iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** V **Month of publication:** May 2018

DOI: http://doi.org/10.22214/ijraset.2018.5077

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue V, May 2018- Available at www.ijraset.com

A Hybrid Parameter Based Composite Web Service Discovery with Underlying Semantic Profile

Sreehara B [1], Lashma K [2]
[1] *M. Tech, Computer Science & Engineering, Sree Buddha College of Engineering, Kerala, India.*
[2] *Assistant Professor, Computer Science & Engineering, Sree Buddha College of Engineering, Kerala, India.*

Abstract: ***Web services are a network based interface to application functionality, built using standard internet technologies. It allows applications to communicate by using standardized protocols with low cost. With the development of SOA, web services have gained wide popularity. Web services are defined as self-contained and self-describing applications that can be published, located and invoked through the web. It is a modular, well-defined and encapsulated function. These are XML based components that can be executed by any application on the World Wide Web irrespective of platform. Web service provides a platform that allows interoperability between software applications running on different platforms and frameworks. Since many web services are available in internet, finding the most appropriate for the user request is difficult. Web service discovery is the process of finding suitable web services for a given task. The discovery process located a web service provider and web service descriptions are retrieved. This work proposes a web service discovery approach, which can mine the semantics of interaction inter face parameters and creates an index library based on the co-occurrence probability. A weather forecast web service discovery and recommendation is implemented, in which the registered users in the system can find the appropriate web services. So the proposed system for the web service discovery is more efficient and fast.***

Keywords: Semantics, ASN, Ontology, interaction interface parameters, Service description.

I. INTRODUCTION

Web services allow application to communicate using standardized protocols with low cost. With the development of SOA, web services have gained wide popularity. Web services are defined as self-contained and self-describing applications that can be published, located and invoked through the web. These are XML based components that can be executed by any application on the World Wide Web irrespective of platform. The primitive web services can be combined to handle complex requirements to form value added composite services. Web service provides a platform that allows interoperability between software applications running on different platforms and framework.

Web services are implemented using standards such as UDDI, SOAP, WSDL, etc. Web services are developed and published by different vendors using UDDI. It is the mechanism to register and discover web services. The details of a web service are provided in the WSDL document. It provides the format to describe the web service and how they are bound to a network address. Definitions, operations and service bindings related to web services are the components of WSDL. XML is used by WSDL to express definitions of a web service. Operations of a web service include four types such as one way message sent without a reply, simple request and reply, solicit response and sending notifications. Since many web services are available in internet, finding the most appropriate for the user request is difficult. Service discovery is a significant activity in the Services Computing paradigm. Service discovery can be informally described as the task of efficiently and accurately finding a relevant set of services that satisfies a user's given service request. Efficient discovery plays a crucial role in conducting further service composition. With the ever-increasing number of services available on the Internet, increased numbers of service consumers can participate in composition activity. Although existing service discovery techniques have produced promising results that are certainly useful, they may not be well aligned with the needs of Internet-scale environments.

The current existing web service discovery approaches can be classified into two categories: syntax-based service discovery methods and semantic annotation-based web service discovery methods. The first category, syntax-based service discovery process primarily depends on selecting appropriate keywords and making a query that matches the selected keywords with the web service descriptions. Because keywords are unable to capture the underlying semantics of the web services, they may miss some results and may return many irrelevant results. Second, many users would like to express their requests more precisely than is possible using keywords alone, the underlying reason that users want to search for Web services is that they need to find operations that offer a particular functionality. Current web service discovery documents usually provide details about these service operations. Hence, service consumers must browse these returned results individually to determine whether each service meets their requirements. Thus,

the greater the number of web services providers that emerge, the heavier the burden that consumers face in finding the services that they need.

The second categories, the semantic annotation-based web service discovery approaches, hold promise for automated service discovery and selection. A majority of the current approaches for web service discovery call for semantic web services that have semantic tagged descriptions through various approaches. However, these approaches have several limitations. First, it is impractical to expect all new services to have semantic tagged descriptions. Second, descriptions of the vast majority of already existing web services are specified using web services Description Language (WSDL) and do not have associated semantics. Also, from the service requester's perspective, the requester may not be aware of all the knowledge that constitutes the domain. Specifically, the service requester may not be aware of all the terms related to the service request. As a result of which many services relevant to the request may not be considered in the service discovery process.

