

INFUSING SEMANTICS IN WSDL WEB SERVICE DESCRIPTIONS TO ENHANCE SERVICE COMPOSITION AND DISCOVERY

Ourania Hatzi, Mara Nikolaidou and Dimosthenis Anagnostopoulos

Department of Informatics and Telematics, Harokopio University of Athens, El. Venizelou 70, Athens, Greece
{*raniah, mara, dimosthe*}@hua.gr

Keywords: Web Services, Service Discovery, Service Composition, Semantic Web Technologies, WSDL, Semantics, Semantic Similarity, AI Planning

Abstract: Semantic information can significantly enhance web service discovery and composition in large domains, such those facilitated by cloud infrastructure. If semantics awareness is achieved, locating the appropriate web services can be performed taking into account the actual meaning of the information included in the web service descriptions. Furthermore, semantic relaxation is possible; in such cases, approximate solutions can be found. In order to perform semantic relaxation, the semantically equivalent and similar concepts to the original concepts describing the web services have to be identified and their semantic distance has to be evaluated. This paper explores the semantics infusion in cases where such information is not inherent in web service descriptions, that is, in the prominent web service description standard WSDL. Based on semantically enhanced WSDL descriptions, a case where semantic relaxation is performed for web service composition through AI planning is presented, and the way the resulting approximate composite services can be assessed in terms of semantic distance is also discussed.

1 INTRODUCTION

Web service provide interoperability between diverse, heterogeneous systems over a network in the form of a "black box", that is, the internal representation and implementation of a software system offering its functionality as a web service is hidden. Interoperability is achieved by providing a well-defined, standard interface that describes, among others, the web service operations functionality, the required inputs and the produced outputs, and the ways to invoke and communicate with the web service.

As web service numbers continuously increase, and their functionality is being offered over cloud infrastructure in the form of SaaS or PaaS, the requirement for effective web service management emerges. Users and developers need to be able to locate web services based on their functionality and combine them to form intricate functionality. The automation of these processes of web service discovery and composition is essential, whether they are offered over the cloud or over the web in general.

Currently, the prominent standard for web service description is the *Web Service Definition Language* (WSDL,). WSDL aims at being machine readable; therefore, it operates at the syntactic level. Auto-

mated web service invocation is possible, even at the syntactic level, as there are tools that automatically parse WSDL descriptions and generate web service clients. However, the automation of other web service related tasks is not a trivial issue (Benatallah et al., 2005)(Dustdar and Schreiner, 2005). Such tasks include web service discovery, which is the automated process of locating a web service with specific capabilities, adhering to client-specific constraints, and web service composition, which involves the selection, combination and interoperation of simple web services in order to perform complex, intricate tasks, given a high-level definition of an objective. Such tasks are significantly facilitated and enhanced by utilising semantics. If semantics are incorporated, locating the appropriate services can be performed taking into account the actual meaning of the concepts describing the web services, even if there are syntactic differences. Moreover, semantic relaxation can take place enabling approximate solutions. For that purpose, semantic web service description languages such as OWL-S and SAWSDL have been introduced. The potential of web service discovery and composition with regard to semantic description standards has been researched and ways to incorporate and exploit the semantic information inherent to these stan-

dards have been proposed. For example, the PORSCÉ framework (Hatzi et al., 2011) provides solutions to the web service composition problem by representing it as an AI planning problem over the OWL-S profiles of the available atomic web services, and it is able to provide approximate composite services by enhancing the planning problem with semantic information acquired by the corresponding OWL ontologies.

Despite the fact that semantic standards provide the aforementioned benefits, currently the vast majority of the available web services is described in WSDL, as most Integrated Development Environments (IDEs) are capable of automatically producing such descriptions. Therefore, it is important to explore ways to enhance WSDL descriptions with semantic information, in order to benefit from the advantages it could offer during service discovery. To the best of our knowledge, there are currently no approaches that attempt to automatically infuse semantics through various sources into WSDL descriptions. SAWSDL (SAWSDL,) is an emerging standard based on WSDL, which permits semantic annotations to WSDL elements; however, the elements are annotated manually. Semantic annotations in SAWSDL reference ontologies; therefore, approaches for semantic enhancement and relaxation that are applied for the OWL-S case (Hatzi et al., 2011) can be used.

