Abstract

The research activity in the semantic web services area is very active in proposals of solutions for calculating similarity between services. The similarity is measured in different dimensions, i.e. input/output parameters, QoS, correlations between preconditions and effects, etc. This work describes an approach to measuring the degree of similarity between semantic web services considering multiple dimensions as a whole. We propose the integration of results obtained from different matching algorithms in a generic and customizable manner.

Key words: Semantic Web Services, Matching, Ontology.

1. Introduction

Service-Oriented Architecture (SOA) defines a software architecture style based on the definition of reusable services, where service providers and client consumers interact each other in a decoupled way.

A service is an abstract specification of one or more business operations, which provides sufficient details for the client to consume it, preferably on an independent platform way, such as the Web. One possibility to describe this specification is using the Web Service Description Language (WSDL) [10], which describes Web Services by the structure of their messages and operations, among other things. However, this description only deals with syntactic aspects and does not allow to describe, for example, what the service actually does or the semantics of its parameters. To address these limitations, the Semantic Web Services technology (SWS) [15, 17] has emerged.

The growing popularity of web services available on the Web poses a new challenge: given a request service, find the service that best fit the user’s needs. In the literature, this problem is referred to as semantic matching.

Example (BirthCertificate services) Suppose an application with the following concepts: Citizen who has a name, an identification number and a birth date, Uruguayan who is a Citizen born in Uruguay; Montevidean who is an Uruguayan born in Montevideo city; BirthCertificate which is a legal document signed by the Uruguayan Government that certifies the date and place where a citizen was born. Additionally, suppose there are two services: $s_1$ who receives as input a Montevidean, returns as output a BirthCertificate for that citizen, and it has a cost of U$S 10; and $s_2$ who receives as input an Uruguayan, returns as output a BirthCertificate and has no cost. We can describe these services using WSDL.

Suppose:

<table>
<thead>
<tr>
<th>Service</th>
<th>Input Concept</th>
<th>Output Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_1$</td>
<td>Montevidean</td>
<td>BirthCertificate</td>
</tr>
<tr>
<td>$s_2$</td>
<td>Uruguayan</td>
<td>BirthCertificate</td>
</tr>
</tbody>
</table>

We know both services ($s_1$ and $s_2$) have as output a birth certificate, but one uses a String type to represent it, while the other uses an array of bytes. In a purely syntactic way these services are not equivalent, when actually in a semantic way they are. This shows that service matching based on WSDL is rather limited. Adding semantic descriptions, i.e., annotating corresponding parameters with Citizen and BirthCertificate concepts, solves the mismatches:

Moreover, these concepts (citizen, uruguayan, montevideoan) must belong to a shared domain ontology (OWL [1]) that represents the subsumption between the concepts.

One of the great challenges of the Semantic Web is to automatically discover, compose and execute services [16]. Services, such as buy books, book flights, convert file formats, etc., are currently implemented as Web Services with semantic descriptions. For this purposes standards like OWL-S [2], WSDL-S [5] and WSMO [6] have emerged, expanding the possibilities to achieve this challenge. Semantic Web Service descriptions may also have non-functional
aspects regarding, for example, quality-of-service (QoS). Therefore, semantic description brings us the possibility to compare services and to decide which ones are more similar considering all these aspects.

The research activity in this area is giving a great number of approaches and solutions for calculating similarity among services. Therefore, we think it would be useful to have a framework to integrate these approaches.

Our proposal, is to provide such a framework for calculating global similarity in a generic and customizable way. This global similarity is the result of integrating single similarities calculated by different approaches that we will call dimensions. This framework is customizable in the sense that the user may determine the relative importance of each dimension and obtain a single rank for candidates services matching a given service. Therefore, our work does not focus on semantic matching algorithms itself, but on an environment in which it could be able to reuse existing matching algorithms.

To achieve this goal, we introduce the concept of similarity vector to measure the degree of similarity between web service descriptions. This vector belongs to a n-dimension space where each dimension represents similarity degree for a specific aspect of the semantic web service description. We also introduce the idea of a similarity profile in which it is possible to indicate the use of certain dimensions and the choice of specific implementations for calculating local and global similarity. We have also defined an ontology to represent our multidimensional semantic web service matching approach, allowing similarity profile specification in a flexible and generic way.

