Semantic Information Interoperability in Open Networked Systems *

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 open networked systems, each node shares part of its informational resources on the network and is responsible of providing their ontological description. To enable semantic information interoperability in networked contexts with a multitude of autonomous ontologies, appropriate matching techniques are required to determine semantic mappings between concepts of different ontologies. In this paper, we describe H-MATCH, an algorithm for dynamically performing ontology matching in multi-ontology contexts. H-MATCH provides several matching models and metrics to work at different levels of depth, with different degrees of flexibility and accuracy, thus supporting semantic interoperability in a flexible way.

1 Introduction

In open networked systems like Peer-to-Peer and Grid, informational resources (e.g., datasets, documents) are provided by many different nodes generally spanned across multiple organizations. Open networked systems provide significant advantages in terms of flexibility, fault tolerance, and extensibility, but, in order to allow resource sharing, they introduce the problem of coping with the intrinsic dynamics of the system and with the need of a semantic description of resources [1–3]. Ontologies are generally recognized as an essential tool for describing resources in order to allow communication and knowledge sharing among distributed users and applications, by providing a common understanding of a domain of interest [4]. However, in open networked systems, where nodes are autonomous in terms of capabilities and resource management, information resource sharing can not be based on a global ontology. In fact, each node shares part of its resources on the network and is responsible of providing their ontological description. To enable resource sharing and semantic information interoperability in networked contexts with a multitude of autonomous ontologies, appropriate ontology matching techniques are thus required to determine semantic mappings between concepts of different ontologies that are semantically

related [5–7]. In this paper, we describe H-Match, an algorithm for dynamically performing ontology matching at different levels of depth, with different degrees of flexibility and accuracy. H-Match performs the matching of two ontologies, and provides, for each concept of one ontology, a ranking of similarity of the concepts in the other ontology. H-Match takes into account different levels of richness in resource descriptions, and allows one to consider various metadata elements of ontology descriptions separately or in combination. H-Match has been developed in the framework of the Helios project, whose aim is to support knowledge sharing and ontology-addressable content retrieval in peer-based systems [8].

The paper is organized as follows. In Section 2, we present the foundations of our approach for ontology matching. In Section 3, we describe the matching process. In Section 4, we discuss applicability issues in open networked contexts. In Section 5, we discuss related work on ontology matching. Finally, in Section 6, we give our concluding remarks.

2 Foundations for semantic information interoperability

The problem of matching ontologies in multi-ontology contexts of open networked systems introduces a number of challenging issues to be addressed for semantic information interoperability. In this paper, we focus on three major requirements:

– The advent of the Semantic Web has produced a large body of research around ontology languages, and many standard proposals that can be used for resource description in open networked systems have emerged (e.g., RDF, DAML+OIL, OWL). Different ontologies can describe the same domain using different descriptions of the same resources, also using the same language. An important requirement of ontology matching techniques is to capture the elements that are relevant for matching purposes in ontology resource descriptions in a language-independent manner, to be applicable in many contexts. H-Match addresses this requirement by exploiting a reference ontology model (see Section 2.1).
– The meaning of ontology concepts depends basically on the names chosen for their definition and on their contexts, namely on their properties and on the relations they have with other concepts in the ontology. We are interested in addressing the fact that these two features can have a different impact in different ontology structures and can play a different relevance in the matching process. In order to address this requirement, H-Match computes a comprehensive value of matching of two concepts, by combining both their linguistic and their contextual features. Furthermore, H-Match allows one to set the relevance of the linguistic and contextual features in the matching process.
– A key requirement of dynamic matching is the capability of coping with different levels of detail and structuring in describing the resources of interest, by considering various ontology elements separately or in combination.

In the remainder of the paper, we will show how H-MATCH addresses this requirement by providing four different matching models for dealing with different levels of depth and different degrees of flexibility and accuracy of results. These matching models are used for dynamically suiting the matching process to different levels of richness in ontology descriptions.

In the following, we describe the foundations of ontology matching in open networked systems.

