Matching Techniques for Resource Discovery in Distributed Systems Using Heterogeneous Ontology Descriptions *

S. Castano, A. Ferrara, S. Montanelli, G. Racca
Università degli Studi di Milano
DICO - Via Comelico, 39, 20135 Milano - Italy
{castano,ferrara,montanelli,racca}@dico.unimi.it

Abstract

In distributed environments like Peer-to-Peer networks and Grids, where the resources to be shared are provided by many different nodes generally spanned across multiple organizations, dynamic resource discovery is a central problem. In this paper, we propose a semantic approach for the dynamic resource discovery based on ontologies for a semantically rich description of the metadata characterizing the resources to be shared, and on semantic matching techniques for dynamically and flexibly comparing ontological resource descriptions against a target resource request.

1. Introduction

In distributed environments like Peer-to-Peer networks and Grids, where the resources to be shared are provided by many different nodes generally spanned across multiple organizations, dynamic resource discovery is a central problem. By *dynamic resource discovery* we mean the capability of finding the existing resources in the network that best match the requirements of a given resource request, either for a computational task or for answering a user query. For example, in Peer-to-Peer networks the problem of sharing distributed contents is crucial, and recent research is devoted to provide techniques for evolving from basic P2P networks supporting only file exchanges using simple filenames as metadata, to more complex systems like schema-based P2P networks, capable of supporting the exchange of complex resources (e.g., documents, services) by using explicit schemas to describe their resources, usually using RDF and thematic ontologies as metadata [10, 13]. In such systems, queries asking for a given content can be more precise and flexible since they are posed against such schemas. As another example, the resources matching problem in Grids involves assigning resources to tasks, given task requirements and resource policies [5, 14]. A common issue in resource discovery both in P2P and Grids is related to the fact that data and resources need to be described in a way that is understandable and usable by the community that is target user. This is the goal of ontologies, part of the Semantic Web effort, which are generally recognized as an essential tool for allowing communication and knowledge sharing among distributed users and applications, by providing a common understanding of a domain of interest [2, 11].

In this paper, we address the dynamic resource discovery problem, and we propose a semantic approach based on: i) *ontologies* for a semantically rich description of the metadata characterizing the resources to be shared; ii) *semantic matching techniques* for comparing ontological resource descriptions against a target resource request. The approach originates from the query matching techniques we have developed in the framework of the HELIOS project, where we have developed an infrastructure for supporting knowledge sharing in a schema-based P2P network, using ontologies as metadata for content descriptions [7, 8]. The proposed semantic matching techniques are conceived to operate in a dynamic fashion, to cope with the inherent dynamism of resource discovery in distributed contexts.

The paper is organized as follows. In Section 2, we discuss the dynamic resource discovery problem, and we present a reference ontology model for resource metadata representation. In Section 3, we describe the H-MATCH algorithm for ontology-based resource matching. In Section 4, we provide an example of application of resource matching in a P2P schema-based network and we discuss applicability issues for Grids. In Section 5, we discuss related work. Finally, in Section 6, we give our concluding remarks.

*This paper has been partially funded by the WEB-MINDS FIRB Project, funded by the Italian Ministry of Education, University, and Research.

2. The proposed approach to resource discovery in distributed environments

A node provides different kind of resources to the network: i) *data/contents*, such as structured, semi-structured, or unstructured data (e.g., relational databases, XML documents): ii) *services*, such as Web services or remote procedures; iii) *physical resources*, such as computational resources (e.g., processor types, storage capabilities). In our approach, metadata related to the resources that a node exposes to the network are described by means of a node ontology. The use of an ontology for specifying resource metadata allows us to exploit Semantic Web techniques for knowledge representation, to provide a semantically rich representation of the resources of interest in terms of concepts, properties, and semantic relations. A node ontology is organized in a two-layer architecture (see Figure 1) where the upper layer represents the *resource knowledge* and the lower layer represents the *network knowledge* of a given node, respectively.

