

Semantic-enriched Service Discovery*

Devis Bianchini, Valeria De Antonellis, Michele Melchiori and Denise Salvi
Università di Brescia
Dip. di Elettronica per l'Automazione
Via Branze, 38
25123 Brescia - Italy

{bianchin|deantone|melchior|salvi}@ing.unibs.it

Abstract

Automated techniques and tools are required to effectively locate services that fulfil a given user request. To this purpose, the use of semantic descriptions of services has been widely motivated and recommended for automated service discovery under highly dynamic and context-dependent requirements in distributed environments. Our aim in this work is to propose a semantic-enriched framework to describe services and an ontology-based hybrid approach where such framework is exploited combining together different kinds of comparison strategies to provide a flexible and efficient matchmaking between service descriptions. For service discovery two matchmaking strategies are proposed: a deductive strategy based on Description Logics, with a reasoning procedure exploiting ontology knowledge to assess the type of match between services; a similarity-based strategy, exploiting retrieval metrics to measure the degree of match between services.

1 Introduction

In the recent years a growing number of services has been provided on the web and it is more and more difficult to promptly access and retrieve needed web information. In particular, a challenging problem is to improve servicing of the web user requests. Totally or partially automated techniques and tools are required to effectively locate services that fulfil a given user request. To this purpose, the use of semantic descriptions of services has been

widely motivated and recommended for automated service discovery under highly dynamic and context-dependent requirements in distributed environments. Semantic-enriched frameworks are considered a key issue to enforce timely discovery and dynamic composition of services. The ontology description languages OWL, OWL-S and, more recently, WSML have been proposed with the capability of supporting Description Logics like *SHOIN(D+)*, that is, a knowledge representation formalism with a sound, complete and decidable inference procedure [1]. In the literature, several approaches based on these languages are being developed. In [9] a service matchmaking strategy based on the DAML+OIL language used for service description, on the OWL-S service ontology and on a DL reasoner is proposed. The DAML+OIL expressions representing Service Profiles, according to the OWL-S ontology, are consistently mapped into the corresponding DL expressions and DL-based reasoning facilities are applied to check if the request description is equivalent, subsumed or consistent with the descriptions of service advertisements. The approach described in [11] also use a service capability representation based on the OWL-S Service Profile, with inputs, outputs, preconditions and effects. Semantic relationships between input and output parameters in the domain ontologies are exploited to verify if: (a) all the outputs required are also provided by the advertisement and (b) all the inputs needed for advertisement execution are provided by the requester. A logical formalism based on Horn clauses is also provided to express preconditions and effects and to verify their logical implication or equivalence. In [15] a DL-based reasoning algorithm extending the non standard inference mechanisms of *abduction* and *contraction* are applied for the matchmaking between the requested service profile and the provided one, expressed by means of DL expressions.

*This work has been partially supported by the MAIS (Multichannel Adaptive Information Systems [13]) FIRB Project funded by the Italian Ministry of Education, University and Research, and by NoE INTEROP [10] IST Project n. 508011 - 6th EU Framework Program.

The compared descriptions could be incomplete or not fully compatible, so when an element in the request not consistent with an element in the offer is found, it is removed (contraction) and each required element that is not present in the offer is added (abduction). Each time an element is removed or added, a penalty value is increased. The higher is the penalty, the lower is the compatibility between the request and the advertisement. The METEOR-S Web Service Discovery Infrastructure presented in [14] extends traditional UDDI Registries, including also semantic aspects of service descriptions by means of shared domain specific ontologies. While searching for services, relevant domain ontologies can be referred to, enabling semantic matchmaking of services. Moreover, registries are organized into domains to enable domain-based classification of all services.

Our aim in this work is to propose an ontology-based hybrid approach where different kinds of comparison strategies are combined together to provide a flexible and efficient matchmaking between service descriptions. We extend the keyword-based approach of UDDI Registry by considering semantic matchmaking on the basis of a Domain Ontology (that provides the general knowledge about concepts of the business domain in which services are used) and a Service Ontology (where services are organized by means of semantic relationships at multiple levels of abstraction). For service discovery two matchmaking strategies are proposed: a deductive strategy based on Description Logics, with a reasoning procedure exploiting ontology knowledge to assess the type of match between services [2]; a similarity-based strategy, exploiting retrieval metrics to measure the degree of match between services [3]. The similarity approach is applied after the deductive one in order to rank selected services according to the measured matching degree. Also in [7] a mixed matchmaking approach has been proposed: however, the similarity-based part of this approach exploits IR techniques that are not tailored to service description comparisons and, during the deductive matchmaking procedure, no terminological relationships are considered between names of elements used for service descriptions. Our approach overcomes these limitations.

