Semantic Driven Service Discovery for Interoperability in Web Information Systems [⋆]

Devis Bianchini, Valeria De Antonellis, Michele Melchiori and Denise Salvi

Università di Brescia
Dip. Elettronica per l'Automazione
Via Branze, 38
25123 Brescia - Italy
{bianchin|deantone|melchior|salvi}@ing.unibs.it

Abstract. Semantic integration and interoperability in Web Information Systems is a major crucial issue of the recent years. Semantics is particularly important to share and integrate information and services in open P2P environments, where the lack of a common understanding of the world generates the need for explicit guidance in discovering available resources. Nowadays, ontologies supply a common basis for various research areas, wherever semantics is involved. Their use as means to share descriptions of available resources is being investigated for content discovery also in P2P systems and, in particular, ontology-based techniques are at the core of many proposals related to service discovery. In this paper we propose a semantic-driven service discovery approach in a P2P scenario, where peers are organized in semantic communities for integration and interoperability purposes. Specific ontologies are defined to add semantics to service descriptions and to guide service discovery among peers.

1 Introduction

Semantic integration and interoperability in Web Information Systems (WIS) is a major crucial issue of the recent years. Distributed provisioning and invocation of WIS functionalities is addressed through the use of services, whose integration must be obtained by solving semantic heterogeneity of their functional interface. Moreover, each networked enterprise provides different functionalities/services, among which the required one should be semi-automatically detected by solving semantic heterogeneity in their functional interfaces. Semantics is particularly important to share and integrate information and services in open P2P environments, where the lack of a common understanding of the world generates the need for explicit guidance in discovering available resources. Nowadays, ontologies supply a common basis for various research areas, wherever semantics is involved. Their use as means to share descriptions of available resources is

being investigated for content discovery also in P2P systems and, in particular, ontology-based techniques are at the core of many proposals related to service discovery. In fact, ontology-based approaches constitute a step towards semantic service discovery, offering the benefits of formal specifications and inferencing capabilities. In open P2P systems, difficulties mainly arise due to the highly dynamic nature of peer interoperability, the lack of any agreed-upon global ontology, as well as the necessity of distributing the computation among peers when processing queries and searching services. Hence, effective service discovery methods and techniques under highly dynamic and context-dependent requirements are primary needs for a unified framework supporting semantic service discovery in a flexible fashion, exploiting service semantic description and flexible ontology-based matchmaking.

In this paper we propose the Semantic Driven Service Discovery approach in an open P2P networked scenario, P2P-SDSD, where peers are organized in a semantic community for integration and interoperability purposes. Ontologies over the Web are introduced to express domain knowledge related to service descriptions and to guide service discovery among peers. For community constitution, we assume that a peer called *promoter* spreads out a *manifesto* containing a suitable portion of its peer ontology, expressing a core domain knowledge for possible member aggregation. Each peer that aims at joining the community matches its own peer ontology against the manifesto and replies to the promoter. Otherwise, if a peer is not interested, it forwards the manifesto to the other peers of P2P network. Once the semantic community is established, services can be searched and exchanged between the members of the community by means of ontology-based techniques.

The paper is organized as follows: Section 2 gives an overview of the proposed architecture for the peer community, underlining the role of semantics; Section 3 explains the startup procedure of the community through the manifesto sharing; Section 4 shows how to interoperate in the community to perform service discovery and publishing; Section 5 compares the proposed approach with related work; finally, in Section 6 some final considerations and future work are discussed.

