

Semantics and OWL in UDDI Registry: Improving the Discovery Process of Web Services

Elira Hoxha

University of Tirana, Faculty of Economy, Department of Mathematics, Statistics and Applied Informatics
E-mail: elirahoxha@yahoo.com, elira.hoxha@unitir.edu.al

Dhimitri Tole

University of Tirana, Faculty of Economy, Department of Mathematics, Statistics and Applied Informatics
E-mail: dhimitri.tole@unitir.edu.al

Kreshnik Vukatana

University of Tirana, Faculty of Economy, Department of Mathematics, Statistics and Applied Informatics
E-mail: k.vukatana@gmail.com, kreshnik.vukatana@unitir.edu.al

Doi:10.5901/ajis.2014.v3n1p341

Abstract

The discovery process of web services becomes a problem in terms of finding the most appropriate service in that huge set of various items published throughout the web. In this paper we will describe some of the existing mechanisms, which come as proposals for improving the discovery process of web services. By using one of these approaches, we will propose a semantic-based architecture, composed of many collaborating elements and then we will show a part of the implementation of the matching algorithm, between the user request and the published services. This matching is based on the ontological description of services and will be done by using the owl semantic specification.

Keywords: semantics; ontology; owl; web service discovery; matching algorithm; uddi registry;

1. Semantic Web Services and Some other Approaches for Improving the Discovery Process

Web services improved with semantics, also known as Semantic Web Services, have evolved year after year. Through them has been developed expressive semantic languages which describe services and use logic reasoning and problem-solving methods to make possible the realizing of all the steps involved with the services, as discovery, selection, composition, mediation and invocation. Many efforts and good work is done in this field, in order to build suitable infrastructures for solving problems and helping the users to reduce manual activities, by replacing these activities with intelligent behaviour of software agents. Some of the semantic specifications for web services, used nowadays are OWL-S, Annotations for WSDL and XML Schema (SAWSDL), etc. There exist many works about these specifications which determine algorithms for discovering, composition and mediation of services. As has been mentioned also from Mukhopadhyay and Chougule (2012), these approaches are focused on different aspects of the discovery process of the services.

So, concisely we can say that there are models that consider the context, enabling the personalization and optimization of the request of the user and the resulting services (suggested by Rong and Liu, 2010). According to them, the context should be domain and problem-oriented and therefore they propose the categories of Personal Profile, User History, Process and Other Contexts, where all together would result in finding a more suitable service for the user. The disadvantage of this method is that the architecture of the system becomes more complicated, when new attributes and conditions are added.

Another approach divides the research process of a service into layers (Wen-yue *et al.*, 2010) and the method takes care primarily on matching the category of service, then the functionalities and ultimately the quality of service.

Nawaz *et al.* (2008) propose a model where the subjects are registered, in order to get notifications in the moment that a new service is published in the UDDI registry. In this case the matching between the request and the offered services is done through semantics, which is associated with the registered data from the interested subject for a certain

service. In this case, the time for service discovery is minimized, because the search field is reduced to a specific category, while the probability of finding the most appropriate web service is increased. The drawback of this approach is the added difficulty in the development and maintenance of new components, of the system architecture.

The model proposed by Zhou *et al.* (2008) is based on discovering web services by clustering keywords and concepts expansion. They use bi-directional graphs for finding the degree of matching between the request and the offered services, based on ontology and semantic reasoning.

Another approach for service discovery is based on agents and also considers the different qualities of service (QoS). So, Rajendran and Balasubramanie (2010) suggest a framework which contains agents used for sorting and prioritizing services based on quality certifications earned by the publishers of the services.

A similar model comes from Pathak *et al.* (2005), who propose that during the search of a particular service, the user should have the possibility to specify in his request non-functional requirements, which will be added to the basic search criteria. Thus, the evaluation and ranking of web services will be based on certain criteria determined by the user, such as availability, speed of service, etc.

As we have seen so far, the models for discovering web services are based on keywords or ontology. Tsai *et al.* (2011), use both of these approaches by proposing a hybrid model, which considers the information of web service providers, the description of services by the users and by the providers, the labels and categories, as well as the QoS attributes. From the tests performed by the authors, to compare this method with methods that are based only on keywords, or just in semantics, it appears that the hybrid approach gives better results compared with the use of these approaches separately.

Regarding the existing models for the matching algorithm between user requests and offered web services, we can mention Paolucci *et al.* (2002), who propose a mechanism which uses the degree of matching, between individual parameters of a service and the user request. Subsequently, they add to these criteria the matching of service results and in the end, the matching of the input.