2.1 A reference ontology model

H-MATCH is based on a reference ontology model, called H-MODEL, capable of representing the ontology features that are relevant for matching purposes in a language independent way, in terms of concepts, properties, and semantic relations. In the following we focus on the OWL language, which provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users, with different needs in terms of resource description accuracy [9]. Hereafter, we describe the constructs of H-MODEL, by showing the correspondent elements of OWL that can be mapped on them.

Concept. A concept c in H-MODEL is defined as a pair of the form $c = (n_c, P_c)$, where n_c is the concept name, and P_c is a set, possibly empty, of properties appearing in a property restriction associated with c (i.e., the properties of c). Each concept c in H-MODEL represents a class declaration in OWL. The concept name n_c is set by referring to the RDF ID associated with the class declaration in OWL.

Property. A property p is defined as a pair of the form $p = (n_p, PC_p)$, where n_p is the property name, and PC_p is a set of property constraints. Each property constraint associates a property p with a concept c, by specifying the minimal cardinality and the property value v_p of p in c. A property constraint $pc_p \in PC_p$ is a 3-tuple of the form $pc_p = (c, k_p, v_p)$, where c is a concept, $k \in \{0, 1\}$ is the minimal cardinality associated with p when applied to c, and v_p is the value associated with p when applied to c, and can be a datatype dt_p or a reference name. We call strong properties the properties with $k = 1$, and weak properties the ones with $k = 0$. Each property p represents a property declaration in OWL. The property name n_p is set by referring to the RDF ID associated with the property declaration. The property cardinality k_p as well as the property value v_p are enforced by exploiting OWL property restrictions. In particular, OWL cardinality restrictions are exploited for setting the cardinality of the property, while the AllValuesFrom, SomeValuesFrom, and hasValue OWL clauses are exploited for setting the value of the property, that can be a datatype, or a reference name representing the name of a concept or an individual in OWL.

Semantic relations. A semantic relation sr is defined as a binary relation of the form $sr(c, c')$, where c and c' are concepts and sr is the relation holding between

them. H-MODEL provides a same-as, a kind-of, and a part-of semantic relation. In particular, the same-as and the kind-of relations represent the equivalentClass and the subClassOf relations in OWL, respectively. Moreover, the kind-of relation is exploited for the representation of the intersectionOf, and the unionOf OWL operators. For an intersection clause of the form $A \equiv B \sqcap C$ we set two kind-of relations of the form A kind $-$ of B and A kind $-$ of C. For a union clause of the form $A \equiv B \sqcup C$ we set a two kind-of relations of the form B kind $-$ of A and C kind $-$ of A. Finally, the part-of relation is used for representing an enumerated class defined as a collection by means of the oneOf OWL clause. In particular, we set a part-of relation between each H-MODEL element representing a component of the collection and the H-MODEL concept representing the enumerated class. A H-MODEL graphical representation of OWL ontologies is shown in Figure 1 (see Section 3).

2.2 Linguistic features

Linguistic features refer to names of ontology elements and their meaning. To capture the meaning of names in an ontology in the matching process, we refer to a thesaurus Th of terms and terminological relationships among them. Th is automatically derived from the lexical system WordNet [10]. For the thesaurus construction, we note that in real ontologies, like OWL ontologies, ontology element names can be composed by one or more terms. Terms and terminological relationships to be stored in Th are selected as follows.

Basic terms. Given the set T of terms used as names of ontology elements, a term $t \in T$ is a basic term, denoted as bt, if an entry for bt exists in WordNet. In the thesaurus construction, an entry is defined for each basic terms $bt_i \in T$.

Compound terms. Given the set T of terms used as names of ontology elements, a term $t \in T$ is a compound term, denoted as $ct =< bt_1, bt_2, \ldots, bt_n >$, if ct is composed by more than one basic term bt_i and an entry for ct does not exists in WordNet. For managing compound terms and their terminological relationships, we follow an approach similar to [11]. In a typical compound term ct, one of its constituent basic terms denotes the central concept represented by ct, while the remaining basic terms denote a specification of such a central concept. In particular in English, the basic terms appearing on the left side of ct denote the specification of the meaning of term appearing on the right side. Our thesaurus organization makes explicit these considerations for compound terms, by introducing appropriate terms and terminological relationships entries in Th necessary to: i) correctly capture the meaning of a compound term and ii) to represent all terminological relationships of interest with other terms in Th. Given a compound term $ct =< bt_1, bt_2, \ldots, bt_n >$, an entry is defined in Th for ct and for each constituent basic term $bt_i, i = 1, 2, \ldots, n$.

