Automatic Web Service Discovery Using Ontology Approach

S.Krishna Priya
PG Student,
Rajalakshmi Engineering College, Chennai.

K.Jayashree
Assistant Professor (SS),
Rajalakshmi Engineering College, Chennai.

R.Chithambaramani
Assistant Professor,
Rajalakshmi Engineering College, Chennai.

Abstract - Different set of web services are published in the registry with out having a clear description. The appropriate selection of services based on the user request is difficult. Hence need for web service discovery is increased. Semantic based ontology approach is used for the automated web service discovery. The web services are created and stored in the service registry. To discover the efficient services based on the user request domain ontology is created. The match making algorithm is used to match the request with services from the ontology repository where the domain ontology stored. The best matching is taken by removing the irrelevant services for the user request. The web service composition which provides multiple services have to be discovered and they together matches the service request.

Keywords - Automated discovery, Web Service composition

I. INTRODUCTION

The web is a huge information space filled with interconnected resources. The service is an application that exposes its functionality through an API. A Web Service is a software interface that describes a collection of operations that can be accessed over the network through standardized Extendible markup Language (XML) messaging. It uses protocols based on the XML language to describe an operation to execute or data to exchange with another Web service. A web service is a standard way of integrating web-based applications using the XML, Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), and Universal Description Discovery Integration (UDDI) open standards over an Internet protocol. The web service is a technology which allows the communication of application to each other in a platform and programming language independent manner. The XML and the web solve the traditional middleware blurs. SOAP is used for exchanging structured information in the implementation of web services.

The web services are published by the service providers in service registry having different categorization. The service similarity should be maintained in similar categories. Large numbers of categories are available in the registry and so it is difficult to discover the similar services from categories. The semantic based ontology approach is considered to be efficient for the categorization and selection of services from different categories. Multiple services are discovered and combined here. Combination of similar web services is a crucial approach for web service discovery.

In this paper we present the architecture for semantic based ontology approach for automated service discovery and selection. In Section 2 discusses the related work and Section 3 presents the proposed architecture for Web services discovery and composition. In Section 4, we have discussed the implementation of web service discovery and composition using a sample web service application. Section 5 presents the conclusion and future work.

II. RELATED WORK

In this section we have discussed about the work related to semantic based ontology for discovery of web services.

Aabhas V. Paliwal et al. [1] projected semantic based service discovery. The web service discovery relies on semantic categorization of web services and semantic enhancement of service request. The semantic categorization of web services is achieved by ontology framework as offline in UDDI. Semantic enhancement of web services is achieved by parameter based service refinement and semantic similarity based matching. The matching of increased service request with retrieved service description is achieved by Latent Semantic Indexing (LSI).Ranking of semantic relationships, hyper clique pattern discovery, additionally used for the invention. Solely single web service is taken into account for matching service request. Therefore web service composition is not satisfied.

Jeberson Retna Raj et al. [2] projected web service discovery based on computation of semantic similarity distance and QOS normalization. The author introduces functional and nonfunctional analysis for computing the similarity activity. The WSDL program extracts the Meta information contents from the WSDL file and it will be preprocessed. Presently once preprocessing, acceptable terms are going to be known. Semantic distance between terms is going to be calculated by Normalized Google Distance. Web service similarity may be determined by using bipartite graph algorithm. The score worth is going to be computed based on aggregation of purposeful and nonpurposeful activity values. For a user request, an inventory of candidate services are going to be provided to the client based on degree of matching with the search query from higher to lower order. Similarity measurement algorithm is employed for sheming the similarity between two terms. The QOS attribute like time interval, throughput, dependableness and availableness may be normalized. The similarity distance that may be obtained from Google that provides correct weight to the several terms. Solely trained and novice user will discover the services based on keyword and operation based mostly queries.

Guisy Di Lorenzo et Al. [3] introduces towards semantic driven generation of executable web services compositions. A life cycle is projected for the automated composition of web services and also the description of data relies on the usage of domain ontologies. Verifiable and valid executable method may be created and that is achieved by exploiting formal definitions of composition rules and of BPEL4WS constructs. Operational semantics is that the formal basis of all the life cycle phases. A primary step towards the definition of a unifying life cycle approach to web service composition development. The development method could begin from a composition goal that describes a particular customer's request or additionally from a composition goal that needs building an web service ready to satisfy a category of requests. The life cycle approach consists of automated synthesis of two graph model like the operations flow model and service work flow model. In operational flow model, choose set of (available) service operations that are semantically compactable. And in service workflow model confirm a composition of such operations that semantically matches preconditions and effects of the requester composite service. Approach addresses the generation of each ad-hoc and re-usable compositions. Check and supply the Input/output matching of the chosen operations; the applying of graph transformation techniques so as to specific the composition by suggests that of workflow patterns. Since life cycle model is employed, the standard of services and also the security needs are not considered.

