Intelligent Broker
A Knowledge based Approach for Semantic Web Services Discovery

Mohamed El Kholy and Ahmed Elfatatry
Institute of Graduate Studies & Research, Alexandria University, El horia street, Alexandria, Egypt

Keywords: Web Service, Service Broker, Semantic Web, Service Description, Web Service Description Language (WSDL), Knowledge-based System.

Abstract: Ever since the introduction of the service oriented model of computing (SOA), service discovery has been the major research challenge in SOA. Service consumers usually prefer to express their requirements informally. Expressing requirements in such a way leads to difficulties in the matching procedure, and hence results in poor matching results. In this paper, we present the concept of multi-level search as a solution for matching informal expression of user requirements. In the suggested approach, intermediate brokers receive service requests and suggest suitable services that match the given requests. We present a mechanism by which an intelligent broker utilizes a knowledge based system to overcome the drawbacks of syntactic and semantic discovery. The intelligent broker receives informal user requirements and performs multi-level search. The search starts with key word search, then meaning search, and finally expert search. If the keyword search fails to produce a proper matching, then, the search progresses to the following levels: semantic, and then intelligent search. In this paper we argue that multi-level search could revive the dream of automatic service discovery and present a detailed model for implementation.

1 INTRODUCTION

Service discovery is crucial for the success of service oriented computing. However, it is still posing a research challenge. Service discovery may be performed manually or via automated mechanism. In both cases, the searching interface must be able to compare between the provided capabilities and the required functionality. In this paper, we present the concept of "multi-level search" as a solution for discovery in service oriented systems.

A major problem in service discovery is the informal expression of user requirements. Unclear ideas or ambiguous words in consumer requirements lead to improper matchmaking results. The consumer requirements are sent to an intermediate broker that registers services from different providers. A number of research efforts have focused on enhancement of user requirements. One solution is to use of common ontology to formally describe user requirements (Zhang, 2011), (Baklouti, 2013). Such solution enforces the client to use additional programs and hence affect the platform independence of web services technology.

An alternative approach is to employ intelligent brokers to improve service discovery. However, existing web service brokers have failed to deliver this promise (Zhang, 2011), (Zulkernine, 2011). A significant amount of reasoning is performed by such brokers to fulfill the required functionality. However, existing brokers are still not intelligent enough to deal with the complexity of informal user requirements. Most brokers are capable of applying keyword search and semantic search only (Zhang, 2011).

This paper introduces the "intelligent broker" concept. In our solution, broker receives informal specification of service functionality from the user and then utilizes a knowledge based approach to perform three levels of service discovery. The first level is a syntax search that is based on key word matching between the written user requirements and the described service functionality. In case of no matching, the user requirements are transformed to a controlled English form and are provided with available ontology. Semantic discovery takes place as a second level. In case of semantic search failure, the controlled user requirements are transformed into predicate logic based search. Then, it is

El Kholy M. and Elfatatry A..
Intelligent Broker - A Knowledge based Approach for Semantic Web Services Discovery.
DOI: 10.5220/0005455300390044
In *Proceedings of the 10th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2015), pages 39-44
ISBN: 978-989-758-100-7
Copyright © 2015 SCITEPRESS (Science and Technology Publications, Lda.)

converted to a knowledge based system supported with rules to allow an intelligent search mechanism. These multi levels of service discovery improve the probability of accurate matching.

This paper is organized as follows. Section 2 discusses recent work on search brokers. Section 3 introduces the problems of syntactic and semantic search. Section 4 presents our vision for intelligent service discovery. Section 5 explains the proposed method through a prototype of the intelligent broker.

2 THE BROKER ROLE IN SERVICE DISCOVERY

2.1 Onto Broker

The Onto Broker architecture was introduced as a semantic middleware (Paulheim, 2010). It is capable of sharing meanings (semantics) of information and capturing of complex relationships from heterogeneous data sources. Onto Broker contains a storage layer where ontologies from different sources are stored. Such model requires high storage and complex implementation of collaborating servers. Also, customers have to integrate the Onto Broker API into their own solutions to be connected to onto broker.

2.2 Service Composition Broker

(Zhang) (2011) introduced a method to use the broker to allow flexible semantic web service composition. The broker first applies traditional semantic search to find a service that fulfills the user requirements. In case of matching failure the broker employs an intelligent planner. The intelligent planner generates sub tasks from the user requirements and workflow knowledge. Then the planner maps each task to a web service, and then starts searching again. This process is repeated until the user requirements are satisfied. However, this approach utilizes the artificial intelligence to deal with the user requirements not to preform service discovery.

