

Semantic Web Service Discovery Algorithm based on Constraint Extraction and Structure Analysis

Kun Li, Huayu Li and Jidong Chen

*College of Computer and Communication Engineering
China University of Petroleum, UPC, Qingdao, China*

Abstract

As the Internet Web services increased, how to find a web service which can meet user's requirements quickly and accurately has become a problem to be solved. Now many kinds of web service methods based on semantic are based IO matching. They used other ways to supplement while semantic matching is failed. However the accuracy of semantic web service discovery based on IO matching purely is not high enough, the Algorithm of semantic web service discovery which is based on constraint extraction and structure analysis is proposed. It is divided into two parts. First part is to carry out the concept of semantic matching based on constraint extraction. Then it uses algorithm based on structure analysis while matching is failed, aiming to assess the structural similarity of the desired vs the retrieved services and the semantic similarity of their identifiers. This article uses owls-tc2.0 as test set to test this method. The experimental results show that this method improves service discovery accuracy effectively.

Keywords: *semantic Web; Discovery of semantic web; precision; constraint extraction; structure analysis*

1. Instruction

Web service discovery has become a hot research topic in recent years. Various web service discovery techniques have been proposed, many of which perform the profile based service signature (I/O) matching. However, the service I/O concepts are not sufficient to discover web services accurately. The reason is that, on the one hand, that the traditional web service descriptions lack the necessary semantic information can not accurately describe the function of web services, affect the precision of web service discovery seriously. On the other hand, traditional web service discovery technology is through the syntax-level service description language and an algorithm based on keyword matching to achieve.

Semantic web services (SWS) have attracted a significant amount of attention in recent years. The aggregation, including description and discovery of services plays an important role in various internet-based virtual computing environments. SWS discovery is the process of locating existing web services based on the description of their functional and non-functional semantics. The research of semantic web service describes various components of web services in the semantic level, description language such as, OWL-S [1], WSMO [2], WSDLs [3], SAWSDL [4], etc. Discovery of semantic web usually define a service matcher (Matchmaker), and service provider and service requestor use the same service description language, measuring the degree of matching based on the service matcher request service and semantic web. The technology of discovery of semantic web based on logical semantic matching currently divided into two categories. First, it is the discovery of web service based on IO. the second is the discovery of semantic web based on IOPE [7-8]. Common of these methods are based on the inclusion relation between two concepts to measure the degree of semantic matching. However, the degree of matching web services is determined by the degree of semantic matching service of IO. In other

words, the quality of IO semantic parameter determines performance of the discovery of semantic web algorithm based on semantic matching and IO web services.

To compensate for the lack of semantic description of services IO concept, we extract IO semantic constraints through web services, describe web services by OWL-S to propose a new algorithm. The algorithm is divided into two parts. The first step is to execute discovery of semantic web service based on semantic constraint extraction, when the semantic match fails then execute discovery of semantic web service algorithm based on structure analysis.

The rest of this paper is organized as follows. Section 2 gives the related research of semantic web services discovery. Semantic web service discovery algorithm based on constraint extraction and structure analysis is proposed in section 3. Section 4 evaluates the proposed approach on OWL-S and pallet (API developed at the University of Maryland, <http://www.mindswap.org/2004/owl-s>), use xampp as a local server for testing. The conclusion and future research are given in section 5 finally.

2. Related Research

Service IO-based semantic matching includes OWLSM[10], OWLS-UDDI [11]. Performing semantic web service discovery mainly use the IO concept semantic inclusion relations of service to determine the degree of semantic match between two services. OWLSM defines three types of semantic matching degree: Equivalent, Subsume, Fail. Using DAML-S as web service description language in OWLS-UDDI and presenting semantic matching definitions of semantic web service discovery, Which is defined semantic matching has exact, plug-in, subsumes, fail, *etc.*

The method which is the combination of semantic matching based on logic and information retrieval is used in Owls-mx [6]. Web services are described by owls and semantic reasoning using owl-DL. Firstly, executing semantic matching based on IO logic. When semantic matching fails then execute information retrieval method. Seven semantic matching level are defined during IO semantic matching process. Information retrieval uses cosine similarity measure, extended Jacquard similarity measure and Jensen-Shannon similarity measure. Owls-mx has achieved good results.

