Computing and Informatics, Vol. 31, 2012, 1575–1595

SEMANTIC WEB SERVICE ENGINEERING: ANNOTATION BASED APPROACH

Djelloul Bouchiha, Mimoun Malki

EEDIS Laboratory
Djillali Liabes University of Sidi Bel Abbes
Algeria
e-mail: bouchiha.dj@gmail.com, malki@univ-sba.dz

Abdullah Alghamdi, Khalid Alnafjan

College of Computer and Information Sciences, KSU
Riyadh, Saudi Arabia
e-mail: {Ghamdi, Alnafjan}@ksu.edu.sa

Communicated by Marian Babík

Abstract. Web services are an emerging paradigm which aims at implementing software components in the Web. They are based on syntactic standards, notably WSDL. Semantic annotation of Web services provides better qualitative and scalable solutions to the areas of service interoperation, service discovery, service composition and process orchestration. Manual annotation is a time-consuming process which requires deep domain knowledge and consistency of interpretation within annotation teams. Therefore, we propose an approach for semi-automatically annotating WSDL Web services descriptions. This is allowed by 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, web service, engineering, semantic web services, ontology, SAWSDL, ontology matching, similarity measures

1 INTRODUCTION

A Web service is a software component that provides services via a standardized interface. It uses Internet technology as an infrastructure for describing software components (Web services) by using a syntactic standard, namely the Web Service Description Language (WSDL). This presents a lack which must be handled to have Semantic Web Services (SWS). Semantics allow automatic discovery and composition of Web services.

Several initiatives have been proposed for developing a world-wide standard for the semantic description of Web services, such as OWL-S [25] and WSMO [32]. These proposals add a semantic layer to the existing syntactic description of Web services; so they provide conceptual model and a language to semantically describe all aspects related to the Web services. Another solution, which seems simple but has the same impact as the others, is SAWSDL [13]. SAWSDL defines a mechanism to associate semantic annotations with Web services that are described using WSDL.

Ontologies represent the semantic resources with which WSDL descriptions can be annotated. The annotation process consists particularly in relating and tagging WSDL descriptions with concepts in ontologies. Manual annotation poses several problems: the first one is that of finding the relevant ontology or ontologies. The second problem is the large size of the Web service description and of the ontology or vocabulary. Because of these factors, it is necessary to have a semi-automatic and scalable way for annotating Web services with real world ontologies.

In this paper we propose an approach for semi-automatically annotating WSDL Web services descriptions. The annotation process 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 rest of paper is organized as follows: Section 2 provides the background needed for understanding our work. 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 evaluate its performance. In Section 5, we discuss some other efforts that describe adding semantics and annotating Web services. Finally, Section 6 presents conclusion and future work.

2 BACKGROUND

The ontology is an explicit specification of a conceptualisation [16]. It specifies a conceptualization of a domain in terms of concepts, attributes, relations and assertions [14]. Many formal languages have been proposed to specify ontologies, such as OWL (Ontology Web Langage) [26].

Semantic heterogeneities can appear between two ontologies even if they belong to the same domain. To resolve this problem, ontology matching techniques can be used. The goal of ontology matching is to find the relations (mappings or correspondences) between entities expressed in different ontologies. Very often, these relations are equivalence relations that are discovered through the measure of similarity between the entities of ontologies [12]. A similarity measure aims to quantify how much two entities are alike. Different strategies (e.g., string similarity, synonyms, structure similarity and based on instances) for determining similarity between entities are used in current ontology matching systems. The WordNet thesauri can support improving similarity measures [24]. WordNet is an online lexical database designed for use under program control [27].

Similarity measures relying on WordNet can be classified into three categories:

1. Similarity measures based on path lengths between concepts: *lch* [21], *wup* [34], and *path*. The *lch* measure finds the shortest path between two concepts, and scales that value by the maximum path length in the is-a hierarchy in which they occur. *wup* finds the path length to the root node from the least common subsumer (LCS) of the two concepts, which is the most specific concept they share as an ancestor. This value is scaled by the sum of the path lengths from the individual concepts to the root. The measure *path* is equal to the inverse of the shortest path length between two concepts.

2. Similarity measures based on information content: *res* [31], *lin* [23], and *jcn* [17]. The *lin* and *jcn* measures augment the information content of the LCS of two concepts with the sum of the information content of the individual concepts. The *lin* measure scales the information content of the LCS by this sum, while *jcn* subtracts the information content of the LCS from this sum (and then takes the inverse to convert it from a distance to a similarity measure).