Most of the recent researches in this area depend on the consumer side. They supply the consumer with domain ontology or intelligent plans to improve service provision. Our vision in this paper is more general in that it accepts informal user requirements. Then, the broker carries out a number of transformations to search for suitable matches. This guarantees faster and more accurate matching.

3 SERVICE DISCOVERY PROBLEMS

3.1 Syntax Service Discovery

Syntax keyword discovery has proven to be insufficient for powerful service discovery (Žemlička, 2014). Web services are described by XML file called Web Service Describing Language (WSDL). WSDL allows developers to mainly describe two essential parts of a developed service: its functionality and how it can be invoked. Matchmaking components at the broker side use the functional descriptions to match users' services against their requirements. The XML Schema Definition (XSD) language is used in the WSDL file to express the structure of the message parts and data types. XSD offers simple types (e.g., integer and string). Hence, traditional WSDL file allows only keyword searching and cannot allow semantic reasoning. On one hand, matchmakers commonly preprocess WSDL documents to extract terms that may allow discoverers to find services relevant to their requests. On the other hand, the user requirement is supposed to be unclear and contain ambiguous words. The weakness of service descriptions or the unclearness of user requirements may lead to two different problems. The first is the matchmaking failure, which means that the user request will have no reply at all. The second is to have a long list of candidate services as a reply to the user requirements. The user must analyze all these services to find out the best service that exactly fulfills his requirements.

3.2 Semantic Web Service Description

Semantic web services have limitations in the discovery process due to the complex implementations of their description methods. Moreover, the semantic description methods suffer from the lack of ontology standers among different interacting parties (Rajagopal, 2006). The problem related to semantic web service discovery has been tackled in many directions. Semantic web service description enhances web service discovery by using non-ambiguous concept definitions from shared ontologies (Zaremba, 2007). Thus the semantic search overcomes the keyword search by utilization interrelationships among data (Rajagopal, 2006). The semantic data are machine understandable data, and its relationships can be achieved through shared ontologies (Ma, 2010). In this section we will focus on two of the common semantic descriptions of web which are WSMO-DF and OWL-S.

3.2.1 Web Service Modeling Ontology Discovery Framework (WSMO-DF)

WSMO describes web services in a rich loosely coupled semantic annotation. WSMO consists of four main elements which are ontology, description, goals, and mediator. The description element describes the service according to the ontology defined in the ontology element. Goals define the user requirement. The mediator manages the interaction between WSMO elements. WSMO-DF is based on WSOA to offer an abstract description of semantic services. WSMO-DF suggests two types of Web service descriptions (Georgios, 2010). The first is the abstract web service description which defines a service in terms of its abstract functionality. The second type is the concrete service description which contains the details and constrains needed for the service consumer. For example, a hotel may offer an abstract service for booking rooms. Then there are the detailed descriptions and information about the service. For instance, number of rooms, date, payment, etc. In WSMO-DF the web service discovery mechanism may use light or rich semantic descriptions. In the light abstract description the abstract services are represented as Complex Concepts (CCs). CCS map similar services to the same class. For instance, order class and search class. The rich semantic discovery mechanism is the most fine-grained level, where Web services are modeled in more detailed specification including their state transitions.

3.2.2 Web Ontology Language for Services (OWL-S)

OWL-S offers a conceptual model for semantically annotating Web services. The model is based on four ontologies, which are: Service, Service Profile (SP), Service Process and Service Grounding. SP plays the main role in service discovery while service process and grounding provide information to use the service. SP contains the main descriptive information of the service including the service name, and other useful information about the provider. It describes the service functional properties in terms of inputs and outputs (Niu, 2010). Profile based web service discovery is one of the most commonly used semantic discovery mechanisms. In such approach, the procedure of matchmaking the service requests and service description depends on that both are represented as profile instances. These profiles include inputs and outputs which are annotated with ontology. The matchmaking is performed using semantic rule formalisms. Listing 1 introduces the complex concept and the service profiles in their semantic service description.

