Weight Assignment of Semantic Match using User Values and a Fuzzy Approach

Simone A. Ludwig

Department of Computer Science
University of Saskatchewan
Canada
ludwig@cs.usask.ca

Abstract. Automatic discovery of services is a crucial task for the e-Science and e-Business communities. Finding a suitable way to address this issue has become one of the key points to convert the Web into a distributed source of computation, as it enables the location of distributed services to perform a required functionality. To provide such an automatic location, the discovery process should be based on the semantic match between a declarative description of the service being sought and a description being offered. This problem requires not only an algorithm to match these descriptions, but also a language to declaratively express the capabilities of services. The proposed matchmaking approach is based on semantic descriptions for service attributes, descriptions and metadata. For the ranking of service matches a match score is calculated whereby the weight values are either given by the user or estimated using a fuzzy approach.

1. Introduction

Dynamic discovery is an important component of Service Oriented Architecture (SOA) [1]. At a high level, SOA is composed of three core components: service providers, service consumers and the directory service. The directory service is an intermediary between providers and consumers. Providers register with the directory service and consumers query the directory service to find service providers. Most directory services typically organize services based on criteria and categorize them. Consumers can then use the directory services' search capabilities to find providers. Embedding a directory service within SOA accomplishes the following, scalability of services, decoupling consumers from providers, allowing updates of services, providing a look-up service for consumers and allowing consumers to choose between providers at runtime rather than hard-coding a single provider.
However, SOA in its current form only performs service discovery based on particular keyword queries from the user. This, in majority of the cases leads to low recall and low precision of the retrieved services. The reason might be

that the query keywords are semantically similar but syntactically different from the terms in service descriptions. Another reason is that the query keywords might be syntactically equivalent but semantically different from the terms in the service description. Another problem with keyword-based service discovery approaches is that they cannot completely capture the semantics of a user's query because they do not consider the relations between the keywords. One possible solution for this problem is to use ontology-based retrieval.

A lot of related work on semantic service matching has been done [2,3,4,5,6] however, this approach takes not only semantic service descriptions into account but also context information. Ontologies are used for classification of the services based on their properties. This enables retrieval based on service types rather than keywords. This approach also uses context information to discover services using context and service descriptions defined in ontologies.

The structure of this paper is as follows. The next section describes in detail the matching algorithm, match score calculation with weight values and the fuzzy weight assignment. In section 3, a summary of the findings and directions for future work are described.

2. Matching Algorithm

The overall consideration within the matchmaking approach for the calculation of the match score is to get a match score returned which should be between 0 and 1, where 0 represents a "mismatch", 1 represents a "precise match" and a value in-between represents a "partial match". The matchmaking framework [3] relies on a semantic description which is based on attributes, service descriptions and metadata information. Therefore, the overall match score consists of the match score for service attributes, service description and service metadata respectively:

$M_O = \frac{M_A + M_D + M_M}{3}$, whereby M_O, M_A, M_D, M_M are the overall, attribute, description and metadata match scores respectively.

Looking at the service attributes first, it is necessary to determine the ratio of the number of service attributes given in the query in relation to the number given by the actual service. To make sure that this ratio does not exceed 1, a normalization is performed with the inverse of the sum of both values. This is multiplied by the sum of the number of service attributes matches divided by the number of actual service attributes shown below. Similar equations were derived for service descriptions and service metadata respectively. The importance of service attributes, description and metadata in relation to each other is reflected in the weight values.

$$M_A = \frac{w_A}{(n_{AQ} + n_{AS})} \cdot \frac{n_{AQ}}{n_{AS}} \cdot \frac{n_{MA}}{n_{AS}}, \quad M_D = \frac{w_D}{(n_{DQ} + n_{DS})} \cdot \frac{n_{DQ}}{n_{DS}} \cdot \frac{n_{MD}}{n_{DS}},$$

$$M_M = \frac{w_M}{(n_{MQ} + n_{MS})} \cdot \frac{n_{MQ}}{n_{MS}} \cdot \frac{n_{MM}}{n_{MS}}$$

whereby w_A, w_D and w_M are the weights for attributes, description and metadata respectively; n_{AQ}, n_{AS} and n_{MA} are the number of query attributes, service attributes and service attribute matches respectively; n_{DQ}, n_{DS} and n_{MD} are the number of query descriptions, service descriptions and service description matches respectively; n_{MQ}, n_{MS} and n_{MM} are the number of query metadata, service metadata and service metadata matches respectively.

Match Score with User Weight Assignment (UWA)

The user defines the weight values for service attributes, descriptions and metadata respectively, based upon their confidence in the “search words” used.

Match Score with Fuzzy Weight Assignment (FWA)

Fuzzy weight assignment allows for uncertainty to be captured and represented, and helps the automation of the matching process.

