Enabling semantic search in P2P systems through a three-layer Distributed Service Registry

Devis Bianchini\textsuperscript{1}, Valeria De Antonellis\textsuperscript{1} and Michele Melchiori\textsuperscript{1}

University of Brescia, Department of Electronics for Automation, via Branze, 38
25123 Brescia - Italy

Abstract. Effective searching and access to data and functionalities in P2P environments require platform independency for integrating different sources over the Web and must take into account the presence of high dynamicity and the absence of a common resource conceptualization. Service Oriented Architecture enables to export shared data and functionalities through Web Services, thus enabling platform-independent resource discovery. In P2P networks, available services are published and retrieved on different registries, that are autonomous and heterogeneous. Semantic Web Service technology is suggested to face semantic heterogeneity. We propose a virtual Distributed Service Registry (DSR) and a reference architecture to support semantic search over P2P networks. The DSR is organized on three layers: a logical layer, in which single registries are connected as peers in a P2P network; a semantic layer, where peer registries are semantic-enriched and semantic links between peer registries holding similar services are established; a mapping layer, where mappings between similar services are defined to support service interoperability between heterogeneous peers. In the reference architecture, each peer is equipped with: a service knowledge evolution manager, to update peer local knowledge through interactions with other peers; a semantic search assistant, that exploits the three-layer organization of the DSR to satisfy requested user searches, also suggesting possible alternative services or related services suitable for composition, on the basis of semantic links.

1 Introduction

Effective searching and access to data and functionalities in P2P environments require platform independency for integrating different sources over the Web and must take into account the presence of high dynamicity and the absence of a common resource conceptualization. Service Oriented Architecture enables to export shared data and functionalities through Web Services, thus enabling platform-independent resource discovery. An ever-growing amount of information on the Web today is accessible through Web Service interfaces. In particular, search services are used to access underlying Web sources by requiring users to fill in some fields in a search form and return search results properly ranked \cite{7}. Each search service is described by means of its functional interface, where we
distinguish among inputs and outputs. Inputs correspond to data to be provided by the user to access the underlying Web sources, while outputs represent search results, provided through the execution of the service. In P2P networks, available services are published and retrieved on different registries, that are autonomous and heterogeneous. Semantic Web Service technology is suggested to face semantic heterogeneity. We propose a virtual Distributed Service Registry (DSR) and a reference architecture to support semantic search over P2P networks. The DSR is organized on three layers: a logical layer, in which single registries are connected as peers in a P2P network; a semantic layer, where peer registries are semantic-enriched and semantic links between peer registries holding similar services are established; a mapping layer, where mappings between similar services are defined to support service interoperability between peers. In the reference architecture, each peer is equipped with: a service knowledge evolution manager, to update peer local knowledge through interactions with other peers; a semantic search assistant, that exploits the three-layer organization of the DSR to satisfy requested user searches, also suggesting possible alternative services or related services suitable for composition, on the basis of semantic links. Other approaches based on distributed service registries have been proposed in literature for service discovery purposes. Proposals like the one described in [8, 12] are based on a centralized component that classifies the involved registries according to service categories, which are used to identify the suitable registries to which a service request must be submitted. Semantic heterogeneity, raising from the use of different peer ontologies for semantic service description, has been addressed in [4] through the adoption of mediators components. In [3] the organization of peer registries through inter-peer semantic links has been proposed to allow more efficient request forwarding on the network with respect to traditional flooding-based strategies.

**Contribution of our paper.** The three layers of the DSR proposed in this paper clearly distinguish between: (i) the logical organization of registries in the open and highly dynamic P2P network, where nodes keep joining and leaving at any moment; (ii) the organization of available services through semantic links, that allow more sophisticated requests forwarding policies to find out suitable search services; (iii) mappings between similar services, that enable service interoperability in presence of different resource conceptualizations. The advantages of such a service organization have been highlighted also in [1] with reference to data discovery in P2P networks. Moreover, the DSR is part of a reference architecture, where each peer is also equipped with a knowledge evolution manager, that has been designed to update peer knowledge through local interactions with other peers, without any centralized component. The paper is organized as follows: in Section 2 the reference architecture is presented; Sections 3 and 4 describe the roles and functionalities of the semantic search assistant and the knowledge evolution manager, respectively; in Section 5 some implementation hint and preliminary evaluation are given, while Section 6 concludes the paper.
2 The Semantic Search Framework

In this section, a running example illustrates the considered application scenario and the reference architecture provides the building basis for the semantic search framework.