The a1 service is classified in the Order class and requires a passenger name, departure airport, arrival airport, and flight number as input where the output is e-Ticket which is sent to the user Email. a2 is a more general service in the search class. It takes the specification of a product. Then it searches for this product according to its specifications. When a matching product is found, the result (which is the product name) is sent back to the user.

Complex Concepts

1) *a1* ≡ *Order* Π ∀ *passenger name.Name* Π ∀ *depart.Depart* Π ∀ *arrive.Arrive* Π ∀ *flighno.NO* Π ∀ *ticket.e-Ticket* to { email}
2) *a2* ≡ *Search* Π ∀ *specification.Spesification* Π ∀ *product.Product*

OWL-S Service Profile Instances

1) a1 : Orsder, (a1, Passenger Name): hasInput, (a1, Depart): hasInput, (a1, Arrive): hasinput (a1, Flight No): hasinput, (a1,e-Ticket) hasoutput
(a1, Email) : to
2) a2 : Search, (a2, specification) : hasInput,
(a2, product): hasOutput

Listing 1: Semantic Web Service Description Examples.

The analytical study done in this work about different semantic service description proved that none of them is sufficient to ensure the process of service discovery. Unfortunately, semantic web service discovery suffers from high complexity and the lack of standard ontologies. Furthermore semantic search suffers from the absence of public semantically annotated Web Services (Rajagopal, 2006). Moreover, Semantic web services should be invoked and composed by clients who know business process but are not aware of semantic languages. The drawbacks of the syntactic and semantic Web service discovery ensure us that web services cannot depend on these kinds of search (Žemlička, 2014). Web service needs more powerful and intelligent searching mechanism to improve service discovery.

4 INTELLIGENT SERVICE DISCOVERY

The intelligent approach differs from other approaches in that it accepts traditional informal consumer request. It does not enforce the service consumer to apply any additional constraints to formalize his requirements. The proposed system is considered as a gateway between the syntax based user requirement and the intelligent web world. This approach is based on semantic web service discovery in which all services must be registered at a semantic matchmaker. The matchmaker includes the semantic service description in a given ontology. The first step in the proposed approach is to transfer user requirements from informal English to formal English language. We chose the Attempto controlled English (ACE) to be our intermediate formal language.

4.1 Attempto Controlled English (ACE)

The (ACE) is a type of Controlled Natural Language (CNL) which is used to control the user requirements. ACE has been used for two reasons. The first is that it has an easy structure as it is closer to natural English. The second reason is because of the possibility to be unambiguously translated into predicate logic. Moreover ACE is used in knowledge representation for the Semantic Web. ACE is a formal language that contains only short sentences. The structure of the sentence and its type are well defined. These formal sentences can be converted to first order predicate logic which facilitates reasoning in semantic data.

4.2 Search Mechanism

The proposed system receives user request in XML form. Then, three levels of search are applied to the request. Figure 1 shows the three levels. At first, traditional key word search is performed to find the candidate service. If no matching occurs, the user request is passed to the CNL unit which is supported with English vocabulary. This unit introduces the main elements of the sentence as subject, verb, complements and adjuncts. This step solves the issue of ambiguous sentence structure that could be included in the user requirements. Then, formal sentences are passed to ontology provider which checks every word in the sentence and exchanges it with the corresponding ontology word in the same domain. As a result, the user requirements are transferred from syntactic form to semantic form and the second level of semantic search then takes place. The search is performed at a semantic matchmaker where the services are registered with their semantic description. The matching between requested tasks and the semantically described service is performed as ontology reasoning at the semantic matchmaker. The results of successful matches are passed back to the requester. In case of no matching results, the requester's requirements are transferred from the ACE form to the predicate logic form. Then, the logical requirements are passed to knowledge base center which contains defined rules that improve the service search. It also includes the history of the registered services and knowledge on how to fulfill the requester's tasks. The knowledge base includes the history of the requester and his way to identify his requirements. This knowledge base is connected to an inference engine containing rules that support intelligent search at the semantic matchmaker.

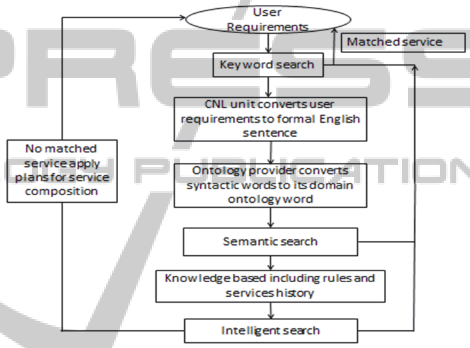

Figure 1: Three levels of service search.

