Comparative Study on Semantic Web Service With and Without OWL-S

Yamuna B R

Department of Computer Science
Govt. Engineering College, Dairy Circle Hassan 573201, Karnataka
yamuna.ramaiah@gmail.com

***Abstract*— The composition of complex services at design time is a well-understood principle which is nowadays supported by many broadly available tools. It's already beyond the human ability to analysis them and generate the composition plan manually. This paper proposes a framework for dynamic and personalized composition of web services using an approach that integrates not only functional attributes of web services, but also nonfunctional attributes such as service requirements, quality of service (QoS), and some implementation details are also presented. In this work we present initial focuses on the issue of automated service composition in conjunction with the semantic web. In this report we propose a composition that automatically handle the integration of web services through the use of an Ontology Web Language (Owl-S). The adhoc queries of user are split into meaningful query terms and each term is evaluated against service available (through WSDL and OWL-S description). The comparison results have been revels that the false positive rate and false negative rate for different numbers of services with and without OWL-S as shown. Analysis rate is high in case of without using OWL-S as compared with OWL-S.**

***Keywords*— Ontology Web Language, Quality of Service, Semantics, Universal Description Discovery and Integration, Web service.**

I. Introduction

One of the most interesting inventions in recent decades is that of web services [1]. These are computer program "applications": self-describing, self-contained applications whose function is to automatically share information over the Internet with other applications. Some weaknesses such as browsing information without taking its meaning into account have recently appeared in web services. This creates a need for a new web with more relevance to the user.

The information on the web can be defined in a way that can be used by computers not only for display purposes, but also for interoperability and integration between systems and applications. One way to enable machine-to-machine exchange and automated processing is to provide the information in such a way that computers can understand it. This is precisely the objective of the semantic web to make possible the processing of web information by computers.

The semantic web is not a separate web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation [2].

Semantic web is actually an extension of the current one in that it represents information more meaningfully for humans and computers alike. It enables the description of contents and services in machine-readable form, and enables annotating, discovering, publishing, advertising and composing services to be automated. It was developed based on ontology, which is considered as the backbone of the semantic web. In other words, the current web is transformed from being machine-readable to machine understandable.

One function of the web is to build a source of reference for information on several subjects, while the semantic web is designed to build a web of meaning. The foundation of vocabularies and effective communication on the semantic web is ontology. "Ontology provides a formal, explicit specification of a shared conceptualization of a domain" [1, 3]. Therefore, it facilitates knowledge sharing over distributed systems; in other words, it allows systems or applications to cooperate, that were not formerly designed to interoperate. Ontology plays a major part in solving the problem of interoperability between applications across different organizations, by providing a shared understanding of common domains.

Recently, ontologies have become a popular research topic in many communities, including knowledge engineering, electronic commerce, knowledge management and natural language processing [4]. Ontologies provide a common understanding of a domain that can be communicated between people, and of heterogeneous and widely spread application systems. In fact, they have been developed in Artificial Intelligence (AI) research communities to facilitate knowledge sharing and reuse.

Ontology Representation

Ontology is comprised of four main components: concepts, instances, relations and axioms [4]. The present research adopts the following definitions of these ontological components:

A Concept (also known as a class or a term) is an abstract group, set or collection of objects. It is the fundamental element of the domain and usually represents a group or class whose members share common properties. This component is represented in hierarchical graphs, such that it looks similar to object oriented systems. The concept is represented by a "super-class", representing the higher class or so called "parent class", and a "subclass" which represents the subordinate or so-called "child class". For instance, a university could be represented as a class with many subclasses, such as faculties, libraries and employees.

An Instance (also known as an individual) is the "ground-level" component of an ontology which represents a specific object or element of a concept or class. For example, "Jordan" could be an instance of the class "Arab countries" or simply "countries".

A Relation (also known as a slot) is used to express relationships between two concepts in a given domain. More specifically, it describes the relationship between the first concept, represented in the domain, and the second, represented in the range. For example, "study" could be represented as a relationship between the concept "person" (which is a concept in the domain) and "university" or "college" (which is a concept in the range).

An Axiom is used to impose constraints on the values of classes or instances, so axioms are generally expressed using logic-based languages such as first-order logic; they are used to verify the consistency of the ontology.

The expressiveness of ontology is based on the degree of explication of the (meta-) knowledge. Several ontologies capture specific domains or certain applications, while others try to capture all terms in natural language. Ontologies that capture extra relations and extra constraints are considered to be more expressive, because they capture knowledge of the domain on a more detailed level. On the other hand, the expressiveness of ontology is restricted by the languages used for describing or specifying it. Ontology languages can be seen as restricting the expressiveness of the ontology [5].

II. OWL-S

OWL-S defines an upper layer for describing the properties and capabilities of web services in OWL. It is intended to enable users and software agents to automatically discover, invoke, compose, and monitor web resources offering services, under specified constraints. Fig. 1 shows, how high level constructs such as a service profile: to represent the interfaces of services including inputs, outputs, preconditions and effects, a service (process) model to represent the details of inner working of a service, and a service grounding to provide information about how to use a service, whereas OWL-S profile model views a service as an atomic process, OWL-S service (process) model captures the state of a service as a complex interaction process are defined [8]. While OWL-S profile defines a model for describing the functional properties of a service via constructs such as inputs, outputs, preconditions and effects. OWL-S service model uses workflow constructs such as sequence, if-then-else, fork, repeat-until and so forth, to define a composite processes. OWL-S grounding model defines the necessary links to web service industry standard, Web Service Description Language (WSDL) to use its invocation model.

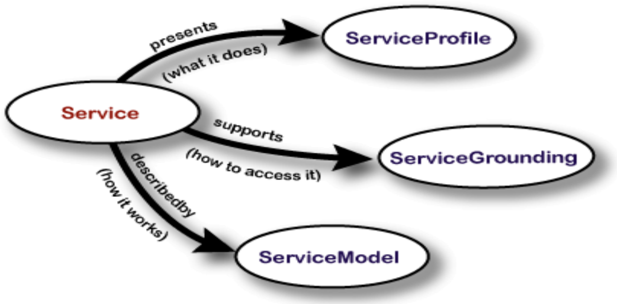

Fig. 1 Top Level of the Service Ontology

The present work however makes use of concepts of the latest version of OWL-S and Universal Description, Discovery and Integration (UDDI) to enhance the added service classification and service product properties. Fig. 2 shows that the query, first converts to WSDL then to OWL-S. The service classification property of OWL-S, similar to the UDDI classification property, is used to represent the categories to which web services belong. The service classification property uses OWL concepts to represent their categories as opposed to syntactic codes used in UDDI. Therefore service classification properties are matched based on their semantic meanings instead of relatively inferior string-based matching. Here, UDDI [6] is an industrial initiative to create an internet-wide network of registries of web services for enabling businesses to quickly, easily, and dynamically discover web services and interact with one another. UDDI allows businesses to register their presence on the web by specifying their points of contact both in terms of the ports used by the service to process requests and in terms of the physical contacts of people who can answer questions about the service.

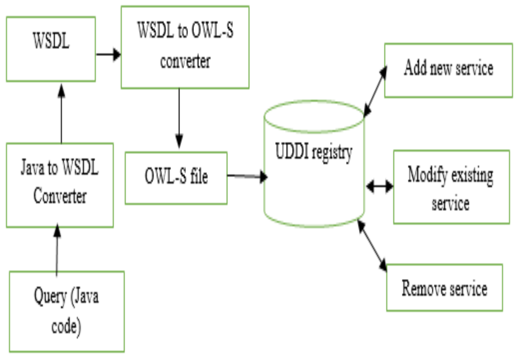

Fig. 2 Semantic Web Service Composition using OWL-S and UDDI

III. Result and Analysis

To make services ubiquitously available, it needs a semantics-based approach such that applications can reason about a service's capability to a level of detail that permits their discovery, deployment, composition and synthesis. In order to allow interoperability and machine-readability, a common conceptual ground must be agreed upon. The first step towards this common ground is standard languages such as UDDI and OWL-S. Industry specific ontologies along with OWL-S can also be used to formally describe web-services. This is the approach taken by the OWL-S language.

It is important to establish the Quality of Service (QoS) of a composite service at runtime based on the quality of individual web service. To make sure of that and to increase the expectation of user, false positive rate and false negative rate has been calculated. The false positive rate is considered as the number of services that are available in the registry but not showing the result and false negative rate is calculated on the services which are not available in registry.

A. *False Positive Rate*

The false positive rate means showing results even though the web services are available, means showing result wrong for the service available in the repository.

The Table. I shows the false positive rate analysis of number of services with and without OWL-S. Without using OWL-S, analysis result is high as compared to with OWL-S.

Table. I False Positive Rate With and Without OWL-S

No. of Services	Without OWL	With OWL
1	0	0
2	0	0
3	1	0
4	2	0
5	2	1
6	3	2
7	4	2

Fig. 3 shows the false positive rate analysis for both with and without OWL-S against the number of services. From three onwards (No. of services), rate is increased without using OWL-S. But using OWL-S, this rate is low.

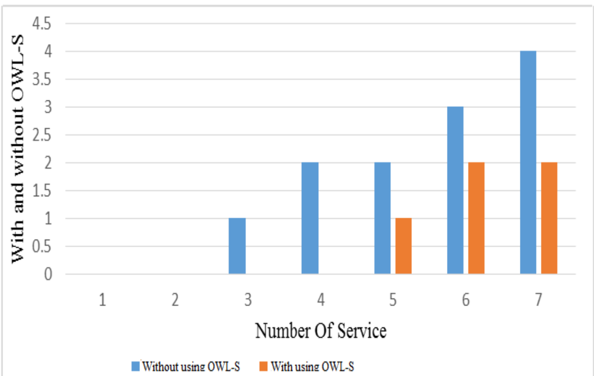

Fig. 3 False Positive Rate Analysis

B. *False Negative Rate*

False negative rate means, showing results for the service which are not available in the repository. The Table II shows the false negative rate for different numbers of services, with without OWL-S. The analysis rate is high in case of without OWL-S as compared to with OWL-S.

Table. II False Negative Rate With and Without OWL-S

No. of services	Without OWL	With OWL
1	0	0
2	0	0
3	0	0
4	0	0
5	1	0
6	2	1
7	2	1

The Fig. 4 shows the graphs for false negative rate. As the number of services increases, with using OWL-S false negativity is less as compared to without using OWL-S.

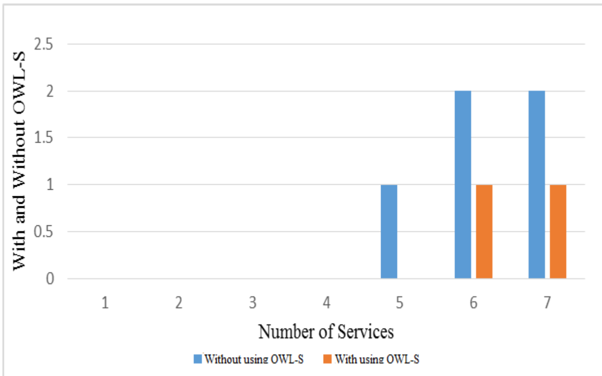

Fig. 4 False Negative Rate Analysis

IV. CONCLUSIONS

The approach presented in the study achieves better results using OWL-S and UDDI for better composition of Semantic web services. It focuses on service description that can capture the functionality more precisely. Without consideration of the pre/post conditions, the composition may result in an inappropriate solution, not satisfying the user goal from the functionality point of view. This also explores the possibility to realize a QoS-driven composition, even if large-scale service registries are considered. Moreover, in practical scenarios, the service composition must deal with the changes in the service environment and must be able to process multiple queries from different users. In real applications, these methods also handle the transactional behaviour of service composition execution.

REFERENCES

[1] D. Tidwell, "Web Services-The Web's Next Revolution", IBM Web Service Tutorial, http://www-106.ibm.com/developerworks/ edu/ws dwwsbasicsi.html, 2000.

[2] T. Berners-Lee, J. Hendler, & O. Lassila, "The Semantic Web. Scientific American", pp.34-43, 2001.

[3] M. Paolucci, T. Kawamura, T.R. Payne, and K.P. Sycara, "Semantic Matching of Web Services Capabilities", In Proceedings of International Semantic Web Conference, 2002, pp.333-347.

[4] Mohammad Mustafa Taye, "Understanding Semantic Web and Ontologies: Theory and Applications", Journal of Computing, Volume 2, Issue 6, ISSN 2151-9617, 2010.

[5] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications", Knowledge Engineering Review, vol. 11, no. 2,1996.

[6] Oasis Consortium, "The UDDI Technical White Paper", http://www.uddi.org, 2000

[7] Nebil Ben Mabrouk, Sandrine Beauche, Elena Kuznetsova, Nikolaos Georgantas, and Val_erie Issarny, "QoS-aware Service Composition in Dynamic Service Oriented Environments", inria-00468178, version 1,2009.

[8] Jing DONG, Yongtao SUN, Sheng YANG, & Kang ZHANG1, "Dynamic web service composition based on OWL-S", Science in China Series F: Information Sciences, Vol.49, No.6, pp. 843-863, 2006.