This paper presents a novel approach for incorporating semantic information from a variety of sources into WSDL descriptions, facilitating both web service discovery and composition, as well as the acquisition of approximate solutions. Semantic distance measures are also explored, in order to enable accuracy assessment, in case of approximate solutions. The presented approach has been incorporated in the PORSCÉ framework (Hatzi et al., 2011) aiming at accommodating web service composition through AI Planning, handling both semantic (OWL-S and SAWSDL) and non-semantic (WSDL) web service descriptions in a uniform way.

The rest of the paper is structured as follows: Section 2 provides the definitions for semantic concept relevance. Section 3 provides the proposed WSDL semantic analysis approach, including hierarchical relationships and semantic relevance metrics. Section 4 presents ways to semantically relax the web service composition problem, in order to acquire approximate solutions, when attempting to solve it as a planning problem. Section 5 proposes techniques to assess the quality of the approximate solutions, in terms of semantic distance. Section 6 briefly discusses the incorporation and application of the proposed approach in the PORSCÉ framework. Finally, conclusions and future directions reside in Section 7.

2 DEFINITION OF SEMANTIC CONCEPT SIMILARITY

Semantic awareness and relaxation is applied on the concepts describing the web service functionality, i.e. inputs and outputs. In order to achieve it, semantically equivalent and similar concepts have to be located and correlated with the original concepts, so that in during discovery and composition, the semantically similar concepts can be used instead of the original ones.

The notion of semantic concept similarity is independent of the underlying implementation and applies to both semantic and non semantic web service descriptions; the difference in each case lies in the source of the semantic information and the method used for calculating the semantic distance.

Intuitively, two concepts are considered semantically similar if and only if

1. they have a specific semantic relationship (including semantic equivalence), and
2. their semantic similarity, in terms of a specific semantic distance / similarity measure, exceeds a user-defined threshold, allowing the adjustment of the concept relevance criterion, enabling the incorporation of different degrees of relaxation.

Let O denote the set of the available ontology concepts, F denote the set of the selected hierarchical relations and a the concept distance threshold. The a threshold, where $a \in [0..1]$, restricts the distance of two concepts, defining the minimum similarity that is acceptable in order for the concepts to be matched.

Definition 1 A concept $C \in O$ is considered relevant to a concept $D \in O$ with respect to a hierarchical relation set F and a concept similarity threshold a , denoted as $C \sim_a^F D$, if they satisfy at least one hierarchical relation in F and the threshold a on their similarity.

Definition 2 For each concept $C \in O$ its concept relevance set R_C , is defined as the set of all the relevant concepts of C , that is, $R_C \equiv \{T \in O : T \sim_a^F C\}$.

Definition 3 For each set A of concepts, its extended set EX_A , is defined as the union of the concept relevance sets of its concepts, that is,

$$EX_A = \bigcup_{\forall C \in A} R_C.$$

Definition 4 Two concept sets A and B are relevant, denoted as $A \sim_a^F B$, if all the concepts of one set have at least one relevant concept in the other set and the two sets have the same size, that is,

$$A \sim_a^F B \Rightarrow \forall C \in A, \exists D \in B : C \sim_a^F D \wedge \forall D \in B, \exists C \in A : D \sim_a^F C \wedge |A| = |B|.$$

3 WSDL SEMANTIC ANALYSIS BACKGROUND

Web service descriptions expressed in WSDL, unlike the ones expressed in OWL-S or SAWSDL, do not contain inherent semantic information attached to the concepts that describe web service inputs or outputs. For this reason, semantic information must be acquired by other sources, such as thesauri or lexical databases. One of the largest and most commonly used lexical thesauri is WordNet (Miller et al., 1990). WordNet, among its other features, groups words into sets of semantically or lexically related words, called synsets, supporting the identification of semantically equivalent or related concepts that can be utilized in the proposed approach for semantic awareness and semantic relaxation (Fellbaum, 2010). Most synsets are connected to other synsets via semantic hierarchical relations.