The main objective is to devise a framework for discovery of web services and recommended the web services to the users based on QoS similarity and location similarity of the services. The main objectives of the system is:To provide more accurate web service discovery and recommendation to users.

To discover the more appropriate web services to users based on their request and underlying semantic concepts.

To develop a new framework for the web service discovery based on the underlying semantics of the interaction interface parameters.

To recommend similar web services on the basis of QoS similarity and location similarity.

A. Scope and Scheme

Service Discovery is an important aspect of the web service lifecycle. Web services need to be discovered and utilized according to functionality. As the growth of web services increases rapidly, a problem arises when identifying and selecting appropriate web services, because of the massive number of web services available over the Internet. Another obstacle that obstructs finding the proper web service is a lack of suitable search mechanisms, as most of the search tools are based on syntax rather than semantics. Additionally, existing search tools may fail to involve non-functional parameters such as Quality of Service and Cost of Service. The mechanism which drives the finding of web services is called service discovery. In SOA, it is a key component and important aspect which leads web services to utilize their functions. In SOA implementation, the primary factor is a higher degree of reusing roles in the form of readily implemented services, and the aim is to minimize development time and costs.

The SOA and web services definition has three primary roles are interacting with each other within SOA architecture. These three main roles are the Service Provider, Service Requester and Service Registry. The roles interact using publish, find and bind operations. The service providers are the business process that provides access to the web service and publishes the service description for consumption. The service description usually uses to bind with the information. The service requester also uses the Meta information in a description to attach and consume a service. The service registry is an optional logical concept where the service discovery method is to locate information about the service provider and obtain the service details. Service discovery provides the functionality to discover capabilities of services automatically. Usually, a service discovery system can help services to register their obtainability, locate a single instance of a particular service and also notify when an instance of a service changes. During the past decade, researchers, practitioners, and academic communities have been proposing different systems, methods, and approaches for the fast-evolving research area of web service discovery systems. This work focuses on SOA and the Service Discovery of different web services across hybrid environments. Both SOAP and REST web service technologies are considered and analysed. The analysis and survey of web service discovery open source standards will be brought into focus by examining several case studies and innovative solutions. The aim of the project is to enable service discovery in hybrid environments platforms, using standards where possible.

The objective of the proposal is to analyse the service discovery mechanisms for different web service standards by various platforms. There exist numerous different kinds of web service discovery protocols; they are responsible for connecting machine to machine to achieve the purpose of a web service from the service providers to the service requester. With the rising number of web services and also to fulfil the requirements of scalability, high availability and maintainability of services, service discovery techniques and pattern are also changing rapidly. There are several standards which are involved in service discovery, namely UDDI, WS-Discovery, and ebXML. Much more are also available for micro-service architectures.

In the proposed system retrieve the underlying semantic concepts from service metadata, and attempts to assist service consumers in finding not only similar service operations but also potentially compassable ones using the given requests with high precision/recall rate and low time costing. For all of above reasons, a much simpler, more efficient service discovery method is

required both to meet the needs of the user centric, demand-driven Internet environment and to lower the barriers to entry for web service consumers. The web service input and output parameters contain the underlying functional knowledge that is mined for improving service discovery. This fundamental knowledge conveys the semantic relationships and meaningful associations between the operation parameters. These semantic relationships provide the service domain context needed for service discovery. In this paper, an approach to an efficient and easy-to-use web service discovery approach that attempts to retrieve the underlying semantic concepts from existing service metadata is proposed with above syntax and semantic annotation-based service discovery approaches, this work tries to retrieve the underlying semantic concepts from service metadata, which attempts to assist service consumers in finding not only similar service operations but also potentially compassable ones using the given requests.

