A Semantic Distance Measure for Matching Web Services

Arif Bramantoro, Shonali Krishnaswamy, and Maria Indrawan

School of Computer Science and Software Engineering,
Monash University, Victoria, Australia
Bramantoro@gmail.com
{Shonali.Krishnaswamy, Maria.Indrawan}@infotech.monash.edu.au

Abstract. A key issue in web services is matching that involves comparing user requests with advertised services and finding the best available ones. In semantic web services, an ontology is used by the matching system to determine the semantic relationship between the requests and the registered services. In this paper, we propose that the semantic relationship can be measured quantitatively in order to provide a more precise similarity measures between the requested and advertised services and to produce a better ranking of relevant services. We proposes and develops a Semantic Distance Measure that is tailored to provide a quantitative measure that indicates similarity between advertised and requested services. We establish that such a measure is an effective means of discriminating services at a level of granularity that is able to enhance the matching process in semantic web services.

1 Introduction

Web services are loosely coupled and reusable software components that can be distributed over internet technologies and open standards [1]. A critical step in the process of web services is matching. The main task of web services matching is to compare user requests with advertised services and to find the best available ones. To bring the matching of web services to its success, there is a need for a language to describe web services content and a matching algorithm that is able to recognize when a user request matches an advertised service [3]. The existing work for web services is based on XML syntax to describe web services content and to provide keyword-based matching. This work lacks well-defined semantics and has therefore led to the on-going research based on the semantic web services [4].

In semantic web services, data have a structure and an ontology describes the semantic of the data [5]. An ontology defines a conceptualization of a domain related to concepts, attributes, and relations [6]. The concepts provide model entities of interest in the domain. They are typically structured into a taxonomy tree where each node represents a concept and each concept has its parent as general concepts [5].

In semantic web services matching, service providers can advertise their web services via a well-defined description language and ontology, such as DAML-S [7] and OWL-S [8]. The matching system then allows services requesters to upload their requests which are encoded in specific description language and ontology as well. From this point, the matching system determines the relationship between the requests and the registered services in an ontology.

M. Dean et al. (Eds.): WISE 2005 Workshops, LNCS 3807, pp. 217 – 226, 2005.

There have been several matching techniques developed for matching semantic web services such as Colucci et al. [9], Elgedawy [10], Wang and Stroulia [11], Paolucci et al. [3] and Pahl and Casey [13]. These techniques differ in their support for variant result sets, extent of semantic support, degrees of matching and presentation of ranked results. These matching strategies exploit the semantic relation available between advertised and requested services. However, the main limitation of these works is the lack of quantitative measures for specifying the extent of similarity. Current semantic web services matching techniques have degrees of matching which are discrete and at a coarse level of granularity. A quantitative measure provides a more precise measurement and therefore can produce a better ranking of relevant services. Another advantage of measuring the similarity between services quantitatively is a finer level of granularity in matching result. In fact, while similarity and distance between services in ontology implicitly exist, none of the observed techniques is able to provide quantitative measures of similarity between concepts in the matching process.

This paper proposes that matching in semantic web services can be enhanced through the use of measures that quantify the “semantic distance” [12] between concepts in web services ontology. We demonstrate that a matching process based on Semantic Distance Measures will overcome the issues discussed above by refining and quantifying the degrees of the matching.

2 Semantic Distance Measures in Web Services Matching

In this paper, we contend that semantic distance can represent in quantitative terms the degree of matching between a service request and a service advertisement. There are four degrees of similarity in [3] determined by minimal distance between concepts in ontology. We now illustrate how these degrees of similarity can be strengthened by precise quantitative measures of similarity.

In Paolucci’s model [3], the first degree of similarity is *exact match*. It returns an advertised service that is the same as a requested service. For example, if a user is looking for the service that sells Sedan, then an advertised service that sells Sedan is considered as *exact match*. *Exact match* in [3] also returns an advertised service which is the parent of the requested service. For example in figure 1, if a user is looking for the service that sells a Sedan, then the advertised service that sells a Car is also considered as an *exact match*. Paolucci et al. in [3] argue that this case of matching can be considered as *exact match* since by advertising a Car, provider will commit to provide every service which is subclass of that service. However, this assumption is not always true as it is possible to provide only certain sub-classes (e.g. only Sedans and no SUVs).

The second degree of matching is termed *plug-in match*. It returns an advertised service that is a grandparent of a requested service. It is not specified in [3] whether the great-grandparent or great-great-grandparent services are considered also as plug-in or not. For example in figure 1, if an SUV is requested, then according to the definition of plug-in, a service which advertises its capability as Vehicle will be returned.

The third degree of matching is termed subsumes match. It returns an advertised service that is a child of a requested service. For example in figure 1, if a Car is requested, then according to the definition of subsumes match, a service which advertises its capability as Sedan will be returned. In this degree of matching, there is no difference between advertised services which are a child and a grandchild of requested service, regardless of which level the advertised services is at.

The last degree of matching is termed *fail match*, which shows no subsumption relation (parent-child relation) between an advertised service and a requested service. Even though an advertised service and a requested service have the same parent (or grand parent) in ontology, they are still considered as *fail match* in [3]. For example in figure 1, if the requested service is a Sedan and the advertised service is a Bus, then a *fail match* is returned.

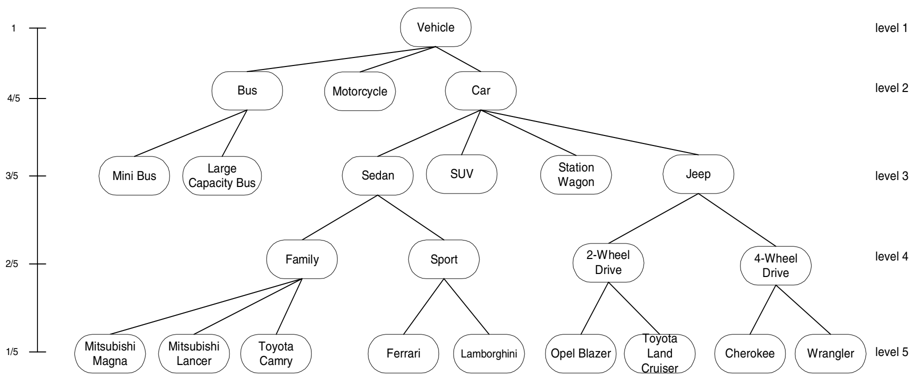

Fig. 1. The Example of Vehicle Ontology

We identify from above discussion of Paolucci's model that there are certain limitations. Firstly, a more precise measurement can produce a better ranking of relevant services. The matching result should not be only in the form of four discrete degrees, but should be at a finer level of granularity. For example, the Vehicle ontology in figure 1 shows that Sedan is semantically closer than Ferrari with respect to Vehicle. However, in [3] both services which sell Sedan and Ferrari are considered as *subsumes match* if the requested service is Vehicle.

Secondly, we identify that not all of the services without a subsumption relation should be considered as fail match. Especially, the services which have the same parents should not be considered as fail match. For example, the service which sells Lamborghini should be ranked as some degree of matching to a service which sells Ferrari. This is because there is possibility that a user who wants to buy Ferrari may also considers buying Lamborghini since they are both Sports cars.

We anticipate that a matching process based on Semantic Distance Measures will overcome these issues by refining and quantifying the degrees of the matching.

Therefore, we formulate the notions of our proposed model as follows.

1. The highest degree of matching is *exact match* where the requested service and the advertised service are exactly the same.
2. A requested service and an advertised service which has the child-parent relationship in the ontology should be higher in similarity than these services

which have the parent-child relationship. For example, if Sports car is requested, then a service which advertises Sedan should be higher in similarity than a service which advertises Ferrari. It can be seen from the Vehicle ontology in figure 1 that Sport is a child of Sedan and Sport is a parent of Ferrari.
3. The services in a lower level of the ontology should be ranked relatively closer to one another than the services in the higher level of ontology. According to Sussna's observation in [14, 15], concepts which are located deeper in the tree tend to be more closely related to one another than those higher in the tree. Therefore, we consider assigning a different weight for different level in the ontology. For example in figure 1, the services which sell Ferrari and Lamborghini (level 5 in the ontology) should be closer to one another than the services which sell Sedan and SUV (level 3 in the ontology).
4. The services which have the same parent should be considered closer than the services with the grandparent-grandchild relation. For example, the relationship between Ferrari and Lamborghini should be ranked closer than the relation between Ferrari and Sedan.

These above considerations form the basis of our SDM for matching web services. As we can see from the example ontology in figure 1, the similarity or the distance between services implicitly exists. Therefore, there is a need to exploit this similarity or distance by the use of quantitative and explicit measures. By using semantic distance, the similarity or the relatedness between concepts can be measured. The concepts refer to a particular sense of words or services in term of web services. According to Budanitsky in [12], there are three principal approaches that can be used to measure the distance between concepts dictionary-based, Thesaurus-based, and semantic networks. [12]. According to Lee et al. in [16], semantic networks are defined as "any representation interlinking nodes with arcs, where the nodes are concepts and the links are various kinds of relationships between concepts." Since this view of a semantic network has structured similarity to web services ontologies, we will focus the discussion only on this approach. In semantic networks, WordNet [17] is widely used as the encoding of lexical knowledge.

2.1 Incorporating Semantic Distance Measures in the Matching Process

As it is not always the case that an *exact match* to a requested service exists, every advertised service in the ontology should be able to be measured against requested services quantitatively. However, we need to satisfy all the requirements for matching web services as discussed previously. There are several measures of Semantic Distance that have been developed. In this research, we modify the SDM proposed by Hirst and St-Onge in [18] to facilitate application in the context of semantic web services. There are three major relations in Hirst and St-Onge's framework, i.e.: extra-strong, medium-strong, and strong. The definition of medium-strong is defined by the following formula:

$$\text{weight} = C - \text{path length} - k \times \text{number of changes of direction} \tag{1}$$

where C and k are constants, C set to 8 and k set to 1[1]. By using the formula in equation 1, Hirst and St-Onge emphasize on the length of the path and the direction changing. It can be inferred from the formula that the longer the path the lower the weight and the more changes of direction the lower the weight.

We now explain our proposed modification of the Semantic Distance formula in equation 1 proposed by Hirst and St-Onge's in [18]. In their paper, the formula is used to measure the relations between nouns in Word-Net. However, our SDM algorithm applies the measurement for services in web service ontology. The concepts of nouns in [18] is replaced with web services inputs and outputs parameters while the taxonomy tree of WordNet in [18] is replaced with the ontology of web services.

The following formula can be used to have semantic distance measurement as well as satisfying all the requirements for matching web services.

- Let S_1 be the requested service from user and S_2 be the advertised service.
- Let lw be the level weight for each path in ontology. It depends on the depth of the ontology. To count the level weight lw, we use the following formula: $lw = \frac{(n - \ln - 1)}{n}$ where ln is the level of the node in the ontology.

  For example, the Vehicle ontology in figure 1 has five levels. Therefore, we weigh the edges in the topmost level as 1, the second level as 4/5, the third level as 3/5, the fourth level as 2/5 and the lowest level as 1/5.
- Let C be the constant. We follow Hirst and St-Onge in setting C to 8. The logical explanation for this value is for not neglecting other values in the formula
- Let $PathLength$ be the number of edges counted from service S_1 and S_2 in the ontology. For example the service Ferrari and SUV have the value of $PathLength = 4$
- Let $NumberOfDownDirection$ be the number of edges counted between service S_1 and S_2 which direction is downward. For example figure 1, Ferrari and SUV have 1 value of $NumberOfDownDirection$, while SUV and Ferrari have value of 3

Our proposed formula for measuring the semantic distance between service S_1 and S_2 is:

$$sdm(s1,s2) = C - lw * PathLength - NumberOfDownDirection \quad (2)$$

The formula in equation 2 only works for measuring the distances between services in the single inheritance ontology. While this research focuses on the matching between services in a single ontology, we present an extension to our proposed SDM to facilitate its use in multiple inheritance ontologies. The formula to count Semantic Distance Measure for matching web services which use multiple inheritance ontologies is as follow:

$$sdm(s1,s2) = C - lw * PathLength - NumberOfDownDirection + mcp(s1,s2) \quad (3)$$

[1] We communicated personally with Hirst to obtain these values.

The difference between the formula in equation 2 and 3 is the additional `mcp(s1,s2)` in equation 4. The `mcp(s1,s2)` is used to measure the number of most common parents in multiple inheritance ontologies. The formula is as follow:

$$\text{mcp (s1;s2)} = -\log \sum_{l=1}^{n} \frac{NumberOfMostCommonParents(s_1, s_2)_l}{l \times n} \tag{4}$$

where n is the depth of ontology which is measured in nodes and `NumberOfMostCommonParents` is the number of parents in level l. This formula is useful to measure the distance between two services which are not only related in one ontology, but also present relations in other ontologies.

3 Evaluation

We implemented a matching engine based on our proposed SDM to conduct evaluations. The matching engine uses the following technologies in its implementation:

There are four supporting technologies used by SDM Matching Engine. The supporting technologies used in our SDM Matching Engine architecture are outlined as follows:

- **JDOM** [21]is an XML parser for Java. It provides a simple API for XML (SAX) and XML Document Object Model (DOM).
- **Jena [22]**is a Java framework for supporting semantic web applications. *Jena* is an open source software developed by HP Semantic Labs. *Jena* provides a programmatic environment for RDF, RDS and Web Ontology Language (OWL).
- **Jess** [23] s a Java expert system shell (*Jess*) and serves as a rule engine and scripting environment written in Java. *Jess* is developed by the Sandia National Laboratories. We use *Jess* to store the web service ontologies and to query the relationship between web services in the ontology.
- **OWLJessKB [24]** is a successor to *DAMLJessKB* which is a description logics reasoner for ontologies written in DAML+OIL. *OWLJessKB* provides extended reasoning for OWL by utilizing *Jena* and *Jess*. It provides a Java API that supports the reading and retrieval of web service ontologies (OWL).

To perform the evaluation, we only consider one parameter of each service for one matching process even though our SDM application can accommodate multiple parameters and combine the matching result from input and output matching. The reason for matching only one parameter is we want to focus on the matching process. By matching one parameter we hope that we can simplify comparison of the matching results between our approaches.

We use the simple example of a Vehicle Selling Service (shown in Figure 1) to show how SDM is used to match between a requested service and an advertised service. The Vehicle Selling Service is also used in Paolucci's model [3]. In our SDM, the Vehicle Ontology used by the Vehicle Selling Service is extended with several concepts. A larger ontology is needed to demonstrate the benefits of our SDM. Figure 1 shows the Vehicle ontology used by Vehicle Selling Service. The root concept of the Vehicle ontology is *Vehicle*. Each concept in this ontology has a level weight. The deeper a concept is in the ontology the level weight tends to decrease as shown in figure 1. For example, the concept *Bus*, *Motorcycle* and *Car* have level weight of $^4/_5$. On the other hand, the level weight for *Sedan*, *SUV* and *Station Wagon* is $^3/_5$. In order to focus on the matching process, we only examine the output parameters which are based on the Vehicle ontology. We assume that all cases use *Price* as the input of the services. It means that to be able to run the services, the user should input the *Price* value of the required vehicle.

3.1 Case 1

Table 2 shows the result of SDM matching between a requested and advertised service in case 1.

Table 2. Matching Result for Case 1

Requested service:	
- input:	Price
- output:	Sedan
Advertised service:	
- input:	Price
- output:	Sedan
Result from matching:	
- SDM input:	100 %
- SDM output:	$(8-(^3/_5+^3/_5)*0-0)/8 * 100\% = (8/8) * 100\% = 100\ \%$
- In Paolucci's model:	exact match

In this case, the user wants to buy a *Sedan* while the advertised service sells *Sedan* also. Since both requested and advertised service have the same input and output services, it returns a perfect value (100%). This result also shows that the concepts of input and output parameters are located in the same position in the ontology. According to Paolucci [3], this case returns a matching degree of *exact match*.

3.2 Case 2

Table 3 shows the result of SDM matching between a requested and advertised service in case 2. In this case, the user wants to buy *Sedan* while the advertised

Table 3. Matching Result for Case 2

Requested service:	
- input:	Price
- output:	Sedan
Advertised service:	
- input:	Price
- output:	Car
Result from matching:	
- SDM input:	100 %
- SDM output:	$(8-(^3/_5+^4/_5)*1-0)/8 * 100\% = (6.6/8)*100\% = 82.5\%$
- In Paolucci's model:	exact match

service sells *Car*. Since *Sedan* is a child of *Car* in the Vehicle ontology, it returns 82.5%. According to Paolucci [3], this case still returns a matching degree of *exact match*. It is clear that a quantitative measure based on Semantic Distance, allows explicit articulation between differences such as case 1 and this case.

3.3 Case 3

Table 3 shows the result of SDM matching between a requested and advertised service in case 4. In this case, the user wants to buy *Sedan* while the advertised service sells *Sport*.

Table 3. Matching Result for Case 3

Requested service:	
- input:	Price
- output:	Sedan
Advertised service:	
- input:	Price
- output:	Sport
Result from matching:	
- SDM input:	100 %
- SDM output:	$(8-({}^{3}/_{5}+{}^{2}/_{5})*1-1)/8 * 100\% = (6/8)*100\% = 75\%$
- In Paolucci's model:	subsumes match

Since *Sedan* is a parent of *Sport* in the Vehicle ontology, it returns 75%. It is noteworthy that in comparing this result with case 2, we see that through SDM, the relationship of similarity between a *Sedan* and a *Sports* car is weaker than the relationship between a *Sedan* and a *Car*. It is evident that a *Sports* car has certain specific semantics that can not be generalized to all *Sedans*, while in case 2 a *Sedan* is a *Car* and a *Sedan* is a less specific requirement than a *Sports* car.

3.4 Case 4

Table 4 shows the result of SDM matching between a requested and advertised service in case 5. In this case, the user wants to buy *Ferrari* while the advertised

Table 4. Matching Result for Case 4

Requested service:	
- input:	Price
- output:	Ferrari
Advertised service:	
- input:	Price
- output:	Sport
Result from matching:	
- SDM input:	100 %
- SDM output:	$(8-({}^{1}/_{5}+{}^{2}/_{5})*1-0)/8 * 100\% = (7.4/8)*100\% = 92.5\%$
- In Paolucci's model:	exact match

service sells *Sport*. Since *Ferrari* is a child of *Sport* in the Vehicle ontology, it returns 92.5%. This result has a higher degree of match than case 2 since the concepts are located in a lower part of ontology. According to Paolucci [3], this case returns a matching degree of *exact match*. Again, it can be seen that we are able to in this case establish a high degree of similarity without terming it as an *exact match*, which it is not.

We performed several other evaluation cases that we are unable to present in this paper due to space considerations. We also performed evaluation with other ontologies that illustrate similar results of the benefits of quantitative similarity measures in matching semantic web services.

4 Conclusion and Future Work

In this paper, we have proposed a formula to measure the Semantic Distance between services through concepts specified in an ontology. We established that Semantic Distance Measures are suitable for determining similarity between requested and advertised web services, thereby facilitating in performing the task of matching in the selection process. Our approach exploited service profiles that are available and provides a quantitative means of specifying the extent to which an advertised service meets the requirements of a user's request. This is significant because it allows us to return a finer level of granularity for each degree of matching. In summary, the primary contribution of our research is that we have developed a Semantic Distance Measure to provide a quantitative similarity measures to support matching in semantic web services. Currently, our model does not support multiple inheritance ontologies. Therefore, the enhancement of our model to incorporate Semantic Distance Measurement between services in multiple inheritance ontologies is a proposed extension.

References

1. Wang H, Huang JZ, Qu Y, et al. Web Services: Problems and Future Directions. In *Web Semantics: Science, Services and Agents on the World Wide Web, April 2004*. Vol1(3), pp. 309-320.
2. McIlraith SA, Son TC, Zeng H. Semantic Web Services. In *IEEE Intelligent Systems, 2001*. Vol16(2). pp 46-53.
3. Paolucci M, Kawmura T, Payne T, et al. Semantic Matching of Web Services Capabilities. In *First International Semantic Web Conference, June 2002, Sardinia, Italy*. pp. 333-347.
4. Berners-Lee T, Hendler J, Lassila O. The Semantic Web. In *Scientific American, 2001*. Vol284 (5). pp. 34–43.
5. Doan A, Madhavan J, Dhamankar R, et al. Learning to Match Ontologies on the Semantic Web. In *VLDB Journal, Special Issue on the Semantic Web, November 2003*. Vol12(4), pp. 303-319.
6. Fensel D. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. *Springer-Verlag, 2001, Berlin*.
7. DAML Services Coalition (alphabetically A. Ankolekar MB, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, H. Zeng). DAML-S: Semantic Markup for Web Services. In *Proceedings of the International Semantic Web Working Symposium (SWWS), July 30-August 1, 2001, Stanford University, CA, USA*.

8. OWL-S Coalition. OWL-S 1.0 Release. Available at http://www.daml.org/services/owl-s/1.0/ (last accessed November 05, 2005).
9. Colucci S, Di Noia T, Di Sciascio E, et al. Description Logics Approach to Semantic Matching of Web Services. In *Proceedings of the 25th International Conference on Information Technology Interfaces 2003 (ITI 2003)*, *June 16-19, 2003, Dubrovnik, Croatia*. pp. 545-550.
10. Elgedawy I. A Conceptual Framework for Web Services Semantic Discovery. In *Proceedings of On The Move (OTM) to meaningful internet systems, pages 1004–1016, Italy, 2003. Springer Verlag*.
11. Wang Y, Stroulia E. Flexible Interface Matching for Web-Service Discovery. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering 2003 (WISE 2003), December 10-12 2003, Roma, Italy*. pp. 147-156.
12. Budanitsky A. Lexical Semantic Relatedness and Its Application in Natural Language Processing. Technical Report, *CSRG-390, August 1999*.
13. Pahl C, Casey M. Ontology support for web service processes. Proceedings of the 9th European software engineering conference held jointly with 10th ACM SIGSOFT international symposium on Foundations of software engineering. pp. 208 - 216 2003.
14. Sussna M. Word Sense Disambiguation for Free-Text Indexing Using a Massive Semantic Network. In *Proceedings of the Second International Conference on Information and Knowledge Management 1993 (CIKM-93), Arlington, Virginia*. pp. 67-74.
15. Sussna MJ. Text Retrieval Using Inference in Semantic Metanetworks. PhD thesis, University of California, San Diego, 1997.
16. Lee JH, Kim MH, Lee YJ. Information Retrieval Based on Conceptual Distance in IS-A Hierarchies. In *Journal of Documentation, June 1993.* Vol49(2). pp. 188-207.
17. Miller GA. Nouns in WordNet. In *Christiane Fellbaum, editor, WordNet: An Electronic Lexical Database*, chapter 1, pp. 23-46. The MIT Press, Cambridge, MA, 1998.
18. Hirst G, St-Onge D. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In *Christiane Fellbaum, editor, WordNet: An Electronic Lexical Database, chapter 13, pp. 305-332*. The MIT Press, Cambridge, MA. 1998.
19. http://www.daml.org/services/owl-s/1.1/BravoAirService.owl (last accessed April 25, 2005).
20. http://www.daml.org/services/owl-s/1.1/CongoService.owl (last accessed April 25, 2005).
21. Hunter J, McLaughlin B., JDOM API Javadoc Specification. Technical Report, 2004. Available at http://www.jdom.org/docs/apidocs/ (last accessed June 2, 2005).
22. McBride B. Jena: Implementing the RDF Model and Syntax Specification. In *Proceedings of the Semantic Web Workshop (WWW2001), May 2001, Hongkong, China*.
23. Friedman-Hill E. Jess in Action. Manning Publications Company, October 2003.
24. Kopena J, Regli W. DAMLJessKB: A Tool for Reasoning with the Semantic Web. In *IEEE Intelligent System, May/June 2003*. Vol18(3). pp. 74-77.