Behavioral Similarity of Semantic Web Services*

Zijie Cong and Alberto Fernández

CETINIA, Universidad Rey Juan Carlos, Madrid, Spain
zijie@ia.urjc.es, alberto.fernandez@urjc.es

Abstract. Service matchmaking is an integral link of service discovery, composition, invocation and other similar task under Service-Oriented Architecture (SOA). Most current approaches measure the degree of match of two service based merely on their I/O pairs which could leads to false result. This paper presents an approach for matchmaking in Semantic Web Services (SWS) that considers each service as a sub-graph of semantic network, which is formed by inputs, outputs, pre-conditions and post-condition, with contribution of syntactical information such as keywords from the service description. Thus the similarity between services is defined as the similarity between two sub-graphs. The aim of this approach is to reveal the internal work flow and intention of services, i.e. behaviors, thus it agrees with human intuition to a larger extent than previous approaches.

1 Introducction

The original intention of adding semantic annotations to web services is to improve the automation of services discovery, selection, invocation and interoperation by letting service descriptions to be machine-processable [7]. One integral part of such automation is matchmaking among services.

Various approaches have been proposed in previous studies. Without concerns about semantics of its components, one primitive method to calculate the similarity of services is based on the syntactical information - e.g. keywords, tag-clouds and textual descriptions.

For services with semantic information, inputs/outputs (I/O) matching is a common method for measuring the similarity. Inputs and outputs of a semantic service are instances of ontological concepts, the similarity of two services is determined by the minimal distance in the taxonomy tree between corresponding concepts of I/O pair, the result is a degree of semantic simialrity, such as exact, plug-in and subsumes [6]. Some studies, such as [5], aimed to achieve higher robustness and precision by combining both.

More recently, various graph based approaches have been proposed. In [4], a service was considered as a composition of processes and thus could be represented as a finite-state machines (FSMs), the similarity between services was

* Work partially supported by the Spanish Ministry of Science and Innovation through grants TIN2009-13839-C03-02 and CSD2007-0022(CONSOLIDER-INGENIO 2010)

defined as the similarity between two FSMs. Like other similar graph-based approaches [[3,2],[3,2]], it concentrated on structural similarity of services instead of the semantic similarity of atomic units of functionality.

This paper presents a novel but preliminary approach for service matchmaking. The main notion behind this approach is that a service could be considered as a sub-graph of a semantic network, which maps input concepts to output concepts via elements specified in preconditions, post-conditions or retrieved from textual description, it reveals the behavior of service that could be a more intuitive option for calculating the degree of match of services.

2 Motivation

Although an appropriate measurement of degree of match is difficult to define, it is consensus that the result of matching should agree with human intuition. Inputs and outputs sometimes may not provide sufficient information about service's behavior, and replying solely on them may lead to false result. An example is presented in the rest of this section.

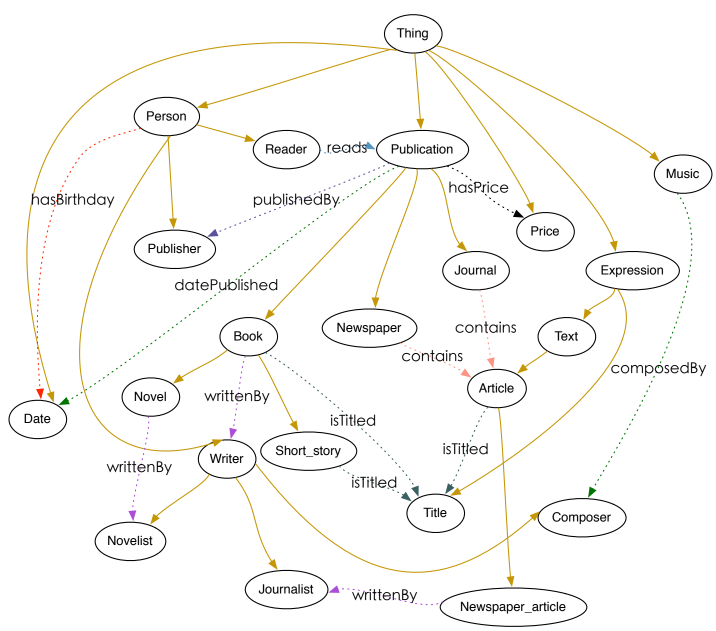

Fig. 1. An ontology of publications with 22 concepts and 10 relations, brown solid lines represent subsumption relations.