2.1 Running example

Consider the classical case of a traveller who is willing to organize a week holiday in one of the European cities and is searching for a low-cost flight. He/she has not chosen the destination yet and his/her constraints regard the cost of the holiday and the departure and return dates. First, he/she looks for available low-cost flights and, on the basis of the offers, decides which city to visit. Secondly, he/she has to find a cheap accommodation for the final destination of the journey. The traveller joins a network of peers, with registries that provide search services to find flight reservation and hotel booking facilities and display search results ordered by price. Examples of search services are depicted in Figure 1, where samples of search results are also shown. Figure 1 shows the organization of services in the DSR as explained in the next section, where the reference architecture will be presented.
For example, service FlyCheap on peer RX provides the travel information of a low-cost flight. Service AirLow on peer RY provides similar information: the price and arrival city of a low-cost flight, given the departure and return dates, the departure city and the admitted flexibility. Service HotelSearching on peer registry RZ (to which peer registry RY is connected in the P2P network) provides the name, cost and stars of hotels in a specified city for which a given type of room is available, from a check-in to a check-out date.

Our reference architecture can support the user search in two ways: (i) given a low-cost-flight search submitted on peer registry RX, the system is able to suggest the local service FlyCheap and also services with similar interfaces, registered on a different registry (for example, service AirLow on registry RY); (ii) once selected a service for low-cost-flight search, say AirLow, the system is able to suggest possible related services which accept as INPUT what is OUTPUT of AirLow, to which the user can be interested in, such as hotel booking facilities (for example, service HotelSearching on registry RZ). In the following, we present such a reference architecture.

2.2 Reference architecture

The proposed reference architecture for each peer is based on:

- the local portion of the virtual Distributed Service Registry (DSR), where available services are registered and organized on three layers;
- a semantic search assistant, that exploits the three-layer organization of the DSR to satisfy requested user searches, also suggesting possible alternative services or related services suitable for composition;
- a service knowledge evolution manager, that updates peer local knowledge through local interactions with other peers.

Both the semantic search assistant and the service knowledge evolution manager work according to the three-layer structure of the DSR.

At the bottom layer (logical layer) registries are connected as peers that can leave and join the network, as usual for P2P environments. At this layer, each registry maintains the functional interfaces of local services. In Figure 1, three peer registries are highlighted, named RX, RY and RZ, connected through logical links.

At the middle layer (semantic layer), each registry maintains a semantic description of locally available services. The semantic service description is expressed according to recently proposed languages for semantic annotation of service functional interface (e.g., WSDL-S [2]). To this aim, each peer is also equipped with an OWL-DL peer ontology, that provides a conceptualization of I/O parameters through concepts and semantic relationships between them (e.g., subClassOf or equivalentClass). Finally, according to UDDI standard [11],
registered services are associated to categories from the UNSPSC standard taxonomy. Besides this local conceptualization of service functional interfaces, each registry maintains a set of semantic links towards registries where similar services are published.

We distinguish among two kinds of semantic links between Web Service descriptions:

*functional similarity links*, denoted with $S_1 \approx S_2$, that relate similar semantic service descriptions $S_1$ and $S_2$ registered on two different peers, that is, services with similar functionalities (e.g., AirLow and FlyCheap services in Figure 1);

*coupling similarity links*, denoted with $S_1 \rightarrow S_2$, to assert that outputs of $S_1$ are semantically related to the inputs of $S_2$: $S_1$ and $S_2$ are defined as coupled and can be taken into account as suitable for composition (e.g., AirLow and HotelSearching services in Figure 1).