2 Network architecture

The semantic community is organized like a P2P network and it is constituted by n peers, each of them exporting its own WIS functionalities by means of services. Each peer can play different roles: (i) to search for a given service (*requester*); (ii) to propose a set of suitable services when a service request is given, through the application of matchmaking techniques (*broker*); (iii) to provide a selected service for its invocation and to publish a new service (*provider*). In an evolving collaborative P2P community, a peer can contain the description of a required service, while a different peer acts as a provider for that service, or a peer can be both a requester and a broker. According to this general view, each peer presents the architecture shown in Figure 1. In the following we focus on the Semantic Peer Registry and the Service MatchMaker, without details on the Application Program Interface, the Service Invoker and the P2P module handling inter-peer communications.
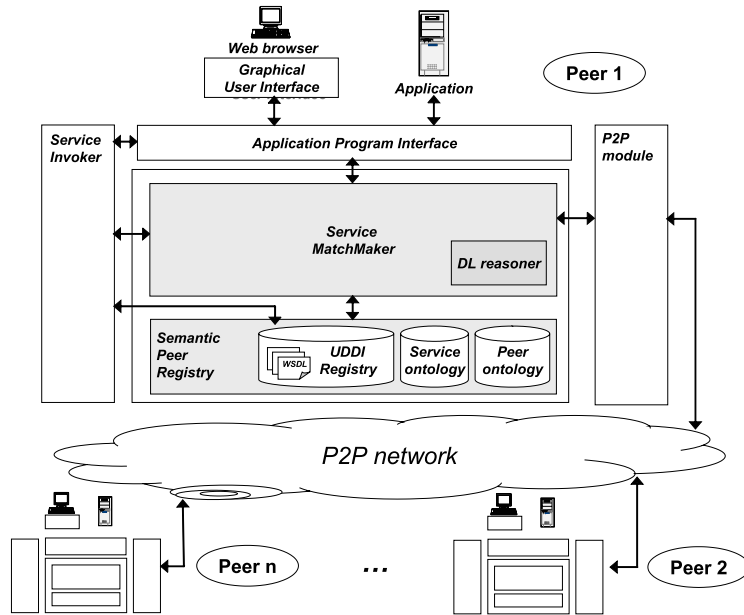
Fig. 1. Peer architecture.

2.1 Semantic Peer Registry

Service descriptions represent functional aspects of a service, based on the WSDL standard for service representation, in terms of service category, service functionalities (operations) and their corresponding input/output messages (parameters). Services are stored in an extended UDDI Registry, called *Semantic Peer Registry*, where besides the UDDI registry and WSDL descriptions, a peer ontology provides semantic knowledge related to service descriptions and a service ontology contains semantic service descriptions with reference to the peer ontology.

The peer ontology is constituted by:

- a *Service Functionality Ontology (SFO)*, that provides knowledge on the concepts used to express service functionalities (operations);
- a *Service Message Ontology (SMO)*, that provides knowledge on the concepts used to express input and output messages (parameters) of services.

Concepts in the peer ontology are organized according to `subclass-of` and `equivalent-to` semantic relationships. Furthermore, the peer ontology is extended by a *thesaurus* providing terms and terminological relationships (as synonymy, hypernymy and so on) with reference to names of concepts in the ontology. In this way, it is possible to extend matchmaking capabilities when looking

for correspondences between elements in service descriptions and concepts in the ontology.

In the service ontology, services are semantically represented by DL logic expressions [4], whose elements (service category, operation names, input/output parameter names) are properly mapped to the peer ontology. Both peer ontology and service ontology are expressed in OWL-DL.

2.2 Service MatchMaker

The Service MatchMaker is in charge of comparing service descriptions, combining together different matchmaking models [5]: (i) a deductive model, exploiting deduction algorithms for reasoning on service descriptions, (ii) a similarity-based model, where retrieval metrics are applied to measure the degree of match between services. In this paper we will combine these matchmaking strategies in the context of P2P semantic communities to improve service discovery.

Both the matchmaking models are based on jointed use of the peer ontology and the thesaurus to classify the type of match between services (exact, partial and mismatch) and to quantify it by means of suitable similarity coefficients, that is, by evaluating the service similarity. Terminological relationships considered in the thesaurus are SYN for synonymy, BT (resp., NT) for broader term (resp., narrower term) and RT for related term. The thesaurus is exploited to compute the *Name Affinity* coefficient between names of input/output parameters and operations.

Definition 1 (Name Affinity coefficient). *Given the thesaurus $\mathcal{TH}$, the Name Affinity coefficient between two terms $t, t' \in \mathcal{TH}$, denoted by $NA(t, t')$, is: (i) 1.0 if $t = t'$; (ii) $max_l(\tau(t \rightarrow^l t'))$ if $t \neq t' \wedge t \rightarrow^l t', l \geq 1$, where $t \rightarrow^l t'$ denotes a path of terminological relationships from t to t'; (iii) 0.0 otherwise. A weight $\sigma_{tr} \in [0, 1]$ is associated to each kind of terminological relationship tr, in order to evaluate its implication for name affinity; in our experimentation, $\sigma_{SYN} = 1$, $\sigma_{BT/NT} = 0.8$ and $\sigma_{RT} = 0.5$. The function $\tau(t \rightarrow^l t') = \prod_{k=1}^{l}(\sigma_{tr_k}) \in [0, 1]$ defines the strength of $t \rightarrow^l t'$ as the product of the weights of all terminological relationships in the path. Since between two terms in the thesaurus there can exist more than one path, the one with the highest strength is chosen.*