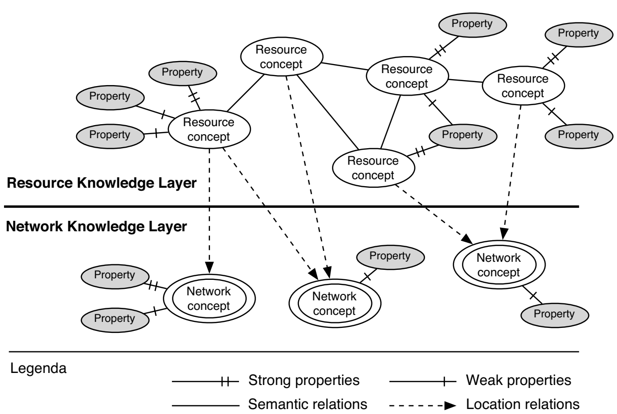

Figure 1. Reference ontology architecture of a generic node N

For a given node N, the Resource Knowledge Layer describes the metadata knowledge of the resources N exposes to the network. The Network Knowledge Layer describes the knowledge that N has of other nodes of the system for the effective routing of target requests. Such knowledge depends on the goal to be pursued (e.g., knowledge on semantic neighbors of N in a P2P network to support dynamic thematic communities/semantic routing protocols [8]).

2.1. Resource description model

Resource metadata are organized in the ontology layers according to a model based on *properties*, *concepts*, *semantic relations* and *location relations*, formally defined as follows.

Given a set $\mathcal{N}$ of names and a set $\mathcal{T}$ of pre-defined basic data types (e.g., basic data types of XML Schema or RDF [1, 4]), a node ontology $\mathcal{NO}$ is defined as a 4-tuple of the form $\mathcal{NO} = (P, C, SR, LR)$, where:

- P is a set of properties. A property $p \in P$ is defined as a pair of the form $p = (n_p, dt_p)$, where $n_p \in \mathcal{N}$ is the property name, and $dt_p \in \mathcal{T} \cup RC$ is the property data type, either a basic type or a resource concept. A set of property constraints PC is defined for properties. Each property constraint associates a property p with a concept c, by specifying the minimal cardinality of p in c. A property constraint $pc \in PC$ is a 3-tuple of the form $pc = (p, c, k)$, where $p \in P$ is a property, $c \in C$ is a concept, and $k \in \{0, 1\}$ is the minimal cardinality associated with p when applied to c. We call strong properties the properties with $k = 1$, and weak properties the ones with $k = 0$.
- $C = RC \cup NC$ is a set of concepts of $\mathcal{NO}$, where RC is a set of *resource concepts* in the resource knowledge layer, and NC is a set of *network concepts* in the network knowledge layer. A concept $c \in C$ is defined as a pair of the form $c = (n_c, PC_c)$, where $n_c \in \mathcal{N}$ is the concept name, and $PC_c \subseteq PC$ is a set, possibly empty, of property constraints.
- SR is a set of semantic relations. A semantic relation $sr \in SR$ is defined as a binary relation of the form $sr(c, c')$, where c and $c' \in RC$ are the resource concepts and sr is the relation holding between them. Semantic relations state the kind of semantic dependency/connection holding between two concepts in the ontology. Example of semantic relations are same-as, kind-of, part-of, associates.
- LR is a set of location relations. A location relation $lr \in LR$ is defined as a binary relation of the form $lr(c, c')$, where $c \in RC$ is a resource concept in the resource knowledge layer and $c' \in NC$ is a network concept in the network knowledge layer, respectively.

3. Matching techniques for resource discovery

The ontology resource description model is exploited to perform dynamic matching at different levels of depth, with different degrees of flexibility and accuracy of results by taking into account different levels of richness in resource descriptions and considering various metadata elements (e.g., resource name, resource properties) separately or in combination. In this Section, we first describe the requirements and foundations of the H-MATCH algorithm.

Then, we illustrate H-MATCH and its matching models. The matching techniques presented in this paper rely on our experience in developing i) schema matching techniques in the ARTEMIS data integration system [6] for the aspects related to linguistic analysis, and ii) query matching algorithms in the HELIOS system for content retrieval in Peer-to-Peer environments with ontological requirements from autonomous peers [8].

3.1. Matching requirements and foundations

In order to enforce dynamic resource discovery, we require a flexible algorithm with the aim of facing two different requirements of the dynamic matching process. The first requirement regards the balance between the linguistic and the contextual features of concept descriptions. The meaning of ontology concepts in a node ontology depends basically on the names chosen for their definition and on the relations they have with other concepts in the ontology. We are interested in addressing the fact that those features can have a different impact in different ontology structures and with respect to different matching purposes. A second requirement regards the context evaluation, in which we distinguish between properties and concepts. The role of the properties in the concept definition might have a different relevance in ontologies of different nodes. As an example, if an ontology is defined describing high structured resources, such as a database, the properties which describe the structure of each concept have a strong impact on the concept meaning evaluation. Furthermore, the composition of the context and its extension in terms of number of adjacents has an impact on the matching accuracy and on its performance, in terms of information to be processed for matching evaluation. The aim of H-MATCH is to allow a dynamic choice of the kind of features to be considered in the matching process.