Another approach is presented by Binder *et al.* (2007), which bases the rankings of services, in numerical expressions obtained as part of the request. Basically, these expressions support a combination of arithmetic operators such as min, max, +, etc., as well as the set operators such as union, intersection, etc., which are used in the set of inputs and outputs of the offered services and the users requirements. Although this model is very flexible and allows the interested subject to specify a certain type of evaluation ranking, the expressions are somewhat non-intuitive, which makes this approach not fully adequate.

A compromise of the two aforementioned models is proposed by Vaculin (2009), which uses the degree of matching between individual parameters of offered services and user requests, as a basic ranking function, where the matching algorithm runs in two phases: the registration phase and the control one.

We have chosen to focus on models that utilize the semantics of the requests made by the user. Thus, we propose a model which compares the ontology created on the request, with ontologies created for each web service registered by the publishers.

2. The Proposed Architecture for the System

Actually the interested subject for a specific web service, searches directly in the lists of UDDI registry, for finding the appropriate service and publisher, and the discovery process is based only on existing taxonomies that do not contain semantics, like North American Industry Classification System (NAICS, 2011). On a previous paper (Hoxha *et al.*, 2012) we have proposed a semantic-based architecture, which added an ontology intermediation layer, with the purpose of an improved process of discovering the best web service which fits the user needs. Now, we are presenting the same system architecture (Figure 1) with some changes in the components interaction, which we are going to describe later. The workflow in our architecture, from the user point of view would be as follows:

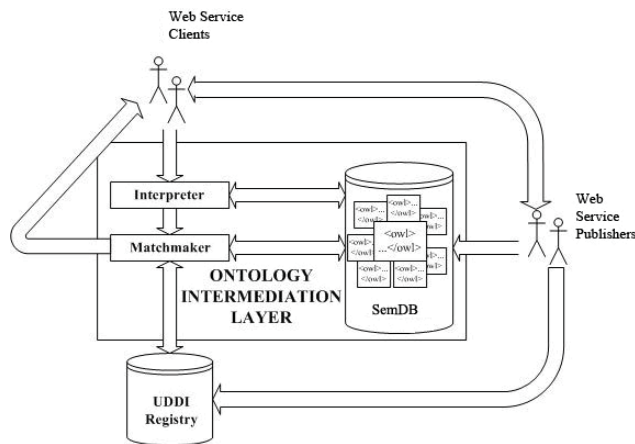


Figure 1. Ontology-based architecture (Hoxha *et al.*, 2012)

The web service client uses a subject, or a certain word for which he thinks would best describe the service he is looking for. This request is sent to the so-called Interpreter, which is a software based on taxonomy, with a great set of English words, synonyms, or other grammatical forms, as well as their hierarchical categorizations. Based on this knowledge, the interpreter component chooses the appropriate term, which will then be used to build the hierarchical tree of this request, using ontologies as proposed by OWL-S (Web Ontology Language for Services) (Martin *et al.*, 2004), which is also designed as a standard by the W3C. To accomplish this, the interpreter will refer to the ontological database component (named SemDB), which will be used for two different approaches in our architecture. The first one consists in storing the owl files for the full ontological trees, of a large number of domains, including the various services offered. The ontologies placed in this section need to be standardized for the specific domain, so that the implementation of the requests trees from the interpreter, as well as the services trees from the publishers, can be done using the same model. This would lead to a successful matching process, between the two trees. The interpreter component, will build a partial tree, based on the selected term of the client request, as well as on the ontological tree of the domain. This tree will include nodes that belong to the parent, grandfather and great-grandfather of the search term, as well as nodes that represent children and grandchildren of this term.

To increase the chance of finding the most appropriate service or similar services related to the newly created tree, we propose a second approach for the SemDB component. It will have a storage function for the owl files, registered there by the publishers of web services. Each owl file is associated with a particular web service and is described by the ontological structure. Besides the description and categorization of web services, these owl files have an identifier in the ontological database, which makes possible the identification of the related services in UDDI registry.

The UDDI registry (UDDI, 2004) (shown as a component of our architecture) is an existing mechanism, that provides a method for the publication and discovery of web service descriptions. The focus of our work was the improvement of the efficiency of this component. Each service in UDDI registry is identified by a unique ID, which refer (as we mentioned earlier) to the owl files of the services, stored by the publishers in the SemDB. These files need to be interpreted and matched against the ontological request tree created by the interpreter, therefore is proposed a new component, based on the idea mentioned by Srinivasan *et al.* (2004). It is called the Matchmaker and will play the role of a mediator between the interpreter, the SemDB and the UDDI registry. So, it will take from the interpreter the ontological tree of the client request and by using a matching algorithm, it will search between the owl files in the ontological database, to find the most appropriate ontological tree and then, it will identify the corresponding web services in UDDI registry; in the end this mechanism will return to the user, the list of possible web services, which can complete their request. The services in this list are sorted in the way that those that meet more criteria from the user request appear first in the list. The user can choose a service from this list and establish the communication with the respective publisher.

