2012 IACSIT Hong Kong Conferences
IPCSIT vol. 30 (2012) © (2012) IACSIT Press, Singapore

QOS Based Service Search and Composition Algorithm

Maria Allauddin Shah[1+] and Farooque Azam [2]

[1] College of Electrical and Mechanical Engineering (CEME)

[2] National University of Sciences and Technology (NUST) Islamabad, Pakistan

Abstract. Service Oriented applications are becoming very popular due to ease of Web services Usage. A significant amount of work has been done by industries, universities and individual researchers to locate a web service and compose them. But still there is a space to fill out for particular requirements. There is a need of a strategy to locate web services across multiple heterogeneous registries which still is a challenging task because it has issues like performance, flexibility, reliability. We have using Google Custom Search API for service search and then matrix match making for composition.

Keywords: web service, discovery, google API, composition.

1. Introduction

A wonderful trend in technology of the age is that Web services serve on internet as replacement of applications. Services are small components present on internet that can cooperatively make a complete application environment.

Services are small components present on internet that cooperatively make a complete application environment. Web Services are applications that can be published to be found on internet and then invoked to give result of the operation defined in it.

A Web service can be an application component like: currency conversion, weather reports, or even dictionary as service. They also solve interoperability problems by providing a way to exchange data between different applications with different platforms. So they are gradually attaining preference as a technology among developers and businesses. Let's take an example of online software that deals with tourism: a tourist portal or a travel guide. This tourist portal may need more than one service at a time. For example, ticket booking, area information, hotel reservation, and weather information.

We have used Google Custom Search API to provide the user search its required service. Though the API is not specific for web service search but we have customized it to produce results that are only web services. With the increasing number of web services, it is becoming more difficult to find the specific service. Composition process is interface based i.e. a matchmaking if performed for number of input output parameters. User is prompted if the quantity of parameters is not same. The user has option to enter other parameters or select some from first output as required.

1.1. Contribution

We have proposed a framework to overcome some web service discovery and composition problems. Using Google Custom Search API provides the flexibility to search the user query on more than one heterogeneous registry at a time. We have programmed to retrieve only relevant WSDL files that are valid and available. The limitation of UDDI search is also defeated as whole web is searched for the user query

[+] Corresponding author. Tel.: + 925822444451.
E-mail address: *marya256@gmail.com.*

word, so there is no more specification of searching by service name/category only. It provides a reliable and trust-worthy service discovery. And further it provides up to date information.

Using simple xml messaging we have provided a flexibility of communication for integration. Also it's helpful for complex type services where most frameworks fail to integrate services. There is no need to worry about data types and hence complex data types. Compositional correctness is guaranteed by matchmaking and final input decision by user.

2. Related Work

A web service search engine [1] has its basis on the study that centrally maintained repositories are not enough to service search and keyword search does not provide full matching requirements for user query. At first there is a focused crawling for WSDL. They have considered the information provided in WSDL documentation. In a next step they have refined the results on user's explicit feedback from users. They used HeritrixWeb crawler by adding some rules to crawl only relevant pages. In next stage they removed duplicate results. However they could not achieve a relative accuracy in the retrieval.

Mydhili K Nair Dr. V.Gopalakrishna [2] "A Survey of Web Service Discovery" stated different methods of service discovery. As today WSDLs, are abundant, scattered across the WWW so the count of Web Services already deployed with similar functionality is mammoth in number. There is an increasing need to evolve Service Discovery Methods that help the Consumer to find the right kind of services for his requirements.

Woogle [3] is a web service search engine. They have done extraction of information about WSDL functionality descriptions, inputs and outputs. They used clustering of parameters, matching of input output and operations, and stored the results in a database. They compared their method with Func and Comb. Comparison of words only with operation names is done by Func method. Whereas in Comb method web service names, parameters names and descriptions are also used for matching; in contrast to Woogle, both of the mentioned keywords are used.

Faisal Mustafa, T. L. McCluskey [4] made a sketch of major challenges tackled by automated web services composition. The problems are associated with distributed, dynamic and unsure disposition of web service. Their model is semi-automatic but fixes few problems of fully automated service composition. They pointed out that internet has huge repository of services it is not possible to automatically analyze them. And hence integration is difficult. They said that second difficulty for automated service composition is that web services are updated frequently. So there is a need to have current information and decision of composition should be based on that. Their technique has few drawbacks. If server is not working input output problems occur. Also their repository does not contain updates information.

Farhan Hassan Khan, M.Younus Javed, Saba Bashir "QoS Based Dynamic Web Services Composition & Execution" [5] presented a framework for dyamic web service composition and execution. At first they discussed major problems of dynamic composition. Then proposed an algorithm for dynamic web service composition. They mentioned composition issues like reliability, availability, data distribution. They introduced the concept of multiple repositories for system reliability. Avalibility is also guaranteed by this concept. An aging factor is used to retrieve uptodate information. They claim that their system is reliable, fault tolerant and performs fats data retrival.

