ABSTRACT
In this paper, we propose a flexible graph-based approach for matching composite Semantic Web services that are semantically described by OWL-S. To this end, we introduce an innovative framework that performs matches at both the atomic and structural level of services, and retrieves ranked lists of advertise services that can be used to compose complex Semantic Web services from basic ones.

Categories and Subject Descriptors
H.3 [Information Storage and Retrieval]: Online Information Services

General Terms
Algorithms

Keywords
Semantic Web, Matching Semantic Web Services, OWL-S

1. INTRODUCTION
In the Semantic Web, Web Services are semantically annotated via using concepts from domain ontologies [2] in order to facilitate and improve their automatic discovery and composition. Despite this, Semantic Web Services discovery and composition is still a challenging task, either when performed at design time or, especially, at online time. Also, since (Web service) reusability is a key concern in service-oriented architectures, discovering existing composite (Semantic Web) services plays a critical role. Starting from this main motivation, in this paper we focus the attention on composite Semantic Web services described by means of OWL-S [3], a popular and well-understood solution to support ontology-based Semantic Web. In particular, OWL-S provides different amenities for explicitly describing profile and model of a service, respectively. The service profile is mainly aimed at supporting service discovery, and it includes functional parameters of the service that are typically used for the matchmaking between different service descriptions.

The service model is instead primarily aimed at specifying composite services via describing internal components and the control flow of composite processes shaping the service itself. In a typical service discovery scenario, a query is formulated as the description of a desired service and the result is a ranked list of advertised services such that their description match the request, according to a given matchmaking algorithm that employs one or more matching criteria [4]. Hence, it follows that existing discovery methods do not differentiate between atomic and composite services. This severely reduces the accuracy of results, and introduces both false positives and false negatives [4].

To address this problem, in this paper we propose a flexible graph-based approach for matching composite Semantic Web services whose main idea consists in performing the matchmaking on the basis of the service model rather than the service profile (like in traditional initiatives – e.g. [2]), via taking advantages from mining the structure of composite services. In more detail, the proposed method performs the matchmaking on two levels. It matches both atomic services of composite processes and the way these services have been composed to create them, i.e. the structure of processes. To this end, an innovative graph-based representation of OWL-S processes that is meant to support the service matchmaking method above is introduced. To specify how component services are combined together, in OWL-S processes a set of control constructs is also provided, similarly to typical control structures found in high-level programming languages. These allow services to be executed sequentially, in parallel, conditionally or in a loop, hence typical OWL-S processes can be composed by atomic or simpler-composite services, and allow several levels of nesting [4].

2. REPRESENTING OWL-S PROCESSES VIA GRAPHS
In our proposed matchmaking approach, composite OWL-S processes are represented as graphs in order to facilitate their matching. Given a composite OWL-S process $P$, the corresponding graph-based representation, denoted as $G_P$, is derived as follows. Let $\mathcal{C}$ be the set of control constructs supported in OWL-S. Each occurrence of a control construct $C \in \mathcal{C}$ in $P$ is represented by a pair of nodes $(C_b, C_e)$ that denote its beginning and ending part, respectively. The sets of such nodes in the whole $P$ are denoted as $C_b$ and $C_e$, respectively. These nodes allow us to represent the partition of $P$ that is enclosed by $C \in \mathcal{C}$, hence to represent the nesting of processes. Moreover, each atomic service $s$ in a composite
process is represented by a graph node. For the sake of simplicity, we use the same symbol to refer both the node and the service it represents. The set of all the atomic services that are contained in $P$ is denoted as $SP$. Thus, a composite OWL-S process $P$ is finally represented by a graph $G_P = (V,E)$, with node set $V = C_e \cup C_s \cup SP$ and edge set $E \subseteq (C_e \times C_e) \cup (C_e \times SP) \cup (SP \times SP) \cup (SP \times C_s) \cup (C_s \times C_s)$. Edges in $G_P$ denote the control flow modeling $P$.