Terminological relationships. Terminological relationships in Th are defined by considering terminological relationships among synsets provided by WordNet. Terminological relationships considered in Th together with their corresponding relationships in WordNet is shown in Table 1. In the thesaurus construction, for each basic term entry $bt_i \in Th$, a terminological relationship entry is defined

Terminological relationship in Th	Relationship in WordNet
SYN (synonymy)	synonymy
BT/NT (broader/narrower terms)	hyponymy hypernymy
RT (related terms)	meronymy coordinate terms

Table 1. Terminological relationships for ontology matching

in Th (i.e., SYN, BT/NT, or RT) if bt_i belongs to one of the WordNet synsets associated with the other basic term entries bt_j by a terminological relationship, according to the correspondences reported in Table 1. For each compound term entry $ct_i = <bt_{i_1}, bt_{i_2}, \ldots, bt_{i_n}> \in Th$, we define a terminological relationship BT/NT between ct_i and bt_{i_n}, to capture the fact that ct_i is a specification of bt_{i_n}. Furthermore, a RT terminological relationship between ct_i and each remaining basic term $bt_{i_j}, j = 1, 2, \ldots, n-1$ is also defined in Th, for capturing the fact that these terms qualify ct_i.

Weighting terminological relationships. In H-MATCH, a weight W_{tr} is associated with each terminological relationship $tr \in \{SYN, BT/NT, RT\}$ in Th in order to express its implication for semantic affinity. Different types of relationships have different implications for semantic affinity. In particular, we set $W_{SYN} \geq W_{BT/NT} \geq W_{RT}$. Synonymy is generally considered a more precise indicator of affinity than other relationships, consequently $W_{SYN} \geq W_{BT/NT}$. The lowest weight is associated with RT since it denotes a more generic relationship than BT/NT. Weights for terminological relationships are defined as shown in Table 2.

Terminological relationship	Weight
SYN	1.0
BT/NT	0.8
RT	0.5

Table 2. Weights associated with terminological relationships

2.3 Contextual features

Contextual features refer to the properties and concepts directly related to a given concept in an ontology. The importance of considering contexts when matching heterogeneous information is well-known [12]. For ontology matching,

a notion of context is essential, since semantic relations and properties play a key role in ontology specification. Given a concept c, we denote by $P(c)$ the set of properties of c, and by $C(c)$ the set of concepts that participate in a semantic relation with c (in the following referred to as *adjacents*), respectively. The context of a concept is defined as the union of the properties and of the adjacents of c, that is, $Ctx(c) = P(c) \cup C(c)$.

Weighting contextual features. In H-MATCH, a weight W_{sr} is associated with each semantic relation to denote the strength of the connection expressed 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 $W_{\mathsf{sp}} \geq W_{\mathsf{wp}}$ to capture the importance of the property in characterizing the concept for matching. In fact, strong properties are mandatory related to a concept and are relevant 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. Weights considered in H-MATCH for properties and semantic relations are summarized in Table 3 [1].

Context element	Weight
same_as relation	1.0
kind_of relation	0.8
part_of relation	0.5
strong_property	1.0
weak_property	0.5

Table 3. Weights associated with contextual features

2.4 Basic matching functions

Term affinity function. The aim of the term affinity function $\mathcal{A}(t, t') \rightarrow [0, 1]$ is to evaluate the affinity between two terms t and t' with respect to Th. $\mathcal{A}(t, t')$ of two terms t and t' is equal to the value of the highest-strength path of terminological

---

[1] Weight definition relies on our extensive experience in developing schema matching techniques. In particular, terminological relationships weights have been extensively experimented in the ARTEMIS integration system [13]. Following similar considerations, we have defined weights for contextual features, which have been tested on a number of real ontologies, with satisfactory results. However, the matching support tool under development allows the designer to modify such default values, by choosing a different configuration interactively, if necessary.