The paper is organized as follows: in Section 2 we present the semantic-enriched framework for service modeling using ontologies; Section 3 and 4 describe the deductive and similarity-based matchmaking strategies and the hybrid approach that exploits the framework. Finally, conclusions are presented in Section 5.

2 Ontology-based service modeling

Starting from the WSDL document of a service and mapping it into a Description Logic expression, a service functional interface can be formally represented as a conjunction of:

- a concept in the form $\exists \text{hasCategory}.CAT$, where CAT is a concept representing the associated service categories;
- one or more concepts in the form $\exists \text{hasOperation}.OP$, where OP is a concept representing an operation of the service; each concept OP is described as a conjunction of:
 - the operation name, expressed by means of an atomic concept;
 - one or more concepts $\exists \text{hasInput}.IN$, where IN is a concept representing an input parameter of the operation;
 - one or more concepts $\exists \text{hasOutput}.OUT$, where OUT is a concept representing an output parameter of the operation.

IN and OUT are specified in the form $\exists R.C$, where R represents the name of the parameter and C is a concept representing admitted values (that is, the range of the parameter). C can be defined as an atomic concept, an enumeration $\{o_1, o_2, \dots, o_n\}$ of nominals or a complex concept obtained by applying the *intersection* operator (\sqcap), the *union* operator (\sqcup) and the *negation* operator (\neg).

The Description Logic family used corresponds to the \mathcal{ALCCO} one, that is, the \mathcal{ALC} logic enhanced with constructs to manage nominals. However, we serialize our service descriptions using OWL-DL, that can be consistently mapped into $\mathcal{SHOIN}(\mathcal{D}^+)$, including also role hierarchy (\mathcal{H}), number restriction (\mathcal{N}) and concrete datatypes (\mathcal{D}^+). This choice is twofold: to use OWL-DL as a recommended Web-based language to represent our ontologies and to allow for future extensions of service descriptions, enhancing expressiveness.

2.1 Adding semantics to service description

In order to add semantics to service descriptions, we exploit in a combined way a Domain Ontology $DomONT$, that contains the OWL-DL definitions of the elements used to describe services (input/output parameters, operation names, service categories) in terms of *concepts* and *semantic relationships* between them, a thesaurus automatically

derived from the lexical system WordNet [6], that contains *terminological relationships* between names of ontology elements (e.g., *synonymy*, *hypernymy*, etc.) to better capture their meaning, and a Service Ontology $ServONT$, that organizes services on different layers of abstraction as exposed in the Section 2.2. Ontologies, as well as service descriptions, are described using the OWL-DL language, allowing for the creation of an overall ontological framework in which service descriptions are completed with the semantics provided by ontologies.

The Domain Ontology provides several kinds of semantic relationships, in particular: *concept specialization*, *concept equivalence*, *concept disjunction* and *concept composition*. Moreover, given a set T of terms used as names of ontology elements and following a procedure similar to that proposed in [5], the thesaurus is built according to the following rules: (i) if $t \in T$ and an entry for t exists in WordNet, then t is defined as *simple term* and added to the thesaurus; (ii) if $t \in T$ is not a simple term, but it is a concatenation of simple terms $t = \langle st_1, st_2, \dots, st_n \rangle$, then t is defined as *composite term*. In this case an entry is defined in the thesaurus for t and for each $st_i, i = 1, \dots, n$.

Terminological relationships between simple terms in the thesaurus are set by considering relationships among synsets in WordNet, as shown in Table 1. In addition, for each composite term t we follow the intuition in [12] and add to the thesaurus the relationships st_n BT t (to denote that st_n is a Broader Term than t or, equivalently, that t is a Narrower Term NT than st_n) and st_i RT $t, i = 1 \dots n-1$ (to denote that st_i and t are Related Terms). In fact, as usual in English, in a composite term t the last simple term st_n denotes the central concept represented by t , while the remaining simple terms are used to better specify the meaning of st_n .

2.2 Case study

Our approach is being exploited in the MAP Project, conceived by the Industrial Association of Brescia -AIB- in cooperation with Chamber of Commerce of Brescia, University of Brescia and University Cattolica del Sacro Cuore, aiming at giving to local firms a set of services to better know the surroundings and the impact of firms' activities on the environment, to compare industrial environmental impact with the impact of other human activities like traffic and to display all the information on interactive maps. Here we focus on this last aspect to provide application examples of the approach. Figure 1 shows a portion of the Service Ontology used for geographical information services. We can distinguish three layers of abstraction. In

the middle layer we have the Abstract Services, that are introduced to summarize the functionalities of sets of similar Concrete Services; these ones are positioned in the lower level of the ontology and are directly invocable services that implement the functionalities represented by Abstract Services. Finally, at the top layer of the ontology Subject Categories organize Abstract Services into standard available taxonomies (in particular, we considered the UNSPSC classification) to provide a topic-driven access to them; these categories are the same that are used in the UDDI Registry to classify published Concrete Services (in this sense, our Service Ontology is built upon traditional UDDI Registry structure); in particular, each Abstract Service is associated to the set of Subject Categories related to the Concrete Services represented by that Abstract one.