The method [12] which combined semantic matching based on logic and structural analysis has proposed by Rodrigo Amorim *et. al.* when semantic matching fails then execute structural analysis algorithms to improve the accuracy of semantic web discovery. D. Wei *et. al.* proposed logic-based semantic matching, when conducting IO semantic matching, first extracting concept's semantic constraints, and describing web service as triples form, then matching on the body, the operation and object respectively, return match result at last[9].

3. Semantic Web Service Discovery Algorithm Based on Constraint Extraction and Structure Analysis

3.1. Semantic Matching of Web Services

Semantic matching algorithm is semantic matching for service IO, that is, service matching degree is determined by the logic matching degree of service input and output parameters, and obtaining concept of service on IO by analysis service semantic description file. Then reasoning on concept of tree body by pallet which has logical reasoning ability. Semantic matching level is divided into Exact, Plug-in, Subsumes, Fail. Assume c is a domain-specific concept, $parents(c)$ refers to the parent node that return c , $Children(c)$ refers to the child node that return c . Simple concept of hierarchy tree is defined as follows :

(1)Exact: the relation between the concept s of service and the concept r of request is complete matching when and only when $r=s$ that means the relation between the concept of service and the concept of request is completely logical equivalence . If the concept tree of I in Figure 1 is aircraft, then I of request is aircraft.

(2)Plug-in: the relation between the concept s of service and the concept r of request is plug-in when and only when $r \in parents(s) \vee s \in children(r)$ that means the concept r is the parent node of concept s or the concept s is the parent node of concept r . If the concept tree of I of service in Figure 1 is aircraft, then I of request is Transportation.

(3)Subsumed-by: the relation between the concept s of service and the concept r of request is subsumed-by when and only when $s \in parents(r) \vee r \in children(s)$ that means the concept s is the parent node of concept r or the concept r is the parent node of concept s . If the concept tree of I of service in Figure 1 is automobile, then I of request is Bus.

(4)Fail: It is considered as semantic matching failure that the above three conditions are not met when a service is semantic matching request service.

Two disjunctive clause semantics of plug-in and subsumed-by defined above appears to be the same, but in reasoning process of the actual body, semantic reasoning results often depend on the base knowledge. If the reasoning knowledge is different, you may get different results. In semantic matching process, first let's make an exact match,then execute plug-in ,the third step is executing subsumed-by, return the result set based on semantic matching at last.

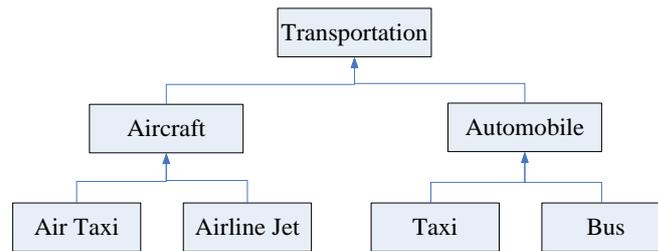


Figure 1. A concept of Hierarchy Tree of Transport Domain

3.2. The Concept Semantic Constraints of Web Service

The description texts in web services are important knowledge sources for semantic service discovery. The constraints of a certain concept can be extracted from the description texts according to the definition of concept in domain ontology, especially the definition of property whose subject is the concept .However, in practice, few domain ontologies are comprehensive enough to provide plenty constraints information, as every concept in the domain ontology has only limited kinds of properties. In this section, an extraction method based on the parse trees of description text is proposed to obtain the semantic constraints of service I/O concepts. The aim of service concept semantic constraints extraction is minimizing shortages which existed in semantic description of service IO concept as far as possible ,that is increasing IO concept semantic relations description and reducing semantic bias and ambiguity of semantic web service, thereby enhancing the accuracy of semantic Web service semantic matching. At present, semantic web services ontology description exists following problems:

(1) Not specified concept domain. The current concept is to describe some of more abstract things in ontology library under normal circumstances, such as price. However, abstract concept often associated with context, only in particular semantic environment, it has real meaning.