II. METHODOLOGY

A. *Existing System*

The emergence of Web Services (WSs) provides new opportunity for resource reuse and sharing. WSs have undergone rapid development for some time. Thus far, a large number of WSs have emerged, and the Internet has developed into a large-scale WS library. With the rapidly increasing volume of information contained in regular web pages, how to rapidly and accurately search for WSs on the Internet has become one of the problems that urgently needs to be addressed in the field of WS. With the ever-increasing number of services available on the Internet, increased numbers of service consumers can participate in composition activity. Although existing service discovery techniques have produced promising results that are certainly useful, they may not be well aligned with the needs of Internet-scale environments. The growing number of web services available within an organization and on the web raises a new and challenging search problem: locating desired web services. The current existing web service discovery approaches can be classified into two categories: syntax-based service discovery methods and semantic annotation-based web service discovery methods.

The first category, syntax-based service discovery process primarily depends on selecting appropriate keywords and making a query that matches the selected keywords with the web service descriptions. Because keywords are unable to capture the underlying semantics of the web services, they may miss some results—and may return many irrelevant results. Second, many users would like to express their requests more precisely than is possible using keywords alone, the underlying reason that users want to search for web services is that they need to find operations that offer a particular functionality. Current web service discovery documents usually provide details about these service operations.

he second category, the semantic annotation-based web service discovery approaches, hold promise for automated service discovery and selection. A majority of the current approaches for web service discovery call for semantic web services that have semantic tagged descriptions through various approaches. However, these approaches have several limitations. First, it is impractical to expect all new services to have semantic tagged descriptions. Second, descriptions of the vast majority of already existing web services are specified using Web Services Description Language (WSDL) and do not have associated semantics. Also, from the service requester's perspective, the requester may not be aware of all the knowledge that constitutes the domain. Specifically, the service requester may not be aware of all the terms related to the service request. As a result of which many services relevant to the request may not be considered in the service discovery process.

B. *Drawbacks of Existing System*

1) Keywords are unable to capture the underlying semantics of the web services.
2) They may miss some results and may return many irrelevant results.
3) In semantic annotation based method, all the services must be semantically tagged.

C. *Proposed System*

To address the challenges involved in searching for web services, this work tries to retrieve the underlying semantic concepts from service metadata, and attempts to assist service consumers in finding not only similar service operations but also potentially compassable ones using the given requests with high precision/recall rate and low time costing. For all of above reasons, a much simpler, more efficient service discovery method is required both to meet the needs of the user centric, demand-driven Internet environment and to lower the barriers to entry for web service consumers. The web service input and output parameters contain the underlying functional knowledge that is mined for improving service discovery. This fundamental knowledge conveys the semantic relationships and meaningful associations between the operation parameters. These semantic relationships provide the service domain context needed for service discovery. In this paper, an approach to an efficient and easy-to-use web service discovery approach that attempts to retrieve the underlying semantic concepts from existing service metadata is proposed with above syntax

and semantic annotation-based service discovery approaches, this work tries to retrieve the underlying semantic concepts from service metadata, which attempts to assist service consumers in finding not only similar service operations but also potentially compassable ones using the given requests.

In the proposed system, geographical location of web service is considered as a new parameter for composite service discovery algorithm. The parameter decides the proximity of service using geographical coordinates. It helps to utilise location aware service (minimum distance). It also increases bandwidth utilization. Second parameter is the quality of service is always a factor of web service discovery. The quality parameter is decided by the service users through a feedback mechanism. The feedback (QoS) value affects the overall ranking in service discovery algorithm.

In the existing system, one of the more efficient and accurate web service is discovered based on the user request and underlying semantics of the interaction interface parameters. Here web service discovery process has more importance. But in the enhancement part, web service recommendation has been employed.