In the example, we consider the Abstract Services:

- FindFirm, that, starting from a firm type and a geographical zone, returns a list of firm addresses situated in the desired zone; this service can be further specialized by providing also some photos of the desired firms (by means of FindFirmWithPhotos service); roughly speaking, we say that the FindFirmWithPhotos service is a *specialization* of the FindFirm service if the first one provides at least the same functionalities of the second one, possibly adding new functional features;

- CreateMap, that prints the geographical map of a given address; this service is specialized in the FindNearestFeatures service, that represents on a geographical map a list of requested features closer to a given address;

- FindFirmOnMap, that, starting from a firm category, represents a list of firms on a geographical map; we say that this service is composed of the FindFirm and CreateMap services, that is, roughly speaking, the functionalities of FindFirmOnMap includes the union set of functionalities of FindFirm and CreateMap; the first service is called the *composite* one, while the other ones are the *component* services.

As an example, the $SHOIN(D+)$ expression for the FindFirm service is the following:

```
FindFirm ≡
  ∃hasCategory.GeographyInformationService ⊓
  ∃hasOperation.(firmRetrieval ⊓
    ∃hasInput.(∃industrialArea.location) ⊓
    ∃hasInput.(∃type.firmType) ⊓
    ∃hasOutput.(∃firmAddresses.addressList))
```

The specialization and composition relationships constitute

Thesaurus entries	Relationships in WordNet	Relationships added to the thesaurus
st_i, st_j simple terms	st_i, st_j belongs to the same synset	st_i SYN st_j
	a hyponymy/hyponymy relationship exists between synsets of st_i, st_j	st_i BT/NT st_j
	a meronymy relationship exists between synsets of st_i and st_j or st_i and st_j are coordinate terms	st_i RT st_j
$t = \langle st_1, st_2, \dots st_n \rangle$ composite term		st_n BT t st_i RT $t, i = 1 \dots n - 1$

Table 1. Terminological relationships between the names of Domain Ontology elements.

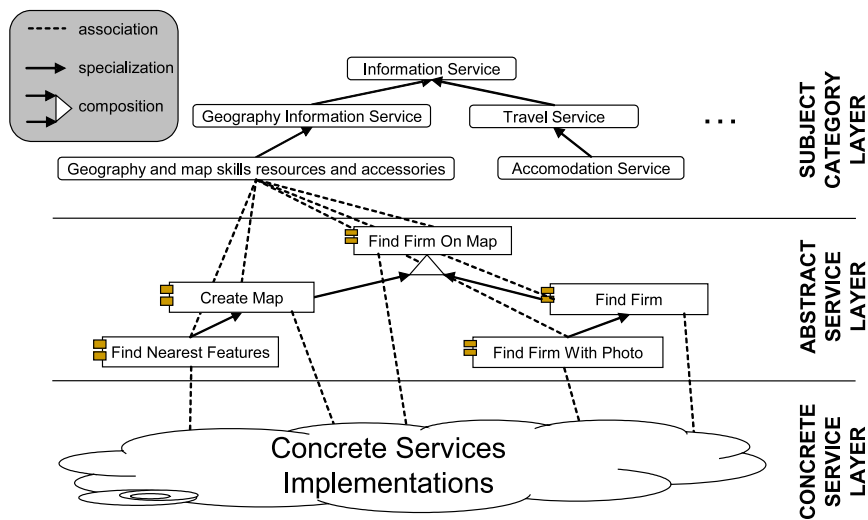


Figure 1. A portion of Service Ontology for geographical information services.

two kinds of semantic relationships between Abstract Services that will be exploited to speed up the discovery algorithm, as shown in the Section 4.

3 Semantic-enriched matchmaking

3.1 Deductive matchmaking

The thesaurus introduced in Section 2.1 is exploited to compute the *Name Affinity* coefficient between names of input/output parameters and operations. A weight $\sigma_{tr} \in [0, 1]$ is associated to each kind of terminological relationship $tr \in \{\text{SYN}, \text{BT/NT}, \text{RT}\}$ in the thesaurus, in order to evaluate its implication for affinity; in particular, we have $\sigma_{\text{SYN}} > \sigma_{\text{BT/NT}} > \sigma_{\text{RT}}$ (in our experimentation, $\sigma_{\text{SYN}} = 1$, $\sigma_{\text{BT/NT}} = 0.8$ and $\sigma_{\text{RT}} = 0.5$). The thesaurus can be viewed as a matrix, where each cell is represented in the form $\langle (t, t'), tr, \sigma_{tr} \rangle$, where t, t' are two entries of the thesaurus (respectively, the source and the target terms of the relationship tr). Two terms can be related by one or more chains of terminological relationships: we call *path* of length l between two terms t, t' , denoted with $t \rightarrow^l t'$, a finite ordered sequence of terminological relationships $\langle tr_1, tr_2, \dots, tr_l \rangle$, where the source term of tr_1 is t and the target term of tr_l is t' . The *strength* of $t \rightarrow^l t'$ is the product of the weights of all the relationships belonging to $t \rightarrow^l t'$, that is, $\tau(t \rightarrow^l t') = \prod_{k=1}^l (\sigma_{tr_k}) \in [0, 1]$. Since between two terms in the thesaurus there can exist more than one path, the one with the highest strength is chosen. The *Name Affinity* coefficient between t and t' , denoted by $NA(t, t')$, is computed as follows:

$$NA(t, t') = \begin{cases} 1 & \text{if } t = t' \\ \max_l(\tau(t \rightarrow^l t')) & \text{if } t \neq t' \wedge \\ & t \rightarrow^l t', l \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

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.

We introduce a definition of non standard subsumption between concepts based on the semantic relationships in $Dom\mathcal{ONT}$ and name affinity according to the thesaurus \mathcal{TH} . Given a pair of concepts C and D with names c_n and d_n , respectively, we say that C is subsumed by D with respect to \mathcal{TH} , denoted by $C \sqsubseteq_{\mathcal{TH}} D$, if and only if one of the following conditions holds:

- $C, D \in Dom\mathcal{ONT}$ and $(C \sqsubseteq D)$ is satisfied in $Dom\mathcal{ONT}$, where \sqsubseteq denotes the usual subsumption in Description Logics;
- only $C \in Dom\mathcal{ONT}$ and $GC(C) \cap D_{\mathcal{TH}} \neq \emptyset$, where $GC(C) = \{X \in Dom\mathcal{ONT} \mid C \sqsubseteq X\}$ and $D_{\mathcal{TH}}$ is the set of terms that have name affinity with d_n , that is, $D_{\mathcal{TH}} = \{y \in \mathcal{TH} \mid (d_n \sim y)\}$;
- only $D \in Dom\mathcal{ONT}$ and $SC(D) \cap C_{\mathcal{TH}} \neq \emptyset$, where $SC(D) = \{X \in Dom\mathcal{ONT} \mid X \sqsubseteq D\}$ and $C_{\mathcal{TH}}$ is the set of terms that have name affinity with c_n , that is, $C_{\mathcal{TH}} = \{y \in \mathcal{TH} \mid (c_n \sim y)\}$;
- if $C, D \notin Dom\mathcal{ONT}$, there exists name affinity between c_n and d_n , that is, $(c_n \sim d_n)$.

Obviously, $C \equiv_{\mathcal{TH}} D$ holds if and only if $C \sqsubseteq_{\mathcal{TH}} D$ and $D \sqsubseteq_{\mathcal{TH}} C$. This kind of subsumption checking between concepts is exploited to verify the kind of match between the Description Logic expressions that describe an advertisement \mathcal{S} and a request \mathcal{R} . Following an approach similar to the one exposed in [9], we define five kinds of matches, that will be described in more details. To verify the kind of match, service description components are considered separately. Firstly, we consider the service categories $CAT_{\mathcal{R}}$ of the request and $CAT_{\mathcal{S}}$ of the offer and we verify if $CAT_{\mathcal{R}} \sqsubseteq_{\mathcal{TH}} CAT_{\mathcal{S}}$. If this condition is not verified, then the match fails, otherwise the other kinds of matches are investigated.

Exact match denotes that \mathcal{R} and \mathcal{S} present the *same* functionalities, that is, what is required by \mathcal{R} is exactly provided by \mathcal{S} . Formally, we say that

$$\begin{aligned} match(\mathcal{R}, \mathcal{S}) = \text{'exact'} \Leftrightarrow \\ \forall op_{\mathcal{R}}^i \exists op_{\mathcal{S}}^j \text{ such that} \\ (\forall in_{\mathcal{S}}^{jk} \exists in_{\mathcal{R}}^{ih} : in_{\mathcal{S}}^{jk} \equiv_{\mathcal{TH}} in_{\mathcal{R}}^{ih}) \wedge \\ (\forall out_{\mathcal{R}}^{ih} \exists out_{\mathcal{S}}^{jk} : out_{\mathcal{R}}^{ih} \equiv_{\mathcal{TH}} out_{\mathcal{S}}^{jk}) \wedge \\ (Name_op_{\mathcal{R}}^i \equiv_{\mathcal{TH}} Name_op_{\mathcal{S}}^j). \end{aligned}$$

Note that this is a very strong condition. Each operation of \mathcal{R} is compared with each operation of \mathcal{S} . In the match-making process (for each kind of match) we require that for each comparison between two operations (respectively, between corresponding parameters of two operations) when a kind of match is established for a pair of operations (respectively, corresponding parameters) such operations (respectively, parameters) do not participate in further comparisons.