relationships between them in Th if at least one path exists, and is zero otherwise. A path strength is computed by multiplying the weights associated with each terminological relationship involved in the path, that is:

$$\mathcal{A}(t,t') = \begin{cases} \max_{i=1\ldots k} \{W_{t\rightarrow_i^n t'}\} & \text{if } k \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where: k is the number of paths between t and t' in Th; $t \rightarrow_i^n t'$ denotes the ith path of length $n \geq 1$; $W_{t\rightarrow_i^n t'} = W_{1_{tr}} \cdot W_{2_{tr}} \cdot \cdots \cdot W_{n_{tr}}$ is the weight associated with the ith path, where $W_{j_{tr}} \mid j = 1, 2, \ldots, n$ denotes the weight associated with the jth terminological relationship in the path.

Datatype compatibility function. A datatype compatibility function is defined to evaluate the compatibility of data types of two properties according to a pre-defined set CR of compatibility rules. The datatype compatibility function $\mathcal{T}(dt, dt') \rightarrow \{0, 1\}$ of two data types dt and dt' returns 1 if dt and dt' are compatible according to CR, and 0 otherwise, that is:

$$\mathcal{T}(dt, dt') = \begin{cases} 1 & \text{iff } \exists \text{ a compatibility rule for } dt, dt' \text{ in } CR \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For instance, with reference to XML Schema datatypes, examples of compatibility rules that hold between datatypes are: xsd:integer $\Leftrightarrow$ xsd:int, xsd:integer $\Leftrightarrow$ xsd:float, xsd:decimal $\Leftrightarrow$ xsd:float, xsd:short $\Leftrightarrow$ xsd:int.

Property and relation closeness function. A closeness function $\mathcal{C}(e, e') \rightarrow [0, 1]$ calculates a measure of the distance between two elements of concept contexts (i.e., two properties, two semantic relations, or a semantic relation and a property, respectively). $\mathcal{C}(e, e')$ exploits the weights associated with context elements and returns a value in the range [0,1] proportional to the absolute value of the complement of the difference between the weights associated with the elements, that is:

$$\mathcal{C}(e, e') = 1 - \mid W_e - W_{e'} \mid \quad (3)$$

where W_e and $W_{e'}$ are the weights associated with e and e', respectively. For any pairs of elements e and e', the highest value (i.e., 1.0) is obtained when weights of e and e' coincide. The higher the difference between W_e and $W_{e'}$ the lower the closeness value of e and e'. We note that the closeness of a property and a semantic relation can be evaluated in order to capture the structural heterogeneity among different ontological descriptions.

3 Ontology matching with H-Match

The general goal of ontology matching techniques is to find concepts that have a semantic affinity with a target concept, by producing a measure of their affinity. To satisfy the main requirements for ontology matching described in the

introduction, we have defined four matching models for H-MATCH, by exploiting the basic matching functions of Section 2.4. The matching models have been conceived to span 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 real ontologies (e.g., OWL ontologies). Each model calculates a semantic affinity value $SA_{c,c'}$ of two concepts c and c' which expresses their level of matching. $SA_{c,c'}$ is produced by considering linguistic and/or contextual features of concept descriptions. In a matching model, the relevance of the linguistic and the contextual features of c and c' in the matching process can be properly set. $W_{la} \in [0,1]$ is a weight used to set the impact of the linguistic affinity (and consequently of the contextual affinity) in the semantic affinity evaluation process.

3.1 Surface matching

The surface matching is defined to consider only the linguistic features of concept descriptions. Surface matching addresses the requirement of dealing with high-level, poorly structured ontological descriptions. Given two concepts c and c', surface matching provides a measure $SA_{c,c'}$ of their semantic affinity by exploiting the terminological affinity function (1), that is:

$$SA_{c,c'} = \mathcal{A}(n_c, n_{c'}) \tag{4}$$

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

3.2 Shallow matching

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 linguistic features but also information about the presence of properties and about their cardinality constraints. For property comparison, each property $p_i \in P(c)$ is matched against all properties $p_j \in P(c')$ using (1) and (3), and the best matching value $m(p_i)$ is considered for the evaluation of $SA_{c,c'}$, as follows:

$$m(p_i) = \max\{\mathcal{A}(n_{p_i}, n_{p_j}) \cdot \mathcal{C}(p_i, p_j)\}, \forall p_j \in P(c') \tag{5}$$

where n_{p_i} and n_{p_j} denote the names of p_i and p_j, respectively. $SA_{c,c'}$ is evaluated by the shallow matching as the weighted sum of the linguistic affinity of c and c', calculated using (1), and of their contextual affinity, calculated as the average of the property best matching values computed using (5), that is:

$$SA_{c,c'} = W_{la} \cdot \mathcal{A}(n_c, n_{c'}) + (1 - W_{la}) \cdot \frac{\sum_{i=1}^{|P(c)|} m(p_i)}{|P(c)|} \tag{6}$$