Puwel Wang et al. [4] introduce building toward capability specifications of web service based on environment ontology. Automated web service discovery needs web service capability specifications of web service supported on environment ontology. The most ideas of environment ontology are the environment entities during a explicit application domain and their interactions. For every environment entity, there is a tree like hierarchic state machine modeling the consequences that are to be achieved by the web services on this environment entity. Algorithms made for domain environment ontology and match making between the web service capabilities shows however a web service discovery supported. The algorithm for constructing environment ontology from general domain ontology has additionally been delineated that the environment ontology is made. Associate effect-based capability profile that creates the web service specification a lot of correct while not exposing the web service's realization details and also the structured effects of the sharable environment entities modeled by FCHMs that build the web service specification a lot of comprehensible with one another and build the web service capability communicatory. The algorithm has additionally been developed to transfer the effect-based profile into a capability specification mechanically. This framework is not ready to develop an effective service discovery support system.

Jing Dong et al. [6] have enforced OWL-S ontology framework extension for dynamic web service composition. Web service may be composed at the abstract service level rather than the concrete level by raising the amount of abstraction and therefore they need to project an abstract service layer. This abstract services are connected with an instant pool, contains all instances of the abstract service that facilitate fail-over and dynamic. The service instance pool permits filtering and plugging in candidate services at runtime. To automatically generate the method on the fly, they used a planner prototype which supported Java Theorem Prover (JTP). The OWL-S extension includes the data on the input/output, flow management, semantic property, and candidate instance pool of the abstract service within the ontology hierarchy. The "instance" section is more to the OWL-S ontology framework. This instance provides the date concerning the candidate concrete service instances for this abstract service. So as to attach the instance section to its service, a replacement predicate "implemented by" is introduced to the basic level OWL-S ontology framework. This links associate instance class to its service class. The planner prototype will take the user's goals and also the service ontology and feed them into the backend interface engine to come up with the results that are abstract services rather than concrete services. The instance class contains the instance pool data for the abstract service. This OWL-S ontology framework couldn't support a lot of complicated optimization.

Georgios Meditskos et al. [8] projected an approach Structural and Role Oriented Web service discovery with taxonomies in OWL-S. They projected web service discovery framework by OWL-S advertisements, combined with the excellence between service and web service of the WSMO Discovery framework. They need used a web service matchmaking algorithm that extends object-based matching techniques in structural case-based Reasoning, the existence of profile taxonomies, incorporating domain information through profile instance class membership relationships. The WSMO Discovery Framework (WSDMO-DF) is used for web service discovery. The mixture of object-based structural matching techniques that are employed in the domain of structural case-based reasoning(SCBR), with Description Logic(DL) reasoning over profile instances, enhancing the invention with services that can't be retrieved using solely logic-based reasoning. WSMO-DF and OWL-S SP are two frameworks they permit the existence of profile taxonomies that change the annotation of I/Os with ontology roles and outlined a matchmaking algorithm that exploits the structural information of ontologies. Web service composition ought to be the vital one to consider.

III. PROPOSED ARCHITECTURE

In this paper we presented semantic ontology for the automated web service discovery and the composition of web services. The semantic based service discovery involves semantic categorization of services and semantic enhancement of service request. The services published in the registry should be categorized and this will do as offline. The service selection includes refinement of web service with input, output, description parameters and enhancement of service request with ontology concept. The OWL-S is an ontology built on top of Web Ontology Language (OWL) based framework of the Semantic Web, for describing Semantic Web Services. The semantic ontology based web service discovery and the web service composition is done by using OWL-s. The architecture for Semantic Based Automated Service Discovery and composition is given in Figure 1.