Plug-in match, that is, \mathcal{S} presents a plug-in match with \mathcal{R} when \mathcal{S} presents *at least* the functionalities of \mathcal{R} (what is required by \mathcal{R} is provided by \mathcal{S}), but \mathcal{S} provides also other capabilities or information that are not required. Formally, we say that

$$\begin{aligned} \text{match}(\mathcal{R}, \mathcal{S}) = \text{'plug-in'} \Leftrightarrow \\ \forall op_{\mathcal{R}}^i \exists op_{\mathcal{S}}^j \text{ such that} \\ (\forall in_{\mathcal{R}}^{jk} \exists in_{\mathcal{S}}^{ih}: in_{\mathcal{R}}^{ih} \sqsubseteq_{\mathcal{TH}} in_{\mathcal{S}}^{jk}) \wedge \\ (\forall out_{\mathcal{R}}^{ih} \exists out_{\mathcal{S}}^{jk}: out_{\mathcal{R}}^{ih} \sqsubseteq_{\mathcal{TH}} out_{\mathcal{S}}^{jk}) \wedge \\ (Name_op_{\mathcal{R}}^i \sqsubseteq_{\mathcal{TH}} Name_op_{\mathcal{S}}^j) \end{aligned}$$

Subsume match, that is, \mathcal{S} presents a subsume match with \mathcal{R} when \mathcal{S} presents *some but not all* the functionalities of \mathcal{R} . Formally, we say that

$$\begin{aligned} \text{match}(\mathcal{R}, \mathcal{S}) = \text{'subsume'} \Leftrightarrow \\ \forall op_{\mathcal{S}}^j \exists op_{\mathcal{R}}^i \text{ such that} \\ (\forall in_{\mathcal{R}}^{ih} \exists in_{\mathcal{S}}^{jk}: in_{\mathcal{S}}^{jk} \sqsubseteq_{\mathcal{TH}} in_{\mathcal{R}}^{ih}) \wedge \\ (\forall out_{\mathcal{S}}^{jk} \exists out_{\mathcal{R}}^{ih}: out_{\mathcal{S}}^{jk} \sqsubseteq_{\mathcal{TH}} out_{\mathcal{R}}^{ih}) \wedge \\ (Name_op_{\mathcal{S}}^j \sqsubseteq_{\mathcal{TH}} Name_op_{\mathcal{R}}^i) \end{aligned}$$

Intersection match denotes that \mathcal{R} and \mathcal{S} have *some common* functionalities, that is, there are some capabilities or information that are required, but not provided and some capabilities or information that are provided, but not required (this is the most general case). Formally, we say that

$$\begin{aligned} \text{match}(\mathcal{R}, \mathcal{S}) = \text{'intersection'} \Leftrightarrow \\ \exists op_{\mathcal{R}}^i, op_{\mathcal{S}}^j \text{ such that} \\ (\exists in_{\mathcal{S}}^{jk}, in_{\mathcal{R}}^{ih}: in_{\mathcal{S}}^{jk} \sqsubseteq_{\mathcal{TH}} in_{\mathcal{R}}^{ih} \vee \\ in_{\mathcal{R}}^{ih} \sqsubseteq_{\mathcal{TH}} in_{\mathcal{S}}^{jk}) \vee \\ (\exists out_{\mathcal{R}}^{ih}, out_{\mathcal{S}}^{jk}: out_{\mathcal{R}}^{ih} \sqsubseteq_{\mathcal{TH}} out_{\mathcal{S}}^{jk} \vee \\ out_{\mathcal{S}}^{jk} \sqsubseteq_{\mathcal{TH}} out_{\mathcal{R}}^{ih}) \vee \\ (Name_op_{\mathcal{R}}^i \sqsubseteq_{\mathcal{TH}} Name_op_{\mathcal{S}}^j \vee \\ Name_op_{\mathcal{S}}^j \sqsubseteq_{\mathcal{TH}} Name_op_{\mathcal{R}}^i) \end{aligned}$$

If all the previous comparisons fail, then the match fails (mismatch). A qualitative ranking among the kinds of matches can be considered, that is, $\text{exact} > \text{plug-in} > \text{subsume} > \text{intersection} > \text{mismatch}$. To verify these kinds of matches, a DL-based reasoner is used (RACER [8]). Note that, from the request viewpoint, the first two kinds of matches could be considered equivalent, since in both cases the offer fulfils the request; in the case of subsume and intersection match, otherwise, the offer satisfies only partially the request.