The rest of the paper is organized as follows. Section 2 gives a brief review of related works that complements and positions our work. Section 3 presents our form to express similarity. Section 4 presents our proposed ontological model for Multi-Dimensional Matching, called MDM-Onto. This section also presents an example of tailoring the MDM-Onto to be used with two concrete dimensions. Finally, Section 5 brings some conclusions and describes future work.

2. Multidimensional Semantic Matching

The most common strategies to establish a semantic matching between two web services are the study of the following aspects of their semantic descriptions: (1) semantic correspondence between their input/output parameters, (2) correlation between their preconditions and effects, (3) QoS’s values (4) Web Service categorization and (5) to compare the existing relationships between input and output parameters. However, since it is yet an open problem, others aspects may be defined as well. In our proposal, we model these aspects as dimensions. In the rest of this paper we will use the term dimension to refer to a specific aspect of web services semantic description.

Most of the existing proposals address the matching problem from a single point of view, as mentioned in [14], taken into account one dimension or an hybrid approach combining some of them. The first and most studied matching dimension is the analysis of semantic correspondence between input/output parameters of request and advertisement services. Some seminal work on this dimension are [19], [21], [12]. A most recent work by D. Hull et al. [13] introduces a new dimension for stateless web services, establishing a semantic relationship between it’s input and output parameters using SWRL. This approach is meant to be used in conjunction with the classic inputs/outputs matching. The authors argue that considering only semantic correspondence between input/output parameters is not enough to achieve an acceptable degree of precision and recall in the matching.

Another dimension on functional service parameters is the correlation between preconditions and effects. Many works on semantic rules languages based on logic like SWRL [4] and RuleML [3] are contributing to the development of implementations for this dimension. In particular, [18] uses preconditions and effects of Web services to explore automatic composition. This approach compares services by their compositional characteristics, such as: if the execution of a service enables the execution of another, if two services may be replaceable or if the services offered by a service are included by another. To achieve this, traditional approaches verify if the compliance of a condition implies the compliance of another condition.

A dimension that is starting to have more relevance in semantic matching, is the quality of service (QoS). Multiple approaches are made in this area and all of them depend on different factors such as the business domain, the requirements of the users, the way users define metrics and of course the development of the evaluation process. All of these characteristics define the Quality Model of a Service, which covers the aspects that identify, characterize and group the elements that are used in the quality evaluation process. This model also defines the parameters, metrics and functions that will allow to make an expression of the QoS [20]. Although there are several works that focus on defining these models, there is not a standard model to describe the QoS yet, which difficult the inclusion of this aspect in the categorization of services [8].

On the other side, works like Pernici et al. [9] and Yuchen Fu et al. [11] propose a multistrategic approach based on several aspects of Web Services. Pernici’s work proposes an ontology based methodology for service discovery analyzing the correspondence between input/output, service categorization, QoS and context of the user. Yuchen’s work proposes a similar multistrategic
matching also using Web Services’ parameters and its Quality characteristics, but includes the correlation between preconditions and effects, instead user context and service categorization matching. Both of them takes into account individual similarities to then obtain a global similarity. The main disadvantage of both approaches is that they provide a rigid strategy to get the global similarity.

Our proposal, is to provide a framework that calculates the global similarity through the integration of local similarity values in a generic and customizable manner. Each local similarity is obtained by applying an algorithm on one of the different dimensions.

3. Dimensions & Degree of Similarity

Given a request service, in order to discover and choose an appropriate advertised service, we need to calculate the degree of similarity between the advertised and the requested services. As it was introduced in Section 2, there are different dimensions or aspects to be considered. In this section we present a way to express the degree of similarity taking into account multiples dimensions.