3.3 Deep matching

The deep matching model is defined to consider concept names and the whole context of concepts, in terms of properties and semantic relations. Each element $e_i \in Ctx(c)$ is compared against all elements $e_j \in Ctx(c')$ using (1) and (3) and the best matching value $m(e_i)$ is considered for the evaluation of $SA_{c,c'}$, as follows:

$$m(e_i) = \max\{\mathcal{A}(n_{e_i}, n_{e_j}) \cdot \mathcal{C}(e_i, e_j)\}, \forall e_j \in Ctx(c') \tag{7}$$

where n_{e_i} and n_{e_j} denote the names of e_i and of e_j, namely the names of a property or of an adjacent, respectively. $SA_{c,c'}$ is evaluated by the deep matching as the weighted sum of the linguistic affinity of c and c', calculated using (1), and of their contextual affinity, calculated as the average of the matching values for the elements of the context of c using (7), that is:

$$SA_{c,c'} = W_{la} \cdot \mathcal{A}(n_c, n_{c'}) + (1 - W_{la}) \cdot \frac{\sum_{i=1}^{|Ctx(c)|} m(e_i)}{|Ctx(c)|} \tag{8}$$

3.4 Intensive matching

The intensive matching model is defined to consider concept names, the whole context of concepts, and also property values, for the sake of a highest accuracy in semantic affinity evaluation. In fact, by adopting the intensive model not only the presence and cardinality of properties, but also their values have an impact on the resulting semantic affinity value. Given two concepts c and c', the intensive matching calculates a comprehensive matching value for the elements of the context of c such as in (7) and, moreover, calculates a matching value $v(p_i)$ for each property $p_i \in P(c)$. The matching value $v(p_i)$ is calculated as the highest value obtained by composing the affinity of the name n_{p_i} and the value v_{p_i} of p_i with the name n_{p_j} and the value v_{p_j} of each property $p_j \in P(c')$, respectively. For property values comparison, we exploit the datatype compatibility function (2) in the case of datatype properties, and the terminological affinity function (1) in the case of object properties, where the property values are constituted by reference names of concept or individuals, that is:

$$v(p_i) = \begin{cases} \max\{\mathcal{A}(n_{p_i}, n_{p_j}) \cdot \mathcal{T}(v_{p_i}, v_{p_j})\}, \forall p_j \in P(c') \text{ iff } v_{p_i} \text{ is a datatype} \\ \max\{\mathcal{A}(n_{p_i}, n_{p_j}) \cdot \mathcal{A}(v_{p_i}, v_{p_j})\}, \forall p_j \in P(c') \text{ iff } v_{p_i} \text{ is a reference name} \end{cases} \tag{9}$$

$SA_{c,c'}$ is evaluated by the intensive matching as the weighted sum of the linguistic affinity of c and c', calculated using (1), and of their contextual affinity, calculated as the average of the matching values for the elements of the context of c using (7) and for the property values calculated using (9), that is:

$$SA_{c,c'} = W_{la} \cdot \mathcal{A}(n_c, n_{c'}) + (1 - W_{la}) \cdot \frac{\sum_{i=1}^{|Ctx(c)|} m(e_i) + \sum_{j=1}^{|P(c)|} v(p_j)}{|Ctx(c)| + |P(c)|} \tag{10}$$

3.5 Example of matching two OWL ontologies

As an example of the ontology matching problem, we consider two real OWL ontologies describing different domains. The first ontology (Ka) describes research projects, while the second ontology (Portal) describes the contents of a Web portal. These ontologies are heterogeneous in terms of language specification (i.e., OWL Lite and OWL Full, respectively) as well as in terms of contents, although both of them provide a subset on concepts describing publications. Small portions of Ka and Portal describing publications are shown in Figure 1(a) and Figure 1(b), respectively. We show the H-MODEL representation of the OWL el-