These three levels of search syntactic, semantic and intelligent increase the probability of matching the required service. If no matches are found, a failure report is sent to the requester advising him to divide his requirements to individual tasks and try to perform service composition to fulfill his main task.

5 PROPOSED BROKER STRUCTURE

Figure 2 shows the structure of the intelligent multi-level search broker. The proposed broker is based on the semantic web services assumption where all the services are semantically described. Service brokers invoke the matchmaker to find service capabilities that are included in the requester's requirements. The proposed broker consists of five main units: communicator, controller semantic search unit, intelligent search unit and search engine unit.

A) Communicator: provides the basic interface to communicant with services requesters. It also monitors the results of each level of service search

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

until the matchmaking process is performed successfully. It replies to the requester with the service provider and the basic service description.

B) Controller: controls the execution of the broker. It receives the request from the communicator converting it to a task that has three levels of processing. It manages the execution of these three levels and their communication with the search engine.

C) Semantic Search Unit: consists of two units ACE converter and Ontology provider.
ACE Converter: is an intelligent engine supported with English dictionary that is responsible for converting the user requirements to control formal English sentence. These ACE sentences are used to perform ontology mapping in the semantic search. Then, it is further converted to predicate logic enabling intelligent search.
Ontology Provider: is responsible for matching every word in the formal user sentence to its corresponding ontology at the same domain.

D) Intelligent Search Unit: contains Knowledge-base and services history. This unit stores service requirements parameters represented in predicate logic form. It also contains the history of invoked services and the data to be transformed, including inputs and outputs. The available knowledge is controlled by rules supplied by the inference engine of the knowledge-based system. Such rules and knowledge allow the broker to perform intelligent search.

E) Search Engine: The search engine calls the matchmaker of semantic WS registry to find appropriate services for a consumer's requirements. The search engine performs traditional key search and accepts data from the semantic search unit to perform the semantic search at the matchmaker.

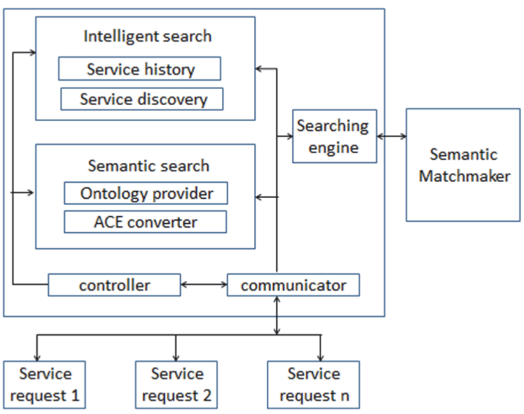

Figure 2: Proposed broker architecture.

6 A SEARCH CASE

Consider a case where a service consumer while writing his code needs to invoke a service that calculates the GPA for a specific student. He writes his requirements using the words: college, subject, and degrees. When these words are passed to the intelligent broker, it will understand that the College may have the same meaning as Faculty and will start representing the user requirement in predicate logic in a form such as:

$$\forall X: \text{subject}(X) \Rightarrow \text{course}(X)$$

$$\forall X: \text{course}(X)\ \text{gpa}(X, \text{student})$$

In semantic repositories the data is saved in triples called Resource Description Framework (RDF). RDF consists of subject, predicate and object. In the introduced example the data about a faculty is saved in a semantic form in RDF triples as shown in Figure (3). So, the intelligent broker can use the predicate logic to search the web semantically. Hence, the intelligent search can catch any triple that gives the same meaning of the predicate logic representation of the user requirements.

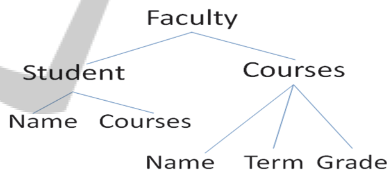

Figure 3: RDF graph representation in semantic web.

7 CONCLUSIONS

We introduced a mechanism for web service discovery that overcomes the problems of keyword search and search and the difficulties in semantic matchmaking. The proposed intelligent broker differs from other approaches in its searching technique. Most of the existing approaches utilize Natural language Processing (NLP) to enhance semantic discovery. The proposed broker goes one step further by utilizing ACE and predicate logic to apply intelligent search. Predicate logic is a promising technique to explore semantic web, since the semantic web is composed of RDF triples based on predicate logic.