3. Relatedness measures based on relations type between concepts: *hso* [19], *lesk* [3], and *vector* [29]. The *hso* measure is path based, and classifies relations in WordNet as having direction. It establishes the relatedness between two concepts by trying to find a path between them that is neither too long nor that changes direction too often. Each concept (or word sense) in WordNet is defined by a short gloss. The *lesk* and *vector* measures use the text of that gloss as a unique representation for the underlying concept. The *lesk* measure assigns relatedness by finding and scoring overlaps between the glosses of the two concepts, as well as concepts that are directly linked to them according to WordNet. The *vector* measure creates a co-occurrence matrix from a corpus made up of the WordNet glosses. Each content word used in a WordNet gloss has an associated context vector. Each gloss is represented by a gloss vector that is the average of all the context vectors of the words found in the gloss. Relatedness between concepts is measured by finding the cosine between a pair of gloss vectors

3 ANNOTATION APPROACH

For our purpose the ontology matching problem has been reformulated to accomplish the Web service annotation. It becomes a matching problem between two taxonomies. On one hand, a WSDL description is organized as taxonomy of elements, notably data types, interface, operations and messages. On the other hand, the ontology concepts are typically organized into a taxonomy tree where each node represents a concept and each concept is a specialization of its parent. We aim at finding mappings between the elements of the first taxonomy and the concepts of the second one.

Notice that mappings between different types of elements are possible. For example, a WSDL operation can be mapped to an ontology relation. An input output parameter of a service can be mapped to an attribute of a concept. Extending matching to ontology assertions is more complex. It is the subject of ongoing research.

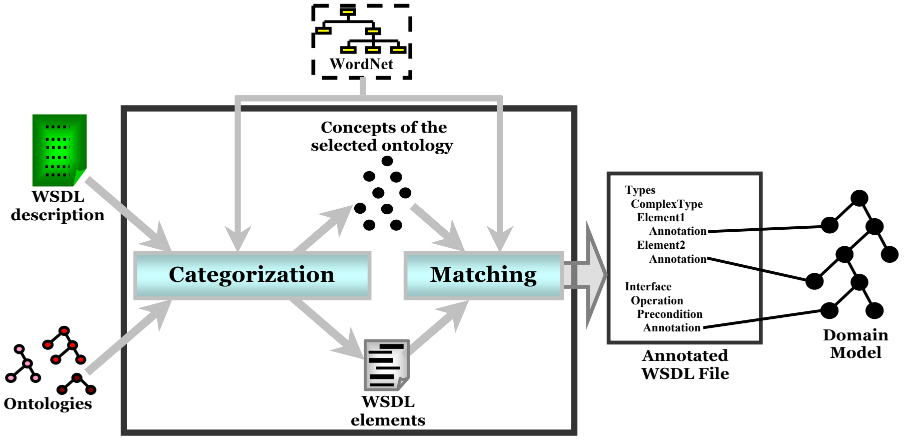


Fig. 1. The annotation approach

As shown in Figure 1, the annotation approach consists of two main processes: categorization and matching. Both categorization and matching rely on ontology matching techniques which in turn are based on similarity measures.

Formally, a similarity measure is defined as follow:

Definition 1 (Similarity)**.** Given a set O of entities, a similarity $\sigma : O \times O \to R$ is a function from a pair of entities to a real number expressing the similarity between two objects such that:

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

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

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

Our system uses WordNet based similarity measures [30]. 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.

Before creating mappings, the WSDL document must be assigned to its appropriate domain. 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, the system 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. Several strategies can be adopted for computing similarities between two sets. Single linkage, Full linkage and Average linkage strategies are defined as follows:

Definition 2 (Single linkage)**.** Given a similarity function $\sigma : O \times O \to R$, the single linkage measure between two sets is a similarity function $\Delta : 2^O \times 2^O \to R$ 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 $\sigma : O \times O \to R$, the full linkage measure between two sets is a similarity function $\Delta : 2^O \times 2^O \to R$ 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 $\sigma : O \times O \to R$, the average linkage measure between two sets is a similarity function $\Delta : 2^O \times 2^O \to R$ such that:

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

Next we detail the two processes involved in our approach.

3.1 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).

```
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
  A WSDL document assigned 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 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 stores 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 doc is assigned to the corresponding ontology to the Final_Sim
   End_if
End_Algo
```

Listing 1. The Categorization algorithm

3.2 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.

```
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
```

Listing 2. The Matching algorithm

As result of the two algorithms, an annotated WSDL document will be generated. The system user has the possibility of withdrawing some mappings, or validating the result as it is generated.

4 RESULTS AND EMPIRICAL TESTING

The algorithms presented above are general 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, has been implemented.

4.1 Implemented Tool

SAWSDL generator[1] can be used to do semi-automatic annotations. It takes in a WSDL document which has to be annotated and a given 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 [13].