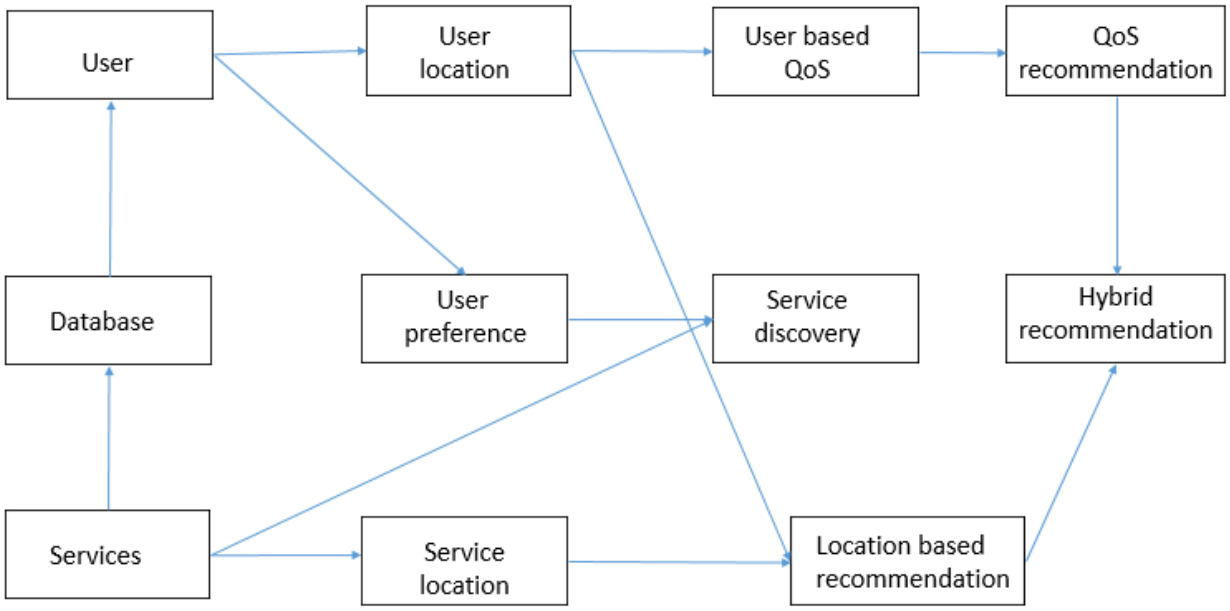


Fig 1: System Architecture

III.SYSTEM DESIGN

A. *Modules And Their Functionalities*

The project A Hybrid Parameter Based Composite Web service Discovery with Underlying Semantics Profile consists of the following modules.

1) Admin Module: Admin module consists of the following links. They are service creation, service request view, user management, service request processing, dataset creation, QoS similarity, location similarity and evaluation.
2) *Service Creation* : First the service provider companies register the system and they requested the admin for entering the system. Then the admin can create the id and password for the service provider and send it to the provider's page. Using this id and password provider can enter the system. Admin can creates the services in the service creation page. Here the details of the services like service name, provider company, IP address, latitude, longitude are also given.
3) *Service Request View:* Here the admin can view the details of the users who requested the web services.
4) *User Management*: Admin can manages all the users and the admin have the provision to delete the users.
5) Service Request Processing: Admin can also see the feedback or rating of the users. Here the admin viewed the userid, search place, parameter and the discovered web services and the corresponding user's rating values.
6) *Dataset Creation*: One of the major function of admin module is the dataset creation. The dataset contains place, parameter, web service name, search date and the output values. The web Service input and output parameters contain the underlying

functional knowledge that is mined for improving service discovery. This fundamental knowledge conveys the semantic relationships and meaningful associations between the operation parameters. These semantic relationships provide the service domain context needed for service discovery. In this paper, we propose an approach to an efficient and easy-to-use web service discovery approach that attempts to retrieve the underlying semantic concepts from existing service metadata. This work tries to retrieve the underlying semantic concepts from service metadata, and also create an index libraries to provide a high-efficiency interface that can match with "Single" operations and "Composite" operations for web service, which attempts to assist service consumers in finding not only similar service operations but also potentially composable ones using the given requests.

Propose a conceptual web service description model that considers basic metadata such as inputs, outputs, and operations within a message that resides in SOAP-based WSDL documents or REST-based WADL documents. The model accomplishes this goal by importing the type path (pPath) for parameters to comprehensively extract semantic information from the service interaction interface, enabling it to exploit a combination of service descriptions and service input and output parameters to accurately discover relevant web service operations that match the service requesters' functionality needs.

describe the web service more precisely, we extend the traditional description model by importing the type path of the interaction interface parameters. The operations and their parameters provide more comprehensive interaction details than the description itself. Therefore, the most effective information to express the interaction devolves to the names of operations and their parameters. By analysing the web services description files such as WSDL, we noticed that the type paths for interaction parameters also contain important semantic information that can help apply the semantics gleaned from the interaction interface parameters.

