Service-oriented Semantic Collaboration in Distributed Information Systems *

Devis Bianchini,1 Valeria De Antonellis,1 Michele Melchiori1 and Denise Salvi1

Abstract. Collaborative information systems in a P2P scenario are characterized by a set of independent peers that dynamically need to cooperate by sharing data and services. For effective collaboration, in absence of a global view of the resources shared across the information systems, semantic interoperability tools are required. In the paper we focus on interoperability issues for service discovery and sharing. We propose an architectural framework for the construction of a service semantic overlay in which peers storing similar services are considered semantic neighbors, that is potential collaboration partners, and are related by semantic links. In particular, semantic neighbors can be exploited to enforce a semantic service request forwarding protocol and to provide a scalable infrastructure for peer communications. The proposed semantic forwarding protocol and policies are described and preliminary experimentation results are discussed.

1 Introduction

Collaborative information systems in a P2P scenario are characterized by a set of independent peers that dynamically need to cooperate by sharing data and services. For effective collaboration, in absence of a global view of the resources shared across the information systems, semantic interoperability tools are required. In literature, P2P semantic-driven resource discovery has attracted much attention and relies on several efforts in related research fields, such as data integration and emergent semantics in P2P environments [1, 10]. Semantic heterogeneity and scalability issues must be faced to go beyond limitations of centralized service-oriented architectures. Structured solutions [2, 12] try to map services that are semantically close to nearby positions, efficiently confining the search space. However, structured approaches are mainly based on DHT structures, that are less flexible and require much efforts for maintenance.

Our approach focuses on the service discovery problem in unstructured networks. The same problem has been studied in [3–5, 14]. Unstructured peer-based systems for service discovery often relies on super-peers for service request routing [3] or on centralized structures to classify peer registries [14] or peer functionalities exported as Web services [5]. Few efforts organize the network through links between peers based on average similarity of services they store [4]. WSPDS [4] describes a P2P network where peers have local DAML-S

* This work has been partially supported by the ESTEEM PRIN Project (http://www.dis.uniroma1.it/~esteem/index.html) and TEKNE FIRB Project (http://www.tekne-project.it/), funded by the Italian Ministry of Education, University and Research.
ontologies to provide service semantics and links with other peers based on an average 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 linked peers, independently from the current request or the local results of the query. In our approach a service semantic overlay is built by relating similar services through inter-peer semantic links in a network of collaborative peers. Like WSPDS, no common ontology is required and no super-peer is defined. However, in our approach, combined use of intra-peer and inter-peer semantic links allows for application of fine-grained request forwarding strategies.

In the paper we focus on interoperability issues for service discovery and sharing. We propose a P2P-based Semantic Driven Service Discovery (P2P-SDSD) architectural framework for the construction of a service semantic overlay over a P2P collaborative network. In the semantic overlay peers storing similar services are considered semantic neighbors, that is potential collaboration partners, and are related by semantic links. In particular, semantic neighbors can be exploited to enforce a semantic service request forwarding protocol and to provide a scalable infrastructure for peer communications. The proposed semantic forwarding protocol and policies are described and preliminary experimentation results are discussed.

Innovative aspects of this work are: the development of the P2P-SDSD architectural framework implementing techniques for semantic link definition and organization in the Service Ontology; the support given by this architecture to the process of effective and efficient service discovery, according to optimization policies, in the internetworked collaborative scenario. The paper is organized as follows: Section 2 discusses the problem statement in a collaborative scenario. Section 3 presents the peer reference architecture. In Section 4 experimental results of the service request propagation are shown. Finally, Section 5 presents concluding remarks and future work.

