A Literature Survey on Efficient Web Service Discovery and recommendation Framework by integrating various Web services using QoS, QoE and Service Social Network

Gadekar P. R.[1], Chavan V. G.[2], Pratibha Pawar[3], Pingle S. V.[4]
[1]Assistant Professor, *SKNSCOE, Korti, Pandharpur, Solapur University, Solapur- India*
[2]Assistant Professor, *SKNSCOE, Korti, Pandharpur, Solapur University, Solapur- India*
[3]PG Student, *SKNSCOE, Korti, Pandharpur, Solapur University, Solapur- India*
[4]Assistant Professor, *SKNSCOE, Korti, Pandharpur, Solapur University, Solapur, MS, India*
[1]gadekar.prakash@gmail.com [2]Vijaysinh.chavan@sknscoe.ac.in [3] pratpawa16@gmail.com , [4]Sub.pingale83@gmail.com

Abstract-Today is the world of web and services, new services mostly combine with already available services for providing a particular functionality instead of creating a new service for providing new functionalities. Though, Web services are increasingly becoming the most dominating implementation for the service oriented architecture paradigm for enterprises due to their ease of use. Their uptake on a Web scale has been significantly less than initially anticipated. One of the main causes for being the uptake significantly less is that, the isolation of services and the lack of social relationships among related services. Other reason is the lack of semantic information about the web service description at the time of publication. Hence, this paper proposes connecting the isolated service islands into a global social service network to enhance the services sociability on a global scale, considering the QoS parameters and QoE. Once, the social service network of web services is created we can use this for web service discovery as well as for web service recommendation.

Keyword - Web service, semantic information Global service network QoS, QoE

I. Introduction

Data the architecture used for designing the software is very important as it gives the early view of the software being developed. A service-oriented architecture (SOA) is an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network. Services can be combined to provide the functionality of a large software application. SOA makes it easier for software components on computers connected over a network to cooperate.

Web service is the new distributed computing paradigm for Web application, which uses SOA and is the most popular implementation of service oriented architecture. The reasons behind the web services as popular implementation of SOA are their interoperability, they use standardized protocols and have a low communication cost.

Web Services have much to offer towards interoperability of applications and integration of large scale distributed systems. To make Web services accessible to users, service providers use Web service registries to publish their services. In order to integrate these services, one must be able to locate and acquire specified services. Existing Universal Description Discovery Integration (UDDI) technology uses a central server to store information about registered Web services.

Web services use standardized industry standard protocol for the communication. All the four layers (Service Transport, XML Messaging, Service Description, and Service Discovery layers) use well-defined protocols in the web services protocol stack. This standardization of protocol stack gives the business many advantages such as a wide range of choices, reduction in the cost due to competition, and increase in the quality.

Web services use SOAP over HTTP protocol, so you can use your existing low-cost internet for implementing web services. This solution is much less costly compared to proprietary solutions like EDI/B2B. Besides SOAP over HTTP, web services can also be implemented on other reliable transport mechanisms like FTP.

Despite all the advantages of the web services, numbers of web services available on web now-a-days are much less than the expected. While there are trillions of Web pages available on the Web, the number of publicly available Web services in one large composition service system is not greater than 4000, which is very small. On the other hand, a large body of research about Web service discovery has been devoted to keyword or semantic based discovery to improve the quantity and quality of service matching performance. Nevertheless, from a recent survey, most services published on the Web have not been used, and only a few of the services on the Web have been discovered, composed or invoked. Lack of quality mechanism for discovering the web services affects the composition of web services as the required web services can't be discovered in an efficient manner, which in turn affects the creation and publication of the web services.

After investigating for lack in the discovery mechanisms following reasons can be inferred.

Current approaches for the web service discovery only consider services as isolated functional islands with no links to related services.

Services are considered only in terms of their own functional and nonfunctional properties and its social interaction with other peer services is ignored.

Service's sociability is the skill, tendency or property of being sociable or social, and of interacting well with related services, which is supported by the network models we refer to here as service social networks. A service social network is constructed to reflect services' social reality, describe the mutual consciousness of mutual agreement about a social situation and to support the services' future social activities.