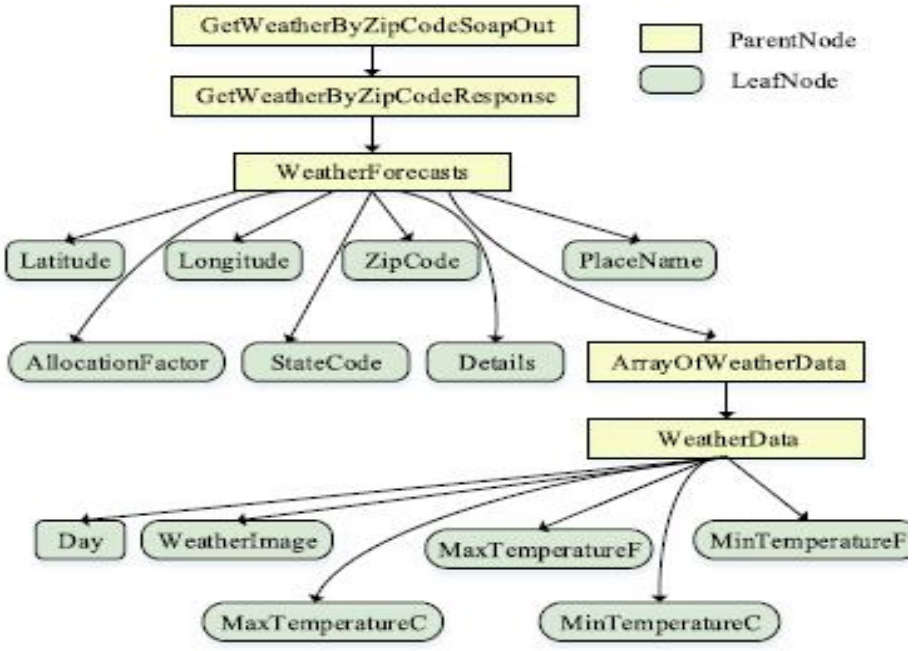


Fig 2: Weather forecast output interface parameters.

Take a weather forecast as an example [11]. One of the output messages from that web service is shown in Fig. 2, where the ellipse nodes express the parameter names, the rectangle nodes express the type names, and the type is the complex type. The single arrows express the traces of defining types and parameters. For example, the parameter name "Place Name" is an element parameter name for the class type "Weather Forecast" and "Weather Data" is an element type name for the class type "Array of Weather Data." Extracting only the parameter names, such as "PlaceName" hardly captures the fact that this service provides weather forecasts. Therefore, we add the type paths of interaction interface parameters to the conceptual models. First formalize the parameters type path. The parameter path pPath is a type-defining path defined as:

$$\text{pPath} = p_1; p_2; \ldots\ldots; p_n;$$

where p_i is the parent node of p_{i+1} , if i = m, and p_i is the name of the parameter in which case name p_m as "LeafNode;" otherwise, p_i is the type name of the parameter, so we name it "Parent Node." Here all of the "Parent Node" nodes contain effective information. Some "ParentNode" nodes contain information with high redundancy, such as "Array Of Weather Data" and

"WeatherData," and some "ParentNode" nodes contain useless information, such as the "of" in "ArrayOfWeatherData." Therefore, the Ppath differs from class diagrams in UML or an OWL-S Profile, and we apply it only to assist in mining the interface semantics.

7) *QoS Similarity:* Values of some QoS properties (e.g., response time) on the same web service vary quite differently from user to user. The response time of a service observed by closely located users usually fluctuates mildly around a certain value. On the other hand, the response time observed by users who are far away from each other sometimes varies significantly. Based on this finding, the recommendation algorithm takes location information into consideration to improve the recommendation accuracy. Here admin can find the similarity of the QoS parameters of different users
8) *Location Similarity :* Find the location information of a user which includes the AS number and the country name according to the user's IP address. Service location information is used to find the location information of web services that includes the AS number and the country in which the web services are located. Here geographical location of web service is considered as a new parameter for composite service discovery algorithm. The parameter decides the proximity of service using geographical coordinates. It helps to utilise location aware service (minimum distance). It also increases bandwidth utilization. Admin has the responsibility to find the location similarity of the web services and users.
9) *Evaluation:* Admin can evaluate the synthetic and real web services based on the users rating / feedback.