From the I/O parameters dimension point of view, the process of matching semantic web services is essentially based on the use of logic inference to check the equivalence or subsumption relationships between the ontology classes that correspond to the parameters in the service description. Typically, five types of matching are identified: exact, if the request is equivalent to the advertisement; plug-in, if the request is subsumed by the advertisement; subsume, if the request subsumes the advertisement; intersection, if the intersection of the request and the advertisement is satisfiable; and disjoint, otherwise.

At the QoS dimension, the process of matching semantic web services is quite different, because it focus on concepts of QoS to find the advertised service that has equal or greater quality value than the requested service. In this case, the degree of matching is not expressed in a discrete scale, such as the five types of matching discussed above, but as a continuous value in the range [0..1], where 1 means the maximum similarity and 0 means there is not similarity at all. This allows to handle cases where a large number of candidate services provides the same type of matching, i.e., partial matches can be appropriately ranked. Moreover, in order to express the similarity among different dimensions we consider a similarity dimensional space. The similarity between two services is then expressed as a vector in this space. The magnitude of this vector represents the measure of the overall similarity between both services.

Definition (Dimensional Similarity) Let $V$ be a $n$-dimensional vector space over $R$. Let $s_1$, $s_2$ and $s_3$ be the semantic descriptions of three services from our universe of discourse $S$, those we want to compare. We define the dimensional similarity in the dimension $i$ as a function:

$$sim_i : S \times S \rightarrow [0..1], \ i : 1..n$$

where: $sim_i(s_1, s_2) < sim_i(s_1, s_3) \iff s_3$ is more similar to $s_1$ than $s_2$ for the dimension $i$

Definition (Similarity Vector) Given n dimensions and two services $(s_1, s_2)$ we define their similarity vector as:

$$sim(s_1, s_2) = (v_1, v_2, \ldots, v_n)$$

where $v_i = sim_i(s_1, s_2), \ i : 1..n$

Under these conditions we can define the concept of similarity between two services as the euclidean norm of the corresponding similarity vector.

Definition (Similarity) We define the similarity between two services $s_1$ and $s_2$ as:

$$\| sim(s_1, s_2) \|_2$$

Definition (Ranking) Extending the semantic of each sim, we can say that:

For $s_1, s_2, s_3$ service descriptions

$s_3$ is more similar to $s_1$ than $s_2 \iff \| sim(s_1, s_2) \|_2 < \| sim(s_1, s_3) \|_2$

The sim function defines an ordering relation between the similarities of all advertised services ($s$) and the requested service ($r$). Therefore, it can be defined a set containing the services most similar to $r$.

$$\mathcal{S} = \{ s \in S/ \| sim(r, s) \|_2 = max( \bigcup_{s' \in S} \{ \| sim(r, s') \|_2 \}) \}$$

Weighting dimensions: The euclidean norm can be taken as a comparison metric under the hypothesis that each dimension has the same weight in comparison to the overall similarity. However, in real world applications some dimensions may be more critical than others. In this sense, weighting means reduce the incidence of one dimension on the calculus of the norm. The dimension that is less weighted contributes less to the norm. We define the weighted norm as follows, but more complex formula could be defined.

Definition (Weighting Norm)

$$\| \vec{v} \|_p = \sqrt{(p_1 v_1)^2 + \ldots + (p_n v_n)^2}$$

Where each $p_i \in [0..1]$ is the weight assigned to the dimension $i$. This weights are parameters of the model.
Note that, although matching algorithms use metrics based on different methodologies, such as ranking, graduated, ordinal, linguistic, and so on, the degree of similarity can always be represented as a numeric value between 0 and 1 through a normalization function.

4. A Model for Multidimensional Matching

In this section we present our proposed Multi-dimensional Matching Ontology called MDM-Onto, and an example of tailoring the MDM-Onto to be used with two concrete dimensions: semantic correspondence between input/output parameters and QoS.