The tool is implemented in JAVA. It interacts with DOMSAX API to parse WSDL documents. It interacts also with the Java WordNet API[2] and Protege-OWL API[3] for ontological parsing and for computing semantic similarity measures between entities and sets. It provides a set of features for personalizing the calculations performed during the categorization and matching processes. Details on the resulting annotated WSDL documents can be also visualized.

The implemented tool performs:

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

[2] `http://jwordnet.sourceforge.net/handbook.html`

[3] `http://protege.stanford.edu/plugins/owl/api/guide.html`

1. *Categorization:* this process consists in classifying a WSDL document into its corresponding domain by considering complex types and operations names. Other constructs, such as elements of complex types and messages parameters, are not considered because they contain less important information, such as code, type, address, etc., which are not specific to a particular domain. When considered, it will burden the categorization algorithm by increasing its complexity and will reduce the quality of this algorithm by reducing the similarity values. The message names are redundant and can exist in the operations.
2. *Matching:* it consists in

   (a) The namespace *xmlns:sawsdl="http://www.w3.org/ns/sawsdl"* is added to the root element of the WSDL document. This namespace references the definitions of the WSDL extensibility elements, notably the attribute *sawsdl: modelReference*.
   (b) The ontologies namespaces, with which the annotation will be done, are added to the WSDL document.
   (c) Annotating the components *element*, *complexType*, *simpleType* and *attribute* of XML schema, as well as the components *operation* and *fault* of WSDL. This annotation is done via the attribute *sawsdl:modelReference*. In the case of WSDL 1.1, the messages parameters (i.e. the WSDL components *part*) can also be annotated.
   (d) Annotating the component interface with the attribute *sawsdl: modelReference* to specify the category of the WSDL document. In the case of WSDL 1.1, it is the component *portType* which will be annotated.

4.2 Evaluation

The evaluation process aims to measure the quality of the WSDL annotated documents that are generated from the developed tool. The process of evaluation embodies two components, both categorization and matching evaluation.

While the categorization evaluation is essential to evaluate the performance of the categorization process, the matching evaluation evaluates the adequacy of the matching process.

A. Categorization Evaluation

To test our categorization algorithm we first obtained a corpus[4] of 424 Web services [18]. 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 plan to extend our testing for remaining Web services in the future.

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

The domain we have selected for testing is Business domain[5]. 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 games domain.

As similarity measure, the path method has been used. It is defined as follows: 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.

Table 1 summarizes the obtained results.

Now, we define the following assistant parameters:

- CN: number of correct WSDL documents that should be assigned to the considered domain ontology;
- EN: number of WSDL documents assigned by the tool;
- CEN: number of the correct WSDL documents assigned by the tool.

As in information retrieval [2], we use two metrics, Precision and Recall[6], 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. It can be presented as "R = CEN/CN".
- Precision (P): proportion of the correctly assigned WSDL documents of all the WSDL documents that have been assigned. It can be presented as "P = CEN/EN".

Precision can be seen as a measure of exactness or fidelity, whereas Recall is a measure of completeness. A perfect Precision score of 1.0 means that every assigned WSDL document was correct (but says nothing about whether all correct WSDL documents were assigned) whereas a perfect Recall score of 1.0 means that all correct WSDL documents were assigned (but says nothing about how many incorrect WSDL documents were also assigned).

Often, there is an inverse relationship between Precision and Recall, where it is possible to increase one at the cost of reducing the other.

Usually, Precision and Recall scores are not discussed in isolation. Instead, either values for one measure are compared for a fixed level at the other measure (e.g. precision at a recall level of 0.75) or both are combined into a single measure, such as the F-measure [20], which is the weighted harmonic mean of precision and recall. F-measure is defined as follows: $F\text{-measure} = (2 * \text{recall} * \text{precision})/(\text{recall} + \text{precision})$.