(2) Not specified property concept. A concept may be associated with multiple attributes in ontology, semantic concepts are usually constrained by properties, and the

same concept can be divided into different parts based different attributes constraints. In particular environments, the semantics of web service IO is not only associated with concept, but also associated with related properties.

(3) Not specified relationship between concepts. Two services with different semantics might have the same IO. In this case the use of service semantic matching will be mistaken for similar.

Therefore, we will express concept constraint into three types, Figure 2 shows the instance represents of services constraint graph.

(1) IsPropertyObjectOf:Triple<A, isPropertyObjectOf, B> said concept A is an attribute of concept B. It indicates concept category that is, the instance of concept A is the property value of the instance of concept B.

(2) HasPropertyObject: This constraint relation is inverse relationship of the constraint isPropertyObjectOf Triple<A, hasPropertyObject, B> said concept A has an object property, and the object of this property is concept B.

(3) Operation: Triple<A, Operation, B> said there are some relationship between the two concepts entity A and B. "Operation" is an abstract word, represents all types of property.

The semantic constraint extraction which is based on the parsing tree of the sentence consists of preprocessing, syntactic parsing and heuristic-based extracting. During preprocessing, some pre-selected key-words representing the service I/O concepts are tagged in the description text. And then, the text is parsed syntactically to identify the constituents modifying the key-words. Finally, several heuristics rules are used to extract constraint triples about the key-words from the syntactic constituents. You can improve the accuracy of semantic matching to some extent by semantic constrained extraction of service concept. In real world, two concepts that are synonyms or very close semantically but their definition is not inclusion relation in ontology, in this case, semantic matching will fail. With this circumstances, we allowed the semantic match fails, when the semantic match fails, carrying on discovery of semantic web based on structural analysis.

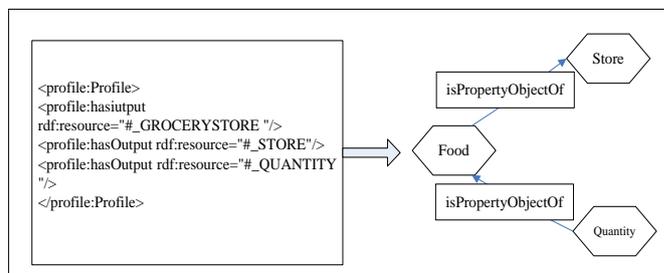


Figure 2. The instance Represents of Services Constraint Graph

3.3. Structural Analysis of Web Services

In this section, we use structural analysis base on synonym dictionary. synonym dictionary is classified for the concept of test set owls-tc. While calculating the similarity between the two types of concepts c_1 and c_2 , if c_1 and c_2 is the same concept, then $M[i][j] = 1$; if c_1 and c_2 is synonyms, then $M[i][j] = 0.5$ and if not, $M[i][j] = 0$. Structural analysis function based thesaurus is defined as follows[9]:

$$sim(c_1, c_2) = p_1 * ancestor(c_1, c_2) + p_2 * descendent(c_1, c_2) + p_3 * sibling(c_1, c_2) + p_4 * leaf(c_1, c_2) \quad (1)$$

Among them, P_1, P_2, P_3, P_4 are the parameters, which can be adjusted between 0 to 1 and $p_1 + p_2 + p_3 + p_4 = 1$; according to user needs, we change the type (1) for each item in the

two concept similarity calculation of proportion of solution, so that the algorithm can achieve the best performance. Ancestor (C1, C2), descendent (C1, C2), sibling (C1, C2), leaf (C1, C2) were used for similarity of parent node, child nodes, brother node, leaf nodes computing concepts of C1 and C2, calculating in the same way for the formula (2) (3) (4).