Figure 1 [1] illustrates an ontology of publication1 with 22 concepts and 10 relations connecting them. Table 1 presents three services using this ontology. Every service description used in this paper is a quintuple (T; I;O; P;Q), where:

- T is the syntactical information of service, such as description, key-words, etc.
- I is a set of input concepts.
- O is a set of output concepts.
- P is a set of predicates that must be true prior to the invocation of the service, i.e. preconditions. These predicates are relations between concepts defined in ontology, such as writtenBy(Book, Writer).
- Q is a set of predicates that must be true after the execution of the service, i.e. post-conditions. Same as for P, these predicates are relations defined in ontology as well.

$$S_1 = \begin{cases} T & returns\ the\ birthday\ of\ a\ given\ novelist \\ I & \{Novelist,\ Novel\} \\ O & \{Date\} \\ P & writtenBy(Novel,\ Novelist) \\ Q & hasBirthday(Novelist,\ Date) \end{cases}$$

$$S_2 = \begin{cases} T & the\ date\ of\ publish\ of\ a\ writer's\ first\ book \\ I & \{Writer,\ Book\} \\ O & \{Date\} \\ P & writtenBy(Book,\ Writer) \\ Q & datePublished(Book,\ Date) \end{cases}$$

$$S_3 = \begin{cases} T & published\ date\ of\ a\ novelist's\ premier\ work \\ I & \{Novelist,\ Novel\} \\ O & \{Date\} \\ P & writtenBy(Novel,\ Novelist) \\ Q & \emptyset \end{cases}$$

Fig. 2. Services using ontology of publication

By using I/O matching approaches, matchmaker will not be able to distinguish between S_1 and S_3 as their inputs and outputs are identical, thus these two services matches exactly, even though the functionality of these two services are different. On the other hand, S_2 would give a lower degree of match against S_3 despite they are more similar behaviorally.

Therefore the aim of our approach is to overcome above limitations by exploiting the behavioral information of services.

[1] This ontology is partially adopted from “books.owl” of OWL-TC3

3 Service Behavioral Graph (SBG)

To exploit the behavioral information of a service, we consider a service as a function that maps its inputs to outputs. In SWS, inputs and outputs are ontological concepts, this mapping is defined by relations in the same domain ontology. As an ontology can be represented by *multi-relational graph* where each vertex denotes a concept and each edge denotes a relation between concepts, a service thus can be further considered as a sub-graph of an ontology. More formally,

Definition 1. *if* $G = (V \times E \subseteq (V \times V))$ *where* V *is the set of concepts and* E *is the set of relations of heterogeneous types, is a multi-relational graph of an ontology, then service* S *is denoted as* $G_S = \left(V^{'}, E^{'}\right)$ *where* $V^{'} \subseteq V$ *and* $E^{'} \subseteq (V^{'} \times V^{'}) \subseteq E^{'}$.

This graph is referred as *service behavioral graph* (SBG) in this paper, it can be discovered from the ontology graph using critical elements and behaviorally correct path defined in the following sections. Algorithm 1 is the pseudo-code for SBG discovery[2], details can be found in Section 3.3.

An example can be seen in Figure 3, the elements in blue of the graph depict the SBG of S_1 defined in Figure 2.

3.1 Critical Elements

We have mentioned in the beginning of this section that the mapping from inputs to outputs is defined by the relations in the domain ontology, this mapping is, in fact, a set of paths from input concepts to output concepts, consisting one or more relations. There may exist multiple paths between a pair I/O concepts, therefore, finding proper paths is critical for describing the the service's behavior correctly.

Such paths is determined by several components in the ontology, which can be concepts or relations, they are referred as *critical elements* in this paper. P(*Precondition*), Q(Post-condition) and *T(Syntactical information)* of service description may offer some clues to these critical elements.