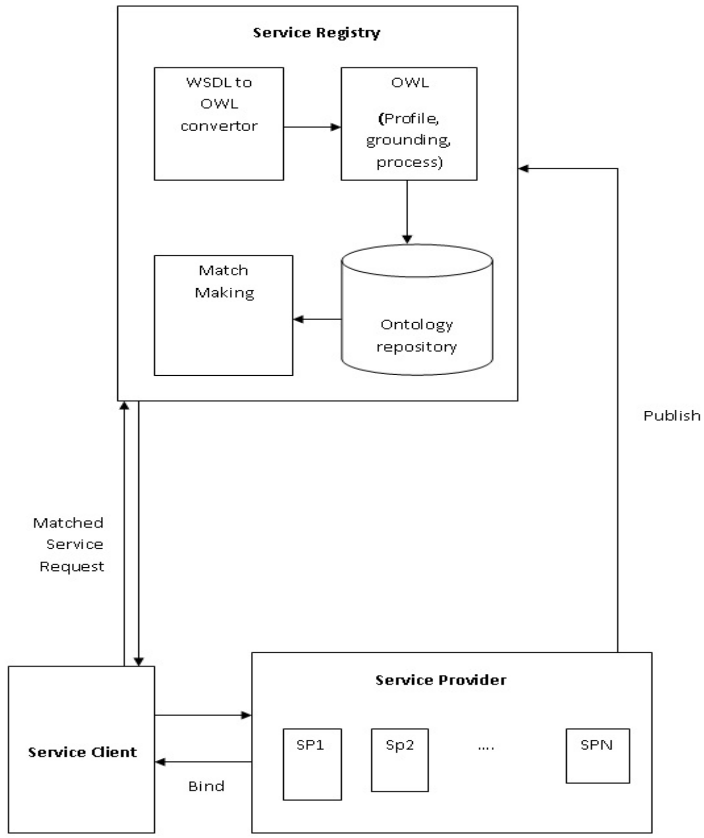

Figure 1: Service Discovery and Composition

The service client makes use of the service. It implements the client side of the client server interface. It uses the registry to discover a service and its description. In the interact operation the service client invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact, and invoke the service.

The service provider implements the server side of the client/server interface. It provides a service to its clients. The services are published within the registry. A service needs to publish its description such that the requestor can subsequently find it. The registry is a network server that maintains details of available services. The registry data is provided by the service providers and queried by the service clients.

The services are created with all user requirements and 'N' no. of service providers publishes their 'N' no. of services in the form of WSDL file to service registry. Service Providers publishes the services along with their Constraints to the service registry. A WSDL to OWL converter tool is used to convert from WSDL to OWL which in turn generates profile, process, and grounding. Using these, a domain ontology is created and that is stored in a repository called ontology repository.

On Service client's request, ontology which was present in ontology repository is parsed to get the information about the services. Services are then searched in the service registry and it is processed through match making algorithm. Using the match making algorithm the services are matched. After filtering out all unmatched services, a best matched service is discovered. A URL is discovered from the service regisrty and it is invoked by the service client and finally binded to the service provider. Here we are using Travel Agency application.

Existing services are combined and it provides multiple services have to be discovered and they together match the service request. Ontology based semantic web uses OWL-s will help the combination of web services. Similar services are combined based on the keywords with user request.

IV. IMPLEMENTATION

Web services are created for the application Travel Agency. The Web Services are created in NetBeans IDE6.1 version. The jbox server version 5 is used as server. It is the easiest way to create the web services in java platform through NetBeans. Hence java development kit 6 is installed along with NetBeans. Neon Toolkit is an OWL editor which used for the creation of semantic models. The neon Toolkit supports the complex class expressions. The Ontograph is generated through the tool protégé.

We have created four basic services as Bus, Train, flight and Banking services. Under these services, sub services are created. Each Services having their own functionalities and uses. The Bus services will give the information about the buses which is having the predefined sources and destinations. Also the bus services will facilitates the services to user like seat availability and booking of ticket. Likewise for Train service provide the user about the Train and the availability of the train for the predefined source and destination. The Flight services provide the information about available flights seat availability for the user given input as source and destination. The Banking services provide services to the user by booking the ticket. The user can book the ticket by using their credit cards as we facilitate credit card services to user. The Booking services are common for Bus Train and Flight services. The sub services are created for providing user needed information. For providing the efficient services to the user we are using ontology. Ontology means set of concepts in a domain and the relationship among these concepts. Concepts are set of classes. Each ontology having the components like individuals, classes attributes and relations. Here we used the Neon Toolkit for the creation of ontology for the application Travel Agency.

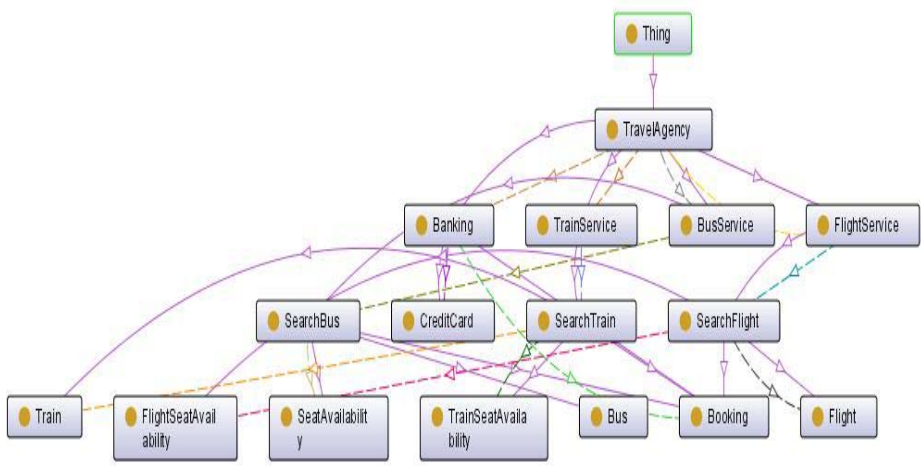

Figure2: onto graph for Travel Agency Application

OWL classes are set of individuals which having similar characteristics. OWL classes are described through class description which may have more than one subclasses. Individuals are the instances or the objects. The way which the classes and individuals are related each other as relations. Here created classes are Travel Agency, Bus service, Train service, Flight service, banking service; the searching of Bus Train Flight also included as class and also included the seat availability check. The booking service is created as common to all other classes. Each classes having their own functionality and characteristics in the travel agency application.