Semantic links are set by the service knowledge evolution manager presented in Section 4. The matching information labelling semantic links are obtained by applying service matchmaking techniques detailed in [5].

Finally, at the upper layer (mapping layer), mappings among I/O of related services are set by the service knowledge evolution manager to enable effective propagation of service invocations throughout the network. Matchmaking techniques are applied to pairs of service inputs and outputs separately, obtaining a set of correspondences that are used to suggest I/O mappings that can be validated by the registry administrator. For example, in Figure 1, I/O mappings between outputs of FlyCheap and AirLow services and I/O mappings from outputs of AirLow service to inputs of HotelSearching service are shown. Complex mappings (i.e., one to many) has to be managed according to suitable techniques as for attribute mappings in data schema integration [10]. We will further work on this aspect in next developments of the approach.

### 3 Service-based semantic search in the Distributed Service Registry

The semantic organization of services in the DSR is exploited by the semantic search assistant during the semantic search process, that is based on service discovery and invocation. Semantic links ensure scalable system performances by guiding in the selection of suitable registries in the DSR for service request propagation. When a new service request $S_R$ is submitted to a registry $R_X$, the service-based semantic search in the DSR is performed according to the phases shown in Figure 2.

1 The United Nations Standard Products and Services Code:  
Identification of local results. In the first phase, the service request $S_R$ submitted to the local registry $R_X$ is matched against the local semantic service descriptions and a list $M_S(S_R)$ of matching service descriptions is obtained. If local results do not completely satisfy the request, $S_R$ can be forwarded to the other nodes of the DSR to extend the list of matching services (expand search branch).

Selection of request recipients. To prune the set of registries to be investigated, thus avoiding time-consuming service search on the P2P network, registries related through functional similarity links are selected as request recipients. Moreover, candidate request recipients can be filtered according to different forwarding policies, that are based on the matching information labeling the semantic links. According to a minimal policy, search over the network stops when matching services which fully satisfy the request have been found. Exhaustive policies can also be applied selecting recipients through semantic links, but the search does not stop when matching services that fully satisfy the request are found: the request $S_R$ is forwarded to other registries to find out services that could present, for example, better non functional features. In [6] a detailed presentation of different forwarding policies based on functional similarity links is provided.

Request forwarding and collection of search results. Once the candidate request recipients have been selected, the search request $S_R$ is forwarded towards
them in order to obtain required search results on the DSR. A token-based strategy is adopted to avoid cycles and network overloading. Each registry receiving the request checks for locally available matching services: if matches have been found, the registry replies to the registry \( R_X \), from which the request came, it consumes a token and, if the remaining number of tokens is not zero, it repeats the forwarding procedure based on its functional similarity links and the current token value (expand search branch). In the running example, if \( \text{FlyCheap} \in MS(S_R) \), according to a minimal policy, the request is forwarded towards \( R_Y \). In fact, because \( \text{FlyCheap} \approx \text{AirLow} \), \( \text{AirLow} \) is also a candidate service for the request. Request propagation towards \( R_Y \) enables the second registry to re-apply the same procedure, thus reaching registries that are not directly connected to \( R_X \) at the logical layer (e.g., \( R_Z \)). Search results are collected and presented to the user.

**Extension of search results.** When the user selects one of the search results, coupling similarity links can be exploited to propose additional results regarding those services that can offer additional functionalities with respect to the user request. In the running example, if \( \text{AirLow} \) service is selected, the system exploits the coupling semantic link (\( \text{AirLow} \rightarrow \text{HotelSearching} \)) to propose the \( \text{HotelSearching} \) service to the user as a service complementary to \( \text{AirLow} \).

**Service invocation.** Finally, once suitable matching services have been identified for a given service request, service invocation can be performed on the DSR following I/O mappings. The user can select one of the search results \( S \) and the service functional interface is displayed as a Web form to enable service invocation. I/O mappings starting from \( S \) are exploited to properly propagate the service invocation towards other registries.

Note that, without the organization of services through semantic links, the discovery process would rely on conventional P2P infrastructures and associated routing protocols for query propagation in the network (e.g., flooding). Exploiting the semantic links, it is possible to enforce query forwarding according to content similarities rather than to the mere network topology.