3. Building the Semantic Database and Implementing the Matching Algorithm

To realize the implementation of the matchmaker component, primarily is needed to build the ontological database, in

order to simulate the storage and selection of owl files, which describe the published web services. Thus the SemDB component will be used for the moment only for its second approach, assuming that the ontological trees, by the client as well as by the publishers are done according to standards. As Database Management System for this case, we have chosen to use Oracle, which offers the possibility of saving data in xmlType. The relation that will contain the ontological trees of publishers will have only two attributes. The first attribute that is the unique identifier of the web service, will be the same ID used by the service to register in the UDDI, while the second attribute will contain the semantic structure of each service, which represents the ontological tree. The equivalence of web services IDs, in the UDDI registry and in the SemDB, is done for the reason of an easier merging of the semantic information from the SemDB and the descriptive information stored in the UDDI registry for each published web service.

Regarding the matching algorithm between the request tree and the existing services trees, we are presenting below the simplified sequence of steps needed:

```
FOR EACH ontological tree of published web services
{
  IF [ ( publisher_tree_root = client_search_term ) OR
    ( publisher_tree_root = father_of_client_search_term AND
      a_child_of_publisher_tree_root = client_search_term ) ]
  THEN ( Priority_for_this_service = 1 )
  ELSE IF [ ( publisher_tree_root = grandfather_of_client_search_term AND
    a_child_of_publisher_tree_root = father_of_client_search_term AND
    a_grandchild_of_publisher_tree_root = client_search_term ) OR
    ( publisher_tree_root = great-grandfather_of_client_search_term AND
    a_child_of_publisher_tree_root = grandfather_of_client_search_term AND
    a_grandchild_of_publisher_tree_root = father_of_client_search_term AND
    a_great-grandson_of_publisher_tree_root = client_search_term ) ]
  THEN ( Priority_for_this_service = 2 )
  ELSE IF [ ( publisher_tree_root = a_child_of_client_search_term_tree ) OR
    ( publisher_tree_root = a_grandchild_of_client_search_term_tree ) ]
  THEN ( Priority_for_this_service = 3 )
  ELSE ( Priority_for_this_service = 0 )
};
SORT the resulting services according to the value of their priorities, from 1 to 3, by
not including services with priority 0;
PRINT the ordered list of services;
```

As it can be seen the algorithm is divided in three phases, in order to differentiate the resulting services according to the "matching degree" with the client request. The services which are "specialized" exactly in what the client is searching for, get the priority 1 and are listed first in the search results. Below is shown one of the queries that implements the matching algorithm. More specifically, this query is part of the first phase of the algorithm, where the priority of the resultant services will be set equal to 1 and it will find all the publishers which offer a service with an ontological tree that has as root node the father node of the client request.

```
SELECT XMLQuery ('
  xquery version "1.0";
  declare namespace xsd="http://www.w3.org/2001/XMLSchema";
  declare namespace rdfs="http://www.w3.org/2000/01/rdf-schema#";
  declare namespace rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#";
  declare default element namespace "http://www.w3.org/2002/07/owl#";
  for $x in //Class
  where $x[@IRI="#client_search_term"] and $x/following-
  sibling::Class[@IRI="#father_of_client_search_term"]
  return count($x) PASSING XML_COLUMN RETURNING CONTENT
) AS RESULT,
KEY_COLUMN AS KEY
FROM SEMDB.OWL_TBL2;
```

XMLQueries using the appropriate *Xpaths*, identify the nodes of different levels in the ontological trees. The other queries needed for the next steps of the algorithm follows the same logic of the query shown above, but differentiate in the *XPath* expression. After getting the list of relevant web services, the system mechanisms will get from the UDDI registry all the other information about these services, which will be shown to the user.

4. Drawbacks

The aim of adding semantics in the process of discovering web services is to improve and facilitate this process. But the selected approach has also its drawbacks. So, maintaining all that information in a centralized place (more specifically the OWL and WSDL documents for each service) convert servers into a bottleneck for the system. Also, since the information in these records is static, it is difficult to maintain the control of coherence and services validity. For this reason we can often encounter dead links that lead us into dysfunctional or missing web services. To avoid this, organizations or individuals should use private UDDIs of a company's intranet within or between enterprises. Maintaining and distributing independent management is a core ideology for improving the UDDI registry.