Syntactical Information Syntactical information is valuable for revealing services' behaviors. For example, even though $S_{1(I,O)} = S_{3(I,O)}$, the textual descriptions *(T)* differ these two services at human-readable level. To find the critical elements, syntactical information and ontological components' identifiers (besides those in I/O sets) need to processed using Information Retrieval techniques to transform into a set of keywords with irrelevant words and morphological variants removed. Then components with keywords appeared in the syntactical information of the service is considered to be a critical element. For example, in S_2, relation *datePublish,* concepts *Date,*

[2] We do not concentrate on finding the shortest behaviorally correct path in this paper, various existing approaches on shortest path problem can be adopted with minimal effort.

Publication, Writer and *Book* are identified as a critical elements as the word "publish", "date", "writer" and "book" have appeared in $S_{2(T)}$.

Preconditions The preconditions is a set of predicates that must be true before the service can be invoked. It is not concerned with the behavior of the service, but indicate the relations among input elements. An *inputs* sub-graph of service behavioral graph can be formed. For example, in S_2, $(Book, writtenBy, Writer)$ is the *inputs* sub-graph of service behavioral graph.

Post-conditions The post-conditions is a set of predicates that must be true after the execution of the service. These predicates often connect input elements with output elements, hence reveal valuable information.

3.2 Behaviorally Correct Path (BCP)

To connect inputs with outputs, a path containing critical elements defined in the previous section needs to be find, we refer this path as a *behaviorally correct path (BCP).*

In semantic networks, concepts are usually connected by heterogeneous links, including hierarchical relations as well as other relations. There may exist multiple paths with same length (in term of number of edges) from on concept to another. For similarity measuring purpose, it is necessary to have unique path between two elements, and such path should not only contain the critical elements, but also be semantically correct.

In [1], Aleksovski et al. considered a path to be semantically correct if and only if no hierarchical links appear after a non-hierarchical one. For example, in Figure 1, a path *{ShortStory, is_ a, Book, writtenBy, Writer}* is semantically correct, while *{ShortStory, is_ a, Book, writtenBy, Writer, is_ a, Person}* is not.

In practice, however, there is a high possibility that no semantically correct path exists between two concept using Aleksovki's definition. Therefore, for the purpose of this paper, we define a behaviorally correct path as:

Definition 2. A behaviorally correct path is a path in semantic network between two concepts containing critical elements with maximum one turn from non-hierarchical relation to hierarchical relation.

And two assumptions must be hold to ensure the existence of a BCP:

1. Any relation in an ontology is at least partially symmetric.

This assumption implies that if a relation exists between two concepts, then there also exists an inverse relation between same two concepts which is at least partially symmetric.

2. All relations are inheritable from a super-concept to a sub-concept.

This assumption implies that if there exists a relation p between concepts x and y, i.e. *p(x, y)*, and *is_ a(z, x)*, then *p(z, y)*. This eliminates sequence of subsumption relation that might be appeared in the beginning of a BCP and also reduces the length of BCP.

Together, Definition 2, Assumption 1 and 2, ensure that there always exist a behaviorally correct path between two concepts.