Fig. 1. Ka and Portal ontology portions

ements which are relevant for H-MATCH by means of a graphical representation. In particular in Figure 1, white ovals represent concepts, grey ovals represent properties, and boxes represent datatypes. Arrows are used for representing kind-of relations, double-stricken and stricken lines for representing strong and weak property respectively, and dashed lines for representing property values. The main features of Ka and portal are summarized in Table 4. We exploit H-MATCH

	Ka	Portal
Language	OWL Lite	OWL Full
# of concepts	251	291
# of properties	154	204
Average # of properties per concept	13	4
Average # of relations per concept	2	2

Table 4. Features of the Ka and Portal ontologies

for matching Ka against Portal in order to automatically discover the affinity between the concepts that describe publications, in spite of concept heterogeneity in the two ontology descriptions. Moreover, we use H-Match also for providing, for each concept of Ka, a measure of semantic affinity with the concepts of Portal. In Table 5, we show the best results obtained from matching two concepts of Ka (i.e., Special_Issue_Publication and Article_In_Book) against Portal using our four matching models, with $W_{la} = 0.6$.

Matching model	SA$_{\text{Special\_Issue\_Publication,Publication}}$	SA$_{\text{Article\_In\_Book,Book}}$
Surface	0.8	0.8
Shallow	0.6357	0.6435
Deep	0.6527	0.6435
Intensive	0.6175	0.6262

Table 5. Example of matching results with H-Match

In our first example, the term Special_Issue_Publication is a specification of the term Publication (i.e., a BT/NT relationship is defined between them in Th). The semantic affinity value calculated using the surface matching model is due to this terminological relationship. Using other matching models, this affinity value based only on linguistic features is revised, by exploiting also contextual features. In particular, the shallow and intensive matching results are lower than the deep one, due to the fact that in Ka Special_Issue_Publication has a large number of properties which are different from the properties of Publication in Portal. Using the deep matching model, the kind-of relation between Special_Issue_Publication and Journal, which is a Publication in Portal has an important role in determining the semantic affinity value. On the opposite, in our second example, Article_in_Book does not have any semantic relation in Ka. For this reason, we obtain the same semantic affinity value with the shallow and the deep model. In this case, the intensive model is more accurate, because property values are considered in the semantic affinity evaluation.

3.6 Considerations

The four different models are exploited for suiting ontology matching to cope with different application scenarios. The main features of the matching models and their corresponding suggested scenarios are summarized in Table 6.

The choice of the appropriate matching model depends on the level of detail of the ontology description as well as on the expected degree of precision of the results. The shallow model is useful when only concept names are to be considered. It requires few computational resources since neither concepts properties nor their relations are considered. This model is well suited, for example, to perform an initial ontology comparison to decide wether it is worth performing a deeper analysis. If the ontology is constituted mainly by concepts with a few

	Surface	Shallow/Deep	Intensive
OWL sub-language	Lite/DL/Full	Lite/DL/Full	Lite/DL/Full
Ontological description	Poorly structured ontologies with very simple resource description	Schematic ontologies with taxonomic resource description	Articulated ontologies with rich resource description
Suggested for	Linguistic-driven matching	Linguistic and context-driven matching	Linguistic, value, and context-driven matching
Advantages	Less computation	More accurate characterization of matching concepts	Finest characterization of matching concepts

Table 6. Applicability of the matching models

number of properties and hierarchical relation among concepts, the shallow and deep model allow a good degree of precision without requiring great amount of computational resources. In presence of an articulated ontology, with rich resource descriptions and where relations among concepts are described through property values, the intensive model guarantees the most accurate results, but is the most expensive in term of computational resources.

4 Applicability to semantic information interoperability in open networked systems

In this section, we discuss the main applicability issues of our ontology matching techniques for semantic information interoperability in open networked systems based on the Peer-to-Peer (P2P) paradigm.