We say that t and t' have *name affinity* ($t \sim t'$) if and only if $NA(t, t') \geq \alpha$, where $\alpha > 0$ is a threshold given by experimental results to select only terms with high values of the *Name Affinity* coefficient. The choice of the actual value of α is done during a training phase where α is set initially to a given value (i.e., 0.5) then this value is increased or decreased until a satisfactory trade-off between recall and precision is obtained. That is, increasing α leads to be more selective by identifying a name affinity between two terms only if they are very similar according to the thesaurus. Viceversa, by decreasing α, name affinities are established also between pairs of terms that are related by a weaker path of terminological relationships. Name Affinity coefficient is used to extend

traditional Description Logic subsumption test (denoted by $\sqsubseteq$) between two generic terms, even if they do not belong to the peer ontology.

Definition 2 (Affinity-based subsumption test). *Given an atomic concept C in the peer ontology $\mathcal{PO}$, we define the set of terms in the thesaurus that have name affinity with the concept C as $C_{\mathcal{TH}} = \{T{\in}\mathcal{TH} \mid T{\sim}C\}$. Analogously, we define the set of concepts of $\mathcal{PO}$ that have name affinity with a term T in $\mathcal{TH}$ as $T_{\mathcal{PO}} = \{C{\in}\mathcal{PO} \mid T \in C_{\mathcal{TH}}\}$.*
Given the peer ontology $\mathcal{PO}$, the thesaurus $\mathcal{TH}$ and a pair of terms T^1 and T^2 used in service descriptions to denote service elements, T^1 is subsumed by T^2 with respect to $\mathcal{TH}$, denoted by $T^1 \sqsubseteq_{\mathcal{TH}} T^2$, if and only if there exists $C \in T^1{}_{\mathcal{PO}}$ and $D \in T^2{}_{\mathcal{PO}}$ such that $C \sqsubseteq D$ is satisfied in $\mathcal{PO}$.
Note that we pose $T^1 \equiv_{\mathcal{TH}} T^2$ if both $T^1 \sqsubseteq_{\mathcal{TH}} T^2$ and $T^2 \sqsubseteq_{\mathcal{TH}} T^1$ hold.

The deductive matchmaking model applies the affinity-based subsumption test to service description elements considered separately (categories, operations, I/O parameters) to classify the match between a service request $\mathcal{R}$ and each supplied service $\mathcal{S}$. In [5] a formal definition of the following kinds of matches is given:

Exact match, to denote that $\mathcal{S}$ and $\mathcal{R}$ have the same capabilities, that is, for each operation in $\mathcal{R}$ there exists an operation in $\mathcal{S}$ that has: (i) equivalent name; (ii) equivalent output parameters; (iii) equivalent input parameters.
Plug-in match, to denote that $\mathcal{S}$ offers at least the same capabilities of $\mathcal{R}$, that is, for each operation in $\mathcal{R}$ there exists an operation in $\mathcal{S}$ that has: (i) an equivalent or more specific operation name; (ii) an equivalent or more specific output parameter for each output parameter of the required operation; (iii) a set of input parameters, each of them is equivalent or more generic than an input parameter of the required operation; the inverse kind of match is denoted as `subsume`; the rationale behind the `plug-in` and `exact` matches is that $\mathcal{S}$ totally fulfills the request $\mathcal{R}$ if it provides all the required outputs, but, on the other hand, $\mathcal{R}$ must be able to provide all the inputs needed for the execution of $\mathcal{S}$.
Intersection match, to denote that $\mathcal{S}$ and $\mathcal{R}$ have some common capabilities, that is, there exist an operation in $\mathcal{S}$ and an operation in $\mathcal{R}$ such that: (i) their names are related in any generalization hierarchy; (ii) there exists a pair of I/O parameters, one from $\mathcal{R}$ and one from $\mathcal{S}$, that are related in any generalization hierarchy.
Mismatch, otherwise.

Service categories are initially exploited to filter out not suitable services: only supplied services whose categories are equivalent or more specific than the request category are selected. We consider a qualitative ranking among the kinds of matches, that is, `exact` > `plug-in` > `subsume` > `intersection` > `mismatch`.