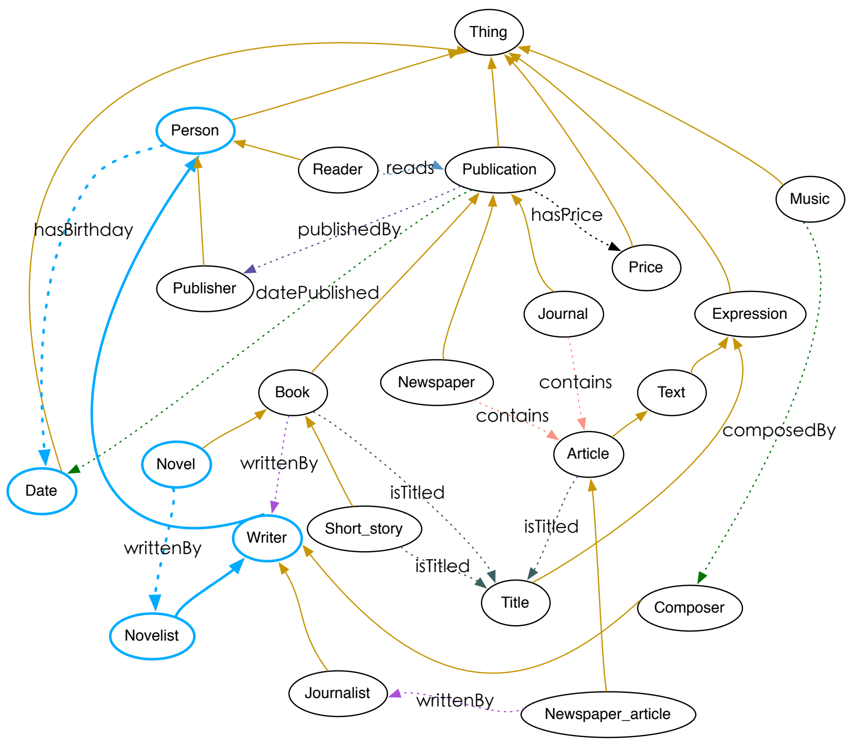

Fig. 3. SBG of service that returns a novelist's birthday in blue

3.3 Algorithm

Algorithm 1 presents how a SGB is discovered. In line 3, inputGraph denotes a graph formed by input concepts and preconditions. CriticalElements in line 4 determines the critical elements described in Section 3.1, this procedure can be implemented using IR techniques. Variable P at line 6 denotes a set of paths from a input concept to output concept via various relations, these concepts and relations are elements of the set of critical elements CES. The final result SBG is a union of all shortest paths connecting critical elements and inputGraph.

4 Calculating Similarity

The similarity of services is defined as the similarity of their corresponding SBGs. As sub-graphs of a semantic network, SBGs are multi-relational graphs, which can be represented by binary 3-way tensor.

A tensor is an object that extends the notion of scalar, vector and matrix to higher orders. A single-relational graph has representation of a adjacent matrix

Algorithm 1 Algorithm for SBG discovery.

procedure SBG(S) ▷ S is the service description quintuple
  $SBG \leftarrow \emptyset$
  $inputGraph = (S_I, S_P)$
  $CES \leftarrow$ CRITICALELEMENTS(S_P, S_T, S_Q) ▷ critical elements
  for $\forall i \in CES \cap S_I,\ R = CES \cap (S_P \cup S_Q),\ o \in CES \cap S_O$ **do**
    $P \leftarrow \mathrm{BCP}(i, r, o)$ ▷ set of behaviroally correct paths
    $SBG \leftarrow (SBG \cup (\underset{p \in P}{argmin}(\mid p \mid))$ ▷ shortest path
  end for
  return $SBG \cup InputGraph(S)$
end procedure

that can be seen as a 2-way tensor, if we consider a multi-relational graph as a union of multiple single-relational graphs, it thus can be represented using a 3-way tensor. A tensor representation of an ontology with n concepts and m relations is

$$A \subseteq \{0, 1\}^{n \times n \times m}$$

where

$$A^k_{i,j} = \begin{cases} 1 & if\ (i, j) \subseteq E_k, k < m \\ 0 & otherwise \end{cases}$$

Figure 4 illustrates such tensor in a visualized form.

An intuitive approach for calculating the similarity between two tensors is subtraction, as two services share same ontology, the order of tensors of services are equal. The result of subtraction will be a tensor of symmetric difference of two SBGs. Under service discovery scenario, a service request is compared against a candidate service advertisement:

$$SBGDiff(S^R, S^A) = SBG(S^R) - SBG(S^A) \quad (1)$$

$$inputGraphDiff(S^R, S^A) = InputGraph(S^R) - InputGraph(S^A) \quad (2)$$

where S^R and S^A are service request and service advertisement tuples respectively. Since the SBGs and InputGraphs are binary three-way tensor, the result tensor is $R = (-1, 1, 0)^{n \times n \times m}$ where each -1 indicates an element appear in service advertisement only, 0 indicates an common elements and 1 indicates an element appear in service request only.