4.1 Resource discovery

In schema-based P2P networks, where resources are described by means of an ontology rather than being identified by file names, H-Match can be exploited to find out resources semantically related to a target request. In a Grid environment, where different nodes expose different kind of services, H-Match can be used to overcome the lack of agreement on the way resources are described, and to help a node in finding nodes which have compatible resource descriptions. In general, a resource request is represented as an H-Model description of the target of interest. When a node receives the request, it compares it against its ontology, by applying the H-Match algorithm. The matching evaluation depends on the expressiveness of the ontological description contained in the request and on the desired level of accuracy (i.e., surface, shallow, deep, or intensive matching). H-Match provides a ranking of the concepts provided by the answering node ordered by the semantic affinity between the target request

description and the concepts in the ranking. A threshold-based mechanism is used for filtering the concepts in the ranking to be returned to the requesting node. A deeper description of ontology-based content retrieval in P2P systems is described in [14].

4.2 Query routing

In P2P networks, query routing policy is one of the main factors that affect the load of the entire network and the effectiveness of the query answers. A semantic routing algorithm can be an alternative to the broadcasting of queries and can reduce the need of a more or less centralized index of the resources present in the network. In fact, H-MATCH results can be used to find the most relevant peers with respect to a given resource. A peer can exploit these associations among resources and peers to find the best recipients for its subsequent queries and to forward appropriately incoming queries to other peers. This semantic routing reduces both the network and the peers load, thus providing a greater scalability than routing techniques based on the topology of the network. We are working on the development of a semantic routing protocol in the framework of FIRB WEB-MINDS project, and some initial results are described in [15].

4.3 Semantic communities

H-MATCH can be adopted to discover semantic mappings among nodes storing similar concepts in a networked system. The ability to define such semantic mappings among independent nodes, allows to introduce the idea of *semantic communities* of nodes on predefined topic of interests described by means of one or more key-concepts. Given a topic of interest, each node applies H-MATCH to its ontology searching for relevant concepts and semantic mappings to nodes storing concepts semantically related to the topic (*semantic neighbors*). Exploiting semantic neighbors ontologies, further mappings can be discovered. Following the chain of mappings, a set of nodes constituting the semantic community on the considered topic of interest is identified. Because of the absence of a centralized organization which provides a complete list of semantic mappings between each pair of nodes, the community definition is a spontaneous process. As a consequence more than one semantic community on the same topic of interest can coexist, especially in large networks.

5 Related work

Work related to ontology matching for multi-ontology networked systems can be grouped into two main families of ontology matching approaches, namely model-based and logic-based approaches. The model-based approach is based on the idea of exploiting the ontology metadata model for working on the ontology structure through a set of techniques of analysis, matching and learning. As an example, the Glue [7] approach exploits machine learning techniques to find

semantic mappings between concepts stored in distinct and autonomous ontologies. Given two distinct ontologies, the mapping discovery process between their concepts is based on the measure of similarity which is defined through the joint probability distribution. The measure of similarity between two concepts is computed as the likelihood that an instance belongs to both the concepts. Another approach for model-based metadata matching in described in [16], where the choice of metadata for classifying data sources according to the requirements of a given application or task is discussed. In this approach, metadata information is organized as a set of categories and concepts, and the matching is enforced through fuzzy metrics.

The logic-based approach is based on the idea of exploiting the semantics associated with ontological descriptions for defining and analyzing mappings through automatic reasoning techniques. In particular, mapping discovering is reduced to the problem of checking a set of logical relations. For instance, in [6] the Ctx-Match algorithm is defined in order to point out semantic mapping between concepts stored in distinct peers of a Peer-to-Peer system. This algorithm compares the knowledge contained in different contexts looking for semantic mappings denoting peers interested in similar concepts. Ctx-Match is based on a semantic explication phase where concepts are associated with the correct meaning with respect to their context, and on a semantic comparison phase where concepts are translated in logical axioms and matched. As another example, in [17] the meaning of mappings is formally defined. The semantics provides a basis for reasoning about mappings (e.g., determining whether two mappings are equivalent or if a certain mapping formula is entailed by a mapping), combining evidence to propose likely mappings, and learning mappings. In particular, the reasoning is used for determining whether two mappings are equivalent, and whether a mapping is minimal (i.e., removing any formula from the mapping loses information).