---

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

[6] `http://en.wikipedia.org/wiki/Precision_and_recall`

Service name	Category	Single link	Full link	Average link
ActionPlanning	Business	0.3333	0.0	0.0472
AutoLoanCalculator	Business	0.5000	0.0	0.0596
BusinessFinderUDDI	Business	0.3333	0.0	0.0223
BasicOptionPricing	Business	0.1111	0.0	0.0091
CompanyMarketData	Business	0.2500	0.0	0.0375
NAICSandSICCode	Business	0.3333	0.0	0.0447
Flash-db.com	Business	0.2000	0.0	0.0326
ForecastTModelInterface	Business	0.2500	0.0	0.0343
ManageNumbers	Business	0.2500	0.0	0.0281
Workflow	Business	0.3333	0.0	0.0277
TrackingAll	Business	0.2500	0.0	0.0199
UNSPSCConvert	Business	0.1000	0.0	0.0236
UPC Database Lookup	Business	0.2500	0.0	0.0477
AsianEarthquakes	Weather	0.2000	0.0	0.0060
AustralianandNewZealand WeatherService	Weather	0.1250	0.0	0.0287
NOAA Weather Station	Weather	0.2500	0.0	0.0252
FastWeather	Weather	0.2500	0.0	0.0068
GlobalWeather	Weather	0.3333	0.0	0.01730
Unisys Weather Forecast	Weather	0.2500	0.0	0.0101
Weather By Zip Code	Weather	0.1666	0.0	0.0043
Weather Fetcher	Weather	0.1666	0.0	0.0050
Weather Forecast By Zip Code	Weather	0.2500	0.0	0.0353
Weather Warnings by State	Weather	0.2000	0.0	0.0103
World Weather By Station ID	Weather	0.1666	0.0	0.0043
World Weather Forecast by ICAO	Weather	0.2500	0.0	0.0353
TemperatureService	Weather	0.1250	0.0	0.0287
FonttoGraphic	Games	0.2000	0.0	0.0286
DiceThrowing	Games	0.1111	0.0	0.0181
EightBall	Games	0.1666	0.0	0.0289
SuperLottoIT	Games	0.2000	0.0	0.0111
Lottery Numbers	Games	0.2500	0.0	0.0152

The "Service name" indicates the service name specified at the third line of the TXT file accompanying the WSDL document in the hierarchically classified Web services of the corpus.
The "Category" indicates the class of the WSDL document according to the considered corpus.
The "Single link" indicates how much the WSDL document and Business ontology are alike according to the Single linkage strategy.
The "Full link" indicates how much the WSDL document and Business ontology are alike according to the Full linkage strategy.
The "Average link" indicates how much the WSDL document and Business ontology are alike according to the Average linkage strategy.

Table 1. Categorization statistics of Web services

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.

Table 2 presents 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.

Threshold	CN	EN	CEN	Recall	Precision	F-measure
0.00	13	31	13	1.00	0.42	0.59
0.01	13	25	12	0.92	0.48	0.63
0.02	13	18	11	0.85	0.61	0.71
0.03	13	9	7	0.54	0.78	0.64
0.04	13	4	4	0.31	1.00	0.47
0.05	13	1	1	0.08	1.00	0.14

Table 2. Precision, recall and f-measure statistics for the categorization algorithm

Figure 2 depicts the corresponding curves to Table 2.

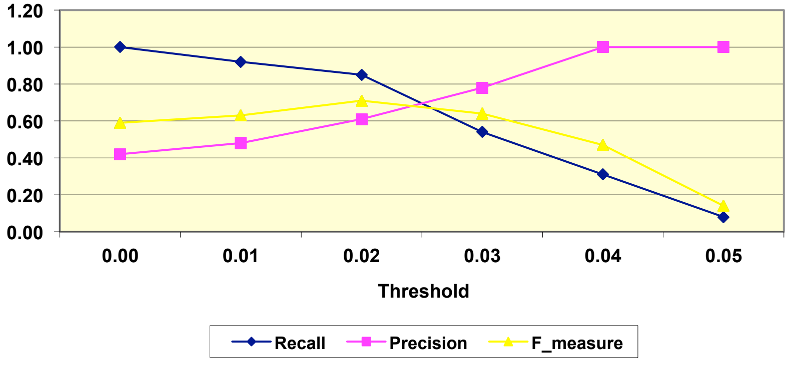


Fig. 2. Precision, recall and f-measure curves for the categorization algorithm

It is very important to choose the threshold value correctly. We can see from Figure 2 that for threshold = 0.02, which corresponds to the topmost value of the f-measure curve, gives the best categorization. However, even with the best threshold, some problems can appear. For example, The Web service "BasicOption-Pricing" 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.

B. Matching Evaluation

To verify the fitness of the obtained result, a reference annotated WSDL document is considered as a valid. The chosen WSDL document was "TrackingAll" and the suggested matching between the WSDL document elements and the business domain ontology concepts are as follows.

WSDL document element	The corresponding concept
set_Customer_Permission	Association
set_Operator_Permission	Association
set_Company_Permission	Association
getSpecified_Tracking_Access	Transition
get_Company_List	Association
getSpecified_Company	Association
showAll_Tracking	Transition
showSpecified_Tracking	Transition
getSpecified_Tracking	Transition
show_Tracking_History	Activity
get_Tracking_History	Transition
getAll_Tracking	Transition
validate_User	protege:PAL-Constraint

Table 3. The manually suggested matching

Table 4 illustrates the obtained results by our matching algorithm.

To evaluate the quality of the matching algorithm, we compare the match result EN returned by our automatic matching process with manually determined match result CN in the reference WSDL annotated document. We determine the true positives, i.e. correctly identified matches CEN.

So: CN is the number of correct WSDL constructs that should be annotated by the domain ontology concepts; EN is the number of WSDL constructs annotated by the tool; CEN is the number of the correct WSDL constructs annotated by the tool.

Table 5 presents the precision, recall and f-measure statistics obtained by applying our matching algorithm on this set of 31 Web services for different threshold values according to the path measure similarity.

Figure 3 depicts the corresponding curves to Table 5. It 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.

WSDL document element	**The corresponding concept**	**Path measure**
set_Customer_Permission	Association	0.1666
set_Operator_Permission	Association	0.1666
set_Company_Permission	Association	0.1666
getSpecified_Tracking_Access	Transition	0.1666
getSpecified_Customer_Access	protege:PAL-Constraint	0.1428
getAll_Customer_Access	protege:PAL-Constraint	0.1428
get_Company_List	Association	0.2500
get_Operators_List	Agreement	0.1111
get_Customer_List	protege:PAL-Constraint	0.1428
getSpecified_Company	Association	0.2500
getSpecified_Customer	protege:PAL-Constraint	0.1428
showAll_Tracking	Transition	0.1666
showSpecified_Tracking	Transition	0.1666
getSpecified_Tracking	Transition	0.1666
show_Tracking_History	Activity	0.2500
get_Tracking_History	Transition	0.1666
get_Customer_Tracking	protege:PAL-Constraint	0.1428
getAll_Tracking	Transition	0.1666
get_Customer_CompanyName	protege:PAL-Constraint	0.1428
get_Company_Name	Agreement	0.1000
add_Company	REA_Element	0.0909
add_Operator	REA_Element	0.0909
add_Customer	REA_Element	0.0909
update_Company	Agreement	0.1666
update_Customer	Agreement	0.1666
update_Status	Agreement	0.1666
add_Status	REA_Element	0.0909
add_Tracking	REA_Element	0.0909
update_Tracking	Agreement	0.1666
validate_User	protege:PAL-Constraint	0.2000

Table 4. The automatically computed matching

Threshold	**CN**	**EN**	**CEN**	**Recall**	**Precision**	**F-measure**
0.00	13	30	13	1.00	0.43	0.60
0.05	13	30	13	1.00	0.43	0.60
0.10	13	25	13	1.00	0.52	0.68
0.15	13	17	13	1.00	0.76	0.87
0.20	13	4	4	0.31	1.00	0.47
0.25	13	3	3	0.23	1.00	0.38

Table 5. Precision, recall and f-measure statistics for the matching algorithm

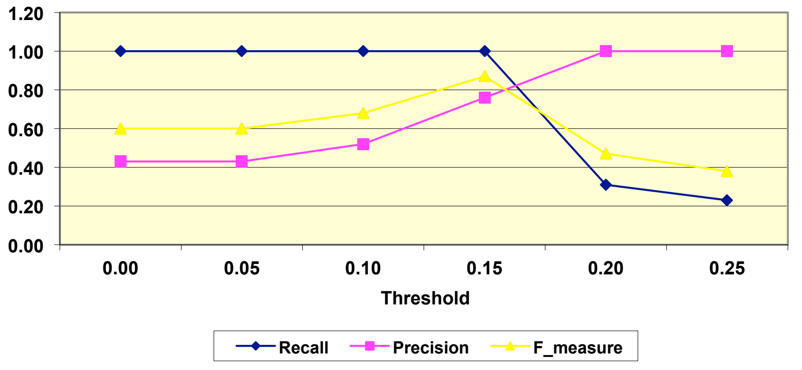


Fig. 3. Precision, recall and f-measure curves for the matching algorithm

Some knowledge specified by a given domain ontology is insufficiently precise for a good annotation of the WSDL document. This is because in some cases the domain ontology is oriented toward a specific task to be carried out in the domain rather than task independent domain knowledge. To improve the quality of the final WSDL annotated document, domain ontology must be greatly improved. For example, a refactoring activity is necessary for removing from domain ontology those concepts that have the same extension as other concepts also present in the ontology.

5 RELATED WORKS

Several proposals have already been suggested for adding semantics to Web services, such as [5, 6, 9, 33]. Other approaches concentrate on the Web service annotation: In a preliminary work Bouchiha et al. propose to annotate Web service with ontology using ontology matching techniques [7]. However, they focus on WSDL-S [1] instead of SAWSDL [13].

Hess et al. present the ASSAM annotator tool which suggests annotations to the user based on a machine learning algorithm [18]. While Hess et al. cast the problem of classifying the WSDL descriptions as a text classification problem [18], we consider it, in our approach, an ontology matching problem. Our decision can be argued by the structured nature of the WSDL descriptions which include data types that can be considered as ontology concepts and operations that can represent ontology relationships.

Patil et al. employ schema matching techniques to select the relevant domain ontology for a WSDL file from a collection of ontologies. Then they annotate the elements of the WSDL file with the concepts of the selected ontology [28]. There are two significant differences in our approach and that suggested in [28]. First, we believe our approach is richer as we consider the XSD data types, interface, operations and messages of WSDL, rather than just the data. Secondly, we use ontology matching techniques for classification as compared to schema matching techniques used by [28]. Ontology matching techniques are more efficient and capture domains more accurately than schema matching techniques, leading to better classification.

Belhajjame et al. show how information can be inferred about the semantics of operation parameters based on their connections to other (annotated) operation parameters within workflows [4]. Bowers and Ludascher present a calculus and two inference algorithms to automatically propagate semantic annotations through workflow actors described by relational queries [8]. Both approaches rely on workflows to create semantic annotations. However, a workflow consists of a sequence of connected services which can include some implementation errors. So this way can drive wrong annotations.

Grcar and Mladenic present a system for semi-automatic annotation of Web service schemas and other resources. The presented system, Visual OntoBridge (VOB), provides a graphical user interface and employs a set of machine learning algorithms to support the user in the annotation task [15]. This approach aims to be applied to WSMO semantic Web services. However, WSMO resolves disparities between concepts with an internal mechanism called mediators. Thus, it does not need any annotation process.

Lerman et al. address the problem of automatically recognizing semantic types of the data used by Web services. They describe a metadata-based classification method for recognizing input data types. Then they use content-based classifiers to recognize semantic types of the output data [22]. In this approach, the WSDL is used as meta-data to annotate the inputs and outputs of services. However, the WSDL file itself needs to be annotated with a semantic model.

Carman and Knoblock introduce a framework for learning Datalog definitions of Web sources. The implemented system actively invokes the sources and compares the data they produce with that of known sources of information. It then performs an inductive logic search through the space of plausible source definitions in order to learn the best possible semantic model for each new source [10]. This framework can contribute to the service discovery more than the service annotation.

Table 6 summarizes the characteristics of the Web service annotation approaches as follows: *(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.

6 CONCLUSION AND FUTURE WORK

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 main goal of this approach is to provide a way to map WSDL

Approach	**Considered elements**	**Annotation resource**	**Techniques**	**Tool**
Belhajjame et al., [4]	Operation parameters	Workflow	Parameter compatibility rules	Annotation Editor
Bouchiha et al., [7]	Complex types and operations names	Domain ontology	Ontology matching	WSDL-S Builder
Hess et al., [18]	Operations, message parts and Data	Domain ontology	Text classification techniques	ASSAM
Patil et al., [28]	Data (Inputs and Outputs of services)	Domain ontology	Schema matching techniques	MWSAF tool
Grcar and Mladenic, [15]	Natural-language query	Domain ontology	Text mining techniques	Visual OntoBridge (VOB)
Lerman et al., [22]	Data (Inputs and Outputs of services)	Meta-data (WSDL)	Machine learning techniques	Semantic labelling tool
Bowers and Ludascher, [8]	Annotation and Query	Workflow	Propagation method	Prolog Implementation
Carman and Knoblock, [10]	Datalog definitions	Source definitions	Inductive logic search	EIDOS

Table 6. Summary of Web service annotation approaches

descriptions to domain ontologies, and therefore to migrate syntactic 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 experiments have been carried out to evaluate this approach and to show the effectiveness of its algorithms. The obtained results were very satisfactory and encouraging, and show that the approach provides a suitable starting point for semantic Web services development.

The nice feature of our algorithms is that they are very generic and can be applied using arbitrary similarity metrics. Thus, other similarity measures, which are not implemented yet, can improve the matching process. Furthermore, statistical techniques, such as *multidimensional scaling strategy* [11], can contribute to improving the categorization process. As a future work, a comparative study will be done between the similarity measures and strategies to see which measure and strategy give best results.

In addition, our algorithms can be adapted for other complex aspects of semantic Web services. For instance, the categorization algorithm can be used to select appropriate ontologies for WSMO [32] and the matching algorithm can be used to generate mediators which resolve disparities between the concepts of these ontologies.

Since elements of more complex Web services would usually refer to concepts from several domain or generic ontologies, the proposed approach can be improved so that several ontologies can be selected and used for annotating the Web services. Thus, the categorization algorithm will compute the similarity between a WSDL document and a set of domain or generic ontologies. The ones for which this algorithm gives a similarity greater than the threshold are candidate to be used in the matching algorithm, which in turn will be modified so that semantic similarities are computed between WSDL elements and all ontologies concepts. Each WSDL document element will be annotated by the nearest concept.