We further define six degrees of match based on this resulting tensor. Let D denotes the symmetric difference tensor between SBGs of service request S^R and service advertisement S^A; G denotes the symmetric difference tensor of InputGraphs. The degrees of match of a service request and service advertisement are,

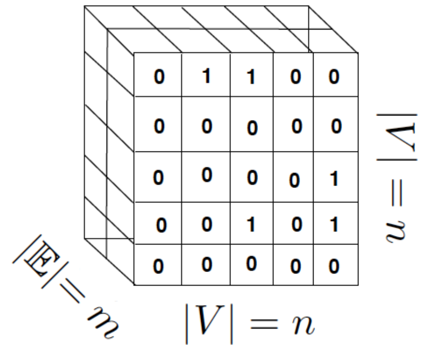

Fig. 4. 3rd-order tensor representation of an ontology with n concepts and m relations

Exact S^R EXACTLY matches S^A $\iff$ $\forall d \in D : d = 0$. This degree indicates that two services match exactly at both behavioral and structural level.

Tail-match S^R is TAIL-MATCH with S^A $\iff$ $\forall d \in D : d \geq 0 \wedge d \in G$. This degree indicates that service request provides extra excessive inputs but matches with service advertisement's behaviors and outputs.

Head-Match S^R is HEAD-ALIGNED with S^A $\iff$ $\forall g \in G : g = 0 \wedge \forall d \in D : d \leq 0 \wedge \forall o_r \in S^R_O \exists o_a \in S^A_O : o_r > o_a$. This degree indicates that service request requires only a subset of advertisement's outputs but matches its behaviors and inputs.

SubgraphOf S^R is a sub-graph OF S^A $\iff$ $\forall d \in D : d \leq 0$. This degree indicates that the SBG of service request is a sub-graph of service advertisement, which cannot be invoked directly but might be padded through service composition.

Subsumes S^R SUBSUMES S^A $\iff$ $\forall d \in D : d \geq 1$. This degree implies that the service advertisement is a subgraph of request, invocation might be done after service composition.

Fail S^R does not matches S^A.

The service advertisements that match with service request on first three degrees are invokable while the following two degrees, subgraphof and subsumes, require extra units of functionality,i.e. services, to be participated.

5 Conclusion and Future work

This paper presents a novel but preliminary approach of calculating the similarity between two services. This approach intends to reveal the behavioral information of services, and by comparing their similarity to achieve higher accuracy, robustness and in agreement with human intuition. The main notion behind this approach is that we consider a service as a sub-graph of semantic network that connects its inputs concepts and output concepts via critical elements, referred as Service Behavioral Graph (SBG). We use syntactical information and conditions to determine the critical elements, and a SBG is discovered by exploiting these elements. The similarity of services is defined as six degrees of match in this paper based on the differences between two SBGs. In practise, services do not often belong to the same ontology, alignment needs to be performed in case of multiple ontologies are participated in matchmaking.

Experiments with actual realistic test cases are necessary to access the practicability of our approach. One expectable limitation of our approach is that it depends on the quality (in term of richness) of the ontology to a large extent, which is highly unstable in practise. Our future work includes implementation, experiments and evaluation of this approach, also solving open issues such as diminish the deviation caused by the instability of the quality of ontologies and refine the degree of match.

References

1. Z. Aleksovski, W. ten Kate, and F. van Harmelen. Exploiting the structure of background knowledge used in ontology matching. In *Ontology Matching Workshop at International Semantic Web Conference (ISWC)*. Citeseer, 2006.
2. J. Corrales, D. Grigori, and M. Bouzeghoub. Bpel processes matchmaking for service discovery. *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, pages 237–254, 2006.
3. D. Grigori, J.C. Corrales, and M. Bouzeghoub. Behavioral matchmaking for service retrieval: Application to conversation protocols. *Information Systems*, 33(7-8):681–698, 2008.
4. A. Günay and P. Yolum. Structural and semantic similarity metrics for web service matchmaking. In *Proceedings of the 8th international conference on E-commerce and web technologies*, pages 129–138. Springer-Verlag, 2007.
5. M. Klusch, B. Fries, M. Khalid, and K. Sycara. Owls-mx: Hybrid owl-s service matchmaking. In *Proceedings of 1st Intl. AAAI Fall Symposium on Agents and the Semantic Web*, volume 142, 2005.
6. M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic matching of web services capabilities. *The Semantic Web (ISWC 2002)*, pages 333–347, 2002.
7. K. Sivashanmugam, K. Verma, A. Sheth, and J. Miller. Adding semantics to web services standards. In *Proceedings of the International Conference on Web Services*, pages 395–401. Citeseer, 2003.