REFERENCES

Blake, M. B., Cheung, W. K., Jäger, M. C., Wombacher,

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

2007. WSC-07: evolving the web service challenge. In *IEEE Joint Conference on E-Commerce Technology (9th CEC) and Enterprise Computing, E-Commerce and E-Services*.

Bleul, S., Weise, T., Geihs, K, 2007. Making a fast semantic service composition system faster. In *IEEE Joint Conference on E-Commerce Technology (9th CEC)*.

Bleul, S., Weise, T., Geihs, K.: The web service challenge: a review on semantic web service composition. In: Wagner, M., Hogrefe, D., Geihs, K., David, K. (eds.) *Service-Oriented Computing (SOC'2009), vol. 17. European Association of Software, Science and Technology (2009)*.

Brogi. A., Corfini, Popescu, 2008. Semantic-Based Composition-Oriented Discovery of Web Service. In *ACM Transaction on Internet Technology* 8(4), 19:1-19:33.

Chiu I, Shu L., 2007. Biomimetic design through natural language analysis to facilitate cross-domain analysis. In *(AI EDAM) Artificial Intelligence for Engineering Design, Analysis and Manufacturing*; 21:45–59.

Georgios Meditskos, Nick Bassiliades, 2010. Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S. In *IEEE Transactions on Knowledge and Data Engineering* 22(2), 278-290.

Heiko Paulheim, Florian Probst, 2010. Application Integration on the User Interface Level: an Ontology-Based Approach. In *Data and Knowledge Engineering Journal (DKE)*.

Huang, S., Wang, X., Zhou, 2005. Efficient web service composition based on syntactical matching. In *IEEE International Conference on E-Technology, E-Commerce, and E-Service (IEEE'05)*.

Luo, S., Xu, B., Yan, Y, 2010. An accumulated-QoS-first search approach for semantic web service composition. In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA'10)*.

Ma, H., Jiang, W., Hu, S., Huang, Z., Liu, Z, 2010. Two-phase graph search algorithm for QoS-aware automatic service composition. In *IEEE International Conference on Service-Oriented Computing and Applications (SOCA'10)*.

Michal Žemlička and Jaroslav Král, 2014. Flexible Business-oriented Service Interfaces in Information Systems. In *Proceeding of ENASE- 9th International Conference on evaluation of Novel Approaches on Software Engineering*.

Muhammad Baqar Raza, Robert Harrison, 2011. Design, Development & Implementation of Ontological Knowledge based System for Automotive Assembly Lines. In *International Journal of Data Mining & Knowledge Management Process*.

Nabil Baklouti, Faten Fakhfakh, Bilel Gargouri, Mohamed Jmaiel, 2013. OWL-LingS Editor A Tool for Semantic Description of Linguistic Web Services. In *CLOSER 2013, the 3rd International Conference on Cloud Computing and Services Science*. SCITEPRESS.

Niu, W., Chang, L., Wang, X., Han, X., Shi, 2010. Semantic Web Service Discovery Based on Context and Action Reasoning. In *Pattern Recognition and Artificial Intelligence* 23(1), 65–71.

Rajagopal, S., Selvi, 2006. Semantic grad service discovery approach using clustering of service ontologies. In *Proceedings of IEEE TENCON 2006*, pp. 1–4.

V. R. Benjamins, 1998. IBROW3 — An Intelligent Brokering Service for Knowledge-Component Reuse on the World Wide Web. In *Proc. 11th Workshop Knowledge Acquisition, Modeling, and Management (KAW 98)*.

Yufeng Zhang, Hong Zhu, 2011. An Intelligent Broker Approach to Semantics-based Service Composition. In *35th IEEE Annual Computer Software and Applications Conference*.

Zaremba, M., Vitvar, T., Moran, M., Brambilla, M., Ceri, S., Cerizza, D., Della Valle, E., Facca, F. M., Tziviskou, 2007. Towards semantic interoperabilty: in-depth comparison of two approaches to solving semantic web service challenge mediation tasks. In *Ninth International Conference on Enterprise Information Systems (ICEIS'07)*.

Zulkernine F. H., Martin, P., 2011. An Adaptive and Intelligent SLA Negotiation System for Web Services. In *Services Computing, IEEE Transactions on* (Volume:4, Issue: 1), March.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS