A Heuristic Approach to Semantic Web Services Classification

Miguel Ángel Corella and Pablo Castells

Universidad Autónoma de Madrid, Escuela Politécnica Superior
Campus de Cantoblanco, 28049 Madrid, Spain
{miguel.corella, pablo.castells}@uam.es

Abstract. Web service technologies, and the vision of semantically-enhanced services, aim to be key enablers to further exploit the potential of the Web as a platform where units of functionality can be deployed, shared, and assembled in a much more flexible way than it is possible today. Due to the expectable growth of the number of services offered in the WWW, the need for service repositories and mechanisms for their (semi)automatic organization and discovery of services is becoming increasingly important. In this paper, we propose a heuristic-based mechanism that enables service publishers to (semi)automatically classify their services in a service taxonomy managed by a service repository.

1 Introduction

Since the emergence of the semantic web [3], many research efforts have been aiming to use semantics to endow web services with a much higher potential for automation. These efforts have resulted in a new research trend called semantic web services [16]. The basis of this trend is to attach some semantic information to current WSDL – based web services descriptions [6] in order to enable their analysis and manipulation by software programs. This manipulation would be useful to enact powerful capabilities such as service automatic selection, invocation, composition or discovery.

Nowadays, UDDI [12] is the most widely accepted and used protocol for publishing and searching services over the web. These actions are usually performed using UDDI registries, which can be seen as service repositories easily accessed through a URL. In these registries, the published services are classified using some taxonomy (e.g. UNSPSC – United Nations Standard Products and Service Code, NAICS – North American Industry Classification System, etc.). Nevertheless, this classification is performed manually by a human publisher. Due to the huge quantity of classes in service taxonomies, the classification process is usually complex and costly. Furthermore, taxonomies are subject to evolution, or even complete replacement by new ones, making even heavier the maintenance effort load on repository administrators.

The central point of our approach is to provide automatic mechanisms to help service publishers in the classification task. We propose a heuristic that provides a ranked list of service categories in which the new published service fits best.

B. Gabrys, R.J. Howlett, and L.C. Jain (Eds.): KES 2006, Part III, LNAI 4253, pp. 598 – 605, 2006.

The paper is organized as follows: Section 2 introduces some previous work in the domain of web service classification and other areas related to the work presented here. Section 3 introduces the problem of web service classification and shows the scenario in which our work takes place. Section 4 presents some ideas demonstrating why web service semantics are needed in our classification approach. The complete presentation and explanation of our heuristic classification approach is described in Section 5. Finally, Section 6 presents some conclusions and outlines future work.

2 Related Work

Existent web service classification proposals may be divided into heuristic (e.g. [13]) and non-heuristic approaches (e.g. [5] and [9]). We briefly discuss next the most relevant initiatives in this scope related to our research.

In [9], two different approaches to web service classification are presented: a) selecting the category based on the service information (using Natural Language Processing, machine learning and text classification techniques) and b) creating categories based on service information (using clustering techniques). In this proposal the classification is based on extracting relevant words from service descriptions, and using them to build term vectors for classification mechanisms (e.g. Naïve Bayes), in such a way that the service classification problem is solved by a text classification approach.

In [5], the classification process follows the same steps than in [9], but Support Vector Machines are used as the classification method. In addition, service publishers are provided with some information (a concept lattice extracted using Formal Concept Analysis over service descriptions) about how the words used in their descriptions contribute to the selection of a specific category.

In [13], a framework to (semi)automate the semantic annotation of web services (i.e. the attachment of semantic information to web service descriptions, such as parameter description based on ontology concepts, service classification, etc.) is presented. A matching algorithm between web service data types and ontology concepts is defined (based on matching element schemas) in order to obtain a degree of similarity between services and domain ontologies. Having as many domain ontologies as service categories, they classify a service by finding the ontology that yields a higher similarity value when compared with the service.

Another related area to our work is that of service matchmaking (e.g. [10] and [14]). It is somehow related to web service classification but differs in the final objectives. Our research aims to find similarity degrees between services in order to assign them to a common category, admitting some degree of fuzziness in the matching. In contrast, work on service matchmaking follows a strictly boolean approach, aiming to find a service that matches some capabilities in order to e.g. invoke it or compose it into a more complex process.

3 The Service Classification Problem

Service categorization is commonly used to facilitate service retrieval, be it by manually browsing service repositories, or by automatic discovery mechanisms. Classification taxonomies can be extremely large, comprising thousands of categories, within multiple hierarchical levels. Additionally, the number of services in a repository can grow quite large. Furthermore, the placement of a service under a proper category requires a considerable amount of knowledge of the taxonomy, the service characteristics, the application domain, the overall organization of the repository, implicit guidelines, etc., in order to make good classification decisions. As a consequence, the task often becomes overwhelming for repository administrators. Our work aims at alleviating the administrator's work by automatically providing her/him with a smaller set of likely appropriate taxonomy categories (ranked by likelihood of appropriateness), when a new service has to be registered in the repository.

We approach the service classification task as follows. Given a set of service descriptions, already classified under some taxonomy, and a new service description, we propose a heuristic for automated service classification, based on the comparison of the unclassified service with the set of already classified services, whereby a measure of the likelihood that the service should be assigned a certain category is computed.

In our approach we make the following assumptions:

- A service taxonomy or classification is available, i.e. a set of different service types (e.g. currency exchange, flight information retrieval, etc.) arranged in some logical way (e.g. a list, a tree with parent-child relations, etc.). These could be standards commonly used for this purpose, such as UNSPSC or NAICS.
- A service registry is in place, where services are stored and classified in advance (manually or under human supervision).This could be an UDDI registry which seems to be the most widely accepted standard. The repository should be populated with classified services, which will serve as a basis for new ones classification.

4 The Need for Service Semantics

WSDL descriptions provide us with details about the operations a service provides, as well as the input and output information involved. While this enables comparisons already, the comparison would be greatly enhanced if the descriptions were enriched with further semantic information, as we discuss next. Consider this example: take a currency conversion service and a service that computes the expected time a trip between two cities would take. Since the WSDL description of a service only includes functional information (i.e. the description of its interface), at this level we can only compare service elements such as operations, messages, data types, etc. Let us say the currency conversion service has only one operation, which takes as inputs an amount of money (of type double) and two currency codes (of type string), and returns as output the converted amount (a double). On the other hand, the trip time calculator has also one operation taking as inputs an average speed (a double) and two city names (strings), and returns as output the estimated trip time (a double). It is easy to see that using only the WSDL descriptions (without using NLP mechanisms) we would get a very high similarity value between these services, since their interface is (syntactically) exactly the same. Since the similarity measure between services is a critical point in our approach, cases like this one would lead to misclassification.

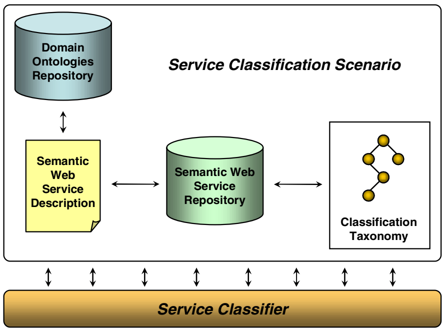
Domain Ontologies Repository
Service Classification Scenario
Semantic Web Service Description
Semantic Web Service Repository
Classification Taxonomy
Service Classifier

Fig. 1. Overview of the main elements involved in our approach to web service classification

Therefore, more information is needed in the descriptions, semantic information enabling to make the difference between e.g. a currency code and a city name. Thus we advocate here for using semantic web services, which raises two further issues: the choice of a semantic description language and the availability of domain ontologies where the concepts involved in a service are defined.

Any of the currently available semantic service description languages, such as OWL-S [11], WSMO [15], WSDL-S [1] and SWSO [2], could be used in our approach, since our heuristic is language agnostic. We only assume a set of domain ontologies is available as a common vocabulary for all the classified services.