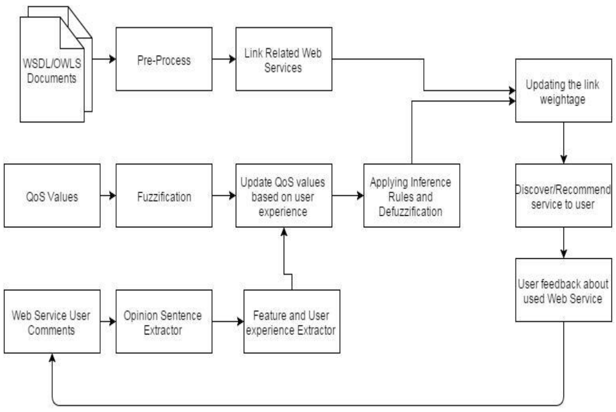
WSDL/OWLS Documents
Pre-Process
Link Related Web Services
Updating the link weightage
QoS Values
Fuzzification
Update QoS values based on user experience
Applying Inference Rules and Defuzzification
Discover/Recommend service to user
Web Service User Comments
Opinion Sentence Extractor
Feature and User experience Extractor
User feedback about used Web Service

Fig 1: System Architecture

WSDL Document Pre-process: The input for this step is WSDL document and output will be a vector consisting name of web service, message names used in the WSDL, list of names of input and output parameters used in each message, description of web service etc.

Link related web services: This module will process the vector of each web service got from pre-process step and will link the related web services.

Opinion Sentence Extractor: Input to this module is user comments on all the web services whose WSDL documents are considered for forming the social network of web services. It extracts the sentences which gives the opinion about the QoS parameter of service.

Feature and User experience extractor: This module takes the opinion sentences as the input and finds the exact feature about which the user commenting and user sentiments behind it.

Fuzzification: This module takes the crisp QoS parameter values and converts those into fuzzy linguistic variable. We have used overlapping trapezoidal membership function for the same. As user gives his experience in natural language, we have to bridge a gap between crisp values and experience in natural language and hence we have used fuzzification.

Update QoS values: This module will update the QoS values based on the user experience got after processing the user comments.

Applying inference rules and Defuzzification: This module will apply the inference rules to get the resultant QoS value in linguistic variable form, which we will further defuzzify to get the resultant crisp value for a QoS parameter. We have used center of gravity mechanism for Defuzzification.

Update link weight-age: This module will calculate the weight-age of link between two web-services by using the functional and non-functional parameters along with the sentiment of user comments.

Discover/Recommend service to user: This module will Discover/Recommend the web services to user based on user query/previous usage pattern of user

II. LITERATURE SURVEY

It proposes the approach to Considers the description of a particular web service checks for the peer services. To give weightage to link of social services this paper considers different functional and non-functional parameters of a web service. The non-functional parameter means the different QoS values. Quality of Service (QoS) has been widely used as a standard way to model and evaluate the non-functional features of a web service. Typical QoS features include reliability, response time, security, and invocation fee. QoS plays an essential role in various web service management tasks, such as selecting a service that fulfils both the functional and non-functional requirement specified by a user. It also serves as the key criterion to differentiate web services that provide similar functionality [1].

The above approach has QoS considered in it but it also has some limitations:

1. It does not always reflect in what users are interested

2. It relies on the QoS information published by service providers.

To overcome above issues this paper suggest to use the sentiment analysis of user review also to extract the feature about which the user review is talking about [2].

The paper presents an attempt to apply Neurofuzzy in the design and implementation of a rule-based scheduling algorithm to solve the shortcoming of well-known scheduling algorithms. A fuzzy-based decision maker has been proposed to compute a new priority of all CPU processes according to the process pre-priority and its execution time. Results given in this paper demonstrate that the average waiting time and the average turnaround time in the proposed algorithm are better than that obtained using pre-emptive priority scheduling, and closed to that obtained from pre-emptive shortest-job-first (SJF) scheduling. The new proposed algorithm is a dynamic scheduling algorithm which deals with both process priority and its execution time, while the pre-emptive SJF scheduling algorithm doesn't. The results obtained, using Neurofuzzy, are approximately the choice as for fuzzy but it responds faster than it. On the other hand the functional Neurofuzzy is the best algorithm compared with structural and fuzzy scheduling algorithms [3].