Similarity analysis is applied to quantify the match between services and it is based on the Name Affinity coefficient. In particular, when `exact/plug-in` match occurs, similarity between services is set to 1 (full similarity); if `mismatch` occurs, the similarity value is set to zero; finally, when `subsume` and `intersection`

ENTITY-BASED SIMILARITY

$$ESim(\mathcal{R},\mathcal{S}) = \frac{2 \cdot A_{tot}(IN_{\mathcal{R}},IN_{\mathcal{S}})}{|IN_{\mathcal{R}}|+|IN_{\mathcal{S}}|} + \frac{2 \cdot A_{tot}(OUT_{\mathcal{R}},OUT_{\mathcal{S}})}{|OUT_{\mathcal{R}}|+|OUT_{\mathcal{S}}|} \in [0,2]$$

$IN_{\mathcal{R}}, IN_{\mathcal{S}}$ - sets of input parameter names of $\mathcal{R}$ and $\mathcal{S}$
$OUT_{\mathcal{R}}, OUT_{\mathcal{S}}$ - sets of output parameter names of $\mathcal{R}$ and $\mathcal{S}$
$A_{tot}(IN_{\mathcal{R}}, IN_{\mathcal{S}}) = \sum_{in_{\mathcal{R}}^i \in IN_{\mathcal{R}}, in_{\mathcal{S}}^j \in IN_{\mathcal{S}}} NA(in_{\mathcal{R}}^i, in_{\mathcal{S}}^j)$
$A_{tot}(OUT_{\mathcal{R}}, OUT_{\mathcal{S}}) = \sum_{out_{\mathcal{R}}^i \in OUT_{\mathcal{R}}, out_{\mathcal{S}}^j \in OUT_{\mathcal{S}}} NA(out_{\mathcal{R}}^i, out_{\mathcal{S}}^j)$

OPERATION SIMILARITY

$$OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j) = NA(name\_op_{\mathcal{R}}^i, name\_op_{\mathcal{S}}^j) + \frac{2 \cdot A_{tot}(IN_{\mathcal{R}}^i, IN_{\mathcal{S}}^j)}{|IN_{\mathcal{R}}^i|+|IN_{\mathcal{S}}^j|} + \frac{2 \cdot A_{tot}(OUT_{\mathcal{R}}^i, OUT_{\mathcal{S}}^j)}{|OUT_{\mathcal{R}}^i|+|OUT_{\mathcal{S}}^j|} \in [0,3]$$

$IN_{\mathcal{R}}^i, IN_{\mathcal{S}}^j$ - sets of input parameter names of the i-th operation of $\mathcal{R}$ and the j-th operation of $\mathcal{S}$
$OUT_{\mathcal{R}}^i, OUT_{\mathcal{S}}^j$ - sets of output parameter names of the i-th operation of $\mathcal{R}$ and the j-th operation of $\mathcal{S}$

FUNCTIONALITY-BASED SIMILARITY

$$FSim(\mathcal{R},\mathcal{S}) = \frac{2 \cdot \sum_{i,j} OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j)}{|OP(\mathcal{R})|+|OP(\mathcal{S})|} \in [0,3]$$

$OP(\mathcal{R}), OP(\mathcal{S})$ - sets of operation names of $\mathcal{R}$ and $\mathcal{S}$

GLOBAL SIMILARITY

$$GSim(\mathcal{R},\mathcal{S}) = w_1 \cdot NormESim(\mathcal{R},\mathcal{S}) + w_2 \cdot NormFSim(\mathcal{R},\mathcal{S}) \in [0,1]$$

w_1, w_2 - weights introduced to assess the relevance of each kind of similarity ($w_1 \in [0,1]$ and $w_2 = 1-w_1$)
$NormESim(), NormFSim()$ - $ESim()$ and $FSim()$ normalized to the range $[0,1]$

Table 1. Similarity coefficients between service descriptions $\mathcal{R}$ (request) and $\mathcal{S}$ (supply).

match occur, similarity coefficients exposed in Table 1 are computed to evaluate similarity between $\mathcal{R}$ and $\mathcal{S}$.

Entity-based similarity coefficient $ESim$ evaluates the similarity of all the I/O parameters of the considered services to measure how much they are based on the same information; *Functionality-based similarity* coefficient $FSim$ compares pairs of operations together with their corresponding I/O parameters to measure how much the two services perform the same functionalities. Each component in $ESim()$ formula in Table 1 produces by construction a value belonging to the [0,1] range, so that $ESim() \in [0,2]$. Similarly, each component in $OpSim()$ formula produces by construction a value belonging to the [0,1] range, so that $OpSim() \in [0,3]$ and, accordingly, $FSim() \in [0,3]$. $ESim$ and $FSim$ are normalized into the [0,1] range and combined in the *Global similarity* coefficient $GSim$. A detailed description of similarity coefficients and their application is given in [6].

