{4}$$

The policy selection factor ensures that the embedded query policy is always selected as far as it is compatible the pre-defined policies at the destination. Otherwise, the answering peer supplies an answer which is the most accurate answer that can be provided given the actual peer workload.

Once a policy P is selected the H-Match algorithm is configured according to the parameters specified by P and the probe query answer is composed by including all matching concepts exceeding the matching threshold (simple query). For conjunctive queries, the answer includes all matching concepts exceeding the

matching threshold if there is at least one matching concept for each concept in the query target. For disjunctive queries, the answer is includes all matching concepts exceeding the threshold for at least one concept in the query target.

5 Applicability issues and considerations

As an example of resource discovery, we consider the probe queries of Figure 2, together with their embedded policies. In the example we assume that three peers, namely P1, P2, and P3, share the same peer ontology, shown in Figure 5, describing knowledge in the tourism domain. Peers are configured with the pre-defined

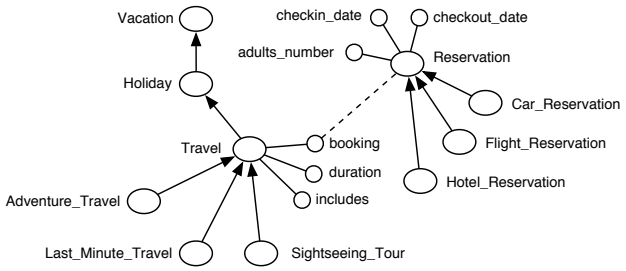

Fig. 5. Example of a peer ontology

policies in Figure 4. In particular, P1 is configured with the policy $P_{LC/LA}$, P2 is configured with the policy $P_{MC/MA}$, and P3 is configured with the policy $P_{HC/HA}$, as shown in Figure 6. The probe queries in Figure 2 search for concepts similar to Journey and to Hotel and Hostel against the peer ontology in Figure 5.

Cost and accuracy evaluation. When the probe queries are processed, the first step is to evaluate their cost and accuracy factors, respectively. In our example, we assume that the compatibility threshold CT is 0.6 for all the peers and we determine the cost and accuracy factors as follows:

$$QueryA = \begin{cases} CF = 33.4 \\ AF = 3 \end{cases}$$
$$QueryB = \begin{cases} CF = 57.35 \\ AF = 3 \end{cases}$$

In the subsequent step, the cost factor and the accuracy factor are calculated for the pre-defined policies associated with each peer as follows:

$$\mathsf{P1} \rightarrow P_{LC/LA} = \begin{cases} CF = 1 \\ AF = 2 \end{cases}$$
$$\mathsf{P2} \rightarrow P_{MC/MA} = \begin{cases} CF = 10.4 \\ AF = 3 \end{cases}$$

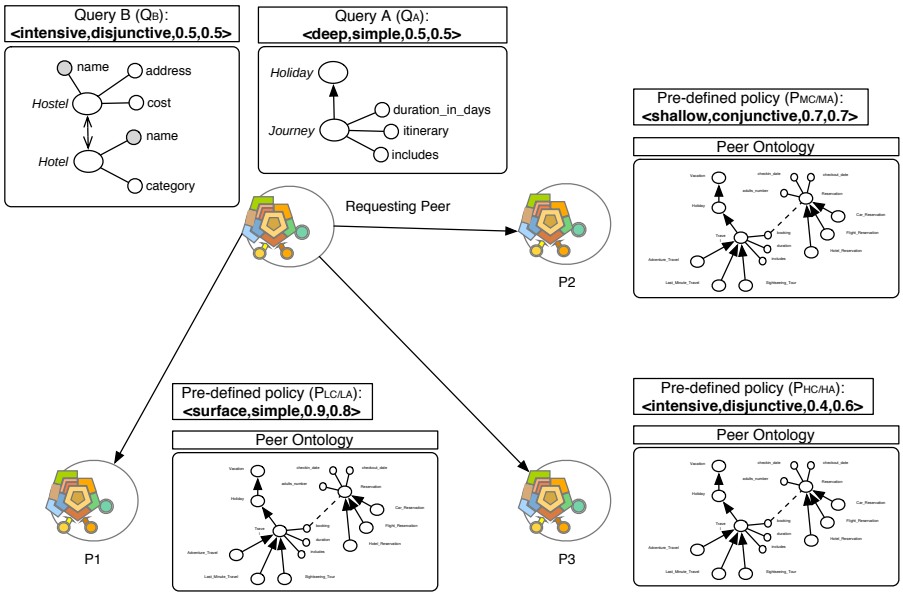

Fig. 6. Example of a resource discovery scenario based on probe query policies

$$\mathsf{P3} \rightarrow P_{HC/HA} = \begin{cases} CF = 57.35 \\ AF = 5 \end{cases}$$

Then, the embedded policies are compared with the pre-defined policies of the peers according to (4). The results of the comparison process are shown in Table 1. Based on such results, the peer P1 cannot apply the embedded policies

Table 1. Compatibility results for policy selection

	$P_{LC/LA}$	$P_{MC/MA}$	$P_{HC/HA}$
Q_A	0.35	0.65	1
Q_B	0.33	0.59	1

for processing the probe queries, because they are incompatible with the policy $P_{LC/LA}$ with respect to the compatibility threshold CT. The peer P2, that is configured with the pre-defined policy $P_{MC/MA}$, can apply the embedded policy only for Q_A, because the compatibility between its pre-defined policy and the embedded policy of Q_A is higher than CT. Finally, P3, that is configured with the pre-defined policy $P_{HC/HA}$, can process both the query Q_A and the

query Q_B by using their embedded policies, because both are compatible with its predefined policy.

The example shows how the peers P2 and P3, which are configured for supporting the processing of probe queries with high computational costs, can effectively answer to the incoming probe queries by adopting the embedded policy, i.e., by guaranteeing the level of accuracy required by the requesting peer, while the peer P1, which is capable to reply only to low cost queries, has to adopt its own pre-defined policy in order to reply to the incoming queries.

6 Related work

The problem of answering queries has been deeply studied in the data integration literature [7–9]. These approaches assume to have mappings over a set of models and address the problem of computing the tuples that satisfy a query in all the models in the set. With respect to the data integration approach, the focus of this paper is on finding concepts that are similar to a set of target concepts expressed in the query, without assuming to have pre-defined mappings among the peer ontologies. Our approach provides an approximate measure of semantic similarity between concepts, instead of a set of data answers to a query. An interesting direction of future work is to combine our approach with the query answering techniques proposed in data integration, with the aim of using H-Match for finding the mappings that are required for query answering in data integration. The problem of answering queries has been studied also in peer-based systems mainly addressing the problem of the efficient routing of queries over the network. For example, Edutella [10] provides an infrastructure for sharing metadata in RDF format. The network is segmented into thematic clusters. In each cluster, a mediator semantically integrates source metadata. A mediator handles a request either directly or indirectly: directly, by answering queries using its own integrated schema; indirectly, by querying other cluster mediators by means of a datalog-based query processing module. With respect to Edutella, we refer to a pure P2P system, where each peer has equal capabilities and functionalities, without mediators. Each peer acquires a knowledge of the network by means of probe queries, and exploits this knowledge for subsequently routing appropriately queries for data retrieval. In the SWAP project [11] (Semantic Web and Peer-to-Peer), each peer implements an *ontology extraction* method to extract from its different information sources an RDF(S) description (ontology). Such ontologies are used by the *SeRQL Query Language* to perform query processing. Peers storing knowledge semantically related to a target concept are localized through SeRQL views defined on specific similarity measures. Views from external peers are integrated through an *ontology merging* method to extend the knowledge of the receiving peer according to a rating model. Our approach has in common with SWAP the idea of using similarity measures for finding knowledge over a P2P network. Contribution of our work with respect to this approach is on one side related to a more flexible way of producing similarity measures with

H-Match and on the other side on the use of policies for configuring the query processing of the peers in the system.

7 Concluding remarks

In this paper, we have proposed the notion of query policy for dynamically configuring the matchmaker of a given node by taking into account its current workload as well as the requested degree of accuracy for the matching process. Our future work on this topic will be devoted to extensively test the policy-based approach, in order to provide a complete set of experimental results regarding the quality of query answers and the overall system efficiency using policy-based matching configuration. To this end, we plan to use a network simulator for testing the accuracy of query results by varying the peer workload. We are combining this work with the semantic routing protocol we are developing [12] in order to make query processing and routing more effective. A further work will be devoted also to define a set of rules for automatically configuring each peer, by taking into account its overload in terms of query received per time unit. The idea is to have peers capable of reacting to the amount of network traffic by re-configuring their query processing policies.

References

1. ACM SIGMOD Record: Special topic section on peer to peer data management. Volume 32. ACM Press (2003)
2. Yang, B., Garcia-Molina, H.: Improving search in peer-to-peer systems. In: Proc. of the 22nd Int. Conf. on Distributed Computing Systems. (2002)
3. Castano, S., Ferrara, A., Montanelli, S., Racca, G.: Semantic Information Interoperability in Open Networked Systems. In: Proc. of the Int. Conference on Semantics of a Networked World (ICSNW), in cooperation with ACM SIGMOD 2004, Paris, France (2004)
4. Castano, S., Ferrara, A., Montanelli, S., Racca, G.: From surface to intensive matching of semantic web ontologies. In: Proc. of the 3rd DEXA Int. Workshop on Web Semantics (WEBS 2004), Zaragoza, Spain, IEEE Computer Society (2004) 140–144
5. Cominardi, F.: Ontology Query Resolution in Peer-to-Peer Systems. Master's thesis, Università degli Studi di Milano (2004)
6. Korfhage, R.: Information Storage and Retrieval. Wiley Computer Publishing (1997)
7. Halevy, A.: "Answering Queries Using Views: A Sourvey". VLDB Journal **10** (2001) 270–294
8. Lenzerini, M.: Data Integration: A Theoretical Perspective. In: Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002), Madison, Wisconsin, USA, ACM (2001) 233–246
9. Calvanese, D., Damaggio, E., De Giacomo, G., Lenzerini, M., Rosati, R.: Semantic Data Integration in P2P Systems. In: Databases, Information Systems, and Peer-to-Peer Computing, First International Workshop, DBISP2P, Berlin, Germany, Springer Verlag (2003) 77–90

10. W. Nejdl et al.: EDUTELLA: a P2P Networking Infrastructure Based on RDF. In: Proc. of the 11th Int. World Wide Web Conference (WWW 2002), Honolulu, Hawaii, USA (2002)
11. J. Broekstra et al.: A Metadata Model for Semantics-Based Peer-to-Peer Systems. In: Proc. of the 1st WWW Int. Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID 2003), Budapest, Hungary (2003)
12. S. Castano and A. Ferrara and S. Montanelli and E. Pagani and G. P. Rossi and S. Tebaldi: On Combining a Semantic Engine and Flexible Network Policies for P2P Knowledge Sharing Networks. In: Proc of the 1st DEXA Workshop on Grid and Peer-to-Peer Computing Impacts on Large Scale Heterogeneous Distributed Database Systems (GLOBE 2004). (2004)