5 The Service Classification Heuristic

As has already been introduced earlier, our approach to service classification is based on the comparison of a new unclassified service with a set of already classified services. Our heuristic can be divided into three different levels of granularity, each one corresponding to the comparison between two elements involved in the classification process: service to category similarity measure, service to service similarity measure, and, finally, concept to concept similarity measure. These are explained in detail next.

5.1 Service to Category Similarity

The comparison between a service and a category should provide evidence that a service should belong to a taxonomy category. The proposed similarity measure works as follows. Let $\mathcal{S}$ be the set of all web services in a repository, and let $\mathcal{C}$ be the taxonomy used to classify the services in the repository. If we allow a service to be classified under several categories of the taxonomy, we may define the classification by $\mathcal{C}$ as a mapping $\tau : \mathcal{S} \rightarrow 2^{\mathcal{C}}$. Given a new service s to be added to $\mathcal{S}$, we want to find the categories in $\mathcal{C}$ that best suit s. Given $c \in \mathcal{C}$, let P(s:c) be the probability that c is an appropriate classification for s. We define an estimate for this probability by comparison of s with all the services classified under c. With this aim, if we take P(s:c) ~ 0 if

$\{x\in \mathcal{S} \mid c\in \tau(x)\} = \emptyset$ (i.e. c is disregarded as a potential category for s if there is no previous service $s\in \mathcal{S}$ classified under c), we can write:

$$\mathrm{P}(s:c) \sim \mathrm{P}\left(s:c \wedge \left(\bigvee_{x\in\mathcal{S}} c\in \tau(x)\right)\right)$$

By rewriting the right hand-side, it can be seen that:

$$\mathrm{P}(s:c) \sim \sum_{A\subset\mathcal{S}} (-1)^{|A|+1} \prod_{x\in A} \mathrm{P}\big(c\in\tau(x)\big)\cdot \mathrm{P}\big(s:c \mid c\in\tau(x)\big)$$

provided that $s{:}c \wedge c\in\tau(x)$ are pairwise independent for all $x\in\mathcal{S}$. Since $c\in\tau(x)$ is true iff $x\in \{x\in\mathcal{S} \mid c\in\tau(x)\}$, and assuming a crisp service classification (i.e. $c\in\tau(x)$ is either true or false, as opposed to fuzzy classification where $\mathrm{P}(c\in\tau(x)) \in [0,1]$), we have: Now we shall estimate $\mathrm{P}(s{:}c \mid c\in\tau(x))$ by a measure of similarity sim (s, x), that is:

$$\mathrm{P}(s:c) \sim \sum_{A\subset\tau^{-1}(c)} (-1)^{|A|+1} \prod_{x\in A} \mathrm{sim}(s,x)$$

whereby the appropriateness of a category for a service is computed in terms of the similarity between the service and the services classified under that category. The measure of the similarity between two services will be defined in the next subsection.

Finally, $\mathrm{P}(s{:}c)$ is taken as a score value for ranking the categories according to their predicted appropriateness for s. Note that $\mathrm{P}(s{:}c) \in [0,1]$ provided that sim $(s, x) \in [0,1]$ and increases monotonically with respect to sim (s, x).

5.2 Service to Service Similarity

The similarity between two services is measured in terms of the similarity of their operations and parameters. Let $\mathcal{P}$ be the set of all the parameters of the services in $\mathcal{S}$, and $\mathcal{OP}$ the set of service operations. If we denote by $P_s \subset \mathcal{P}$ the set of the parameters of service s, and by $OP_s \subset \mathcal{OP}$ the set of its operations, the similarity is defined as:

$$\mathrm{sim}(s,s') = f\left(\mathrm{sim}(P_s,P_{s'}),\mathrm{sim}(OP_s,OP_{s'})\right)$$

Developing a measure for comparing the complete set of parameters (despite the operation they belong to) between services is still work in progress in our research at the time of this writing, so in the meantime we are working with $f(x,y)=y$.