### 4 Managing knowledge evolution in the Distributed Service Registry

The knowledge evolution manager is responsible of setting semantic links by applying service matchmaking techniques defined in [5]. Service matchmaking techniques are applied to: (i) evaluate the degree of matching between services, to the purpose of establishing functional similarity links; (ii) evaluate the degree of interdependency between services, to the purpose of establishing coupling similarity links. Matchmaking techniques work on the basis of knowledge in the OWL-DL peer ontology. Since we do not constrain registries in the DSR to adopt a common shared conceptualization, different peer ontologies can be used on different nodes in the DSR. To bridge the gap between slightly different terminologies, peer ontologies are extended with a thesaurus of terms related by
terminological relationships to the names of ontological concepts. The thesaurus is built from the WordNet lexical system [9]. In this way, it is possible to extend matchmaking capabilities when looking for correspondences between semantic services described in terms of different peer ontologies. A detailed explanation of how the thesaurus is built and used in combination with OWL-DL ontologies has been presented in [5].

Semantic links are labelled with matching information to qualify the kind of match and to quantify the degree of similarity between similar or coupled services. In particular, for functional similarity links the kind of match between semantic service descriptions can be one of the following:

**Exact** to denote that \( S_1 \) and \( S_2 \) have the same capabilities, that is, they have:
- (i) equivalent output parameters;
- (ii) equivalent input parameters;

**Extends** to denote that \( S_2 \) offers at least the same capabilities of \( S_1 \), that is:
- (i) an equivalent or more specific output parameter in \( S_2 \) can be found for each output parameter of \( S_1 \);
- (ii) a set of input parameters in \( S_2 \) can be found, each of them is equivalent or more generic than an input parameter of \( S_1 \);
the inverse kind of match is denoted as **Restricted**; the rationale behind the Extends match is that \( S_2 \) at least fulfils the functionalities of \( S_1 \) if it provides all the \( S_1 \) outputs, but, on the other hand, \( S_1 \) must be able to provide all the inputs needed for the execution of \( S_2 \);

**Intersects** to denote that \( S_1 \) and \( S_2 \) have some common capabilities, that is:
- there exists a pair of I/O parameters, one from \( S_1 \) and one from \( S_2 \), that are related in any generalization hierarchy in the peer ontology.

The degree of similarity is expressed by a similarity coefficient \( \text{Sim}_{IO}(S_1, S_2) \in [0, 1] \) between \( S_1 \) and \( S_2 \), that is evaluated by matching inputs (resp., outputs) of \( S_1 \) with inputs (resp., outputs) of \( S_2 \). For coupling similarity links, a coupling coefficient \( \text{Coupl}_{IO}(S_1, S_2) \in [0, 1] \) is used to quantify the degree of matching between the outputs of \( S_1 \) and the inputs of \( S_2 \).

The knowledge evolution manager is responsible of identifying similar services registered on different nodes by means of a knowledge harvesting process. Specifically, at predefined time intervals, the knowledge evolution manager of peer \( R_X \) sends a probe request for each locally available service that is not yet source of a semantic link. The probe request contains the semantic description of the service. A peer \( R_Y \), receiving the request, applies by its knowledge evolution manager the matching techniques between the probe request and each local semantic service description. For each similar service, a reply is sent back containing: (i) a reference to the matching service, (ii) the corresponding matching information, that is, the kind of match and the value of \( \text{Sim}_{IO} \) coefficient for functional similarity links or the value of \( \text{Coupl}_{IO} \) coefficient for coupling similarity links; (iii) the semantic correspondences between I/O parameters required to set I/O mappings. \( R_X \) collects received information and establishes a semantic link towards \( R_Y \).

In case the matching service on \( R_Y \) has the same reference of a local service in \( R_X \) (that is, they have the same UDDI ServiceKey and provider IP address),
the link is not established since they represent the same service published twice (actually, in a P2P environment service multiple registrations on different registries can not be avoid).