$$m = \frac{\sum_{i=1}^{|\text{ancestor}(c1)|} (\sum_{j=1}^{|\text{ancestor}(c2)|} M[i][j])}{|\text{ancestor}(c1)| \times |\text{ancestor}(c2)|} \quad (2)$$

$$sd = \frac{\sum_{i=1}^{|\text{ancestor}(c1)|} (\sum_{j=1}^{|\text{ancestor}(c2)|} (m[i][j] - m)^2)}{|\text{ancestor}(c1)| * |\text{ancestor}(c2)|} \quad (3)$$

$$vc = \frac{sd}{m} \quad (4)$$

In above formula M is a matrix of |ancestor(c1)| |ancestor(c2)|, (2) is used to calculate average of concepts c1 and c2 parent node, (3) is used to calculate the standard deviation between concepts c1 and c2 parent node. In the calculation process is emptied when the value of vc is less than the threshold.

3.4. Algorithm Description

The proposed algorithm first extract service concept semantic constraints, then execute service semantic matching, while the match fails, to calculate the similarity of two concepts by performing services structural analysis $\text{sim}(c1, c2)$. When the value of $\text{im}(c1, c2)$ is greater than the threshold, we think the two are related services. Service will be expressed into three tuple $\langle A, \text{Operation}, B \rangle$, published service will be expressed as three tuple $\langle sa, so, sb \rangle$, and request services will be expressed as three tuple $\langle ra, ro, rb \rangle$. The algorithm is described as follows:

(1) Computation of Concept semantic similarity

Semantic similarity is the similarity between semantic content to measure documents and terms, it is widely used in the fields of information retrieval and extraction, we use it to calculate similarity between two concepts, and consider the concept of information content (IC) and the distance information of two concepts in the ontology concept tree at the same time, then we will get higher similarity by this algorithm. The concept of IC value refers to the amount of information provided by the concept of IC, expressed as $-\lg(p(c))$, its meaning is that a concept of the emergence of the C frequency is larger, the concept of self-information is smaller.

We adapt IC computing model in [12], this model not only take the nodes of sub nodes into account, but also the depth of each concept is also included, the purpose is to make each concept has an accurate IC value. Because the owls-tc2.0 was used as test set, so the IC value is calculated IC value of the owls-tc2.0 definition of the concept of ontology. As shown in formula (1):

$$IC(w) = k(1 - \frac{\lg(\text{hypc}(w) + 1)}{\lg(\text{node}_{\max})}) + (1 - k) (\frac{\lg(\text{deep}(w))}{\lg(\text{deep}_{\max})}) \quad (5)$$

The $\text{hypc}(W)$ expresses sub concept of returns concept w , $\text{deep}(W)$ says the depth of concept W 's concept tree, nodemax says the total number of concepts in taxonomy tree, deepmax represents the maximum depth of classification tree, K is a adjust parameters, which can be adjusted according to user needs, for change (1) in two in solving IC proportion, that is the change in the proportion of nodes of sub nodes and depth two share in the model. In this text, the K value is 0.5.

When calculating the similarity with the semantic similarity algorithm based on information content and Conrath [12] concept semantic similarity algorithm, the information content and the two concepts in the classification of the position in the tree are taken into account, computational methods in [18] is adapted. As shown in formula (2).

$$sim(w1, w2) = 1 - k \times \left(\frac{\lg(\text{Len}(w1, w2) + 1)}{\lg(2 \times \max_{w \in \text{onto}}(\text{depth}(w)))} \right) - (1 - k) \times ((IC(w1) + IC(w2) - 2 \times IC(LSO(w1, w2))) / 2) \quad (6)$$

$Len(W1, W2)$ is the distance of $W1$ and $W2$ in concept tree, where k is a weight, which can be artificially adjusted, so that the algorithm can achieve the best performance. Here, the value of K is 0.5. $LSO(W1, W2)$ says the minimum common parent concept of $W1$ and $w2$. the path factor of two concepts is also considered in this model, because if the common parent of two concepts have the same value, and the IC values of two concept are the same, or the sum of IC value of two concepts is the same, but the path distance between two concepts is inequality, under this condition, using Resnik[11], Jiang and Conrath[12] semantic similarity algorithm can't distinguish the difference, but the calculation model presented in this paper is very easy to distinguish, which makes the calculation of semantic similarity between concepts is more accurate.