With respect to these approaches, a main advantage of H-MATCH is the capability of dealing with different levels of accuracy in ontological descriptions by considering both the linguistic and the contextual features of ontology concepts. H-MATCH is suitable for dynamic scenarios like Peer-to-Peer and Grid, where ontologies evolves quickly and are characterized by different levels of accuracy in resource description.

6   Concluding remarks

In this paper, we have presented the H-MATCH approach to the problem of ontology matching in multi-ontology contexts such as in open networked systems. H-MATCH has been implemented in the framework of the HELIOS project for supporting knowledge sharing and ontology-addressable content retrieval in peer-based systems [8]. Our future work will be devoted to the intensive experimentation on H-MATCH on real ontology matching cases in order to address the following issues: i) extensively testing the approach actually adopted for compound term management in the thesaurus; ii) refining the matching algorithm,

by adopting an incremental refresh of matching results for operating in dynamic scenarios like Peer-to-Peer; iii) optimizing the matching process, by further reducing the number of matches performed in the deep and intensive model.

References

1. Broekstra, J., 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)
2. Nejdl, W., 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)
3. Halevy, A., Ives, Z., Suciu, D., Tatarinov, I.: Schema Mediation in Peer Data Management Systems. In: Proc. of the 19th Int. Conference on Data Engineering (ICDE 2003), Bangalore, India, IEEE Computer Society (2003)
4. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001)
5. Motik, B., Maedche, A., Volz, R.: A Conceptual Modeling Approach for Semantics-Driven Enterprise Applications. In Springer, ed.: Proc. of Confederated Int. Conferences DOA, CoopIS and ODBASE 2002, Irvine, California, USA (2002) 1082–1099
6. Bouquet, P., Magnini, B., Serafini, L., Zanobini, S.: A SAT-based Algorithm for Context Matching. In: Proc. of the 4th Int. and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 2003), Stanford, CA, USA, Springer Verlag (2003) 66–79
7. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to Map between Ontologies on the Semantic Web. In: Proc. of the 11th Int. World Wide Web Conference (WWW 2002), Honolulu, Hawaii, USA (2002) 662–673
8. Castano, S., Ferrara, A., Montanelli, S., Zucchelli, D.: HELIOS: a General Framework for Ontology-based Knowledge Sharing and Evolution in P2P Systems. In: Proc. of the 2nd DEXA Int. Workshop on Web Semantics (WEBS 2003), Prague, Czech Republic, IEEE Computer Society (2003)
9. Smith, M.K., Welty, C., McGuinness, D.L., (Eds.): OWL Web Ontology Language Guide (2004)
10. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM (CACM) **38(11)** (1995) 39–41
11. Lauer, M.: Designing Statistical Language Learners: Experiments on Noun Compounds. PhD thesis, Macquaire University, Australia (1995)
12. Ouksel, A.M., Naiman, C.F.: Coordinating Context Building in Heterogeneous Information Systems. Journal of Intelligent Information Systems **3** (1994) 151–183
13. Castano, S., De Antonellis, V., De Capitani Di Vimercati, S.: Global Viewing of Heterogeneous Data Sources. IEEE Transactions on Knowledge and Data Engineering **13** (2001) 277–297
14. Castano, S., Ferrara, A., Montanelli, S., Racca, G.: Matching Techniques for Resource Discovery in Distributed Systems Using Heterogeneous Ontology Descriptions. In: Proc. of the Int. Conference on Coding and Computing (ITCC 2004), Las Vegas, Nevada, USA, IEEE Computer Society (2004)
15. Castano, S., Ferrara, A., Montanelli, S., Pagani, E., Rossi, G.P., Tebaldi, S.: 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)

16. Madhavan, J., A. Bernstein, P., Rahm, E.: Generic Schema Matching with Cupid. In: Proc. of the 27th Int. Conference on Very Large Data Bases (VLDB 2001), Rome, Italy (2001) 49–58

17. Jayant Madhavan, Philip A. Bernstein, P.D., Y. Halevy, A.: Representing and Reasoning about Mappings between Domain Models. In: Proc. of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence, Edmonton, Alberta, Canada, AAAI Press (2002) 80–86