The similarity between the sets of operations (OP and OP') of two services is computed as the average of the best possible pairwise similarities obtained by an optimal pairing of the elements from the two sets. We define the similarity between two operations as:

$$\mathrm{sim}(op,op') = \mathrm{sim}\left(I_{op},I_{op'}\right)\cdot \mathrm{sim}\left(O_{op},O_{op'}\right)$$

where I_{op}, $I_{op'}$, O_{op} and $O_{op'}$ are the set of input and output parameters of the operations *op* and *op*' respectively. The similarity between two parameter sets is computed in turn as the average of the best possible pairwise similarities obtained by an optimal pairing of the elements from the two sets. The similarity between two parameters is defined as the similarity of their types, which are assumed to be classes in a domain ontology. The similarity between concepts in an ontology has been widely studied (see e.g. [4], [7], [8]). In the next section we describe our own proposal, suited to the purpose of concept comparison in our context.

Note that the service comparison defined here returns values in [0,1], provided that the similarity between ontology concepts is also within that range.

5.3 Concept to Concept Similarity

The similarity measure between concepts describing service parameters is key to our heuristic, since the previously defined comparisons are layered on top of this measure. We define it as follows.

Let $\mathcal{T}$ denote the set of all concepts in the domain ontology. The similarity between two concepts is measured in terms of their distance in the ontology class hierarchy. Given two concepts $t \in \mathcal{T}$ and $t' \in \mathcal{T}$, let t_0 be the lower common ancestor to t and t' in $\mathcal{T}$, and let $d = \text{dist}(t,t_0) + 1$, $d' = \text{dist}(t',t_0) + 1$ be the number of levels (plus 1) between t, t' and t_0 in the concept hierarchy. We define the similarity between t and t' as:

$$\text{sim}(t,t') = \left(1 - \frac{\alpha}{\text{h}(\mathcal{T})} \cdot \frac{|d-d'|}{d+d'}\right) \cdot \frac{1}{\min(d,d')} \cdot \left(1 - \frac{\max(d,d')-1}{\text{h}(\mathcal{T})}\right)$$

where:

- h($\mathcal{T}$) is the total height of the concept hierarchy, which is introduced to measure the distance between concepts as a proportion of the total depth of the ontology.
- The term $\frac{|d-d'|}{d+d'}$ increases (that is, the similarity decreases) with the difference in the depth level between t and t'. Note that the similarity would seem to increases with the length d + d' of the shortest hierarchical path between them, but this is compensated by dividing by min(d,d'), in a way that the similarity is essentially not sensitive to the depth of the concepts.
- $\alpha \in [0,1]$ is a parameter that ensures a minimum non-zero similarity value, even for the most dissimilar concepts, in a way that the similarity ranges in some interval [*min*,1] above 0, in order to relax the influence of the measure in the heuristic.
- The factor $1 - \frac{\max(d,d')-1}{\text{h}(\mathcal{T})}$ is introduced to reinforce the decrease of the similarity when the concepts are in the same branch of the ontology hierarchy.

Figure 2 shows how the similarity function sim(*t*,*t*') depends on the distance *d* and *d*' of the *t* and *t*' to their lowest common ancestor.

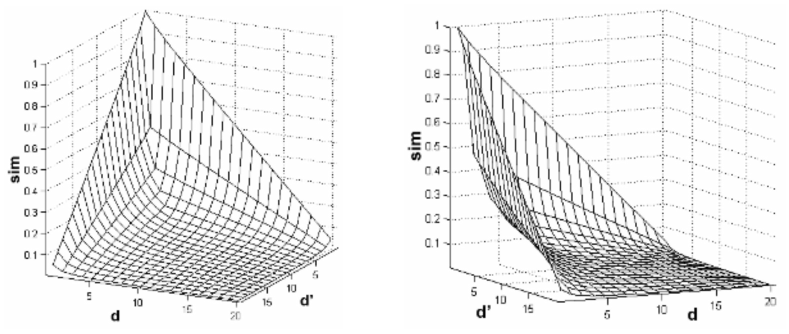

Fig. 2. Concept to concept similarity measure graph (displayed from two angles) showing the behavior of the similarity measure with respect to *d* and *d*' in a twenty depth levels ontology