4.1. The MultiDimensional Matching Ontology

In order to get the values of similarity from different dimensions in a flexible and consistent way, the first step is to specify a formal model that represents the elements involved in the acquisition of the similarity. Our approach to do this challenge is the design of an ontological model inspired by the works of Qurator project on biological data quality [7], and also from work in the area of QoS [20]. However, we differentiate from these projects in the sense that our proposal is to model a generic ontology to capture multidimensional similarities between web services descriptions according to different matching algorithms. We call this ontology: MDM-Ontology (briefly MDM-Onto). Our generic MDM-Onto in spite of its high level abstraction it is easily tailored to different user’s dimensions and different algorithms for each dimension. In addition to the valuable property of checking the consistency among concepts and relationships, the ontological model provides a high level abstraction that allows specifying in simple way relations between factors and metrics.

Figure 1 depicts, in a simplified way, our proposed ontology for representing the context to assess similarity of services. The SimilarityVector is defined on a set of relevant dimensions given by the user. In order to remark the user decision on the desired dimensions we re-called this concept as SimilarityProfile. A similarity profile allows to describe the way a service ranking is made. These profiles group similarity dimensions, that represent aspects of the services that are taken into account to carry out the semantic matching. In turn, each dimension use certain elements from the semantic service descriptions to calculate the degree of similarity on matching algorithms. For example, the Input/Output dimension uses the input and output parameters as basic elements, whereas the quality dimension uses quality factors such as performance, availability, scalability, etc. [22]. Therefore, every dimension of similarity is defined by the elements involved in it.

Thus, we define the concept SimilarityDimension which represents the dimensions taken into account in the profile to make the matching. The relationship between the profile and its dimensions are represented by the property involvesDimension.

The elements involved in the comparison of similarity for each dimension are represented in a general way with the concept SimilarityElement. The property usesElement, represents the link between dimensions and elements described earlier. Each dimension must be related to at least one element with this property.

Each dimension defined in this model performs semantic matching according to a particular dimension (IO, QoS, etc.). The result of this comparison must be provided by a service which is tightly coupled to it and implements it’s matching algorithm. This service has as input two service descriptions (the requested and advertised services) and returns a semantic matching value (the degree of similarity) between them according to it’s related dimension. These concepts are represented by the class DimensionService and the property computeService. Any instance of SimilarityDimension is related to a single instance of DimensionService through the property computeService. As mentioned earlier, the output of this service is the result of the matching by the associated dimension. This result is represented on the ontology by the class SimilarityResult and property hasSimilarityResult. Any service that wants to be classified with this approach is represented by the concept CService. These services can play the role of request or advertisement depending on how they are used. This relationship is represented by the properties hasRequest and hasAdvertisement of the concept DimensionService. These concepts formalize the notion that exists a service that calculates the similarity of the dimension and receives two services as input parameters and returns a result of similarity between them.

The complete MultiDimensional Matching Ontology (MDM-Onto) developed in OWL can be accessed through the home page of CAMALEON Project at http://www.fing.edu.uy/inco/grupos/csi/Proyectos/index.html.
4.2. A MDM-Onto Profile

In this section we show an example of making a profile of the MDM-Onto, i.e., tailoring the MDM-Onto, selecting I/O parameters and QoS dimensions. The tailoring process is basically to extend the class SimilarityDimension with the concrete dimensions and to extend the class SimilarityElement with the concepts should be used to calculate the similarity for each dimension.

4.2.1 Tailoring I/O parameters dimension

In this case we want to add the I/O parameters dimension to the model, so we define IODimension subclass of SimilarityDimension and InputElement and OutputElement subclass of SimilarityElement.

Therefore, an IODimension is a SimilarityDimension characterised by using only InputElement or OutputElement to get the semantic matching result value. Figure 2 shows tailored MDM-Onto with the I/O parameters dimension. The following OWL-DL definition formalize this concept.

\[
\text{Definition} \quad \text{IODimension} \equiv \text{SimilarityDimension} \land \
\forall \text{usesElement} (\text{InputElement} \lor \text{OutputElement}) \land \
\exists \text{usesElement} \text{InputElement} \land \
\exists \text{usesElement} \text{OutputElement}
\]

4.2.2 Tailoring QoS dimension

Similarly as we did with the I/O dimension, we are going to define the QoS dimension in the model. We define QoSCostFactorDimension that extends SimilarityDimension and QoSElement that extends SimilarityElement.

