Semantic Annotation of Web Services

Djelloul Bouchiha & Mimoun Malki
EEDIS Laboratory, Djillali Liabes University of Sidi Bel Abbes, Algeria.
bouchiha.dj@gmail.com, malki@univ-sba.dz

Abstract. Web services are the latest attempt to revolutionize large scale distributed computing. They are based on standards which operate at the syntactic level and lack semantic representation capabilities. Semantics provide better qualitative and scalable solutions to the areas of service interoperation, service discovery, service composition, and process orchestration. SAWSDL defines a mechanism to associate semantic annotations with Web services that are described using Web Service Description Language (WSDL). In this paper we propose an approach for semi-automatically annotating WSDL Web services descriptions. This allows SAWSDL Semantic Web Service Engineering. The annotation approach consists of two main processes: Categorization and Matching. Categorization process consists in classifying WSDL service description to its corresponding domain. Matching process consists in mapping WSDL entities to pre-existing domain ontology. Both categorization and matching rely on ontology matching techniques. A tool has been developed and some experiments have been carried out to evaluate the proposed approach.

Keywords. Annotation; Engineering; Web Service; Semantic Web Services; Ontology; SAWSDL; Ontology Matching Techniques; Similarity Measures.

1 Introduction

Web services are the latest attempt to revolutionize large scale distributed computing. They provide the means to modularize software in a way that functionality can be described, discovered and deployed in a platform independent manner over a network (e.g., intranets, extranets and the Internet). The representation of Web services by current industrial practice is predominantly syntactic in nature lacking the fundamental semantic underpinnings required to fulfil the goals of the emerging Semantic Web Services. SAWSDL defines a mechanism to associate semantic annotations with Web services that are described using Web Service Description Language (WSDL) [20]. The annotation process consists in relating and tagging the WSDL descriptions with the concepts of ontologies.

In this paper we propose an approach for semi-automatically engineering SAWSDL Semantic Web service from an existing Web Service and domain ontology. The proposed approach relies on an annotation process which consists in two phases: (1) Categorization phase, which allows classifying WSDL documents into their corresponding domain (2) Matching phase, which allows associating each entity from WSDL documents with their corresponding entity in the domain ontology. The annotation process relies on ontology matching techniques which in turn use some similarity measures. An empirical study of our approach is presented to help evaluate its performance.

The remainder of paper is organized as follow: In section 2, we discuss some other efforts that describe adding semantics to Web services. In section 3, we present the proposed approach and its underlying concepts and techniques. An empirical study of our approach is presented in section 4 to help evaluate its performance. Finally, section 5 draws some conclusions.

2 Related Works

Several proposals have already been suggested for adding semantics to Web services, such as [18], [5], [6] and [4]. Other approaches concentrate on the Web service annotation: In a preliminary work Bouchiha and al., propose to annotate Web service with ontology using ontology matching techniques [21]. However, they focus on WSDL-S [1] instead of SAWSDL [20].

Table 1. Summary of Web service annotation approaches.

Approach	Considered elements	Annotation resource	Techniques	Tool
[22]	Operation parameters	Workflow	Parameter compatibility rules	Annotation Editor
[21]	Complex types and operations names	Domain ontology	Ontology matching	SAWSDL Builder
[8]	Operations, message parts and Data.	Domain ontology	Text classification techniques	ASSAM
[14]	Data (Inputs and Outputs of services)	Domain ontology	Schema matching techniques	MWSAF tool
[24]	Natural-language query	Domain Ontology	Text mining techniques	Visual OntoBridge (VOB)
[25]	Data (Inputs and Outputs of services)	Meta-data (WSDL)	Machine learning techniques	Semantic labelling tool
[23]	Annotation & Query	Workflow	Propagation method	Prolog Implementation
[26]	Datalog definitions	Source definitions	Inductive logic search	EIDOS

Table 1 summarizes the characteristics of the Web service annotation approaches as follow: (1) The "Approach" column corresponds to the approach in question; (2) The "Considered elements" column describes the considered elements in the annotation process; (3) The "Annotation resource" column indicates the model from which semantic annotations are extracted; (4) The "Techniques" column presents the used techniques for the annotation; (5) The "Tool" column indicates the tool supporting the approach.