6 Conclusions and Further Work

We have developed a set of similarity measures, which assembled together result in an effective heuristic for the (semi)automatic classification of semantic web services. This heuristic offers and alternative mechanism to the ones explored in previous research, such as the usage of NLP techniques [5] [9], and the classification based on WSDL descriptions [13], to which our proposed approach is complementary. Our recursive definition of the similarity measures for service classification by successive levels has the advantage of modularity, in the sense that the measures can be studied and optimized at each level with a reasonable degree of independence. The main limitation of our approach is that it relies on the existence of a critical mass of semantic web service descriptions, which do not yet abound nowadays. Nevertheless, as the semantic web is gaining momentum, we rely on the growth and spread of such corpora, as a hypothesis for our research.

At the time of this writing, we are testing and refining all the measures defined here in order to achieve the best classification success rate. So far, we have performed several experiments over the concept to concept similarity measure obtaining good results. The next steps in our research, apart from finishing complete heuristic test and evaluating the results, include the creation of a sufficient repository of semantic web service descriptions, in order to obtain some realistic performance results of our approach and test its scalability.

Acknowledgements

This research was supported by the Spanish Ministry of Science and Education (TIN2005-0685) and the Ministry of Industry, Tourism and Commerce (CDTI05-0436).

References

1. Akkiraju, R., Farrel, J., Miller, J., Nagarajan, M., Schmidt, M., Sheth, A., Verma, K: Web Service Semantics – WSDL-S, Technical Note, Version 1.0, 2005.
2. Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., Kifer, M., Martin, D., McIlraith, S., McGuiness, D., Su, J., Tabet, S.: Semantic Web Service Ontology (SWSO), Version 1.0, 2005.
3. Berners – Lee, T., Hendler, J., Lassila, O.: The semantic web. In Scientific America, 2001.
4. Bernstein, A., Kaufmann, E., Bürki, C., Klein, M.: How similar is it? Towards personalized similarity measures in ontologies. In the 7. Internationale Tagung Wirtschaftsinformatik. February 2005, pp. 1347 – 1366.
5. Bruno, M., Canfora, G., Di Penta, M., Scognamiglio, R.: An approach to support web service classification and annotation. In Proceedings of the IEEE International Conference on e – Technology, e – Commerce and e – Services (EEE 2005), Honk – Kong, 2005.
6. Christiensen, E., Curbera, F., Meredith, G., Weerawarana, S.: Web Service Description Language (WSDL), v1.1
7. Culmore, R., Rossi, G., Merelli, E.: An ontology similarity algorithm for BioAgent. In NETTAB 02 Agents in Bioinformatics, Bologna, 2002.
8. Ehrig, M., Haase, P., Stojanovic, N.: Similarity for ontologies – a comprenhensive framework. In Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability, at PAKM 2004. December 2004.
9. Heβ, A., Kushmerick, N.: Automatically attaching semantic metadata to Web Services. In Workshop on Information Integration on the Web (IIWeb2003), Acapulco, Mexico, 2003.
10. Li, L., Horrocks, I.: A software framework for matchmaking based on semantic web technology. In the International Journal of Electronic Commerce, 8(4):39 – 60. 2004.
11. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Scycara, K. OWL-S: Semantic markup for web services, v1.1, 2004.
12. OASIS: UDDI: The UDDI technical white paper, 2004.
13. Oldham, N., Thomas, C., Sheth, A., Verma, K.: METEOR-S Web Service Annotation Framework with Machine Learning Classification. In Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC'04), California, July 2004.
14. Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Semantic Matching of Web Service Capabilities. In Proceedings of the First International Semantic Web Conference, 2002.
15. Roman, D., Lausen, H., Keller, U., de Brujin, J., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Kifer, M., König-Ries, B., Kopecky, J., Lara, R., Oren, E., Polleres, A., Scicluna, J., Stollberg, M.: Web Service Modeling Ontology (WSMO), 2005.
16. Terziyan, V. Y., Kononenko, O.: Semantic web enabled web services: State-of-the-art and industrial challenges. In Proc. International Conference on Web Services (ICWS), 2003.