Table 1 specifies, for each OWL-S control construct [3], how edges of $G_P$ are generated. For the sake of simplicity, here we assume that atomic services are simply components inside a control construct. If a nested composite process $P'$ enclosed by a control construct $C$ instead exists, then: (a) an edge $(v,C_b)$ is added in substitution of each incoming edge $(v,s)$; (b) an edge $(C_e,v)$ is added in substitution of each outgoing edge $(s,v)$; (c) the representation of the sub-process $P'$ is computed recursively and added to $G_P$.

### 3. GRAPH-BASED MATCHING OF OWL-S PROCESSES

Consider the first phase of our service matchmaking approach, where atomic services of composite OWL-S processes are taken into account. Let $G_R$ and $G_P$ be the graph-based representations of a requested and a candidate composite service $R$ and $P$, respectively. The degree of match between two nodes $r \in V(G_R)$ and $s \in V(G_P)$, respectively, is computed based on the input and output parameters of the corresponding services $r$ and $s$, as follows. For atomic services, an offer $s$ matches a request $r$ if (a) outputs offered by $s$ match outputs requested by $r$, and (b) inputs provided by $r$ match inputs required by $s$. As a consequence, as regards inputs of $r$ and $s$, the degree of match between $r$ and $s$, denoted by $domIN(r,s)$, is computed as follows:

$$domIN(r,s) = \sum_{u \in IN_r} \sum_{v \in IN_s} \frac{\max\{sim(u,v)\}}{|IN_r| \cdot |IN_s|},$$

such that $IN_r$ and $IN_s$ denote the set of input parameters of $r$ and $s$, respectively. Similarly, as regards outputs of $r$ and $s$, we have:

$$domOUT(r,s) = \sum_{u \in OUT_r} \sum_{v \in OUT_s} \frac{\max\{sim(u,v)\}}{|OUT_r| \cdot |OUT_s|}.$$ 

In previous formulas, function $sim$ computes the similarity between two singleton input or output parameters. In our framework, we make use of two distinct alternatives for defining this function. The first one consists in comparing the corresponding parameter classes $u'$ and $v'$ in the ontology $O$ available at the OWL-S service description, and computing the similarity between $u$ and $v$ on the basis of the number of common ancestors of classes $u'$ and $v'$, as follows:

$$sim(u,v) = \frac{|\{w \in O : u' \subseteq w \subseteq \omega \land v' \subseteq w \subseteq \omega'\}|}{|\{w \in O : u' \subseteq w \subseteq \omega\} \cup |\{w \in O : v' \subseteq w \subseteq \omega'\}|}.$$  

The second alternative consists in comparing parameter names by using some popular string similarity measure, such as Cosine Similarity, Jaccard Similarity, or hybrid measures [1]. The overall degree of match between $r$ and $s$ is finally computed by combining partial degrees of input and output matches.

Consider the second matchmaking phase, where the internal structure of composite OWL-S processes is taken into account. Given two graph-based representations of composite OWL-S processes $G_1$ and $G_2$, we measure the match between $G_1$ and $G_2$ in terms of their overlap via computing their Maximum Common Subgraph (MCS). This matchmaking approach has been inherited by related application contexts such as searching and mining databases of chemical structures, pattern recognition and so forth. The MCS of $G_1$ and $G_2$ is defined as the largest subgraph of $G_1$ that is isomorphic to a subgraph of $G_2$. Based on this concept, we define the degree of match between a query graph $G_R$ and a candidate graph $G_P$, denoted by $domG(G_R,G_P)$, as follows:

$$domG(G_R,G_P) = \sum_{v \in V_G} domG(v,f(v)),$$

such that: (i) $G$ denotes the MCS of $G_R$ and $G_P$; (ii) $V_G$ denotes the set of nodes of $G$; (iii) $f$ denotes a bijective function that maps nodes of $G$ on nodes of $G_P$; (iv) $domG$ denotes a function that computes the degree of match between pairs of graph nodes; (v) $|G_R|$ denotes the number of nodes of $G_R$.

### 4. CONCLUSIONS

We have proposed a flexible graph-based method for matching composite Semantic Web Services based on OWL-S. In contrast to existing approaches, which deal with matching atomic services only, our innovative contribution consists in exploring the internal structure of composite services in order to perform matchmaking at the model level.

This research has been co-funded by the European Commission within the eContentplus targeted project OpenScout, grant ECP 2008 EDU 428016.

### 5. REFERENCES