3 Annotation approach

As shown in Fig 1, the annotation approach consists of two main processes: Categorization and Matching. Both categorization and matching rely on ontology matching techniques. The goal of ontology matching is to find the relations between entities expressed in different ontologies. Very often, these relations are equivalence relations that are discovered through the measure of the similarity between the entities of ontologies.

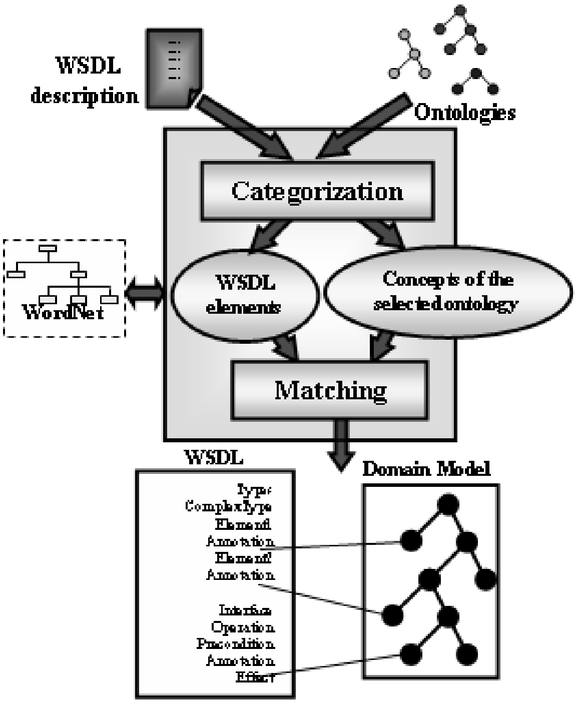


Fig. 1. The annotation approach.

To be accomplished, the ontology matching process uses similarity measures between entities. A similarity measure aims to quantify how much two entities are alike. Formally, it is defined as follow:

Definition 1 (Similarity): Given a set O of entities, a similarity σ : O × O → R is a function from a pair of entities to a real number expressing the similarity between two objects such that:

$$\forall x, y \in O, \sigma(\mathrm{x}, \mathrm{y}) \geq 0 \quad \text{(positiveness)}$$

$$\forall x \in O, \forall y, z \in O, \sigma(\mathrm{x}, \mathrm{x}) \geq \sigma(\mathrm{y}, \mathrm{z}) \quad \text{(maximality)}$$

$$\forall x, y \in O, \sigma(\mathrm{x}, \mathrm{y}) = \sigma(\mathrm{y}, \mathrm{x}) \quad \text{(symmetry)}$$

In our approach, we use WordNet based similarity measures [16]. WordNet is an online lexical database designed for use under program control [13]. So, these measures are computed, and then normalized. Normalisation consists generally in inversing the measure value to obtain a new value between 0 and 1. The value 1 indicates that there is a full semantic equivalence between the two entities.

Similarity measures relying on WordNet can be classified into three categories: (1) Similarity measures based on path lengths between concepts: lch [11], wup [19], and path; (2) Similarity measures based on information content: res [17], lin [12], and jcn [7]; and (3) Relatedness measures based on relations type between concepts: hso [9], lesk [3], and vector [15].

When a set of ontologies are available, similarities between two sets have to be computed by comparing the set of entities of the WSDL file and the set of entities of each ontology. On the basis of such measures, systems will decide between which ontologies to run a matching algorithm. The chosen domain ontology determines the WSDL file category. This process is called the categorization process.

Our approach considers an ontology as a set of entities (concepts), and a WSDL file also as a set of entities (XSD data types, interface, operations, messages). Several strategies can be adopted for computing similarities between two sets. Next we define Single linkage, Full linkage and Average linkage strategies:

Definition 2 (Single linkage): Given a similarity function σ : O × O → R, the single linkage measure between two sets is a similarity function Δ : 2O ×2O → R such that:

$$\forall x, y \subseteq O, \Delta(x, y) = \max_{(e1,e2) \in x^* y} \sigma(e1, e2)$$

Definition 3 (Full linkage): Given a similarity function σ : O × O → R, the complete linkage measure between two sets is a similarity function Δ : 2O ×2O → R such that:

$$\forall x, y \subseteq O, \Delta(x, y) = \min_{(e1,e2) \in x^* y} \sigma(e1, e2)$$

Definition 4 (Average linkage): Given a similarity function σ : O × O → R, the average linkage measure between two sets is a similarity function Δ : 2O ×2O → R such that:

$$\forall x, y \subseteq O, \Delta(x, y) = \frac{\sum_{(e1,e2) \in x^* y} \sigma(e1, e2)}{| x | * | \mathrm{y} |}$$

Next we detail the two processes involved in our approach.

Categorization process. The categorization process aims to classify WSDL service description to its corresponding domain. For this end, the service description is broken down into its fundamental WSDL elements (XSD data types, interface, operations and messages). A list of concepts is also extracted from each ontology. Similarities between two sets based on similarity measure between two entities will be computed to identify which ontology concepts will be kept for the next process. The selected ontology indicates the WSDL domain or category.

We have developed an algorithm (see Listing 1) that implements the categorization process. The algorithm computes the similarity between a WSDL document and a set of domain ontologies. A WSDL document belongs to the category of the domain ontology for which it gives the best similarity (the nearest ontology).

Listing 1. The Categorization algorithm.

```
Algorithm Categorization
 Input
  WSDL document
  A set of domain ontologies
  A similarity measure SM between two entities
  A Similarity SD between two sets
  Threshold
 Output
  An assigned WSDL document to a particular category
 Begin algo
  Filling a vector VE with the WSDL document elements
  For each domain ontology Do
   Filling a vector VC with the domain ontology concepts
   For each element E of the vector VE Do
    For each element C of the vector VC Do
    // Next, Vector Sim is used to store the
    //Similarity between the two vectors VE and VC
     Switch SD of
       Single linkage : If (SM(E,C) > Vector Sim)
                        then Vector Sim • SM(E,C) End if
       Full linkage : If (SM(E,C) < Vector_Sim) then
                      Vector Sim • SM(E,C)
                      End if
       Average linkage : Vector_Sim • Vector_Sim + SM(E,C)
     End switch
    End for
   End_for
   If SD is Average linkage
       then Vector Sim • Vector Sim / (|VC| * |VE|)
   End_if
  // Next, Final Sim is used to store Similarity
  //between VE and the nearest ontology
  If (Final_Sim < Vector_Sim )
      then Final Sim • Vector Sim
  End if
  End_For
  If (Final Sim > Threshold )
     then the WSDL document is assigned to the corresponding
     ontology to the Final_Sim
  End if
End Algo
```

Matching process. The matching process aims to map WSDL elements to ontology concepts. Similarities between a WSDL element and the concepts of the selected ontology will be computed to identify which concept will be attached to the initial WSDL element. This operation is repeated for all WSDL elements.

We have developed an algorithm (see Listing 2) that implements the matching process. The algorithm computes the semantic similarities between WSDL document elements and domain ontology concepts. Each WSDL document element will be annotated by the nearest domain ontology concept.

Listing 2. The Matching algorithm.

```
Algorithm Matching
 Input
  WSDL document
  A domain ontology
  A similarity measure SM between two entities
  Threshold
 Output
  An annotated WSDL document with a domain ontology concepts
 Begin algo
  Filling a vector VE with the WSDL document elements
  Filling a vector VC with the domain ontology concepts
  For each element E of the vector VE Do
   For each element C of the vector VC Do
   //Next, Entity Sim is used to store Similarity
   //between a WSDL element and the nearest
   //ontology concept
    If (SM(E,C) > Entity Sim) then Entity Sim • SM(E,C)
End if
   End_for
   If (Entity Sim > Threshold )
      then assign the element E to the corresponding concept
      of the domain ontology
   End if
  End for
End Algo
```

As result of the two algorithms, an annotated WSDL document will be generated.

4 Results and empirical testing

The algorithms presented above are generic and can be adapted to most domain model languages. The domain model language we have used is the OWL, but we believe that our results could be applied to any similar language. To evaluate and validate our approach a tool, called SAWSDL generator[1], has been developed. SAWSDL generator can be used to do semi-automatic annotations. It takes in a WSDL document which has to be annotated with a set of ontologies. It selects the best ontology for annotating the WSDL document and suggests most appropriate mappings for the XSD data types, interface, operations and messages in the WSDL file. The classification and matching are performed using ontology matching techniques. The tool produces annotated WSDL 2.0 file using extensibility elements and according to the SAWSDL recommendation [20].

To test our categorization algorithm we first obtained a corpus[2] of 424 Web services [8]. Although our initial intention was to test our algorithm on the whole corpus, we have limited our testing to one domain, due to lack of relevant domain specific ontologies. We are in the process of creating new domain ontologies and plan to extend our testing for remaining Web services in the future.

[1] http://www-inf.univ-sba.dz/wsdls/

[2] http://www.andreas-hess.info/projects/annotator/ws2003.html

The domain we have selected for testing is Business domain[3]. Although the ontology used is not comprehensive enough to cover all the concepts in this domain, they are sufficient enough to serve the purpose of categorization. We have taken a set of 31 services out of which 13 are from business domain, 13 from weather domain and 5 from the games domain.

As similarity measure, the path method has been used. It is defined as follow: For two entities e1 and e2, the similarity measure SIM can be given using the WordNet synsets (i.e. term for a sense or a meaning by a group of synonyms) based on the formula: SIM(e1, e2)=1/length(e1, e2), where length is the length of the shortest path between two entities e1 and e2 using node counting.

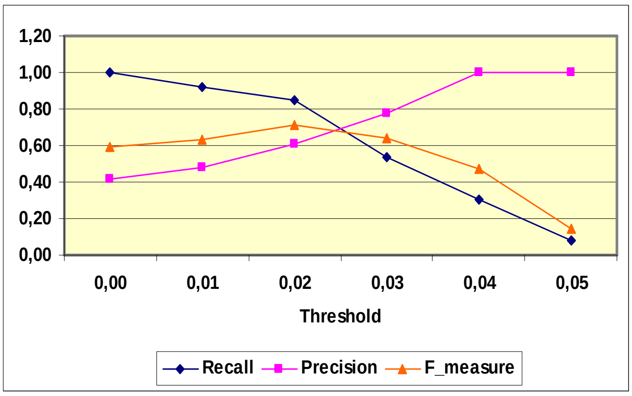
As in information retrieval [2], we use two metrics, Precision and Recall[4], to evaluate the results of our algorithm of categorization.

- Recall (R): proportion of the correctly assigned WSDL documents of all the WSDL documents that should be assigned.
- Precision (P): proportion of the correctly assigned WSDL documents of all the WSDL documents that have been assigned.

Usually, Precision and Recall scores are not discussed in isolation. Instead, they are combined into a single measure, such as the F-measure [10], which is defined as follow: F_measure = (2 * recall * precision)/(recall + precision).

The services are categorized based on the categorization threshold, which decides if the service belongs to a domain. If the best average service match calculated for a particular Web service is above the threshold then the service belongs to the corresponding domain.

Graph 1 depicts the corresponding curves to the precision, recall and f-measure statistics obtained by applying our categorization algorithm on this set of 31 Web services for different threshold values according to the average linkage strategy.



Graph 1. Precision, recall and f-measure curves for the categorization algorithm.

It is very important to choose the threshold value correctly. We can see from Graph 1 that for threshold = 0.02, which corresponds to the topmost value of the f-measure

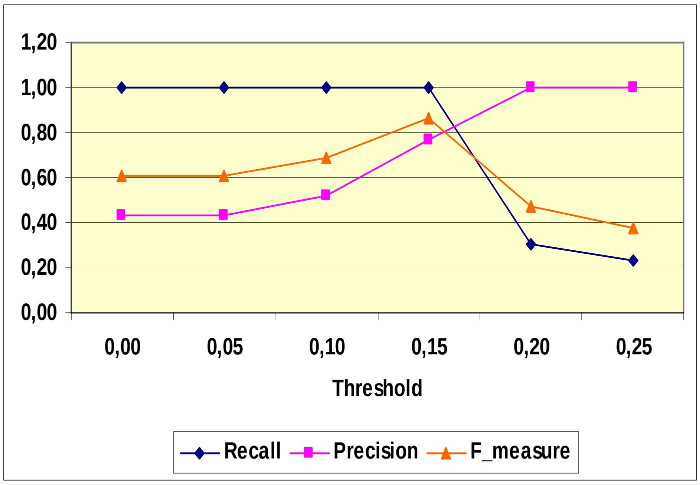
[3] http://www.getopt.org/ecimf/contrib/onto/REA/index.html

[4] http://en.wikipedia.org/wiki/Precision_and_recall

curve, gives the best categorization. However, even with the best threshold, some problems can appear. For example, The Web service "BasicOptionPricing" has not been rightly classified into the business domain, because it includes operations which have not meaningful names. Also, the two Web services "Weather Forecast By Zip Code" and "World Weather Forecast by ICAO" have been wrongly classified into business domain, although they belong to the weather domain. The reason behind this is that the two services include "Forecast" operations which can be shared between both business and weather domain.

To verify the fitness of the obtained result, a reference annotated WSDL document is considered as a valid. The chosen WSDL document was "TrackingAll". Now, to evaluate the quality of the matching algorithm, we compare the match result returned by our automatic matching process with manually determined match result in the reference WSDL annotated document. We determine the true positives, i.e. correctly identified matches.

Graph 2 depicts the corresponding curves to the precision, recall and f-measure statistics obtained by applying our matching algorithm on the chosen Web service for different threshold values according to the path measure similarity.



Graph 2. Precision, recall and f-measure curves for the matching algorithm.

Graph 2 shows that best results of the matching algorithm are obtained with threshold = 0,15. However, even with this threshold, a system user intervention is suggested for withdrawing some matching, or validating the result as it is generated. For example the WSDL elements "update_Company", "update_Customer", "update_Status" and "update_Tracking" have been matched wrongly to the concept "Agreement". The reason behind this is that the WSDL element names include the term "update" which has been treated by the system as name and not as a verb. As a name "update" means "news that updates your information". With a small threshold (<0,15), the user intervention is always necessary for keeping only right matching.

5 Conclusion

In order to harvest all the benefits of Web services technology, an approach has been proposed for annotating WSDL syntactic descriptions of Web services by ontological models. The benefits of such approach are twofold: Firstly, the approach provides a way to map WSDL descriptions to domain ontologies. Secondly, the approach enables the migration of syntactically defined Web services toward Semantic Web Services.

The proposed annotation approach consists of two main processes: Categorization and Matching. At the first process, WSDL service description is classified to its corresponding domain. At the second process the WSDL entities are mapped to pre-existing domain ontology. Both categorization and matching use WordNet based similarity measures.

A tool has been developed to implement the proposed approach. Some validation experiments have been carried out and they showed the usefulness of the proposed approach and highlighted possible areas for improvement of its effectiveness.

The developed approach provides very satisfactory and encouraging results and supports the potential role that this approach can play in providing a suitable starting point for SAWSDL semantic Web services development.

References

1. Akkiraju R., Farrell J., Miller J., Nagarajan M., Schmidt M-T., Sheth A., and Verma K., "Web service semantics – WSDL-S". Tech. rep., W3C. http://www.w3.org/Submission/WSDL-S/. 2005.
2. Baeze-Yates R., and Ribeiro-Neto B., "Modern information retrieval", Addison-Wesley, ACM Press, Reading, MA. 1999.
3. Banerjee S., and Pedersen T., "Extended gloss overlaps as a measure of semantic relatedness". In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence. Pages: 805-810. 2003.
4. Bell D., de Cesare S., Iacovelli N., Lycett M., and Merico A., "A framework for deriving semantic web services". Information Systems Frontiers. Volume 9, Number 1, Pages: 69-84. 2007.
5. Bouchiha D., and Malki M., "Towards re-engineering Web Applications into semantic Web services". The first International IEEE Conference on Machine and Web Intelligence (ICMWI'2010). Algeria, Algiers. 2010.
6. Buitelaar P., and Gmbh D., "Ontology learning for semantic Web services". In Proceedings of ONLINE2003, Düsseldorf, Germany. 2003.
7. Jiang J., and Conrath D., Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings on International Conference on Research in Computational Linguistics, Pages: 19-33. 1997.
8. Hess A., Johnston E., and Kushmerick N., "ASSAM: A tool for semi-automatically annotating semantic Web services". International Semantic Web Conference. Hiroshima, Japan. Pages: 320-335. 2004.
9. Hirst G., and St-Onge D., "Lexical chains as representations of context for the detection and correction of malapropisms". In Fellbaum, C., ed., WordNet: An electronic lexical database. MIT Press. Pages: 305-332. 1998.

10. Larsen B., and Aone C., "Fast and effective text mining using lineartime document clustering", Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Pages: 16-22. 1999.
11. Leacock C., and Chodorow M., "Combining local context and WordNet similarity for word sense identification". In Fellbaum, C., ed., WordNet: An electronic lexical database. MIT Press. Pages: 265–283. 1998.
12. Lin D., "An information-theoretic definition of similarity". In Proceedings of the International Conference on Machine Learning. 1998.
13. Miller G-A., "WordNet: An on-line lexical database". International Journal of Lexicography. Pages: 235-312. 1990.
14. Patil, S. Oundhakar, A. Sheth, and V. Kunal. "METEOR-S Web service annotation framework". WWW 2004, ACM Press. Pages: 553-562. 2004.
15. Patwardhan S., "Incorporating dictionary and corpus information into a context vector measure of semantic relatedness". Master's thesis, Univ. of Minnesota, Duluth. 2003.
16. Pedersen T., Patwardhan S., and Michelizzi J., "WordNet::Similarity - measuring the relatedness of concepts". Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04). Pages: 1024-1025. 2004.
17. Resnik P., "Using information content to evaluate semantic similarity in a taxonomy". In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Pages: 448-453. 1995.
18. Sabou M., Wroe C., Goble C., and Stuckenschmidt H., "Learning domain ontologies for semantic Web service descriptions". Journal of Web Semantics. Volume 3, N 4. Pages: 340-365. 2005.
19. Wu Z., and Palmer M., "Verb semantics and lexical selection". In 32nd Annual Meeting of the Association for Computational Linguistics, Pages: 133–138. 1994.
20. Farrell J., and Lausen H., "Semantic Annotations for WSDL and XML Schema". W3C Recommendation, 28 August 2007. Available at http://www.w3.org/TR/sawsdl/. 2007.
21. Bouchiha D., Malki M., Alghamdi, A., and Alnafjan, K., "An Empirical Approach for Annotating Web Services". The 24th International Conference on Computer Applications in Industry and Engineering. Hawaii, USA. November 16-18, 2011.
22. Belhajjame K., Embury S-M., Paton N-W., Stevens R., and Goble C-A., "Automatic annotation of web services based on workflow definitions". ACM Transactions on the Web (TWEB journal). Number 2, Volume 2. 2008.
23. Bowers S., and Ludäscher B., "A calculus for propagating semantic annotations through scientific workflow queries". Query Languages and Query Processing workshop (QLQP-2006) anised in conjunction with the 10th International Conference on Extending abase Technology, pages 712-723. 2006.
24. Grcar M., and Mladenic D., "Visual OntoBridge: Semi-automatic Semantic Annotation Software". In ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II. LNAI 5782, pages 726-729, Springer-Verlag Berlin, Heidelberg. 2009.
25. Lerman K., Plangprasopchok A., and Knoblock C-A., "Automatically labeling the inputs and outputs of web services". In Proceedings of the National Conference on Artificial Intelligence (AAAI-2006). Boston, Massachusetts, USA. July 2006.
26. Carman M-J., and Knoblock C-A., "Learning Semantic Definitions of Online Information Sources". Journal of Artificial Intelligence Research. Volume 30, pages 1-50. 2007.