If the value of $GSim$ is equal or greater than a threshold δ, then $\mathcal{R}$ and $\mathcal{S}$ are considered *similar*. It is noteworthy to say that these coefficients are specifically oriented toward a comparison between service descriptions expressed in terms of operations and corresponding I/O parameters rather than pure vectors of terms. The result of this similarity evaluation depends on the choice of δ. Actual value of δ is experimentally set during a training phase. Since actual values of $ESim$ and $FSim$ depend on name affinity evaluation and therefore depend also on the α value, we first set this value to obtain a satisfactory name affinity evaluation then we vary the value of δ until an acceptable trade-off between precision and recall is obtained on a set of training services.

The semantic matchmaking techniques can also be applied to compare services stored on the same peer or on different peers, to identify semantic links. In particular, a semantic link between two services is established if the kind of match is not `mismatch` and the $GSim$ value is equal or greater than threshold δ. The semantic link is described by the kind of match and the $GSim$ coefficient and it is stored in the service ontology.

Two types of semantic links can be established: (i) semantic links between services belonging to the same peer (*intra-peer semantic links*); (ii) semantic links between services belonging to different peers (*inter-peer semantic links*); the related peers are referred as *semantic neighbors*. In the following, we briefly illustrate the process of semantic community constitution as defined in the ES-TEEM Project [7], then focusing on inter-peer semantic link definition.

3   P2P semantic community setup

Figure 2 shows the process of semantic community constitution, where the information owned by the peers at each step is listed on the right. We denote as *Global Overlay (GO)* the generic P2P network which peers potentially aiming at joining semantic community belong to. In the P2P network, each peer knows its own IP address. The promoter of the semantic community builds a *startup manifesto*, containing a portion of its peer ontology manually selected, apt to express the core interests of the intended semantic community. The startup manifesto generally will contain the upper level concepts of promoter's Service Message Ontology, since it contains knowledge about the domain in which the sharable services operate. It is important that startup manifesto is limited, so it can be easily flooded over the P2P network starting the setup process. When a peer in GO receives the startup manifesto, it can accept or not to join the community. In case of acceptance, the peer sends back a message to the promoter with its IP address and becomes a candidate member. Otherwise, the peer simply ignores the startup manifesto and forwards it to the other peers in GO, except the one from which the startup manifesto came.

Candidate members constitute a so-called *Semantic Overlay (SO)*, but the community is not established yet. They know their own IP address and the startup manifesto of the community. The promoter sends to all candidate members the complete version of *manifesto* together with the list of candidate members IP addresses. The manifesto contains both the Service Message Ontology
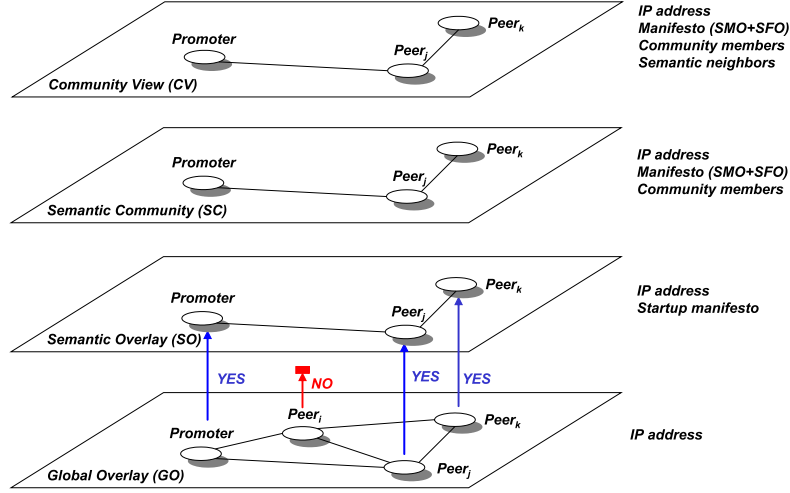
Fig. 2. Setup of the semantic community.

and the Service Functionality Ontology of the promoter and is sent only to candidate members to reduce network overload. At this point, the *Semantic Community (SC)* is established and candidate members become community members. Each peer knows its own IP address, the manifesto and the community member current list.