In [4] paper, they present how to mine product features. The proposed extraction approach is different from the previous methods because it only mines the features of the product in opinion sentences which the customers have expressed their positive or negative experiences on.

In order to find opinion sentence, a SentiWord Net-based algorithm is proposed.

There are three steps to perform our task:

1. Identifying opinion sentences in each review which is positive or negative via SentiWordNet
2. Mining product features that have been commented on by customers from opinion sentences
3. Pruning feature to remove those incorrect features.

In the [5] paper the propose below points

1. To create a synthetic dataset of WSDL/OWL-S documents, QoS parameter values and user reviews.
2. To extract features and sentiment from user comments.
3. To rank the functionally similar web services using fuzzified QoS parameters.
4. To link similar web services and form a social network of web services.
5. To compute the weight-age of each link in the social network of web services.
6. To recommend web services to user based on users query/previous usage pattern of user.
7. To create a platform that allows user to give feedback about used/searched web services.

III. CONCLUSION

The Current UDDI structure is such that a web services is not having any knowledge about the other web services Due to this there is limited scope for the web services Discovery But, as per this paper web service can know about other web services in some way. Then user can have a better web service discovery. Option as when he queries for a specific web service, all the related services can be detected and can be provided to the user improving the user experience. This paper uses the concept of social networking to create a links between the web services using many different functional, non-functional parameters along with the user comments about the web services.

REFERENCES

[1] Wuhui Chen, Incheon Paik, Patrick C.K. Hung, "Constructing a Global Social Service Network for Better Quality of Web Service Discovery", IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 8, NO. 2, MARCH-APRIL 2015

[2] Xumin Liu, Arpeet Kale, JavedWasani, Chen Ding, and Qi Yu, "Extracting, Ranking, and Evaluating Quality Features of Web Services through User Review Sentiment Analysis", 2015 IEEE International Conference on Web Services

[3] ShathaJawad, "Design and Evaluation of Neurofuzzy CPU Scheduling Algorithm", Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference.

[4] Weishu Hu, Zhiguo Gong, JingzhiGuo, "Mining Product Features from Online Re-views", IEEE International Conference on E-Business Engineering 2010

[5] E. Al-Masri and Q.H. Mahmoud, "InvestigatingWeb Services on the World Wide Web," in Proc. 17th IEEE Int'l World Wide Web Conf., 2008, pp. 795-804.

[6] W. Jiang, D. Lee, and S. Hu, "Large-Scale Longitudinal Analysis of SOAP-Based and RESTful Web Services," in Proc. 19th IEEE Int'l Web Serv. Conf., 2012, pp. 218-225.

[7] Z. Maamar, H. Hacid, and M.N. Huhns, "Why Web Services Need Social Networks," Proc. IEEE Internet Comput., vol. 15, no. 2, pp. 90-94, Mar./Apr. 2011.

[8] F.Y. Wang, D. Zeng, K.M. Carley, and W. Mao, \Social Computing: from Social In-formatics to Social Intelligence," IEEE Intell. Syst., vol. 22, no. 2, pp. 79-83, Mar./Apr. 2007

[9] Yen, J., Langari, R.: Fuzzy Logic; Intelligence, Control, and Information. Prentice Hall, Englewood Cliffs (1999)

[10] J. Davies, J. Domingue, C. Pedrinaci, D. Fensel, R. Gonzalez Cabero, M. Potter, M.ichardson, and S. Stincic, “Towards the Open Service Web,” BT Technology Journal, vol. 26, no. 2, 2009.

[11] C. Petrie, “Practical Web Services,” IEEE Internet Computing,vol. 13, no. 6, pp. 94–96, 2009.