Fuzzy logic is derived from fuzzy set theory [7,8,9,10] dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. It can be thought of as the application side of fuzzy set theory dealing with well thought out real world expert values for a complex problem. [11]. Fuzzy logic allows for set membership values between and including 0 and 1, and in its linguistic form, imprecise concepts like "slightly", "quite" and "very". Specifically, it allows partial membership in a set.

A fuzzy set A in a universe of discourse U is characterized by a membership function $\mu_A : U \to [0,1]$ which associates a number $\mu_A(x)$ in the interval $[0,1]$ with each element x of U. This number represents the grade of membership of x in the fuzzy set A (with 0 meaning that x is definitely not a member of the set and 1 meaning that it definitely is a member of the set).

This idea of using approximate descriptions of weight values rather than precise description is used in this approach. First, we have to define a membership function each for w_A, w_D and w_M. The fuzzy subset of the membership function for service attributes can be denoted as such $A = \{(x, \mu_A(x)\} \quad x \in X, \mu_A(x) : X \to [0,1]$. The fuzzy subset A of the finite reference super set X can be expressed as $A = \{x_1, \mu_A(x_1)\}, \{x_2, \mu_A(x_2)\}, ..., \{x_n, \mu_A(x_n)\}$; or $A = \{\mu_A(x_1)/x_1\}, \{\mu_A(x_2)/x_2\}, ..., \{\mu_A(x_n)/x_n\}$ where the separating symbol / is used to associate the membership value with its coordinate on the horizontal axis. The membership function must be determined first. A number of methods learned from knowledge acquisition can be applied here. Most

practical approaches for forming fuzzy sets rely on the knowledge of a single expert. The expert is asked for his or her opinion whether various elements belong to a given set. Another useful approach is to acquire knowledge from multiple experts. A new technique to form fuzzy sets was recently introduced which is based on artificial neural networks, which learn available system operation data and then derive the fuzzy sets automatically.
Fig. 1 shows the membership functions for service attributes, description and metadata respectively. The comparison of the three membership functions shows that it is assumed that service attributes are defined in more detail and therefore there is less overlapping of the three fuzzy sets weak, medium and strong. However, for service description and also metadata the overlap is significantly wider allowing the user a larger “grey area” where the weight values are defined accordingly.

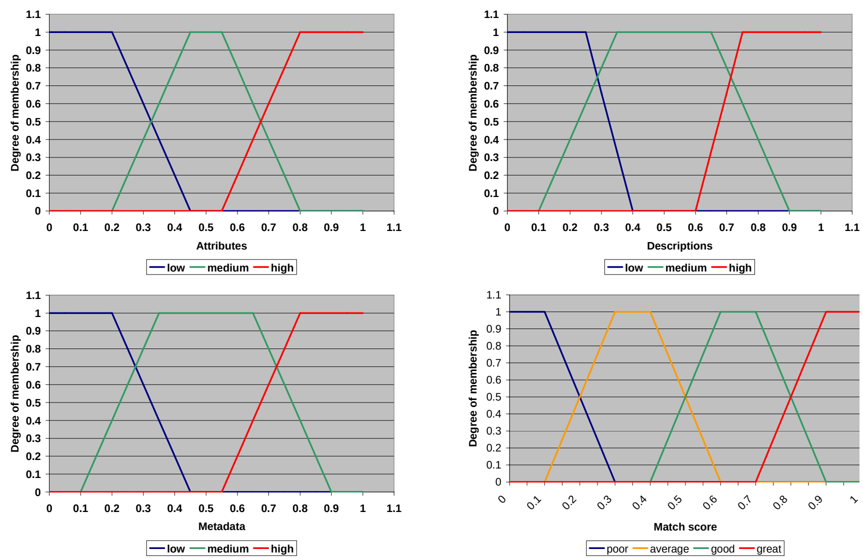


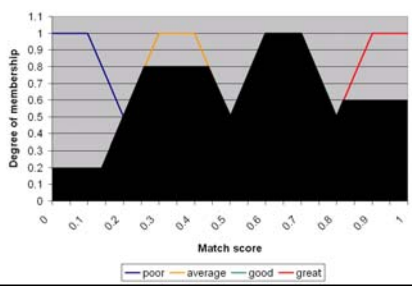
Fig. 1. Membership function of the fuzzy sets for service attributes, descriptions, metadata and match score

In order to do the mapping from a given input to an output using the theory of fuzzy sets, a fuzzy inference must be used. There are two fuzzy inference techniques – Mamdani [12] and Sugeno [13]. The Mamdani method is widely accepted for capturing expert knowledge. It allows describing the expertise more intuitive. However, Mamdani-type inference entails a substantial computational burden. On the other hand, the Sugeno method is computationally effective and works well with optimization and adaptive techniques, which makes it very attractive in control problems. For this investigation, the Mandami inference was chosen because of the fact that it better captures expert knowledge. In 1975, Mandami built one of the first fuzzy systems to control a steam engine and boiler combination by applying a set of fuzzy rules supplied by experienced human operators. The Mamdani-style inference process is performed in four steps which are fuzzification of the input variables, rule evaluation, aggregation of the rule outputs and finally defuzzification.