Example 3.1 Let consider the request to search for metallurgy firms in the TrompiaValley zone; the response to the request should provide a geographical map with the firm locations and the list of their addresses. Using a SHOIN($\mathcal{D}+$) expression, the request looks like:

```
FindMetallurgyFirms ≡
  ∃hasCategory.GeographyInformationService ⊓
  ∃hasOperation.(firmSearching ⊓
    ∃hasInput.(∃area.TrompiaValley) ⊓
    ∃hasInput.(∃type.MetallurgyFirm) ⊓
    ∃hasOutput.(∃map.GeographicalFirmMap) ⊓
    ∃hasOutput.(∃firmAddresses.addressList))
```

If we compare this expression with the SHOIN($\mathcal{D}+$) expression used for the FindFirm service given in the Section 2.2 we see that all the inputs required from the FindFirm service are effectively provided by the request, while there is a required output ($\exists\text{hasOutput}.\text{(\exists map.GeographicalFirmMap)}$) that is not provided. Thus, there is a subsume match between the request and the offer. \square

3.2 Similarity-based matchmaking

The *name affinity* introduced in Section 3.1 is exploited to compute the *similarity coefficients* shown in Table 2. The *Entity-based similarity* coefficient $ESim$ takes into account the I/O parameters of the overall service descriptions to measure how much the two services are based on the same information. The *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. Finally, $ESim$ and $FSim$ are combined in the *Global similarity* coefficient $GSim$. If the value of $GSim$ is equal or greater than a threshold δ given by experimental results, \mathcal{R} and \mathcal{S} are considered *similar*. These coefficients are based on the Dice's formula and widely experimented in [4]. They are more oriented towards a comparison between services and follow the structure of a service description in terms of operations and corresponding I/O parameters better than traditional IR metrics, where service descriptions are simply viewed as vectors of terms.

Example 3.2 Let consider the request and the offer compared in the Example 3.1. If we compute the similarity coefficients in Table 2, we have:

$$ESim(\mathcal{R}, \mathcal{S}) = \frac{2 \cdot (1+1)}{4} + \frac{2-1}{3} = 1.667$$

ENTITY-BASED SIMILARITY
$ESim(\mathcal{R}, \mathcal{S}) = \frac{2 \cdot A_{tot}(IN_{\mathcal{R}}, IN_{\mathcal{S}})}{ IN_{\mathcal{R}} + IN_{\mathcal{S}} } + \frac{2 \cdot A_{tot}(OUT_{\mathcal{R}}, OUT_{\mathcal{S}})}{ OUT_{\mathcal{R}} + OUT_{\mathcal{S}} } \in [0, 2]$ <p> $IN_{\mathcal{R}}, IN_{\mathcal{S}}$ - 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} $A_{tot}()$ - sum of values of the <i>Name Affinity</i> coefficient for all the pairs of parameter names that have affinity; each parameter name participates at most in one $A_{tot}()$ evaluation </p>
OPERATION SIMILARITY
$OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j) = NA(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j) + \frac{2 \cdot A_{tot}(IN_{\mathcal{R}}^i, IN_{\mathcal{S}}^j)}{ IN_{\mathcal{R}}^i + IN_{\mathcal{S}}^j } + \frac{2 \cdot A_{tot}(OUT_{\mathcal{R}}^i, OUT_{\mathcal{S}}^j)}{ OUT_{\mathcal{R}}^i + OUT_{\mathcal{S}}^j } \in [0, 3]$ <p> $op_{\mathcal{R}}^i, op_{\mathcal{S}}^j$ - names of i-th operation of \mathcal{R} and j-th operation of \mathcal{S} $IN_{\mathcal{R}}^i, IN_{\mathcal{S}}^j$ - 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} </p>
FUNCTIONALITY-BASED SIMILARITY
$FSim(\mathcal{R}, \mathcal{S}) = \frac{2 \cdot \sum_{i,j} OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j)}{ OP(\mathcal{R}) + OP(\mathcal{S}) } \in [0, 3]$ <p> $OP(\mathcal{R}), OP(\mathcal{S})$ - sets of operation names of \mathcal{R} and \mathcal{S} </p>
GLOBAL SIMILARITY
$GSim(\mathcal{R}, \mathcal{S}) = w_1 \cdot NormESim(\mathcal{R}, \mathcal{S}) + w_2 \cdot NormFSim(\mathcal{R}, \mathcal{S}) \in [0, 1]$ <p> 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 $[0, 1]$ </p>