[6] Jinghai Rao and Xiaomeng Su et al. "A Survey of Automated Web Services Composition Methods" presented different web service composition methods which include workflow and AI planning. They proposed a five step composition model. Five layers are Presentation, Translation, Process generation, Evaluation and Execution. The author also concludes that although different automatic web services composition techniques are available, it is not true that more automation is better.

[7] Biplav Srivastava, Jana Koehler et al. "Web Service composition: Current Solutions and Open Problems" explored problems of web service composition and analyzed two other approaches and compared them. The industrial approach and Semantic web approach, with each other. The industrial approach is primarily syntactical and is based on XML standards which are used for web services specification. This approach is used for several Businesses to Business and enterprise applications integration.

San-Yih Hwang, Ee-Peng Lim, Chien-Hsiang Lee, and Cheng-Hung Chen [8] formulated the dynamic WS selection procedure in a dynamic environment that is failure prone. They proposed FSM usage to invoke operations of service in an order. They define AR to find a probability that execution of services will terminate successfully. They used Eigen vector to show aggregated reliabilities. The approach can be used only in industrial applications and hence is environment specific.

Liping Liu , Anfeng Liu , Ya Gao [9] used Particle Swarm Optimization for Service Composition. PSO is meaningful for the composition of complex services spread on internet. If there is requirement of multiobjective composition only PSO can do so. A non inferior pareto solution is provided by PSO group search. The solution meets all the required constraints. They used general service overlay model. They say that full automation of service composition is complex rather unachievable task that's us why most of algorithms are semi automated. They said that their algorithm can be applied to specific compositions.

Zhang Hai-tao, Gu Qing-rui [10] presented a dynamic process of domain ontology-based method. Their method considered semantics of service for composition. They verified their work by experimental examples. Their module for service composition makes a portfolio as soon as a service request is received. When all services are determined for integration it starts calling webs services. An OWL-Agent is used to mark functions of the service by forming OWL-S documents that call the services. Shortcomings of this approach is it must have a fixed field of experts in the field service portfolio template firstly, and then selected the user needs to match the template in the service of specific Web services. The limitation of the method is the construction of the field of service composition templates requires human intervention, so degree of automation is reduced.

3. Proposed Framework

General Steps of the proposed framework are given below:-

1. User queries the system for required service
2. Services are searched via Google Custom Search API. Service is selected.
3. Information of selected service is retrieved.
4. User enters required input parameters for first service.
5. User adds the service to composition module. And selects next service.
6. On basis of information of service matchmaking is performed.
7. If parameters match.

i) Composition is performed.

ii) Results are displayed.

8. If parameters do not match. User is prompted to take action.
9. Composition is performed after user's action.
10. Results are displayed.

3.1. Pseudo code

The pseudo code of proposed technique is given as:

- **Algorithm**: Web Service Composition
- **Input:** Request for Web service
- **Output:** Composed Service Results
- User requests a desired service from Database or performs Crawling;
- If Crawling is requested
    - For each input
        - Input goes to Google Custom Search Engine through Google Custom Search API.
        - Engine produces results.
        - Results are parsed to human readable format.
        - Only wsdl link and related information are extracted from the results.
        - Results are displayed to user.

        - Validity check is performed
        - Valid results at present are displayed
- Results are stored in backup database.
    - If no result found for user query word
        - Message dialogue is displayed to enter synonym query word, Or to scale the engine to more links
- User selects a service from list;
- For each Selected Service