In this model, using $\frac{\lg(\text{len}(w1, w2) + 1)}{\lg(2 \times \max_{w \in \text{onto}}(\text{depth}(w)))}$ as a model, the choice comes

from self-information concept in information theory, the physical significance refers to the self-information of path distance between concept $W2$ and $W1$, t The amount of information that is a concept on the path, the range is $[0, 1]$. For (2) model, the range is also $[0, 1]$.

We will meet the following conditions during experiment:

(i) if the concept of $W1$ and $W2$ is the same concept, the value of $IC(W1)$ and $IC(W2)$ is the same by formula (1), then the value of $Len(W1, W2)$ is zero by formula (2), so $sim(W1, W2) = 1.0$. Therefore we first determine the concept of $W1$ and $W2$ are the same concept before calculating based on the IC semantics, if it is , the value of $sim(W1, W2)$ is set to 1 directly, and no calculated based on semantic IC .

(ii) If concept $W1$ and $W2$ are two different field concepts, then the similarity of $W1$ and $W2$ is 0. At this time we directly set $sim(W1, W2) = 0$.

(iii) If $W1$ and $W2$ aren't the same concept, and is in the same field, then we use the formula (1) and the formula (2) to calculate the semantic similarity of $W2$ and $W1$.

(2) Use the algorithm of services semantic matching that presented in 3.1 to calculate semantic matching degree between sa and ra (Exact,Plug-in,Subsumed-by,Fail),and return the value of pa at last.

(3) Compute constraint type of constraint manipulation so and ro (isPropertyObjectOf, hasPropertyObject, Operation),return the value of po at last.

(4)Use the algorithm of services semantic matching that presented in 3.1 to calculate semantic matching degree between sb and rb (Exact, Plug-in, Subsumed-by, Fail), and return the value of pb at last.

(5) According to (1) (2) (3), calculate the value of $sum = n1*pa + n2*po + n3*pb$, if the value of sum is greater than the threshold, match is considered successful, otherwise transfer (5).Wherein, $n1, n2, n3$ values are between 0 and 1, and $n1 + n2 + n3 = 1$.

(6) Use the structural analysis of services algorithm that presented in 3.3 to calculate semantic matching degree of sa and ra, sb and rb . If both the value of $sim(sa, ra)$ and $sim(sb, rb)$ are all greater than a defined threshold service is considered to be relevant.

(7) Repeat the above steps until you find the best service to meet customer needs.

4. Experimental Result and Analysis

The experiment performs analytical and reasoning on ontology by using OWL-S and pallet (API developed at the University of Maryland, <http://www.mindswap.org/2004/owl-s>), use xampp as a local server for testing. Use OWLS-TC2.0 as a test set, which has over 570 services from seven different areas (education, medical care, food, travel, communication, economy, weaponry). In this experiment, 20 queries are evaluated. Two sets of web services using in this experiment have been transferred from OWLS TC2.0 by annotating output concepts with constraints: set1 and set2. In set1, the semantic constraints of the output concepts in request and web service are manually annotated by two people and mainly described by service I/O concepts; while the semantic constraints of concepts in set2 are automatically extracted using the method represented in section 3. Query performance of the experiment used precision and recall rate [13]. Recall rate is an indicator of measuring degree of a retrieval system detected from the literature relevant literature collection successfully, that is the detection percentage of relevant literature and all relevant literature. Precision is the ratio of the number of relevant document has been retrieved and the number of all document has been retrieved. The number of relevant document has been retrieved is Relevant Document, recorded as RelDoc. The number of all document has been retrieved is Retrieved Document, recorded as RetDoc. Precision and Recall is defined as follows.

$$\text{Precision} = \frac{|\text{RelDoc} \cap \text{RetDoc}|}{|\text{RetDoc}|} \quad \text{Recall} = \frac{|\text{RelDoc} \cap \text{RetDoc}|}{|\text{RelDoc}|}$$

OWLS- M4 is reported to be the best-performing matchmaker variant of the OWLS-MX matchmaker [6] which uses Jensen-Shannon information divergence to compare the request and service based on the unfolded concept expressions. We also use the nearest-neighbor as the minimum degree of match and a value of 0.7 as syntactic similarity threshold for OWLS-M4. These values were suggested by the authors of OWLS-MX to obtain better results for OWLS-TCv2.0. M4+InOutConstraint matchmaker uses CGM to filter the results of OWLS-M4 on set1. Owls-m4 matchmaker uses CGM to filter the results of OWLS-M4 on set2. The results shown in Figure 3, Algorithm proposed is the method this paper presented, compared with owls-m4 [6] and M4+InOutConstraint [9] respectively.

Figure 3 shows that M4+InOutConstraint method reduce semantic bias and ambiguity of semantic web services by using extract concepts bound to increase relations describe of the semantic of IO concept to enhance the precision of semantic matching of semantic Web service. The method we proposed to achieve discovery of web service has higher accuracy than other methods, and makes it possible to return more service that meet the needs of the service requester potentially to discover the best services to enhance satisfaction of customer. OWLS-M4 can use the syntactical similarity filter (matching of the unfolded concept expressions) to find relevant services. M4+InOutConstraint can perform the semantic matching more accurately by enriching the description of SWS. The logic based matchmaker for two concepts which differs from that of OWLS-MX variants uses more relaxed notions for matching in order to retrieve more services, and the constraints of I/O concepts can help filter the irrelevant ones. It also indicates that we should use more relaxed semantic based matchmaker when using the constraint graph based matchmaker [9].

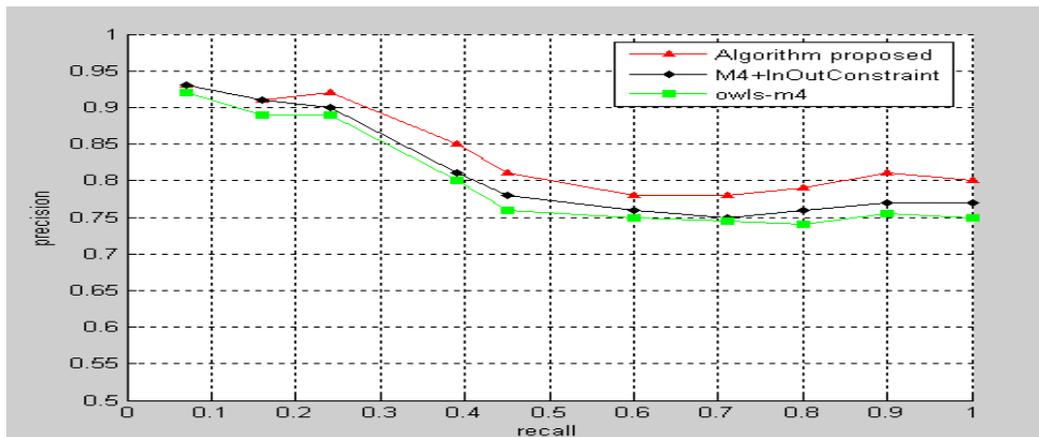


Figure 3 .Results Figure

This experiments result shows that the semantic constraints of service input/output concepts enrich the description of services capabilities and alleviate the unclear problem of semantic web services that described by only I/O concepts. Structure analysis for semantic constraint of service I/O concepts can distinguish the similar web services to improve the precision of discovery Algorithm. The semantic constraint based matchmaker could combine to other SWS matchmakers to improve their performance. The matchmaker CGM [9] explores the constraints information to filter the irrelevant services which probably match the request by pure logical reasoning, while the hybrid matchmaker emphasizes particularly on retrieve those services that fail to retrieve according to pure logic based matching. So they can work with each other without conflict. However, the performance of our matchmaker depends on the performance of constraints extraction. The best performance displayed in Fig.3 is Algorithm proposed which indicates that semantic constraints of service I/O concepts are excellent in SWS discovery task if supported by a good constraints extracting and structure analysis result.