Linguistic features. To capture the meaning of names in a node ontology, we refer to a thesaurus of terms and terminological relationships among them. The terminological relationships that are considered are: synonymy, hyponymy/hypernymy, meronymy. The thesaurus is automatically built by exploiting WordNet [12] as a source of lexical information. The thesaurus can then be interactively enriched by the ontology designer, if additional knowledge has to be taken into account for names in a given node ontology.

Contextual features. The H-MATCH algorithm evaluates the semantic affinity between two concepts by taking into account their contexts. Given a concept $c \in RC$, we denote by $P(c) = \{p_i \mid (p_i, c, k) \in PC_c\}$ the set of properties appearing into a property constraint associated with c (i.e., the properties of c), and by $SR(c) = \{c_j \mid sr_j(c, c_j)\}$ the set of adjacents of c, respectively. The context of a concept is defined as the union of the properties and of the adjacents of c, that is, $Ctx(c) = P(c) \cup SR(c)$.

The *semantic affinity*, that is, the measure of the level of matching of two ontology concepts c and c', is evaluated by weighting both the terminological relationships in the thesaurus and the semantic relations in the contexts of c and c', respectively. A weight W_{tr} associated with a terminological relationship tr expresses its implication for semantic affinity. Different types of relationships have different implications for semantic affinity. In particular, based on our experience in schema matching techniques [6], we have $W_{\mathsf{synonymy}} \geq W_{\mathsf{hyponymy/hypernymy}} \geq W_{\mathsf{meronymy}}$. We assign the highest strength to the synonymy relationship, since it indicates semantic affinity more precisely than remaining terminological relationships. A weight W_{sr} associated with a semantic relation expresses the implication of the connection posed by the relation on the involved concepts for semantic affinity evaluation purposes. The greater the weight associated with a semantic relation, the higher the strength of the semantic connection between concepts. Furthermore, we associate a weight W_{sp} to strong properties, and a weight W_{wp} to weak properties, respectively, with $W_{\mathsf{sp}} \geq W_{\mathsf{wp}}$. In fact, strong properties are mandatory related to a concept and are essential to give its structural description. Weak properties are optional for the concept in describing its structure, and, as such, are less important in featuring the concept than strong properties. The weights associated with the terminological relationships are exploited for performing linguistic affinity evaluation, while the weights associated with properties and semantic relations are exploited for performing contextual affinity evaluation, respectively.

Linguistic affinity. The aim of the linguistic affinity function $LA(c, c')$ is to evaluate the semantic affinity of two concepts c e c' by considering the semantic contents of their names, referring to a thesaurus Th. For this purpose, an affinity function $A(t, t')$ is defined to evaluate the affinity between two terms t and t' in Th. $A(t, t')$ of two terms t and t' is equal to the strength value of the highest-strength path of terminological relationships between them if at least one path exists, and is zero otherwise, where a path strength is computed by multiplying the strengths of each terminological relationship involved in the path, that is:

$$A(t,t') = \begin{cases} W_{tt_1} \cdot W_{t_1t_2} \cdot \ldots \cdot W_{t_nt'} & \text{iff } \exists\, t \Rightarrow t' \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $W_{t_it_j}$ is the strength of the terminological relationship between two terms t_i and t_j in Th, and $\Rightarrow$ denotes a path of terminological relationships. The linguistic affinity

$LA(c,c')$ of two concepts c and c' coincides with the linguistic affinity of their names, that is:

$$LA(c,c') = A(n_c, n_{c'}) \quad (2)$$

where n_c and $n_{c'}$ are the names of c and c', respectively.

Contextual affinity. The aim of the contextual affinity function is to calculate a measure of affinity between concepts based on their contexts. To this purpose, we evaluate the linguistic affinity of properties and adjacents of the considered concepts, as well as the degree of closeness between them. Let $e \in Ctx(c)$ be an element in the context of c and $e' \in Ctx(c')$ be an element of the context of c', respectively. A closeness function $Cl(e,e')$ calculates a measure of closeness between elements of two concept contexts (i.e., two properties, two semantic relations, or a semantic relation and a property, respectively). $Cl(e,e')$ exploits the weights associated with properties and semantic relations and returns a value in the range [0,1] proportional to the absolute value of the complement of the difference between the weights associated to the elements, that is:

$$Cl(e,e') = \begin{cases} T(dt_e, dt_{e'}) \cdot (1- \mid W_e - W_{e'} \mid) \\ \qquad\qquad \text{iff } e \in P(c), e' \in P(c') \\ 1- \mid W_e - W_{e'} \mid \\ \qquad\qquad \text{otherwise} \end{cases} \quad (3)$$

where dt_e and $dt_{e'}$ are the pre-defined data types of e and e', and W_e and $W_{e'}$ are the corresponding weights associated with e and e', respectively. The highest value (i.e., 1.0) is obtained when e and e' have the same weight. The higher the difference between their weights the lower the closeness value of e and e'. When two properties are compared, the compatibility between their data types is also evaluated by exploiting a pre-defined set CR of compatibility rules [1]. A compatibility function $T(dt, dt')$ on two data types dt and dt' returns 1 if dt and dt' are compatible according to CR, and 0 otherwise.

Closeness values are used to calculate the contextual affinity, using the contextual affinity function $CA(c,c')$, which evaluates a matching value $m(e,e')$ between each context element $e \in Ctx(c)$ and each context element $e' \in Ctx(c')$. The value $m(e,e')$ is calculated by composing the linguistic affinity value calculated using (1) and the closeness value calculated using (3), as follows:

$$m(e,e') = A(n_e, n_{e'}) \cdot Cl(e,e') \quad (4)$$

We denote by $\overline{m}(e)$ the best-matching value among the matching values obtained for a given element $e \in Ctx(c)$ using (4), that is:

$$\overline{m}(e) = \max(\{m(e,e')\} \mid e' \in Ctx(c')) \quad (5)$$

The contextual affinity $CA(c,c')$ is finally calculated as the average of all the best-matching values of each element of $Ctx(c)$, that is:

$$CA(c,c') = \frac{\sum_{i=1}^{n} \overline{m}(e_i)}{n} \quad (6)$$

where $n = \mid Ctx(c) \mid$.

Semantic affinity. The semantic affinity function $SA(c,c')$ gives a comprehensive value of affinity of two concepts, combining both the linguistic and the contextual affinity values. The problem of dynamically setting the balance between the linguistic and the contextual features of a node ontology in the matching process is addressed by setting a weight $W_{LA} \in$ [0,1] which measures the degree of the impact of the linguistic affinity in the semantic affinity evaluation process. The semantic affinity $SA(c,c')$ of two concepts c and c' is evaluated as the weighted sum of their linguistic affinity calculated using (2), and then contextual affinity calculated using (6), that is:

$$SA(c,c') = W_{LA} \cdot LA(c,c') + (1 - W_{LA}) \cdot CA(c,c') \quad (7)$$

3.2. The H-MATCH algorithm

H-MATCH provides different degrees of flexibility and different matching models to cope with different levels of detail in describing the target resource of interest. H-MATCH can be dynamically configured by setting the impact of the linguistic and the contextual affinity, and by choosing dynamically which part of concept context has to be considered in the matching process. Three different matching models are proposed to configure H-MATCH.

- *Shallow matching.* The shallow matching is adopted to consider only the linguistic information provided by the concept names and by the reference thesaurus. The quality of the semantic affinity evaluation is influenced by the choice of the concept names in the ontology definition. Meaningful and appropriate names will guarantee more appropriate results.
- *Intermediate matching.* The intermediate matching is performed by considering concept names and also concept properties. With this model, we want a more accurate level of matching by taking into account not only the name but also information about the structure of a concept.
- *Deep matching.* The deep matching model considers concept names and the whole context of concepts. The

[1] CR can be defined like in [6], where we have defined compatibility rules for matching structured data types in the ARTEMIS data integration system.

deep matching guarantees the highest level of accuracy in the semantic affinity evaluation.

The input of the H-MATCH algorithm is constituted by: two concepts c and c'; the matching model; the value of the weight W_{LA}. Deep and 0.5 are the default values for the matching model and W_{LA}, respectively. $W_{LA} = 0.5$ ensures that the linguistic affinity and the contextual affinity have the same impact in the semantic affinity evaluation. The output of H-MATCH is the semantic affinity value of c and c', calculated as the weighted sum of their linguistic affinity and contextual affinity. A high level description of the H-MATCH algorithm is shown in Figure 2.

algorithm	H-MATCH$(c, c', model =$ "deep"$, W_{LA} = 0.5)$
input	the concepts c and c', the matching model $\in$ { shallow; intermediate; deep }, and the weight $W_{LA} \in$ [0,1]
output	$SA(c, c')$ the semantic affinity value between c and c'
begin	
case	$model$ of
	shallow
	Extract n_c and $n_{c'}$;
	intermediate
	Extract n_c and $n_{c'}$ and assign properties of c to $Ctx(c)$ and properties of c' to $Ctx(c')$, respectively;
	deep
	Extract n_c and $n_{c'}$ and assign properties and adjacents of c to $Ctx(c)$ and properties and adjacents of c' to $Ctx(c')$, respectively;
	$LA = LA(c, c')$;
	$CA = CA(c, c')$;
	$SA = W_{LA} \cdot LA + (1 - W_{LA}) \cdot CA$;
return	SA ;
end	

Figure 2. The H-MATCH algorithm

The shallow matching is recommended when only resource names can be specified for discovery process. Since it is based only on linguistic information, the shallow matching guarantees better performance than the other two matching models since requires less computation. On the other side, the intermediate and deep models support a finer comparison, and are well suited when a richer description of the target resource can be provided, where in addition to names, also properties and/or relations are known. These latter two models require more computation, but guarantee a more accurate level of matching.

4. Applicability of the proposed approach

Our approach based on ontologies and matching techniques can be adopted in distributed environments, such as schema-based P2P networks and Grid environments, for resource discovery. In a schema-based P2P network, peers can interact searching for semantically related contents with respect to a target request. We are implementing such an approach in HELIOS, where probe queries can be forwarded in the network with the goal of acquiring data resources and related metadata descriptions from the network. Here, we are developing a semantic routing infrastructure where the information in the network ontology layer is exploited to build an overlay network, enforcing a semantic neighborhood relationship among peers [8]. In a Grid environment, the resource matching problem is affected by the requirement of a resource description agreement. Without such an agreement and considering nodes exposing different kinds of services, our resource discovery approach allows a node N to find nodes which have compatible resource descriptions for a given request. Once compatible resources are discovered and registered trough appropriate location relations, the node N can perform more specific queries in order to exploit their resources, possibly using existing techniques for the resource selection in the Grid, based on property value comparisons [14].

In general, a node N searching for a target resource in the network, submits a request containing an ontological description of the target, according to the reference model described in Section 2. In order to perform the request distribution over the network, suitable communication and routing protocols are employed (e.g., the JXTA framework in P2P networks). Each node compares the incoming request against its ontology, by applying the H-MATCH algorithm. The matching evaluation performed by H-MATCH depends on the expressiveness of the ontological description contained in the request and on the desired level of accuracy (i.e., shallow, intermediate, deep matching).
Replying to the requesting node N, each node returns a list of *candidate resource* descriptions (that is, those which best match the target request using a threshold-based mechanism for filtering), together with their respective matching values. Based on received candidate descriptions and on their matching values, node N selects the relevant resources descriptions to be traced in its ontology. For each relevant resource description, a new concept in the Network Knowledge Layer is created, which describes the sending node which have provided such candidate resource descriptions. In order to maintain information about the nodes where a relevant resource description has been discovered, appropriate ontology concepts and location relations are defined in the node ontology, to trace correspondences between the resource concepts in the Resource Knowledge Layer and

the respective network concepts in the Network Knowledge Layer.

Example of resource discovery. As an example of resource discovery, we consider a Node A and a Node B, and two portions of their ontologies in the Tourism domain, reported in Figure 3. Ontology of Node A describes data

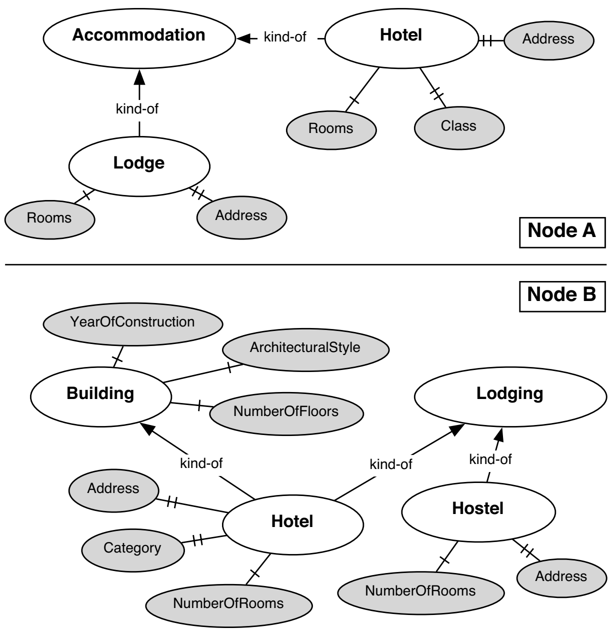

Figure 3. Examples of node ontologies

resources related to accommodation, hotel, and lodge, respectively. Ontology of Node B provides information about lodging and buildings, and, in particular, has a description of the concepts of lodging, building, hotel, and hostel, respectively. Suppose that Node A wants to discover resources matching the target concept hotel of its ontology. For this purpose, it sends a request to Node B, specifying an ontological description of hotel. Node B exploits the H-MATCH algorithm to find the concepts in its ontology that are semantically related to hotel. In Figure 4, we report matching results obtained using the deep, intermediate, and shallow matching models, respectively using the following weights: $W_{\mathsf{synonymy}} = 1.0$; $W_{\mathsf{hyponymy/hypernymy}} = 0.8$; $W_{\mathsf{meronymy}} = 0.5$; $W_{\mathsf{sp}} = 1.0$; $W_{\mathsf{wp}} = 0.5$; $W_{\mathsf{kind-of}} = 0.8$. According to all matching models, the hotel concept in the Node B ontology has the highest semantic affinity with the target concept. However, the H-MATCH distinguishes between the two hotel concepts using both the deep and the intermediate model, respectively. This highlights the fact that they have different properties and adjacents, although they have the same name. In addition, the deep and intermediate model results stress the differences among the concept of hostel, which is another kind of accommodation, and the more general concepts of lodging and building. Such a difference is not captured by applying the shallow model alone.

5. Related work

Ontology-based resource discovery is a research problem studied both in schema-based Peer-to-Peer networks and Grid systems.
In the EDUTELLA project [13], a Peer-to-Peer infrastructure for metadata sharing in RDF format is developed. A declarative language derived from RQL is implemented to formalize semantically rich queries allowing nodes to search for similar contents with respect to a target request. In [3] the matching techniques developed in Edamok are described. Edamok is a research project focused on knowledge sharing in Peer-to-Peer systems. Using an algorithm founded on description logics, the knowledge contained in different contexts is analyzed in order to find semantic mappings denoting peers interested in similar concepts. In [10], an RDF(S) metadata model for encoding semantic information is introduced allowing peers to handle heterogeneous views on the domain of interest. Semantically rich queries formalized in the SeRQL Query Language allow peers to localize nodes storing knowledge semantically related to a target concept. In [5] the authors suggest how ontologies describing Grid resources simplify the systematic building of Grid applications. The paper presents a DAML+OIL ontology for the Data Mining domain representing a taxonomic description of data mining tasks, methodologies, algorithm and software. Such an ontology can be searched and browsed in the *knowledge grid* environment by an application designer in order to improve and speed-up the design of distributed data mining applications on the Grid. In [14], the *matchmaker* approach is presented in order to provide a flexible strategy to the resource matching problem in the Grid. Such an approach is based on three ontologies: a *Resource ontology* describing resources expressed through an extended subset of CIM schema, a *Resource Request ontology* containing a request description and a *Policy ontology* describing resource authorization and usage policies. A set of *Matchmaking Rules* are defined in order to improve the effectiveness of the matching process.