Therefore, a QoSDimension is a SimilarityDimension characterised by using only QoSCostFactorElement to get the semantic matching result value. Figure 2 shows tailored MDM-Onto with the QoS dimension. The following OWL-DL definition formalize this concept.

\[
\text{Definition} \quad \text{QoSDimension} \equiv \text{SimilarityDimension} \land \
\forall \text{usesElement} \text{QoS} \text{CostFactorElement} \land \
\exists \text{usesElement} \text{QoS} \text{CostFactorElement}
\]

4.3. An example

To illustrate these notions we reuse the example presented in Section 1 extended with a third service. \(s_3\) who receives as input a Monteviean, returns as output a BirthCertificate and it has cost of 7.5 dollars. Figure 2 show an instantiated tailored MDM-Onto where we present a SimilarityProfile with the following two dimensions: IODimension and QoSDimension. Each one has its SimilarityElement, i.e. IODimension uses InputElement and OutputElement, whereas QoSDimension uses QoS Element. Suppose we have a DimensionService implementing IODimension using Paolucci et al [19] matching algorithm and another DimensionService implementing QoSDimension using an specific cost metric formula to compare two service costs.

We want to know the degree of similarity that \(s_2\) has with respect to \(s_1\) and the degree of similarity that \(s_3\) has with respect to \(s_1\), and finally compare them to know which one of \(s_2\) and \(s_3\) is the most similar to \(s_1\).

To achieve this, we have to calculate two similarity vectors: \(\text{sim}(s_1, s_2) = \vec{v}_2\) and \(\text{sim}(s_1, s_3) = \vec{v}_3\). Using the instance ontology and algorithm mentioned before it will arise to the vectors: \(\vec{v}_2 = <0.75, 1>\) and \(\vec{v}_3 = <1, 0.75>\). If our model uses the standard euclidean norm to calculate the overall similarity, we arise to the result that both services have the same degree of similarity: \(\|\vec{v}_2\|_2 = \|\vec{v}_3\|_2 = 1, 25\). One solution to this, is using the weighting norm defined in the section 3 which introduces the notion of weight associated to a dimension. Suppose that \(s_1\) is not a critical service for our application and we prefer to spend less money. Under this hypothesis, we can give more weight to the QoSDimension over the IODimension. The final expression is:

\[
\|\vec{v}_2\|_w = \sqrt{(p_{IO}.0, 75)^2 + (p_{QoS}.1)^2}
\]

and

\[
\|\vec{v}_3\|_w = \sqrt{(p_{IO}.1)^2 + (p_{QoS}.0, 75)^2}
\]

Let be \(p_{IO} = 0.8\) the weight of IODimension and \(p_{QoS} = 1\) the weight of QoSDimension.

\[
\|\vec{v}_2\|_w = 1, 20 >\|\vec{v}_3\|_w = 1, 16
\]
As we can see, after the weighting of the dimensions $s_2$ is better than $s_3$, even having worse input output matching. Remember that this is a very simplified example and we assume that we only have the cost of the service as quality factor. In real world we will have a lot of factors to measure the quality of service and we may need to weight each factor individually.

5. Conclusions and Future Work

We developed an OWL ontology for multidimensional matching that models the elements involved in the matching of services among multiple dimensions in a flexible way. Together with this ontology we defined a global concept of similarity among multiple dimensions. The proposed model presents a unified vision that makes it possible to implement a generic framework, which can be used for the following issues: (1) Detect relationships between dimensions, identifying the impact a specific algorithm used in some dimension could have over another dimension; (2) Test that two implementations for the same dimension should lead to the same result; (3) Tune the weights and similarity functions for a given similarity profile taking into account the generated knowledge.

The above issues remark the idea that a formal description of our matching model would allow us to analyze the results looking for a continuous improvement in the model’s quality. At the moment, we are developing a prototype and we plan to make some experimentations with it.

6. Acknowledgments

We thank our colleagues Fernando Carpani and Verónica Peralta at Instituto de Computación, Universidad de la República, for their comments and suggestions. We also thank the financial support from the PDT-Dinacyt program under grant 42/36.

References