Once the semantic community is established, each peer searches for its semantic neighbors to establish the inter-peer semantic links. To do this, it sends a probe service request for each service it wants to make sharable; this probe service request contains the description of the service functional interface (categories, operations, I/O parameters). The probe service request is sent to all the other peers of the semantic community, according to the community member list. Each peer receiving the probe service request matches it against its own service descriptions by applying the matchmaking techniques explained in Section 2.2 and obtains for each comparison the kind of match mt and the similarity degree $GSim$. If mt is not `mismatch` and $GSim$ is equal or greater than threshold δ, they are enveloped in a message sent back to the peer from which the probe service request came. An *inter-peer semantic link* is established between the two peers, that become semantic neighbors with respect to the linked services.

In this way, each peer can build a map of its semantic neighbors, with similar services and semantic links with them. In this phase we say that peers belong

to the *Community View (CV)* and know their own IP address, the community manifesto, the community member current list and semantic neighbors.

Community can evolve when new peers join it or new services are published on community members. To collect the list of new peers that aim at joining the community, promoter sends again the startup manifesto on P2P network and receives the IP addresses of new candidate members, to which the promoter sends the community manifesto, while it sends an updated member list to all the peers in the community. A peer that receives a new member list can repeat the procedure to find semantic neighbors provided that the community changed. When a community member publishes a new service, it advertises the promoter, that triggers the semantic community; in this way, all members update their inter-peer semantic links by means of probe service request mechanism. Note that the semantic community and community view are established before service requests are propagated. As explained in the next section, when a peer receives a request, it applies matchmaking strategies to properly select a subset of its semantic neighbors that are more suitable to serve that particular request.

4   P2P semantic community interoperation

Once the semantic community is established and inter-peer semantic links are defined, services can be searched and exchanged between community members. A peer p of the community can receive a service request either directly from the Application Program Interface or from another community member. Given a service request $\mathcal{R}$, a peer p searches for suitable services in its own Semantic Peer Registry and retrieves a list $CS = \{\langle \mathcal{S}_1, GSim_1, mt_1 \rangle, \ldots \langle \mathcal{S}_n, GSim_n, mt_n \rangle\}$ of services with corresponding similarity values $GSim_i \geq \delta$ and match type mt_i different from `mismatch`.

If a service $\mathcal{S}_i \in CS$ presents an `exact` or a `plug-in` match with the request, then $\mathcal{S}_i$ satisfies completely the required functionalities and it is not necessary to forward the service request to semantic neighbors with respect to $\mathcal{S}_i$. Otherwise, if $\mathcal{S}_i$ presents a `subsume` or an `intersection` match with the request, the peer p forwards the request to those peers that are semantic neighbors with respect to $\mathcal{S}_i$. Peer p does not consider semantic neighbors that present a `subsume` or an `exact` match with $\mathcal{S}_i$, because this means that they provide services with the same functionalities or a subset of $\mathcal{S}_i$ functionalities and they cannot add further capabilities to those already provided by $\mathcal{S}_i$ on the peer p. This phase is repeated for every $\mathcal{S}_i \in CS$. Semantic neighbors which present inter-peer links with any service $\mathcal{S}_j$ stored on p, but not included in CS, are discarded since they are not relevant with respect to $\mathcal{R}$. Each selected semantic neighbor sn presents a set of k inter-peer semantic links with some services on p that are suitable for $\mathcal{R}$. It is described as $\langle sn, \{\langle \mathcal{S}_1, GSim_1, mt_1 \rangle, \ldots \langle \mathcal{S}_k, GSim_k, mt_k \rangle\}\rangle$, where $\mathcal{S}_1 \ldots \mathcal{S}_k \in CS$ and have a semantic link with some services stored on sn, featured by $GSim_1 \ldots GSim_k$ similarity degree and $mt_1 \ldots mt_k$ type of match, respectively. Note that this formulation holds even if sn has more than one service related to the same service $\mathcal{S}_i \in CS$.

Since the relevance of sn does not depend only on the similarity associated to the semantic links between p and sn, but also on the similarity degree between $\mathcal{S}_i \in CS$ and $\mathcal{R}$, the harmonic mean is used to combine these two aspects. Therefore, the relevance of a semantic neighbor sn is defined as:

$$r_{sn} = \frac{1}{k} \sum_{i=1}^{m_{sn}} \frac{2 * GSim_i * GSim(\mathcal{R}, \mathcal{S}_i)}{GSim_i + GSim(\mathcal{R}, \mathcal{S}_i)} \qquad (1)$$

Relevance values are used to rank the set of semantic neighbors in order to filter out not relevant semantic neighbors (according to a threshold-based mechanism) and to further constrain the request forwarding (according to a token-based strategy).

Example. Let consider $\texttt{Peer}_A$ providing three services $\texttt{findDisease}$, $\texttt{findDiagnosis}$ and $\texttt{findLaboratory}$ for which the following semantic neighbors have been found (Figure 3):

$\langle\texttt{Peer}_B, \{\langle\texttt{findDisease}, 0.753, \texttt{intersection}\rangle, \langle\texttt{findDiagnosis}, 1.0, \texttt{exact}\rangle\}\rangle$

$\langle\texttt{Peer}_C, \{\langle\texttt{findLaboratory}, 0.7, \texttt{intersection}\rangle, \langle\texttt{findDisease}, 1.0, \texttt{plug} - \texttt{in}\rangle\}\rangle$
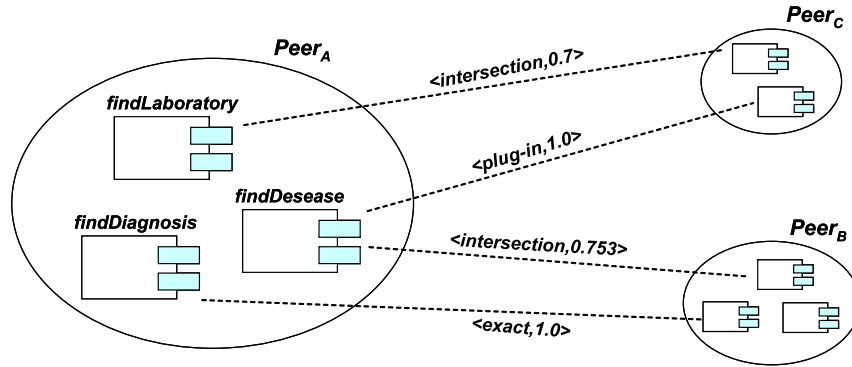

Fig. 3. An example of inter-peer semantic links in the P2P community.

Let suppose that for a given request $\mathcal{R}$ on the $\texttt{Peer}_A$, we obtain the following list of matching services: $\texttt{CS} = \{\langle\texttt{findDisease}, 0.9, \texttt{intersection}\rangle, \langle\texttt{findDiagnosis}, 0.7, \texttt{subsume}\rangle\}$, while $\{\langle\texttt{findLaboratory}, 0.0, \texttt{mismatch}\rangle\}$ is excluded from $\texttt{CS}$. For what concerns $\texttt{findDisease}$, both $\texttt{Peer}_B$ and $\texttt{Peer}_C$ must be considered as semantic neighbors, since they could provide some additional capabilities with respect to $\texttt{Peer}_A$. Moreover, for what concerns $\texttt{findDiagnosis}$, $\texttt{Peer}_C$ is not a semantic neighbor, while $\texttt{Peer}_B$ has a related service, that presents an $\texttt{exact}$ match with $\texttt{findDiagnosis}$. This means that $\texttt{Peer}_B$ has no additional capabilities to offer with respect to those already provided by $\texttt{findDiagnosis}$ in $\texttt{Peer}_A$. The resulting set of selected semantic neighbors with respect to $\mathcal{R}$ is $\{\langle \texttt{Peer}_C, \{\texttt{findDisease}, 1.0, \texttt{plug-in}\} \rangle, \langle \texttt{Peer}_B, \{\texttt{findDisease}, 0.753, \texttt{intersection}\} \rangle\}$. The relevance values for $\texttt{Peer}_C$ and $\texttt{Peer}_B$ with respect to the request $\mathcal{R}$ are then:

$$r_{\texttt{Peer}_\texttt{C}} = \frac{2 * 1.0 * 0.9}{1.0 + 0.9} = 0.947; \quad r_{\texttt{Peer}_\texttt{B}} = \frac{2 * 0.753 * 0.9}{0.753 + 0.9} = 0.82 \qquad (2)$$

5 Related work

Service semantic description in a P2P context and semantic links between peers based on provided services is a crucial aspect to improve effectiveness and performance of P2P service discovery. Several proposals have been made in literature to enable semantic-enhanced service discovery in a P2P context, not necessarily based on semantic communities. In METEOR-S [10] service descriptions are kept in UDDI Registries semantically enhanced with local *domain specific ontologies*, while a centralized *registries ontology* is used to classify peer registries. During the discovery process, *registries ontology* is browsed to find the proper registry to which submit the request. GloServ [1] defines a predefined *skeleton ontology*, represented as a taxonomy of concepts each of them representing a service category, that in turn is associated to a high level server. Each server represents a P2P network of nodes organized via a *Content Addressable Network (CAN)* [9]. These peers provide services belonging to the category represented by ontological concept associated to their own server. The *skeleton ontology* is a centralized structure, replicated and cached in each high level server. Service discovery is organized in two phases: (a) browsing concept taxonomy to find the proper high level server; (b) keyword-based search through a CAN lookup table. Our approach does not constrain peers to use a common ontology: each peer exploits its own peer ontology, eventually mapped to the community manifesto to enhance peer interoperability.