An alternative way to incorporate several ontologies in the annotation process is to keep the categorization process as is, and add directly generic ontologies in the matching process, since they include concepts shared between several domains. However, the priority is given to the domain ontology concepts to refer WSDL elements because they give specific senses of these elements. The matching process can be completed with generic ontologies which include concepts that give generic senses to the WSDL elements when it is necessary.

REFERENCES

[1] Akkiraju, R.—Farrell, J.—Miller, J.—Nagarajan, M.—Schmidt, M.-T.—Sheth, A.—Verma, K.: Web Service Semantics – WSDL-S. Version 1.0, W3C Member Submission, available on: `http://www.w3.org/Submission/WSDL-S/`, November 2005.

[2] Baeze-Yates, R.—Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, ACM Press, Reading, MA. 1999.

[3] Banerjee, S.—Pedersen, T.: Extended Gloss Overlaps As a Measure of Semantic Relatedness. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence 2003, pp. 805–810.

[4] Belhajjame, K.—Embury, S-M.—Paton, N.-W.—Stevens, R.—Goble, C.-A.: Automatic Annotation of Web Services Based on Workflow Definitions. ACM Transactions on the Web (TWEB Journal), Vol. 2, 2008, No. 2.

[5] Bell, D.—de Cesare, S.—Iacovelli, N.—Lycett, M.—Merico, A.: A Framework for Deriving Semantic Web Services. Information Systems Frontiers, Vol. 9, 2007, No. 1, pp. 69–84.

[6] Bouchiha, D.—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.

[7] Bouchiha, D.—Malki, M.—Alghamdi, A.—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.

[8] Bowers, S.—Ludascher, B.: A Calculus for Propagating Semantic Annotations Through Scientific Workflow Queries. Query Languages and Query Processing workshop (QLQP-2006)organised in conjunction with the 10th International Conference on Extending Database Technology 2006, pp. 712–723.

[9] Buitelaar, P.: Ontology Learning for Semantic Web Services. In Proceedings of ONLINE 2003, Dsseldorf, Germany 2003.

[10] Carman, M.-J.—Knoblock, C.-A.: Learning Semantic Definitions of Online Information Sources. Journal of Artificial Intelligence Research, Vol. 30, 2007, pp. 1–50.

[11] Ehrig, M.—Haase, P.—Hefke, M.—Stojanovic, N.: Similarity for Ontologies – A Comprehensive Framework. In 13th European Conference on Information Systems (ECIS), Regensburg (DE) 2005.

[12] Euzenat, J.—Shvaiko, P.: Ontology Matching. Springer-Verlag, Berlin Heidelberg 2007.

[13] Farrell, J.—Lausen, H.: Semantic Annotations for WSDL and XML Schema. W3C Recommendation, 28 August 2007, Available at http://www.w3.org/TR/sawsdl/, 2007.

[14] Fensel, D.: Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer-Verlag 2001.

[15] Grcar, M.—Mladenic, D.: Visual OntoBridge: Semi-Automatic Semantic Annotation Software. In Proceedings of ECML PKDD 2009, Bled, Slovenia, September 2009, Part II, LNAI 5782, pp. 726–729, Springer-Verlag Berlin, Heidelberg 2009.

[16] Gruber, T.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, Vol. 5, 1993, No. 2, pp. 199–220.

[17] Jiang, J.—Conrath D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In Proceedings of International Conference on Research in Computational Linguistics 1977, pp. 19–33.

[18] Hess, A.—Johnston, E.—Kushmerick, N.: ASSAM: A Tool for SemiAutomatically Annotating Semantic Web Services. International Semantic Web Conference 2004, Hiroshima, Japan, pp. 320–335.

[19] Hirst, G.—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, 1998 pp. 305–332.

[20] Larsen, B.—Aone, C.: Fast and Effective Text Mining Using Linear Time Document Clustering. Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 1999, pp. 16–22.

[21] Leacock, C.—Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification. In Fellbaum, C. (Ed.), WordNet: An electronic lexical database, MIT Press 1998, pp. 265–283.

[22] Lerman, K.—Plangprasopchok, A.—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.

[23] Lin, D.: An Information-Theoretic Definition of Similarity. In Proceedings of the International Conference on Machine Learning 1998.

[24] Lin, F.—Sandkuhl, K.: A Survey of Exploiting WordNet in Ontology Matching. In Proceedings of International Federation for Information Processing (IFIP AI 2008), pp. 341–350.

[25] Martin, D.—Paolucci, M.—McIlraith, S.—Burstein, M.—McDermott, D.—McGuinness, D.—Parsia, B.—Payne, T.—Sabou, M.—Solanki, M.—Srinivasan, N.—Sycara, K.: Bringing Semantics to Web Services: The OWL-S Approach. In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, San Diego, California, USA, 2004.