Also, a large number of owl files into a unique database can lead to delayed results from the matchmaking algorithm, which would go through all these lines, looking for the appropriate services. These delays can be avoided by saving the results of different searches from the users, in order to have immediate responses for later same or similar requests for services, under the cache concept.

Another drawback of the semantic approach can be considered the lack of standard ontologies for all the existing domains of web services. This would lead to different ontology trees for the same service, which can cause difficulties in finding the right one. This problem can be improved by the interested subjects themselves. So the large companies which offer services in specific fields should be involved in the process of registering the standard ontologies for their domain. These standard ontologies can be further improved in order to be as complete and comprehensive as possible.

5. Conclusions and future work

Searching and finding the appropriate web service in the UDDI registry can become time-consuming and sometimes inefficient. For this reason, there exist many approaches that help in the improvement of the discovery process of these services, by adding other user specifications, software agents, semantic descriptions, considering the quality of services, non-functional requirements, etc. We focus our work in the advantages of using semantics and so proposed an ontology-based architecture that gives us a mechanism for remodeling the user request in an ontological tree, to be compared with other ontological trees of published web services, in order to find the best service we need. This approach has some weak points, which we mentioned in this paper, along with the proposed improvements.

A future goal for us, is adding software agents in the proposed architecture, whom with their specifics as autonomy, interaction, mobility, event-orientation and the ability to learn, can play a crucial role in the integrated system for a further improvement.

References

- Binder, W., Constantinescu, I., Faltings, B. (2007). A flexible directory query language for the efficient processing of service composition queries. *International Journal Web Service Res*, 2007.
- Hoxha, E., Tole, Dh., Vukatana, K. (2012). Discovering Web Services: An Ontology-Based Architecture. *Economic & Social Challenges 2012 – Economy & Business Doctoral Students Conference*. December 2012.
- Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, Sh., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K. (2004). OWL-S: Semantic Markup for Web Services. November 2004. <http://www.w3.org/Submission/OWL-S/>
- Mukhopadhyay, D., Chougule, A. (2012). A Survey on Web Service Discovery Approaches. *Second International Conference on Computer Science, Engineering & Applications. ICCSEA 2012 Proceedings*. Delhi, India. Springer-Verlag, Germany; ISSN 1867-5662.
- NAICS. (2011). The North American Industry Classification System (NAICS). August 2011. www.census.gov/eos/www/naics.
- Nawaz, F., Qadir, K., Ahmad, H. F. (2008). SEMREG-Pro: A Semantic based Registry for Proactive Web Service Discovery using Publish Subscribe Model. *Fourth International Conference on Semantics, Knowledge and Grid, IEEE Xplore*, 2008.
- Paolucci, M., Kawamura, T., Payne, T. R., Sycara, K. P. (2002). Semantic matching of web services capabilities. *In International Semantic Web Conference*, 2002.
- Pathak, J., Koul, N., Caragea, D., Honavar, V. G. (2005). A Framework for Semantic Web Services Discovery. *WIDM' 2005*, Bremen,

Germany.

- Rajendran, T., Balasubramanie, P. (2010). An Optimal Agent-Based Architecture for Dynamic Web Service Discovery with QoS. *Second International conference on Computing, Communication and Networking Technologies, IEEE Xplore*, 2010.
- Rong, W., Liu, K. (2010). A Survey of Context Aware Web Service Discovery: From User's Perspective. *Fifth IEEE International Symposium on Service Oriented System Engineering*, 2010.
- Srinivasan, N., Paolucci, M., Sycara, K. (2004). Adding OWL-S to UDDI, implementation and throughput. *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*. July 2004.
- Tsai, Y.H., Hwang, S.Y., Tang, Y. (2011). A Hybrid Approach to Automatic Web Services Discovery. *International Joint Conference on Service Sciences, IEEE Xplore*, 2011
- UDDI. (2004). Universal Description Discovery and Integration. <https://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>. 2004.
- Vaculín, R. (2009). Process Mediation Framework for Semantic Web Services. *Doctoral Thesis at Faculty of Mathematics and Physics, Charles University*, June 2009.
- Wen-yue, G., Hai-cheng, Q., Hong, Ch. (2010). Semantic web service discovery algorithm and its application on the intelligent automotive manufacturing system. *International Conference on Information Management and Engineering, IEEE Xplore*, 2010.
- Zhou, J., Zhang, T., Meng, H., Xiao, L., Chen, G., Li, D. (2008). Web Service Discovery based on Keyword clustering and ontology. *IEEE Xplore*, 2008.