An object property is given to each classes created which is a binary relation between two individuals. Each object property we have assigned the domain and its range. The domain and range link individuals from the domain to individuals from the range. The domain of a function, which is set of values it defined, and the range is the set of values it can take. Similarly in OWL, if it specified by a class A as the domain of the property P and class B as the range of P, the informal interpretation of this is that the Property P relates values from the class A to values from the class B. For example for the relation hasbooking having the domain as banking and the range as booking. Also for the relation hasseatavailable having the domain as searchbus and range as seatavailable. Here we can specify the properties also. We are allowed to give the properties like super properties/sub properties. For example in our application for the booking property we are having super properties like hasbanking, hassearchbus, hassearchtrain and hassearchflight. And the sub property is hascreditcard. Likewise for SearchBus having super property as hasbusservice and sub property as hasavailable, hasbooking and hasseatavailable. The equivalent property also mentioned here.The equivalent property in the sense two property have the same extension. Here in our application has Available having the equivalent property of hasSeatAvailable. Also the inverse property can mentioned. Usually the relations are defined from domain to range but in inverse property the relation can be mentioned in both direction. Here in our example the busservice having the inverse property of SearchBus. The relationships are mentioned for every classes. For each services in the travel agency application like Bus, Train and Flight is related to banking service. Ticket booking is a part of banking. Seat availability check is union for Bus train and flight services. The credit card is part of banking. The online booking of ticket is done by using credit card.

The data property are simple types which define the XML schema types. The data properties for each class are specified with its domain and range given. The domain and range of data property link individuals from the domain to RDF literals or XML Schema datatypes. Here for example for the Bus data property having domain as Busservice and range as string. Also for CVCNumber having domain as credit card and range as integer.

V. CONCLUSION AND FUTURE WORK

The efficient web service discovery needs, the user must be able to discover all appropriate web services within the UDDI irrespective of the predefined categories, and all appropriate web services must be successfully discovered even if the user is not aware of all the relevant terms that include all appropriate web services. In this paper we have considered the semantic based ontology approach involves service categorization and selection of services with semantic service description and the composition of web service using OWL-S. We have tested the proposed approach by using a sample web service application. As future work, we extend to explore additional mapping tools to express service request to search for relevant concepts.

REFERENCES

[1] Aabhas V.Paliwal, Jaideep Vaidya, Nabila dam, “Semantic – Based Automated Service Discovery “IEEE Transactions on Service Computing, 2012.

[2] Jeberson Retna Raj, Dr.T.Sasipraba, “Web Service Discovery Based On Computation of Semantic Similarity Distance and QOS Normalization “, Indian Journal of Computer Science and Engineering (IJCSE), 2012.

[3] Guisy Di Lorenzo, Nicola Mazzocca, Francesco Moscato, valeria Vittorini, “Towards Semantics Driven Generation of Executable Web Services Compositions “, Journal of software, 2007.

[4] Puwei Wang, Zhi Jin, “Building toward capability Specifications of Web Services Based on an Environment Ontology “, IEEE Transactions on Knowledge and Data Engineering, 2008.

[5] Miguel Angel Corella and Pablo Castells,” Semi-Automatic Semantic-based Web Service classification “proc.Int’l Conf.Business Process Management workshop, 2006.

[6] Jing Dong, Yongtao Sun, Sheng Yang, “OWL-S Ontology Framework Extension for Dynamic Web Service Composition **“**IEEE Transactions on Knowledge and Data Engineering, 2009.

[7] Jorg Nitzsche, Tammo van Lessen, Dimka Karastoyanova, and Frank Leyman, “BPEL for Semantic Web Services (BPEL4SWS) **“**Springer-Verlag Berlin Heidelberg 2007.

[8] Georgios Meditskos and Nick Bassiliades,” Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S “IEEE Transactions on Knowledge and Data Engineering, February 2010.

[9] Jyotishman Pathak, Neeraj Koul, Doina Caragea and Vasant G Honavar, “A Framework for Semantic Web Services Discovery “, *WIDM’05,* November 5, 2005, Bremen, Germany.

[10] Srividya Kona, Ajay Bansal, and Gopal Gupta and Thomas D. Hite, “Semantics-based Efficient Web Service Discovery and Composition.

[11] www. wikipedia.org

[12] www.w3schools.com