2 Problem statement

The P2P-SDSD framework described in this paper has been mainly developed in the ESTEEM (Emergent Semantics and cooperaTion in multi-knowledge EnvironMents) approach [8, 13], where a comprehensive platform for data and service discovery in P2P systems is proposed, with advanced solutions for trust and quality-based data management, P2P infrastructure definition, query processing and dynamic service discovery in a context-aware scenario. This scenario is characterized by the high dynamic nature of the peer interoperability, the lacking of a reference ontology to which the peers refer to and the need to distribute the computation among peers to process data and service requests. In particular, in the ESTEEM context we have considered the healthcare domain where organizations (laboratories, drug-stores, research centres, hospitals) provide different services (i.e., drug ordering, diagnosis services). In this scenario the peers can share services for collaboration purposes and effective semantic-based service discovery is required. For instance, a medical organization can expose and share with other organizations and doctors its specific medical skill and data by implementing and publishing specialized diagnostic services.
Services are advertised on peers through an abstract description (abstract service), in terms of service functionalities (operations) and input/output messages (parameters), based on the WSDL standard. For each abstract service, a set of concrete implementations is provided on the network, described on registering peer with the URL of the service implementations and binding details. Two distinct peers could register the same abstract service, for which different implementations could be provided, or they could advertise different abstract services that partially overlap. When a peer is looking for a service, it formulates the service request by specifying the expected functional interface and sends the request to one of the peers of the network. According to this vision and to the service-oriented paradigm, each peer can play three different roles: (i) it can store abstract services and references to corresponding concrete implementations (broker); (ii) it can publish on a broker the implementation of a service, described by its functional interface (provider); (iii) it can look for a service (requester). When a peer joins the network, it can act both as a broker/provider or a requester.

In particular, when a peer \( P_X \) joins the collaborative network, it obtains the list of brokers' IP currently available in the network and starts to contact them. When a broker \( P_Y \) replies, \( P_X \) can be connected to it. Successively, if \( P_X \) is a provider, it publishes its services on broker \( P_Y \) and waits for a request to serve. If \( P_X \) is a requester, it sends a service request to the broker \( P_Y \) and waits for the answer. Finally, if \( P_X \) is a broker, it receives from \( P_Y \) a list of brokers known by \( P_Y \).

Brokers are in charge of establishing and maintaining semantic links towards other brokers on the basis of similarity between abstract services they store. Semantic links constitute a service semantic overlay - SSO over the network and linked peers are called Semantic Neighbors. Given the high dynamic considered scenario, the network can evolve continuously and therefore changes to the SSO are possible and have to be managed as explained in the Sections 3.4 and 3.5. In the rest of the paper, we will explain how the semantic overlay is built, maintained and exploited for service discovery purposes.

3 Broker reference architecture in P2P-SDSD

The P2P-SDSD architecture is defined to support cooperation in the collaborative network and to manage the construction and the evolution of the semantic overlay. As shown in Figure 1, the following main components are defined in the P2P-SDSD broker architecture:

- **Peer knowledge infrastructure.** A broker connects to the network by providing an ontology-based representation of the services to provide semantics for service descriptions.
- **Request processor component.** It is responsible for interacting with the user through the Graphical User Interface (GUI) and for satisfying her service discovery requests. It is also invoked to serve discovery requests sent to the broker by its semantic neighbors according to the perspective of collaborative distributed service discovery.
Fig. 1. Architecture of a P2P-SDSD broker.

- **SSO management component.** It is responsible for discovering semantic neighbors among the other peers of the collaborative community and for building and maintaining the service semantic overlay (Semantic link manager). With respect to typical P2P routing, this component provides a semantically enriched forwarding mechanism to effectively implement service requests propagation on the SSO (Semantic forwarding manager).

- **Service matchmaker component.** It is responsible for evaluating the degree and the kind of match when comparing service descriptions. It is invoked by the request processor when the broker needs to answer a request by matching it against the peer services. It is also invoked by the SSO management component to identify semantic neighbors and establishing inter-peer semantic links with brokers providing similar services. A hybrid matchmaking strategy [7] is provided in P2P-SDSD to compare services on the basis of their functional interface.

- **Logical overlay management component.** It is responsible for guaranteeing the connectivity and communication among peers and managing the evolution of the P2P network. The logical level manager is responsible for maintaining the logical overlay (Network evolution) and the associated communication (Membership & Communication). The P2P-SDSD framework organizes the collaborative peers in a service semantic overlay built on the top of a P2P logical network overlay.

### 3.1 Peer knowledge infrastructure

In the P2P-SDSD framework independent peers cooperate in a common domain by sharing services without any a priori reciprocal knowledge. In such a scenario, no centralized authorities are defined to provide a comprehensive view of
the shared services in the collaborative network. In fact, the inner dynamism of the requirements in the cooperation scenario makes it difficult to manage a centralized organization. As a consequence, each broker in the network has a local knowledge infrastructure: (i) UDDI Registry, where implementations of services are registered with their URL and associated to their abstract services through \texttt{tModels}; (ii) Peer Ontology, that provides a conceptualization of abstract service operations (e.g., drug ordering) and I/O parameters (e.g., medicine, price) through concepts and semantic relationships between them; (iii) standard service categorization, denoted with \textit{Service Category Taxonomy} (SCT), extracted from available standard taxonomies (e.g., UNSPSC, NAICS) to conceptualize service domain (e.g., Healthcare Services); abstract services are associated to SCT categories in the UDDI Registry; (iv) \textit{Service Ontology}, where abstract services and semantic links are stored.

This local knowledge is represented using OWL-DL formalism. The peer ontology and the SCT are augmented by a thesaurus containing terms that are related by terminological relationships (as synonymy or hypernymy) to the names of concepts and categories. The thesaurus is automatically built from the general domain independent source of lexical information, WordNet. By means of the thesaurus, broker matching capabilities based on the peer ontology and categories are extended. More details about the combined use of ontologies and thesaurus can be found in [7].

3.2 Service Matchmaker

Each broker in the collaborative network is endowed with an ontology-based matchmaker to support service request processing and SSO management. In [7] we defined a hybrid matchmaking strategy and in [6] the COMPAT architecture that implements it. In the hybrid model, a deductive matchmaking model is combined with a similarity-based model to compare services on the basis of their functional interface. The SCT, peer ontology and thesaurus are exploited by the matchmaker to identify matching services. The deductive matchmaking is used to \textit{qualify} the kind of match \texttt{MatchType} ($S_1$, $S_2$) between two abstract services $S_1$ and $S_2$. According to this matchmaking model, it is possible to state if $S_1$ and $S_2$ provide the same functionalities ($S_1$ \texttt{Exact} $S_2$), if $S_1$ provide additional functionalities with respect to $S_2$ ($S_1$ \texttt{Extends} $S_2$) or viceversa, if there is a non empty intersection between functionalities provided by $S_1$ and $S_2$ ($S_1$ \texttt{Intersects} $S_2$) or if $S_1$ and $S_2$ have nothing in common ($S_1$ \texttt{Mismatch} $S_2$). In case of partial overlapping among service functionalities (\texttt{Extends} or \texttt{Intersects}) the similarity-based matchmaking model is used to \textit{quantify} service similarity $Sim(S_1, S_2) \in [0, 1]$ through coefficients properly defined to compare service interfaces. These coefficients evaluate similarity distance between service interfaces by exploiting terminological relationships in the thesaurus. Otherwise, if $S_1$ \texttt{Exact} $S_2$ or $S_1$ \texttt{Mismatch} $S_2$, $Sim(S_1, S_2) = 1.0$ or $Sim(S_1, S_2) = 0.0$, respectively. Two abstract services are \textit{similar} (denoted with $S_1 \approx S_2$) when \texttt{MatchType}(S1,S2) is not \texttt{Mismatch} and $Sim(S_1, S_2) \geq \delta$, where $\delta$ is a similarity threshold experimentally set. A detailed presentation of the hybrid matchmaking model with presentation of experimental results about the
matchmaker is given in [7].

3.3 Request processor

This module is in charge of receiving incoming service requests and collecting searching results according to threshold-based criteria. The request processor applies semantic service comparison exploiting the capabilities of the service matchmaker in order to find relevant services described in local UDDI registry for a given request. After a local search, the request processor forwards the request over the network to semantic neighbors. We distinguish two cases:

– **service request from a requester**, the request processor receives a service request $S_R$ either locally from the user assistant-GUI or from another peer; the service request is matched against local abstract services by exploiting capabilities of the service matchmaker module; not relevant matching results are filtered out according to a threshold-based mechanism; searching results are collected into a list $MS(S_R)$; the list $MS(S_R)$ is sent to the semantic forwarding manager; which applies different forwarding policies of the request towards semantic neighbors that store similar services; the request processor collects URLs of all concrete implementations of matching services in $MS(S_R)$ and sends back the search answers to the peer from which the request came or directly to the GUI, if the service request has been generated locally;

– **probe service request from a broker**, the request processor receives the functional interface of an abstract service $S_X$ by a broker in the collaborative network, that aims at establishing inter-peer semantic links with the current peer; the request processor matches $S_X$ against local abstract services by exploiting capabilities of the service matchmaker; not relevant matching results are filtered out according to a threshold-based mechanism; matching abstract services are collected in a list $\{\langle S_1 X, Sim_1, mt_1 \rangle, \ldots, \langle S_n X, Sim_n, mt_n \rangle\}$ such that $S_X \approx S_i X$, where $Sim_i = Sim(S_X, S_i X)$ and $mt_i = \text{MatchType}(S_X, S_i X)$; this list is returned to the broker from which the probe service request came that can use it to update its service ontology.

In particular, probe service requests are sent according to a Time-To-Live (TTL) mechanism with a low TTL value to avoid network overload. Experimentation is being performed to establish the best value of TTL.

3.4 Service semantic overlay management

This module is in charge of discovering the semantic neighbors of a peer and maintaining inter-peer semantic links with them. Moreover, it participates to the process of answering a request by forwarding it to selected semantic neighbors. To these purposes, the service semantic overlay management includes a semantic link manager and a semantic forwarding manager.
Semantic link manager  Dealing with P2P network, inner dynamics must be taken into account. In our scenario, network evolution must be considered both at the SSO and at the logical overlay. To this purpose, the semantic link manager is in charge of managing SSO and its evolution with respect to changes that can occur to the participating peers.

Changes to SSO include these events: a new broker join the network, a service is published/removed, the IP address of a new peer is acquired at the logical network overlay, a semantic neighbor disconnects from the collaborative network.

If a new broker \( P_X \) joins the network, the semantic link manager of \( P_X \) generates a probe service request for each abstract service \( S_i_X \) of \( P_X \). The probe service request contains the interface of \( S_i_X \) and the IP address of \( P_X \). This probe service request is sent on the collaborative network as explained in Sect. 3.3. A broker \( P_Y \) that receives the probe service request, replies to \( P_X \) with the list of \( S_j_Y \) such that \( S_i_X \approx S_j_Y \), together with the MatchType and the similarity value. The semantic link manager establishes an inter-peer semantic link \( isl_{P_X \rightarrow P_Y} (S_i_X, S_j_Y) \) for each \( S_j_Y \). Hence, \( P_Y \) is recognized as a semantic neighbor and a semantic link towards \( P_Y \) is inserted into the service ontology of \( P_X \).

If a new service \( S_i_X \) is published on a broker \( P_X \), a probe service request is sent to semantic neighbors of \( P_X \) and inter-peer semantic links are established on the basis of obtained answers as in the previous case. If a service is removed, the inter-peer semantic links based on it are removed on the broker on which it was published.

If the IP of a new broker \( P_Y \) is acquired by \( P_X \), due to the shuffling-based protocol at the logical network overlay, \( P_X \) sends probe service requests to \( P_Y \) and possibly inter-peer semantic links between \( P_X \) and \( P_Y \) are established based on the received answers.

Finally, a broker \( P_X \) recognizes that a semantic neighbor \( P_Y \) becomes unavailable if a given number of messages sent to \( P_Y \) are not answered. In this case the inter-peer semantic links toward abstract services published on \( P_Y \) are removed from \( P_X \).

Semantic forwarding manager  This module receives from the request processor component a list of abstract services \( S_i_X \in MS(S_R) \) that match with the service request \( S_R \) and applies different selective forwarding policies based on inter-peer semantic links to improve service discovery process and reduce network overload.

Minimal policy. Search over the semantic overlay stops when matching services which fully satisfy the request have been found; this strategy is performed according to the following rules:

R1. for each \( S_i_X \in MS(S_R) \) such that \( S_i_X \) Exact | Extends \( S_R \), it is not necessary to forward the request to semantic neighbors, since concrete implementations of \( S_i_X \) already satisfy completely the request;

R2. for each \( S_i_X \in MS(S_R) \) such that \( S_R \) Intersects | Extends \( S_i_X \), the request is not completely satisfied by \( S_i_X \) and is forwarded to semantic neighbors, that could add further functionalities to those already provided by \( S_i_X \); however, broker \( P_X \) does not consider semantic neighbors that provide
services \( S_{jY} \) such that \( S_{iX} \text{Extends} | \text{Exact} S_{jY} \), because this means that \( S_{jY} \) does not provide additional functionalities with respect to those already provided by \( S_{iX} \):

R3. if no semantic neighbors exist for any abstract service \( S_{iX} \in MS(S_{R}) \), the request is forwarded to one of the brokers that are connected to \( P_X \) in the logical network overlay (randomly chosen);

\[ \text{Fig. 2. Exploitation of inter-peer semantic links during distributed search.} \]

**Example 1.** An example of how the minimal policy is applied by the semantic forwarding manager is shown in Figure 2. According to the inter-peer semantic links established among the brokers in the figure, if a request \( S_{R} \) is submitted to broker \( P_X \) and the following kind of match has been discovered \( S_{3X} \text{Intersects} S_{R} \), then the request is forwarded both to broker \( P_Y \) (1) and \( P_Z \) (2). This is done according to the rule R2 of the policy. On broker \( P_Z \), the matchmaker is applied and \( S_{1Z} \text{Exact} S_{R} \), then the search stops (3) according to rule R1 and \( P_Z \) replies to \( P_X \) with the URLs of concrete implementations associated to \( S_{1Z} \). On broker \( P_Y \), the matchmaker is applied and \( S_{2Y} \text{Intersects} S_{R} \), then \( P_Y \) replies to \( P_X \) with the URLs of concrete implementations associated to \( S_{2Y} \), but also in this case rule R1 is applied and the request is not further forwarded to \( P_H \) (4), since \( S_{5H} \) does not add additional capabilities with respect to \( S_{2Y} \).

**Exhaustive policy.** This policy follows the same rules of the previous one, but it does not stop when matching services that fully satisfy the request are found; according to this policy, also for each \( S_{iX} \in MS(S_{R}) \) such that \( S_{iX} \text{Exact} | \text{Extends} S_{R} \), the request \( S_{R} \) is forwarded to semantic neighbors to find other services that could present, for example, better non functional features (not discussed in this paper). A broker which receives \( S_{R} \) from a semantic neighbor applies one of the two forwarding policies; the search stops according to a Time To Live mechanism.
Example 2. If the exhaustive policy is applied to the scenario in Figure 2, the request is forwarded also from broker $P_Z$ to the broker $P_T$. Broker $P_Z$ collects results from $P_T$ together with results found locally and then replies to $P_X$, from which the request came.

3.5 Logical overlay management

The Logical overlay management component manages the peer connectivity at the logical level.

Membership & communication. The membership & communication component is responsible for initializing and maintaining the peer communication with the other peers. When a peer $P_X$ wants to join to the collaborative network, the membership/communication component obtains from a reference peer a list of addresses of peers currently available and starts to contact them. In case that a peer $P_Y$ acting as broker replies, $P_X$ can connect to it and communicate the $P_Y$ IP address to semantic link manager to allow SSO evolution.

Network evolution. In order to guarantee connection between peers an Overlay Management Protocol (OMP) is used, which defines some specific procedures to join, leave and modify the logical network overlay. In our approach a shuffling-based OMP is chosen in order to allow more effective information diffusion among peers. The shuffling protocol is quite simple: each peer continuously changes the set of its neighbors in the logical network by occasionally contacting a random neighbor, then they exchange some of their neighbors. This mechanism allows that a peer eventually become aware of new peers that are made available on the network also if initially they are not directly connected.

4 Experimental evaluation

In this section, we present a preliminary evaluation of performances of our approach. In particular, we are interested in evaluating experimentally the semantic forwarding strategy. To this purpose, we performed a set of simulations based on NeuroGrid [11], an extensible network overlay simulator in which we have implemented the P2P-SDSD service semantic overlay and the minimal forwarding policy explained in Section 3.4.

The simulations we have run compares P2P-SDSD forwarding policy with the Gnutella one [9] both in terms of efficiency and scalability. Actually, Gnutella is oriented to document discovery, but for purpose of comparison with P2P-SDSD we have implemented in the simulations a web service discovery process that exploits the Gnutella forwarding policy. The choice of a comparison with Gnutella is due to the fact that both P2P-SDSD and Gnutella define an overlay network built on top of an unstructured P2P network. Apart from the architectural similarities, we have considered Gnutella since its message forwarding strategy is well-known and it is frequently considered as a reference example. Some other P2P forwarding strategies have been also considered for a comparison with P2P-SDSD and we plan to perform additional experiments in future work. In particular, P2P-SDSD and Gnutella service discovery have been compared on these parameters:
hit ratio: with respect to a submitted request it is the ratio among the number of services retrieved by the distributed search policy and the number of services retrieved if all offered services would be available on the node receiving the request and the local search applied; the higher the hit ratio, the more effective the distributed search by providing an answer similar to a local search; this parameter depends on the TTL value;

generated messages: it is the number of overall messages generated to answer a service request; this parameter depends on the TTL (Time To Live) value and on the peer’s average number of connections to its neighbors; the lower the generated messages, the better the scalability, since a lower number of messages per request reduces the possibility of network congestion.

<table>
<thead>
<tr>
<th>Simulation Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Total number of brokers on the network</td>
</tr>
<tr>
<td>NS</td>
<td>Total number of available services</td>
</tr>
<tr>
<td>Initial TTL</td>
<td>Initial TTL associated to a request message</td>
</tr>
<tr>
<td>Simulation Type</td>
<td>P2P-SDSD or Gnutella</td>
</tr>
<tr>
<td>γ</td>
<td>Probability that two brokers has established semantic links (∈ (0,1))</td>
</tr>
<tr>
<td>ASP</td>
<td>Average number of services per broker</td>
</tr>
</tbody>
</table>

Fig. 3. Simulation Parameters

The experiments have been performed: (i) to demonstrate the better hit ratio results of our distributed search with respect to Gnutella search; (ii) to confirm that the use of the P2P-SDSD request forwarding policy results in an improved scalability. The different experimental settings for the simulation are obtained by setting some operative parameters shown in Figure 3. These parameters determine how the simulated logical overlay network is built and run. The simulator generates a random P2P network [11] of N peers and NS artificial services. Services are randomly assigned to each peer according to the ASP parameter. Given a pair of artificial services $S_1$ and $S_2$ the kind of match between them is established according to the following distribution of probability: $P(S_1$ MISMATCH $S_2) = 50\%$, $P(S_1$ INTERSECT $S_2) = 25\%$, $P(S_1$ EXTENDS $S_2) = P(S_2$ EXTENDS $S_1) = 10\%$, $P(S_1$ EXACT $S_2) = 5\%$.

The γ value is the probability that a pair of peers has already exchanged probe queries and therefore have possibly established inter-peer semantic links. The Initial TTL determines the maximum number of times a message can be forwarded on the network and therefore the higher this value, the higher the number of peers involved in answering the request. In particular, for the results discussed in the following, the initial TTL has been varied between 1 and 12. Moreover, we have supposed a medium sized network having N=100 brokers in which NS=50 services are globally available and ASP=20 services are on average available on each broker. Finally, the probability that two broker have already exchanged probe queries and consequently established semantic links has been set to 60\% (γ = 0.60).

Figure 4 compares P2P-SDSD and Gnutella approaches with respect to the hit ratio by setting different values for the Initial TTL parameter. The figure
shows how P2P-SDSD overperforms the Gnutella even with low TTL values. This can be explained by the fact that inter-peer semantic links allow to selectively reach the most of the peers that provide relevant services with a low number of request forwardings. As TTL gets higher also the Gnutella performs better since the request reaches the most of the peers in the network, but the network overload increases.

The analysis of generated messages has been performed on a series of 13 simulations with Initial TTL value set to 5 and a network of 100 peers. In each simulation we ran, a request has been submitted and we have collected the results obtained with both P2P-SDSD and Gnutella on the same simulated network. In these simulations the average number of generated messages for P2P-SDSD is nearly 19.31, that is about 50% lower than the value of 38.32 obtained by the Gnutella approach.

5 Conclusions

In this paper, we proposed the P2P-based Semantic Driven Service Discovery (P2P-SDSD) architectural framework to enable cooperation and communication based on a semantic overlay that organizes semantically the discovery space. Service discovery in P2P-SDSD is based on a service semantic overlay, over the logical network overlay, built by establishing semantic links among peers that offer similar services. The semantic overlay is exploited to optimize the service discovery process and improve its efficacy keeping low the generated network overload. Preliminary experiments have been performed to confirm the advantages derived from the exploitation of semantic overlay if compared with traditional Gnutella approach. Further experimentation will evaluate the impact of the proposed approach on open P2P networks, where additional issues must be considered during the construction of semantic overlay, and concrete applications in those kinds of networks will be investigated. Future work includes the
definition of discovery and forwarding policies based also on user preferences and Quality of Service.

References