5. Conclusion

Semantic web service discovery is a hot topic in the fields of both semantic web and web service. This paper mainly works on enhancing the semantics of web service through introducing semantic constraints and structure analysis to service I/O concepts and concerns only the function based matchmaking. We use semantic web service discovery method based on constraint extraction and structure analysis for discovery of semantic web, First part is to carry out the concept of semantic matching based on constraint extraction .,then it uses algorithm based on structure analysis while matching is failed. Experimental results demonstrate the effectiveness of the proposed algorithm. In the next study, we will focus on improving the performance of the algorithm and perfect discovery of semantic web tool further, providing support platform for discovery of semantic web.

References

- [1] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, P. Terry, E. Sirin, N. Srinivasan and K. Sycara, "Owl-s: Semantic markup for web services", <http://www.w3.org/Submission/OWL-S/>, (2004).
- [2] J. de Bruijn, C. Bussler, D. John, D. Fensel, M. Hepp, M. Kifer, B. K"onig-Ries, J. Kopecky, R. Lara, E. Oren, A. Polleres, J. Scicluna and M. Stollberg, "D2v1.3.web service modeling ontology (wsmo)", <http://www.wsmo.org/TR/d2/v1.3/>, (2006).
- [3] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.-T. Schmidt, A. Sheth and K. Verma, "Web service semantics - WSDL-S", <http://www.w3.org/Submission/WSDL-S/>, (2005).

- [4] J. Farrell and H. Lausen, "Semantic annotations for WSDL and XML schema", <http://www.w3.org/TR/sawSDL/>, (2007).
- [5] M. C. Jaeger, G. Rojec-Goldmann, C. Liebetrueth, G. M"uhl and K. Geihs, "Rankedmatching for service descriptions using owl-s", *KiVS 2005, Informatik Aktuell*, (2005), pp. 42-113.
- [6] M. Klusch, B. Fries and K. Sycara, "Automated semantic web service discovery with owls-mx", 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Hakodate, Japan, ACM, New York, (2006), pp. 915-922.
- [7] F. Kaufner and M. Klusch, "Wsmo-mx:a logic programming based hybrid service matchmaker", 4th European Conference on Web Service, Zurich, Switzerland, IEEE CS Press, Los Alamitos, (2006), pp. 161-170.
- [8] M. Stollberg, U. Keller, H. Lausen and S. Heymans, "Two-phase web service discovery based on rich functional descriptions", In: Franconi, E., Kifer, M., May, W. (eds.) *ESWC 2007. LNCS*, Springer, Heidelberg, vol. 4519, (2007), pp. 99-113.
- [9] D. Wei, T. Wang, J. Wang and Y. Chen, "Extracting Semantic Constraint from Description Text for Semantic Web Service Discovery", *Text*, no. 2005, (2008), pp. 146-161.
- [10] M. C. Jaeger, G. Rojec-Goldmann, C. Liebetrueth and G. Muhl, Kurt Geihs, "Ranked Matching for Service Descriptions Using OWL-S. KiVS", Springer, (2005), pp. 91-102.
- [11] M. Paolucci, T. Kawamura, T. R. Payne and K. P. Sycara, "Semantic Matching of Web Services Capabilities", *International Semantic Web Conference*. Springer, *Lecture Notes in Computer Science*, vol. 2342, (2002), pp. 333-347.
- [12] R. Amorim, D. Barreiro Claro, D. Lopes, P. Albers and A. Andrade, "Improving Web service discovery by a functional and structural approach", 2011 IEEE International Conference on Web Services DOI 10.1109/ICWS.2011.14.
- [13] C. van Rijsbergen. *Information Retrieval*. 1979.