Semantic links are established and updated when registries join or leave the P2P network and when services are published/removed from peers. In particular, semantic links from \( R_X \) to \( R_Y \) are removed if a given number of service requests sent to \( R_Y \) are not answered. If a new service description \( S_i_X \) is added due to the publication of a new service on \( P_X \), a probe service request is sent and semantic links are established on the basis of obtained answers as previously explained. Finally, if a service description is removed on \( R_X \), the semantic links starting from it are removed without sending any messages on the network. The case of a registry joining the DSR is managed as follows: when the IP address of a new registry \( R_Z \) is acquired by registry \( P_X \) it sends a probe service request to \( P_Z \) and semantic links between \( P_X \) and \( P_Z \) are possibly established based on the received answers. Finally, a registry \( P_X \) recognizes that a linked registry becomes unavailable if a given number of messages sent to it are not answered. As a consequence, corresponding semantic links are removed from \( P_X \).

5 Usage scenario and preliminary experimental evaluation

Figure 3 displays the Web interface that assists the final user during semantic search on each registry in the DSR. In figure, the running example and the discovery phases presented in Section 3 have been considered. In the “Search for” area of the Web interface, the user searches for services that provide information about the available destinations and corresponding fares of low-cost flights, given the departure and return dates and the departure city (first phase in the figure). In this example, the request is submitted on registry \( R_X \). Locally available services with corresponding inputs and outputs, the kind of match and the similarity degree with respect to the request are displayed. In the “Search for” area, the user can also specify a similarity threshold to filter out not relevant services. In the running example, only the FlyCheap service is listed among the local results. Note that I/O names in the list of the retrieved services are displayed together with (possible) corresponding terms used while compiling the search form (e.g., FromCity and DepartureCity in figure). At this point, one of the available forwarding policies (exhaustive or minimal) can be applied to propagate the service request to the other registries of the DSR following functional semantic links, to find additional similar services (second phase). In the example, the AirLow service on registry \( R_Y \) is added to the retrieved ones. The collected services are listed on the interface (third phase). After selecting one or more retrieved services, the user can visualize (through the “Show related services” button) a list of services that are related to the selected ones by means of coupling similarity links (fourth phase). Finally, through the “Invoke selected services” button, the user can start the invocation phase (fifth phase): a Web form is dynamically built starting from the WSDL documents of the selected
services and enables the user to provide the required inputs and obtain the expected search results.

Fig. 3. Semantic Search assistant Web interface.

Preliminary experimental evaluation. This work started from previously efforts on semantic service matchmaking and P2P semantic-driven service discovery. Experimentations on these building blocks have been presented in [5] and [6], respectively. In particular, in [6] we performed a set of simulations by implementing a DSR simulator designed on the basis of the Neurogrid P2P simulator\(^2\). The simulator is able to build the DSR semantic layer, run over it the service discovery process according to the minimal and exhaustive forwarding policies and collect statistics useful for the evaluation of the approach. Performed simulations are in particular devoted to measure the recall of the DSR and its scalability. The experiments have demonstrated that the semantic forwarding strategies generate lower traffic than traditional non semantic ones, generally proving better scalability and allowing a better pruning process on suitable services with which mappings at the upper level of the DSR must be set.

6 Concluding remarks

In this paper we proposed a reference architecture to enable scalable semantic searching in P2P systems. The proposed architecture is based on a virtual Dis-

\(^2\) http://www.neurogrid.net
tributed Service Registry, where each peer registry is equipped with a knowledge infrastructure for semantic description of locally available services and semantic links towards similar services registered on different nodes. Additional modules on each peer are in charge of exploiting the organization of services in the registry and of keeping update the peer local knowledge through interactions with other peers. Currently, a prototype and a simple Web interface to use the DSR have been developed and preliminary experimental evaluation have been performed. Future efforts will be devoted to complete the proposed architecture with the development and testing of the knowledge evolution manager and to evaluate the effectiveness of the three-layer organization of the DSR in supporting the user to define mappings throughout the network for semantic search purposes. The resulting application will be validated on real case scenarios.

References