Table 2. Similarity Coefficients between service descriptions \mathcal{R} (request) and \mathcal{S} (supply).

```

FC-MATCH
(1) Inputs: a request  $\mathcal{R}$ ;
(2)    $ServONT; DomONT; TH$ ;
(3)   a threshold  $\delta$ ;
(4) Outputs: a list  $Results$  of candidate concrete services  $CS_S$ 
      with  $GSim(\mathcal{R}, CS_S) \geq \delta$ , ranked with respect to values of  $GSim()$ ;

(5) begin
(6)   load from  $ServONT$  into  $\mathcal{AS}$  the list of abstract services;
(7)   Candidates :=  $\emptyset$ ; Results :=  $\emptyset$ ;
(8)   foreach  $S \in \mathcal{AS}$  do
(9)     if(SUBSUMPTION-CHECKING( $CAT_{\mathcal{R}}, CAT_S, DomONT, TH$ ) == true) then
(10)    if( $S \notin$  Candidates) then
(11)     remove  $S$  from  $\mathcal{AS}$ ;
(12)      $TypeOfMatch_S :=$  CLASSIFY-MATCH( $\mathcal{R}, S, DomONT, TH$ );
(13)     if( $TypeOfMatch_S <>$  'mismatch') then
(14)       $GSim(\mathcal{R}, S) :=$  EVALUATE-SIMILARITY( $\mathcal{R}, S, DomONT, TH$ );
(15)      add  $\langle S, TypeOfMatch_S, GSim(\mathcal{R}, S) \rangle$  to Candidates;
(16)      foreach  $S_i \in \mathcal{AS}$  do
(17)       if  $S_i$  is a specialization of  $S$  in  $ServONT$  AND
          ( $TypeOfMatch_S =$  'exact' OR  $TypeOfMatch_S =$  'plugin') then
(18)         $TypeOfMatch_{S_i} :=$  'plugin';
(19)         $GSim(\mathcal{R}, S_i) := GSim(\mathcal{R}, S)$ ;
(20)        add to Candidates the tuple  $\langle S_i, TypeOfMatch_{S_i}, GSim(\mathcal{R}, S_i) \rangle$ ;
(21)        remove  $S_i$  from  $\mathcal{AS}$ ;
(22)       endif
(23)      endforeach
(24)     else
(25)      foreach  $S_i \in \mathcal{AS}$  do
(26)       if  $S_i$  is a generalization of  $S$  in  $ServONT$  then
(27)        remove  $S_i$  from  $\mathcal{AS}$ ;
(28)       endif
(29)      endforeach
(30)     endif
(31)    endif
(32)   endforeach
(33)   foreach  $\langle S, TypeOfMatch_S, GSim(\mathcal{R}, S) \rangle \in$  Candidates do
(34)    foreach concrete service  $CS_S$  associated with  $S$  in  $ServONT$  do
(35)     add  $\langle CS_S, TypeOfMatch_S, GSim(\mathcal{R}, S) \rangle$  to Results;
(36)    endforeach
(37)   remove from Results the tuples with similarity  $< \delta$ ;
(38)   order tuples in Results on the basis of similarity values;
(39)   return results;
(40) end

```

Figure 2. The FC-MATCH hybrid matchmaking algorithm.

$$OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j) = 1 + 1.667 = 2.667$$

$$FSim(\mathcal{R}, \mathcal{S}) = OpSim(op_{\mathcal{R}}^i, op_{\mathcal{S}}^j) = 2.667$$

$$GSim(\mathcal{R}, \mathcal{S}) = 0.5 \cdot \frac{1.667}{2} + 0.5 \cdot \frac{2.667}{3} = 0.862 \quad \square$$

4 Hybrid matchmaking

The deductive and similarity-based matchmaking approaches are combined together to enhance precision and flexibility of the service discovery process. The algorithm for hybrid matchmaking is shown in Figure 2. The solution is primarily based on the functional comparison of service descriptions, so we called the algorithm FC(Functional Comparison)-MATCH. The algorithm starts with some initialization instructions (rows 6-7), loading from the service ontology the list of abstract services that constitute the available services to be compared with the request. After performing subsumption checking $CAT_{\mathcal{R}} \sqsubseteq_{\mathcal{TH}} CAT_{\mathcal{S}}$ between service categories (SUBSUMPTION-CHECKING function on row 9), the comparison is performed in three steps (rows 10-30). Firstly, through the deductive approach, the kind of match between the request \mathcal{R} and each available service \mathcal{S} is classified (CLASSIFY-MATCH function on row 12), according to the five kinds of matches proposed in Section 3.1. Similarity evaluation is then applied (EVALUATE-SIMILARITY function on row 14) to further refine and quantify the functional similarity between \mathcal{R} and \mathcal{S} , according to the following rules:

- if `exact` or `plug-in` match occurs, from the request viewpoint the service \mathcal{S} provides all the required functionalities, so $GSim(\mathcal{R}, \mathcal{S})$ is set to 1 (full similarity) without directly computing the similarity coefficients;
- if `mismatch` occurs, $GSim(\mathcal{R}, \mathcal{S})$ is directly set to zero;
- if `subsume` or `intersection` match occurs, the offer fulfils the request only partially and similarity coefficients are computed to quantify how much the service \mathcal{S} satisfies the request \mathcal{R} ; in this case, $GSim(\mathcal{R}, \mathcal{S}) \in (0, 1)$.