WordNet noun semantic relations include:

- **hypernym:** Y is a hypernym of X if every X is a (kind of) Y (canine is a hypernym of dog)
- **hyponym:** Y is a hyponym of X if every Y is a (kind of) X (dog is a hyponym of canine)
- **coordinate term:** Y is a coordinate term of X if X and Y share a hypernym (wolf and dog are coordinate terms)
- **holonym:** Y is a holonym of X if X is a part of Y (building is a holonym of window)
- **meronym:** Y is a meronym of X if Y is a part of X (window is a meronym of building)

Semantic similarity between two concepts s_1 and s_2 can be computed by a variety of measures (Budanitsky and Hirst, 2006) (Wang and Liu, 2008).

(Leacock et al., 1998) take into account the length of the shortest path connecting two concepts *length*, measured as the number of nodes included in the path, and the maximum depth of the taxonomy D :

$$Sim(s_1, s_2) = -\log \frac{length}{2D},$$

Most measures involving common ancestors of the two concepts incorporate the *Information Content* of the deepest concept that can subsume both concepts s_0 ; that is, the *least common subsumer*:

$$IC(s_0) = -\log P(s_0),$$

where $P(s_0)$ is the probability of occurrence of the concept s_0 in a large corpus.

Such measures include (Jiang and Conrath, 1997):

$$Sim(s_1, s_2) = \frac{1}{IC(s_1) + IC(s_2) - 2 \cdot IC(s_0)},$$

and (Lin, 1998):

$$Sim(s_1, s_2) = \frac{2 \cdot IC(s_0)}{IC(s_1) + IC(s_2)}.$$

For the proposed approach, we incorporate a more sophisticated measure of semantic relatedness, called *Omiotis* (Tsatsaronis et al., 2010), which captures relatedness in multiple granularity levels, i.e. between two concepts, as well as between groups of words; therefore, it can be also used for composite service accuracy assessment. *Omiotis* calculates semantic relatedness by utilizing the semantic network constructed by taking into account all semantic relations between concepts. It considers the path length, captured by compactness, and the path depth, captured by semantic path elaboration. Semantic relatedness between two groups of words A and B is calculated as

$$Omiotis(A, B) = \frac{|\zeta(A, B) + \zeta(B, A)|}{2},$$

where

$$\zeta(A, B) = \frac{1}{|A|} \left(\sum_{a \in A} \lambda_{a, b_*} \cdot SR(a, b_*) \right),$$

$$a_* = \arg \max_{a \in A} (\lambda_{a, b} \cdot SR(a, b)) \text{ and}$$

$$b_* = \arg \max_{b \in B} (\lambda_{a, b} \cdot SR(a, b)).$$

The lexical relevance $\lambda_{a, b}$ between terms $a \in A$ and $b \in B$ is calculated the harmonic mean of the respective terms' TF-IDF values, as determined by the standard TF-IDF scheme (Salton et al., 1982):

$$\lambda_{a, b} = \frac{2 \cdot TFIDF(a, A) \cdot TFIDF(b, B)}{TFIDF(a, A) + TFIDF(b, B)}.$$

4 SEMANTIC RELAXATION

In cases when no exact solutions are available, semantic relaxation is able to provide approximate ones. In order to elaborate on this mechanism, this section will study the way semantic relaxation is performed during web service composition, when it is handled as an AI planning problem.

In order for the web service composition to be solved as a planning problem, the following steps have to be performed:

1. The descriptions of the available web services have to be transformed into a planning domain. More specifically, each available web service is transformed into a planning action A .
2. User requirements regarding the desired composite service have to be transformed into a planning problem. More specifically, the available inputs

are represented as the initial state of the planning problem I , while the desired outputs of the composite service are represented as the goal state of the planning problem G .

3. Both the planning domain and problem have to be encoded into a standard language, such as PDDL (Ghallab et al., 1998) and forwarded to a planning system for solving.
4. The solutions acquired by the external planning system, optionally, can be transformed back to a web service standard.

Successful representation of a web service composition problem as a planning problem requires the planning system to be aware of possible semantic similarities among syntactically different but semantically equivalent concepts. Semantic awareness enables the planning systems to match preconditions and effects correctly during the planning process, even if the terms used to refer to them in the web service descriptions differ (Paolucci et al., 2002).

Furthermore, in cases where no exact matching of concepts is possible, semantic analysis is able to provide, apart from equivalent concepts, semantically similar concepts as well, according to the measures of similarity defined in the semantic enhancement step. In this case, input concepts are matched to semantically similar output concepts approximately. Semantic relaxation enables the formulation of composite web services that are less accurate; nevertheless they serve the purpose of the user in the best possible way.

There are two different ways that semantic awareness and relaxation can be implemented. The first way is to alter the planning system itself so that it advises the semantic analysis module whenever required, e.g. when the applicability of an action on a given state must be determined. The second way is transparent to the planning system. It involves enhancing the planning domain and problem description with all semantic information required, at an intermediate step, so that then the planning system can handle it as if it was a classical planning problem. The proposed approach adopts the enhancement of the planning problem with semantic information, since it provides independence of the planner, so that any PDDL-compliant planner can be used.

In the intermediate step, all semantically equivalent and similar concepts for both facts of the initial state and outputs of the available actions are acquired through semantic analysis, according to *Definition 1*.

An initial approach for the translation of the original problem $\langle I, A, G \rangle$ to the enhanced problem $\langle EIS, EAS, EGS \rangle$ is based on the following rules:

- The original set of concepts in the initial state I

Algorithm 1 Computes the *Extended Action Set (EAS)* - Version 1

Input: The set A of the domain actions

Output: The extended set EAS of the domain actions

```

1   $EAS \leftarrow \emptyset$ 
2  for each action  $A_i \in A$  do
3    for each  $F_i \in add(A_i)$  do
4       $D \leftarrow theconceptrelevance\ set\ of\ F_i$ 
                                     // According to Definition 2
5      define array of actions  $XA_{[sizeof(D)]}$ 
6       $k \leftarrow 0$ 
7      for each  $d \in D$  do
8         $XA[k] \leftarrow A_i$ 
9         $add(XA[k]) \leftarrow add(XA[k]) \cup \{d\} - \{F_i\}$ 
                                     // Replace  $F_i$  with  $d$ 
10      $EAS \leftarrow EAS \cup \{XA[k]\}$ 
11      $k \leftarrow k + 1$ 
12  return  $EAS$ 

```

was replaced by its extended set (*Extended Initial State - EIS*), according to *Definition 3*: $EIS \equiv EX_I$.

- Each action produces a set of semantically similar actions (clone actions) by taking into account all possible combinations of the action effects and their relevant concepts. The original actions together with the clones constitute the *Extended Action Set (EAS)* as described in *Algorithm 1*.
- The goals of the problem (*Extended Goal Set - EGS*) remain the same: $EGS \equiv G$. Since semantic relaxation takes place for the concepts of the initial state and the outputs of each action, approximate matching is feasible without the need to alter the goals of the problem.

In the aforementioned approach, despite managing to successfully incorporate the semantic information into the planning problem, two key issues emerged. Due to most state-of-the-art planning systems not being optimal, the resulting plan in some cases contained sequences of the same action (redundant duplicates), either in its original form or in a semantically equivalent; therefore, the system had to perform additional steps to provide the user with a sound description of the composite service. Second, inclusion of the semantically similar actions in the description of the problem yields a very large set of actions, usually containing hundreds or even thousands of actions. Especially in problems with many available services, the large number of actions places an important overhead to the planning system and slows down the execution of the planning process.

Taking into account these results, the approach was optimized and the semantic relaxation step was revised using a different method, which manages to improve performance. Instead of enhancing the set of actions with clones, the Enhanced Action Set (EAS) is produced, by altering the description of each oper-

Algorithm 2 Computes the *Extended Action Set (EAS)* - Version 2

Input: The set A of the domain actions

Output: The extended set EAS of the domain actions

```

1  $EAS \leftarrow \emptyset$ 
2 for each action  $A_i \in A$  do
3    $D \leftarrow \text{theconceptsetofadd}(A_i)$ 
   // According to Definition 3
4    $\text{add}(A_i) \leftarrow D$ 
5 return  $EAS$ 

```

ator, while preserving the initial size of the set. More specifically, the effects list of each action is replaced by its extended set according to *Definition 3*, as described in *Algorithm 2*.

The above is clarified by a concrete example. Suppose that the initial state I and the operators of the problem are the following:

```

I = {debitcard(X), dates, motel}
ActivateCard: prec={creditcard(X), disabled(X)},
  effects(+)={enabled(X)}, effects(-)={disabled(X)}
BookHotel: prec={dates, hotel},
  effects(+)={bookinginfo}, effects(-)={}

```

Semantic relaxation for a given distance metric and threshold yields the following relevant concepts:

```
debitcard~creditcard, motel~hotel, active~enabled
```

The problem definition is transformed to the following:

```
EIS: {debitCard(X), dates, motel, creditCard(X), hotel}
```

```

EOS:
ActivateCard: prec={creditCard(X), disabled(X)},
  effects(+)={enabled(X), active(X)},
  effects(-)={disabled(X)}
BookHotel: prec={dates, hotel},
  effects(+)={bookinginfo}, effects(-)={}

```

The new problem $\langle EIS, EAS, EGS \rangle$ can be consequently forwarded to external planning systems in order to acquire solutions.

5 COMPOSITE SERVICE ACCURACY ASSESSMENT

Composite service accuracy can be assessed using a semantic distance quality metric. In order to calculate the semantic distance quality metric, each concept appearing in the plan is annotated with a semantic distance d_i with respect to the original concept it was derived from and the selected similarity metric, with a concept distance of 1 revealing equivalence. Additionally, it is annotated with a weight w_i based on the kind of hierarchical relationship to the original concept. The plan similarity metric is calculated as a weighted product of the similarities of all n concepts

appearing in the plan. For all n concepts the plan similarity metric is calculated as:

$$PSM = \prod_{i=0}^n w_i d_i$$

If there is an exact input to output matching, or only equivalent concepts are used, then plan accuracy is 1, decreasing as the plan becomes less accurate.

6 INCORPORATION IN THE PORSCE FRAMEWORK

The PORSCE Framework (Hatzl et al., 2011) accommodates automated web service composition for both semantic and non-semantic descriptions of web services. PORSCE transforms the web service composition problem to an AI planning problem, encodes it to standard PDDL (Ghallab et al., 1998) and utilizes external planning systems to acquire solutions. Furthermore, it exploits semantics to perform semantic awareness and relaxation, in order to provide approximate solutions. Figure 1 depicts a screenshot of the PORSCE interface.

The PORSCE Framework is designed and implemented modularly; each step of the process is implemented as a discrete and separate subsystem. This facilitates the incorporation of additional modules. Initially, PORSCE incorporated only modules for parsing, transformation and semantic relaxation for OWL-S descriptions. Currently, the parsing and transformation modules for the non-semantic WSDL case are also incorporated, and experiments are being performed with a test set created from WSDL descriptions collected by crawling the web with a custom crawler targeted to such descriptions. Also, the module for semantic relaxation using WordNet has been developed and is being tested. The goal of the experiments performed is to identify the most representative semantic distance measure for each case, and to explore the impact of difference threshold for semantic distance to the semantically relaxed composition.

It should be noted that, although the initial available web service descriptions are expressed in different standards, the internal representation is uniform; therefore, the resulting composite service could include at the same time atomic web services described in any of the two standards.