Fig. 1: Qos Based Service Search and Composition Algorithm

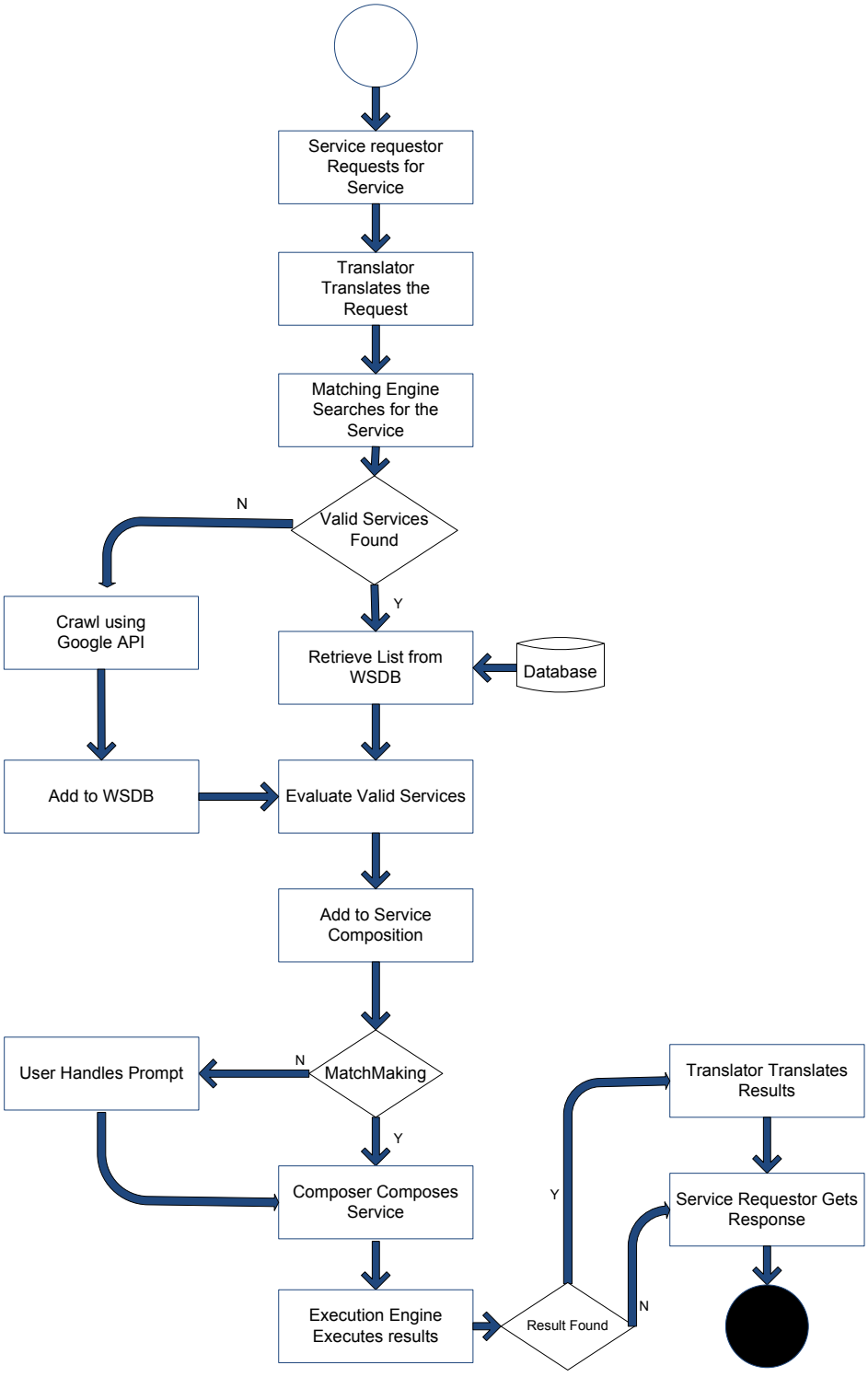
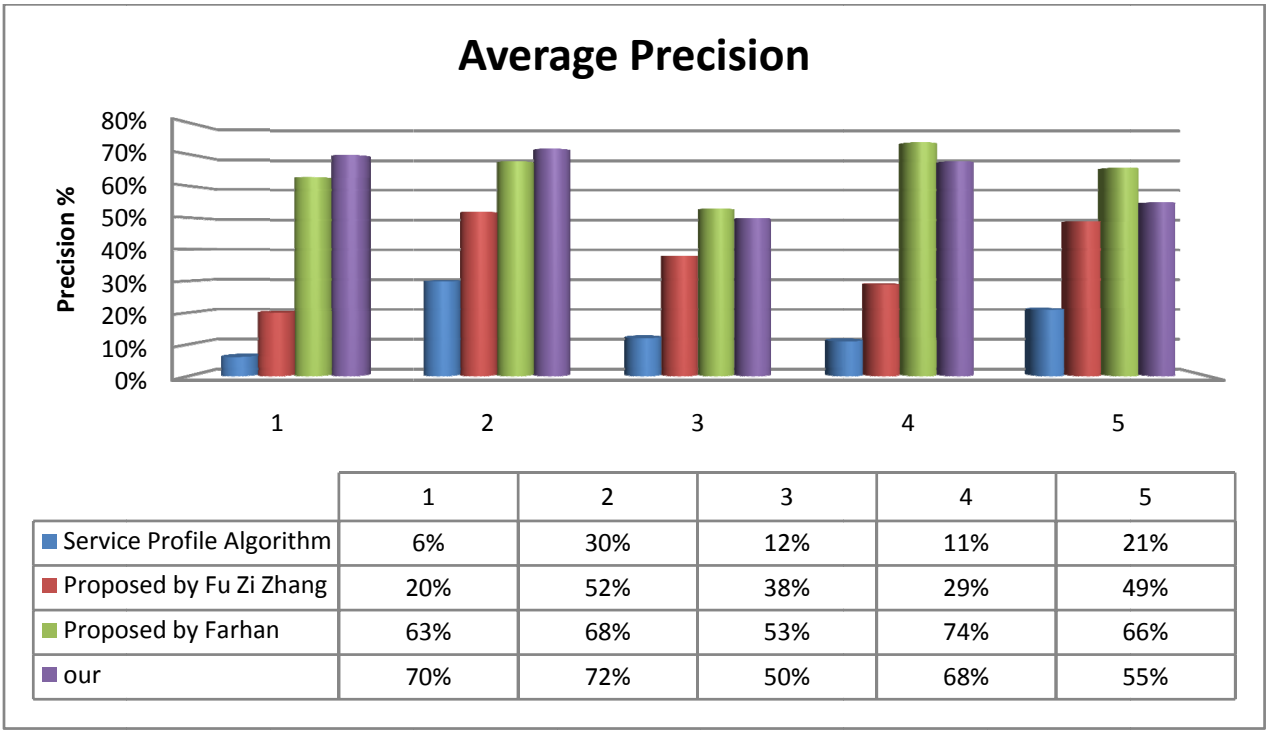


Fig. 2: Flow Chart for Qos Based Service Search and Composition Algorithm

4. Results and Discussion

We compare our framework with couple of other techniques. First technique "QoS Based Dynamic Web Services Composition & Execution", Farhan Hassan et al, "A Web service discovery algorithm based on dynamic composition" is proposed by Fu Zhi Zhang et al. and "A Software Framework for Matchmaking Based on Semantic Web Technology" is proposed by Lei Li and Ian Horricks. The analysis of the results clearly shows that our technique has improved the precision.

After service discovery we performed service composition by matrix matchmaking. In matrix match parameters of service are put on row and second service are put in column to perform matching. Matrix matchmaking reduces time spent on parameters matchmaking as well as takes less memory space. We will perform analysis on composition time.



	1	2	3	4	5
Service Profile Algorithm	6%	30%	12%	11%	21%
Proposed by Fu Zi Zhang	20%	52%	38%	29%	49%
Proposed by Farhan	63%	68%	53%	74%	66%
our	70%	72%	50%	68%	55%

Fig. 3: Average Precision Comparison

5. Conclusion

Our proposed framework covers the limitations of current techniques; it actively obtains user required web service by crawling among different repositories. The framework for service crawling using Google Custom Search API is flexible, scalable, efficient and reliable. In our approach the requester always gets up to date services the retrieval is fast and efficient. Also the client is able to add more repositories from where the services can be crawled. We made the system to be flexible in terms of automation hence we include user involvement at few steps for example selection of service and matchmaking decision. Matrix Matchmaking of input and output parameters guarantees compositional correctness and transactional support.

6. References

[1] Holger Lausen and Thomas Haselwanter, "Finding Web Services" 2007.

[2] Mydhili K Nair, Dr. V.Gopalakrishna, "Look Before You Leap: A Survey of Web Service Discovery" International Journal of Computer Applications (0975 – 8887) Volume 7– No.5, September 2010

[3] Xin Dong Alon Halevy Jayant Madhavan Ema Nemes Jun Zhang, "Similarity Search for Web Services" Proceedings of the 30th VLDB Conference,Toronto, Canada, 2004

[4] Faisal Mustafa, T. L. McCluskey "Dynamic Web Service Composition" 2009 International Conference on Computer Engineering and Technology

[5] Farhan Hassan Khan, M.Younus Javed, Saba Bashir, “QoS Based Dynamic Web Services Composition & Execution”, (IJCSIS) International Journal of Computer Science and Information Security,Vol. 7, No. 2, February 2010

[6] Jinghai Rao and Xiaomeng Su et al. “A Survey of Automated Web Services Composition Methods”

[7] Biplav Srivastava, Jana Koehler “Web Service Composition - Current Solutions and Open Problems”, ICAPS 2003

[8] San-Yih Hwang, Ee-Peng Lim, Chien-Hsiang Lee, and Cheng-Hung Chen ,“Dynamic Web Service Selection for Reliable Web Service Composition” Ieee Transactions On Services Computing, Vol. 1, No. 2, April-June 2008

[9] Liping Liu , Anfeng Liu , Ya Gao , “Improved Algorithm for Dynamic Web Services Composition”, The 9th International Conference for Young Computer Scientists

[10] Zhang Hai-tao, Gu Qing-rui, “A Dynamic Web Services Composition and Realization on the Base of Semantic”, 2010 IEEE