Furthermore, semantic relationships between concrete services, abstract services and subject categories in the Service Ontology are exploited to make more efficient the service discovery procedure (rows 17-22 and 26-27). In fact, the discovery algorithm is applied at the category and abstract layers and can be made faster by considering the semantic relationships between abstract services, according

to the following intuition: if an abstract service \mathcal{S}_a matches with a given service request \mathcal{R} , then also abstract services that provide the same functionalities of \mathcal{S}_a (as expressed by means of semantic relationships) match with \mathcal{R} . The resulting candidate abstract services are then replaced with their corresponding concrete ones (rows 33-36) and only concrete services that present a similarity degree equal or greater than the threshold δ are proposed as search results (row 37). Finally, the selected concrete services are ranked with respect to their similarity degree and proposed to the user (rows 38-39).

5 Conclusions and future work

The paper presents a novel approach to service discovery that combines a fully deductive matchmaking based on Description Logics and a similarity-based matchmaking derived from the information retrieval area. The approach also exploits semantic organization of services in a Service Ontology to speed up semantic discovery. The combination of two types of matchmaking strategies enhances flexibility and effectiveness of the matchmaking process, while use of Service Ontology improves efficiency. The experimentation of the approach is in progress and is being performed in different application domains.

References

- [1] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook. Theory, Implementation and Applications*. Cambridge University Press, 2003.
- [2] D. Bianchini and V. De Antonellis. Ontology-based Semantic Interoperability Tools for Service Dynamic Discovery. In *Proc. of the First Int. Conference on Interoperability of Enterprise Software Applications (INTEROP-ESA'05)*, Geneva, Switzerland, 2005.
- [3] 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*, 2004.
- [4] S. Castano and V. De Antonellis. A Framework for expressing Semantic Relationships between Multiple Information Systems for Cooperation. *Information Systems*, 123(3-4):253–277, 1998.
- [5] S. Castano, A. Ferrara, S. Montanelli, and G. Racca. Semantic Information Interoperability in Open Net-

- worked Systems. In *Proc. of the Int. Conference on Semantics of a Networked World (ICSNW)*, in cooperation with ACM SIGMOD 2004, pages 215–230, Paris, France, June 2004.
- [6] C. Fellbaum. *Wordnet: An Electronic Lexical Database*. MIT Press, 1998.
- [7] B. Fries, M. Khalid, M. Klusch, and K. Sycara. OWLS-MX: Hybrid OWL-S Service Matchmaking. In *Proc. of First International AAI Symposium on Agents and Semantic Web*, Arlington, VA, USA, 2005.
- [8] V. Haarslev, R. Moller, and M. Wessel. *RACER User's Guide and Reference Manual Version 1.7.19*, April 2004. <http://www.sts.tu-harburg.de/~r.f.moeller/racer/>.
- [9] I. Horrocks and L. Li. A Software Framework for Matchmaking Based on Semantic Web Technology. *Special Issue on Semantic Web Services and Their Role in Enterprise Application Integration and E-Commerce*, Michael Wellman and John Riedl, editors, *Int. Journal of Electronic Commerce (IJEC)*, 2004.
- [10] The INTEROP NoE Portal. <http://www.interop-noe.org/>.
- [11] T. Kawamura, J.-A.D. Blasio, T. Hasegawa, M. Paolucci, and K. Sycara. Preliminary Report of Public Experiment of Semantic Service Matchmaker with UDDI Business Registry. In *Proc. of First Int. Conf. on Service Oriented Computing (ICSOC 2003)*, pages 208–224, Trento, Italy, 2003.
- [12] M. Lauer. *Designing Statistical Language Learners: Experiments on Noun Compounds*. PhD thesis, Department of Computing Macquarie University NSW 2109, Australia, December 1995.
- [13] The MAIS Project Home Page. <http://www.mais-project.it>.
- [14] J. Miller, S. Oundhakar, A. Patil, A. Sheth, K. Sivashanmugam, and K. Verma. 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.
- [15] S.Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, M. Mongiello, and M. Mottola. Concept abduction and contraction in Description Logics. In *Proc. of DL 2003. CEUR Electronic Workshop Proceedings*, 2003. <http://ceur-ws.org/Vol-81/>.