The four fuzzy rules for service attributes (A), description (D), metadata (M) and match score (MS) are defined as:

R1: IF *A=low* AND *D=low* AND *M=low* THEN *MS=poor*
R2: IF A=medium AND D=low AND M=medium THEN MS=average
R3: IF A=medium AND D=medium AND M=medium THEN MS=good
R4: IF A=high AND D=high AND M=high THEN MS=great

Let us assume a user's query results in the match values M_A=0.4, M_D=0.5 and M_M=0.7 with $w_A = w_D = w_M$ =1.

1. Fuzzification: $\mu_{(a=low)} = 0.2$ $\mu_{(a=medium)} = 0.8$ $\mu_{(d=medium)} = 1$ $\mu_{(m=medium)} = 0.8$ $\mu_{(m=high)} = 0.6$	2. Rule Evaluation: $\mu_{A \cap D \cap M}(x) = \min[\mu_A(x), \mu_D(x), \mu_M(x)]$ R1: $\mu = 0.2$ R2: $\mu = 0.8$ R3: $\mu = 1.0$ R4: $\mu = 0.6$
3. Aggregation  	4. Defuzzification using centroid technique: $COG = \frac{\int_a^b \mu_A(x) x dx}{\int_a^b \mu_A(x) dx} = 0.614$

The evaluated match score is 0.614 for the given example.

3. Conclusion

The contextual information enhances the expressiveness of the matching process, i.e. by adding semantic information to services, and also serves as an implicit input to a service that is not explicitly provided by the user. The introduction of match scores serves as a selection criterion for the user to choose the best match. Two different approaches to calculate the match score were shown whereby one used precise weight values assigned to service attributes, description and metadata, and the second approach showed the usage of fuzzy descriptions for the weight values. The first

approach is semi-automatic as the user needs to provide the weight values by entering the query, resulting in a confidence value of how good the user thinks the entered query attributes were chosen. The second approach with the fuzzy weight assignment allows for uncertainty to be captured and represented. The benefit of the second approach is that user intervention is not necessary anymore which helps the automation of the matching process.
For further research, an evaluation will be conducted by an experiment to calculate precision and recall rates for both approaches. Furthermore, an investigation will be done to compare how predefined and hard coded weight values influence the precision and recall values. In addition, due to the computational burden of the Mamdani inference, the Sugeno inference might work better in this area where quick response times are important. However, the advantage of capturing expert knowledge might be compromised. This also needs to be explored further.

4. References

1. J. McGovern, S. Tyagi, M. Stevens, S. Mathew, "The Java Series Books - Java Web Services Architecture", Chapter 2, Service Oriented Architecture, 2003.
2. H. Tangmunarunkit, S. Decker, C. Kesselman, "Ontology-based Resource Matching in the Grid - The Grid meets the Semantic Web", Proceedings of the International Semantic Web Conference, Budapest, Hungary. May 2003.
3. S. A. Ludwig and S. M. S. Reyhani, "Semantic Approach to Service Discovery in a Grid Environment", Journal of Web Semantics, vol. 4, no. 1, pp. 1-13, 2006.
4. D. Bell and S. A. Ludwig, "Grid Service Discovery in the Financial Markets Sector", Journal of Computing and Information Technology, vol. 13, no.4, pp. 265-270, 2005.
5. S. A. Ludwig, O. F. Rana, J. Padget and W. Naylor, "Matchmaking Framework for Mathematical Web Services", Journal of Grid Computing, vol. 4, no. 1, pp. 33-48, 2006.
6. T. Gagnes, T. Plagemann, E. Munthe-Kaas, "A Conceptual Service Discovery Architecture for Semantic Web Services in Dynamic Environments", Proceedings of the 22nd International Conference on Data Engineering Workshops, April 2006.
7. L. A. Zadeh, "Fuzzy sets", Information and Control, 8:338-383, 1965.
8. E. Cox, "The Fuzzy Systems Handbook", AP Professional, 1995.
9. L. H. Tsoukalas and R.E. Uhrig, "Fuzzy and Neural Approaches in Engineering", John Wiley & Sons Inc., New York 1996.
10. B. Kosko, "Fuzzy Engineering", Prentice Hall, Upper Saddle River, New Jersey 1997.
11. G. J. Klir, U. H. St. Clair, and B. Yuan, "Fuzzy Set Theory: Foundations and Applications", Englewood Cliffs: Prentice-Hall, 1997.
12. E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller", *International Journal of Man–Machine Study* (1975), pp. 1–13.
13. M. Sugeno, "Industrial Applications of Fuzzy Control", North-Holland, Amsterdam, 1985.