7 CONCLUSIONS

An approach for infusing semantics in non-semantic web service descriptions expressed in WSDL was

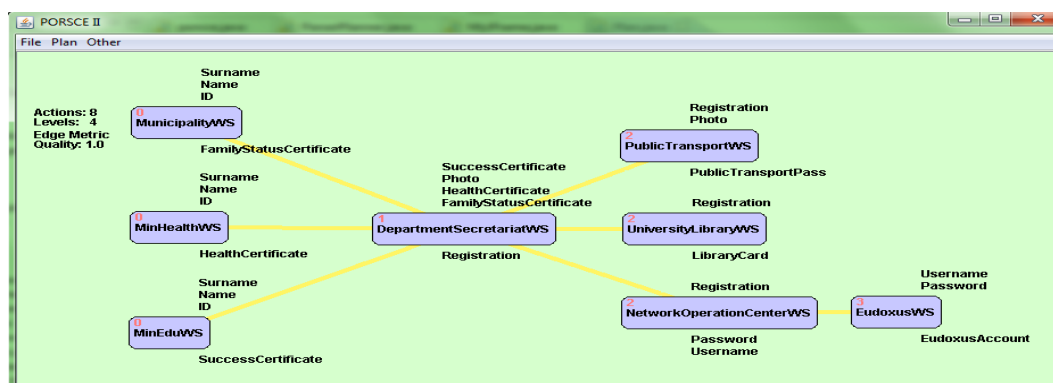


Figure 1: Screenshots of the PORSCCE interface.

presented, enhancing web service discovery and composition. In both cases, search and matching should be detached from syntactic differences and performed under semantic awareness. Semantic relaxation can be also be performed, and approximate solutions can be found. For semantic awareness and relaxation, two factors should be taken into account: the hierarchical relationships between concepts and appropriate semantic distance measures. This paper employs the hierarchical relationships between concepts in WordNet, and explores a variety of semantic distance measures, achieving semantic awareness & relaxation.

The proposed approach was incorporated as a combination of modules in the PORSCCE Framework, and experiments are currently being performed. Future goals include further experimentation with semantic distance measures in order to identify the best one in each case. Furthermore, we also intent to extend semantic awareness and relaxation for non-functional web service properties, such as availability, following the same approach presented in this paper, and research the way web service composition is affected.

REFERENCES

- Benatallah, B., Hacid, M.-S., Leger, A., Rey, C., and Toumani, F. (2005). On automating web services discovery. *The VLDB Journal*, 14:84–96.
- Budanitsky, A. and Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32:13–47.
- Dustdar, S. and Schreiner, W. (2005). A survey on web services composition. *Int. J. Web Grid Services*, 1(1):1–30.
- Fellbaum, C. (2010). Wordnet. In Poli, R., Healy, M., and Kameas, A., editors, *Theory and Applications of Ontology: Computer Applications*, pages 231–243. Springer Netherlands.
- Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D., and Wilkins, D. (1998). PDDL – the planning domain definition language. Technical report, Yale University, New Haven, CT.
- Hatzi, O., Vrakas, D., Bassiliades, N., Anagnostopoulos, D., and Vlahavas, I. (2011). The PORSCCE II framework: Using AI planning for automated semantic web service composition. *The Knowledge Engineering Review*, , to appear, 2011.
- Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008.
- Leacock, C., Chodorow, M., and Miller, G. A. (1998). Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165.
- Lin, D. (1998). An information-theoretic definition of similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann.
- Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244.
- Paolucci, M., Kawamura, T., Payne, T. R., and Sycara, K. P. (2002). Semantic Matching of Web Services Capabilities. In Horrocks, I., Hendler, J. A., Horrocks, I., and Hendler, J. A., editors, *International Semantic Web Conference*, volume 2342 of *Lecture Notes in Computer Science*, pages 333–347. Springer.
- Salton, G., Buckley, C., and Yu, C. T. (1982). An evaluation of term dependence models in information retrieval. In *In Research and Development in Information Retrieval (LNCS 146)*.
- SAWSDL. Semantic Annotations for WSDL. <http://www.w3.org/2002/ws/sawSDL/> [accessed 27/10/10].
- Tsatsaronis, G., Varlamis, I., and Vazirgiannis, M. (2010). Text relatedness based on a word thesaurus. *J. Artif. Intell. Res. (JAIR)*, 37:1–39.
- Wang, L. and Liu, X. (2008). A new model of evaluating concept similarity. *Knowledge-Based Systems*, 21(8):842 – 846.
- WSDL. Web Service Description Language. <http://www.w3.org/TR/wsdl> [accessed 12/05/11].