DAML-S for P2P [8] considers a P2P network, where each member stores a local DAML-S ontology to describe services; service semantic descriptions are kept in local UDDI registries extended with DAML-S ontologies. However, no semantic links are found between peers that provide similar services and when a peer does not satisfy a request, a flooding mechanism is used to find suitable services on the other peers of the network. ARTEMIS [2] defines a P2P network, where each peer has a coarse-grained *Service Functionality Ontology (SFO)* to classify services and a fine-grained *Service Message Ontology (SMO)* to annotate services with medical concepts, based on medical information standards. Peer store in a reference mediator super-peer the services they provide. A peer sends a request to its reference mediator expressed in terms of its own ontologies; mediator uses ontology mappings to find matching services in its local registries and also forwards the request to other mediators. No semantic links are exploited to prune the set of peers to which forward the request.

WSPDS [3] describes a P2P network where peers have local DAML-S ontologies to provide service semantics and semantic links with other peers based on similarity between services they provide. When a request is submitted to a peer, it searches for local matching results and forwards the request to all the semantic neighbors, independently of the current request or the local results of the query. Original contribution of our approach is related to the flexibility of the

matchmaking process based on both deductive and similarity-based approaches and on the definition and exploitation of inter-peer semantic links.

6 Concluding remarks

In this paper, we proposed a semantic-based approach for service discovery in a P2P scenario, where peers are organized in communities. Specific ontologies (called *peer ontologies*) are used to add semantics to service descriptions and are exploited during community setup to find inter-peer semantic links. These links are used to propagate a service request between the peers of the community in an efficient way. Implementation and experimentation of the proposed approach are being performed in the application scenario of scientific collaboration in medicine [7].

References

1. K. Arabshian and H. Schulzrinne. An Ontology-based Hierarchical Peer-to-Peer Global Service Discovery System. *Journal of Ubiquitous Computing and Intelligence (JUCI)*, 2006.
2. The ARTEMIS Project: A Semantic Web Service-based P2P Infrastructure for the Interoperability of Medical Information Systems. `http://www.srdc.metu.edu.tr/webpage/projects/artemis/`.
3. F. Banaei-Kashani, C. Chen, and C. Shahabi. WSPDS: Web Services Peer-to-Peer Discovery Service. In *Proc. of the Int. Conference on Internet Computing (IC'04)*, pages 733–743, Las Vegas, Nevada, USA, 2004.
4. D. Bianchini and V. De Antonellis. Ontology-based Semantic Interoperability Tools for Service Dynamic Discovery. In *IFIP Int. Conf. on Interoperability of Enterprise Software Applications, INTEROP-ESA'05*, Geneva, Switzerland, 2005.
5. D. Bianchini, V. De Antonellis, M. Melchiori, and D. Salvi. Semantic-enriched Service Discovery. In *IEEE ICDE Int. Workshop on Challenges in Web Information Retrieval and Integration, WIRI 2006*, Atlanta, Georgia, USA, 2006.
6. D. Bianchini, V. De Antonellis, B. Pernici, and P. Plebani. Ontology-based Methodology for *e*-Service discovery. *Journal of Information Systems, Special Issue on Semantic Web and Web Services*, 31(4-5):361–380, June-July 2006.
7. ESTEEM Project Home Page: Emergent Semantics and cooperaTion in multi-knowledge EnvironmEnt. `http://www.dis.uniroma1.it/∼esteem/index.html`.
8. M. Paolucci, K.P. Sycara, T. Nishimura, and N. Srinivasan. Using daml-s for p2p discovery. In *Proceedings of the Int. Conference on Web Services (ICWS2003)*, 2003.
9. S. Ratnasamy, P. Francis, M. Handley, R.M. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proc. of the ACM SIGCOMM'01 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication*, pages 161–172, San Diego, CA, USA, August 2001.
10. K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, and J. Miller. METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services. *Journal of Information Technology and Management, Special Issue on Universal Global Integration*, 6(1):17–39, 2005.