[26] McGuinness, D.-L.—van Harmelen, F.: OWL Web Ontology Language, Overview. W3C Recommendation, 10 February 2004, Available at `http://www.w3.org/TR/owl-features/`, 2004.

[27] Miller, G.-A.: WordNet: An On-Line Lexical Database. International Journal of Lexicography 1990, pp. 235–312.

[28] Patil, A.—Oundhakar, S.—Sheth, A.—Verma, K.: METEOR-S Web Service Annotation Framework. WWW 2004, pp. 553–562, ACM Press 2004.

[29] Patwardhan, S.: Incorporating Dictionary and Corpus Information Into a Context Vector Measure of Semantic Relatedness. Master's Thesis, Univeersity of Minnesota, Duluth 2003.

[30] Pedersen, T.—Patwardhan, S.—Michelizzi, J.: WordNet::Similarity – Measuring the Relatedness of Concepts. Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), pp. 1024–1025.

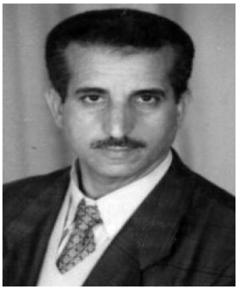
[31] Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In Proceedings of the 14$^{\text{th}}$ International Joint Conference on Artificial Intelligence, pp. 448–453.

[32] Roman, D.—Keller, U.—Lausen, H.—de Bruijn, J.—Lara, R.—Stollberg, M.—Polleres, A.—Feier, C.—Bussler, C.—Fensel, D.: Web Service Modeling Ontology. Applied Ontology, Vol. 1, 2005, No. 1, pp. 77–106.

[33] Sabou, M.—Wroe, C.—Goble C.—Stuckenschmidt, H.: Learning Domain Ontologies for Semantic Web Service Descriptions. Journal of Web Semantics, Vol. 3, 2005, No. 4, pp. 340–365.

[34] Wu, Z.—Palmer M.: Verb Semantics and Lexical Selection. In 32$^{\text{nd}}$ Annual Meeting of the Association for Computational Linguistics 1994, pp. 133–138.

Djelloul Bouchiha received his Engineer degree in computer science from Sidi Bel Abbes University, Algeria, in 2002, and M. Sc. in computer science from Sidi Bel Abbes University, Algeria, in 2005, and Ph. D. in 2011. Between 2005 and 2010, he joined the Department of Computer Science, Saida University, Algeria, as a Lecturer. He became an Assistant Professor since January 2011. Currently, he is an Assistant Professor at the University Centre of Naama. His research interests include semantic web services, web reverse-engineering, ontology engineering, knowledge management and information systems.



Mimoun Malki graduated with Engineer degree in computer science from National Institute of Computer Science, Algiers, in 1983. He received his M. Sc. and Ph. D. in computer science from the University of Sidi Bel-Abbes, Algeria, in 1992 and 2002, respectively. He was an Associate Professor in the Department of Computer Science at the University of Sidi Bel-Abbes from 2003 to 2010. Currently, he is a Full Professor at Djillali Liabes University of, Sidi Bel-Abbes, Algeria. He has published more than 50 papers in the fields of web technologies, ontology and reverse engineering. He is the Head of the Evolutionary Engineering and Distributed Information Systems Laboratory. Currently, he serves as an editorial board member for the International Journal of Web Science. His research interests include databases, information systems interoperability, ontology engineering, web-based information systems, semantic web services, web reengineering, enterprise mash up and cloud computing.

Abdullah Alghamdi is a full time professor, SWE Department, College of Computer and Information Sciences, KSU, Riyadh, KSA. He holds a Ph. D. in software engineering from the department of computer science, Sheffield University, UK, 1997. He got a Post-Doc certificate from University of Ottawa, Canada, where he conducted a joint research at the MCRLab during academic year 2004/2005. He worked as a full and part time consultant with governmental and private organizations in the field of IS strategic planning and defense systems and headed a number of committees inside and outside KSU. He recently published a number of papers in C4I and Enterprise Architecture Frameworks fields. Currently he is the Head of Software Engineering Department, KSU and Director of the national C4I Center for Advanced Systems (C4ICAS).

Khalid Alnafjan received his Bachelor degree in information systems from King Saud University, Saudi Arabia, in 1991, his M. Sc. in computer science from Sheffield University, United Kingdom, in 1994, and PhD in computer science in 1998 from the same university. From 1999 to 2009 he joined the Department of Computer Technology, Riyadh College, GOTEVOT as an Assistant Professor, and from 2010 to date he is an Associate Professor in the Department of Software Engineering, King Saud University. His research interests include software engineering education, software quality assurance, C4i quality assurance and semantic web services.