Original contribution of our work. The H-MATCH algorithm is able to discover the location of compatible resources with respect to a target request, also in heterogeneous environments with lack of agreement on resource descriptions. Without requiring a complete description of re-

0-7695-2108-8/04 $ 20.00 © 2004 IEEE

Deep model		
SA(hotel,hotel)	=	0.8
SA(hotel,hostel)	=	0.63
SA(hotel,building)	=	0.55
SA(hotel,lodging)	=	0.54

Intermediate model		
SA(hotel,hotel)	=	0.83
SA(hotel,hostel)	=	0.65
SA(hotel,building)	=	0.5
SA(hotel,lodging)	=	0.4

Shallow model		
SA(hotel,hotel)	=	1.0
SA(hotel,building)	=	0.8
SA(hotel,hostel)	=	0.8
SA(hotel,lodging)	=	0.8

Figure 4. Example of matching results using H-MATCH

sources, the algorithm supports the use of combined semantic affinity evaluation strategies to obtain a flexible and dynamic matching process.

6. Concluding remarks

In this paper, we have presented matching techniques for ontology-based resource discovery in distributed environments. An implementation of the proposed approach for discovering information resources in schema-based Peer-to-Peer networks has been developed within the HELIOS (Helios Evolving Interaction-based Ontology knowledge Sharing) framework, conceived for supporting dynamic ontology-based knowledge sharing and evolution in P2P systems, and currently under development in the WEB-MINDS FIRB project. A description of the knowledge sharing and resource discovery process considering information resources in HELIOS is given in [7, 9]. In particular, we are currently testing the H-MATCH algorithm on several real ontology descriptions of information resources, with the goal of collecting results regarding its accuracy and its performance.

Future research issues in HELIOS regard the following aspects:

Resource metadata representation. We are developing a metadata repository implementing the ontology model and wrapping techniques to import real ontologies developed according to Semantic Web compatible representations (e.g., DAML+OIL, OWL).

Effective request distribution in the network. Node ontologies and H-MATCH are exploited in order to find nodes which can satisfy a given request by discovering semantic relationships among the ontologies they own. In particular in HELIOS, we are studying the problem of exploiting matching techniques for performing a more precise and focused routing of queries, in order to avoid the broadcast diffusion of queries which is generally typical of Peer-to-Peer systems.

References

[1] XML schema part 2: Datatypes. 2001. http://www.w3.org/-TR/xmlschema-2/.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

[3] P. Bouquet, B. Magnini, L. Serafini, and S. Zanobini. A SAT-based algorithm for context matching. In *Proceedings of the 4th International and Interdisciplinary COnference on Modeling and Using Context (CONTEXT-03)*, pages 66–79. Springer Verlag, 2003.

[4] D. Brickley and R. Guha. RDF vocabulary description language 1.0: RDF schema. 2003. http://www.w3.org/TR/rdf-schema/.

[5] M. Cannataro and C. Comito. A data mining ontology for grid programming. In *Proceedings of SemPGRID '03*, Budapest, May 2003.

[6] S. Castano, V. De Antonellis, and S. De Capitani Di Vimercati. Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering*, 13(2), 2001.

[7] S. Castano, A. Ferrara, and S. Montanelli. H-MATCH: an algorithm for dynamically matching ontologies in peer-based systems. In *Proceedings of the First Workshop on Semantic Web and Databases (SWDB-03), co-located at VLDB 03*, Berlin, Germany, September 2003.

[8] S. Castano, A. Ferrara, S. Montanelli, E. Pagani, and G. Rossi. Ontology-addressable contents in P2P networks. In *Proc. of WWW'03 1st SemPGRID Workshop*, Budapest, Hungary, May 2003.

[9] S. Castano, A. Ferrara, S. Montanelli, and D. Zucchelli. HELIOS: a general framework for ontology-based knowledge sharing and evolution in p2p systems. In *Proc. of DEXA'03 2nd Web Semantics Workshop*, Prague, Czech Republic, September 2003.

[10] J. Broekstra et al. A metadata model for semantics-based peer-to-peer systems. In *Proc. of WWW'03 1st SemPGRID Workshop*, Budapest, Hungary, May 2003.

[11] J.Heflin and J.Hendler. A portrait of the Semantic Web in action. *IEEE Intelligent System*, 16:54–59, May 2001.

[12] A. Miller. WordNet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[13] Nejdl et al. EDUTELLA: a P2P networking infrastructure based on RDF. In *Proc. of the Eleventh International World Wide Web Conference, WWW2002*, Honolulu, Hawaii, USA, May 2002.

[14] H. Tangmunarunkit, S. Decker, and C. Kesselman. Ontology-based resource matching in the grid – the grid meets the semantic web. In *Proceedings of SemPGRID '03*, Budapest, May 2003.