B. Service Provider Module

Service provider module consists of the following links. They are configure services, service parameter setting and availability updating. Service provider can configure the web services and upload the new web services.

1) *Configure Services:* Service provider configures the web services by using the provider id and password. While configuring the service provider can enter the type, availability date and service specifications.
2) *Parameter Setting:* Provider can set the service parameters like throughput, response time and fault tolerance, reliability, performance and security of the synthetic web services.
3) *Availability Updation:* Also update the availability date of the selected web services. Here the service provider can upload their own services by using the provider id and password.

C. User Module

User module consists of the following links. They are profile updating, find weather, parameter analysis, and show my cluster, QoS recommendation, location recommendation and hybrid recommendation.

1) *Profile Updation:* Registered users can update their profile by adding their mobile number and zip code.
2) *Find Weather:* It can be classified as two types. They are:*:* In the case of real dataset, select the count of the parameters and the corresponding web services. Then store this values in a matrix format. Compare the values in the matrix and select the web service which has the maximum values. Here user can rate web services. In the case synthetic dataset, if a user requested the synthetic web service first check the availability of the web service (availability from date and availability to date). Then select the count values of the place, parameter and the corresponding web services. Then creates the synthetic web service that access the service which has the highest count value. Here user can also rate the services. Admin can analyse the ratings of the web service in both real and synthetic datasets. Here the web service discovery process is done.
3) *Parameter Analysis:* User can analyse the service parameters on the basis of the requested date. Analysis details are shown in a graphical format. Users can also select the required parameter, then the weather details of the corresponding parameters are displayed in a graphical way.
4) *Show My Cluster:* User can view all other users who requested the web service with the same place and parameters.
5) *QoS Recommendation:* Finds similar web services for a target web service by considering both the user's QoS values and their corresponding locations. Similarity computation is done with cosine similarity equation. QoS predictions can be generated from both service regions and user regions. With the compressed QoS data, searching neighbors and making web service QoS predictions for an active user can be computed faster than conventional methods.
6) *Location Recommendation:* User location information used to find the location information of a user which includes the AS number and the country name according to the user's IP address. Service location information is used to find the location information of web services that includes the AS number and the country in which the web services are located. It finds similar service users who are similar to the active user by considering both the user's QoS values and their corresponding locations. Here web services are recommended on the basis of the user's location and the service's location.

7) *Hybrid Recommendation:* Here the QoS similarity values and location similarity values are combined. The final recommendation is obtained by combining the user based QoS prediction and the location -based recommendation.

D. Enhancement

In the existing system, one of the more efficient and accurate web service is discovered based on the user request and underlying semantics of the interaction interface parameters. Here web service discovery process has more importance. But in the enhancement part, web service recommendation has been employed. As the number of web services with similar functionality has increased rapidly over the internet the web service discovery is not a challenging task but selection and recommendation are becoming more important. Optimality of a web service depends on its performance and it is measured through Quality of Service i.e. QoS. In Services Computing, QoS [13] is the set of non-functional properties of a web service which includes response time, availability, price, failure rate etc. which are the factors for service users to distinguish web services with similar functionality.

Existing QoS prediction methods rarely finds the similarity of users, services and location into consideration [14]. There are many sites like Programmable Web, Yahoo! Pipes, Google Developers etc. that are used by developers for recommendation on web services but none of them will give a location-based QoS information for the service user. System initially searches for the list of web services those having similar functionality which the user requested and finally the optimal web services are recommended to users.

QoS prediction methods assume that the co invoked web services have equal contribution weights when computing similarity between two users [13]. Argument is that the personalized characteristics (e.g., QoS variation) of both web services and users should be incorporated into measuring the similarity among users and services. Web service QoS factors, such as response time, availability and reliability, are usually user-dependent. From different web services, we can derive different personalized characteristics, based on their QoS values, as perceived by a variety of users. Some web services may have a very good QoS for all users. For example, the availability is always 100 bandwidth. These web services are also likely to have small variation of QoS values over different users. Many other web services may have a relatively large variation of QoS over different users. For example, the availability varies from 50.

The proposed method uses both the location of users and web services on selecting similar neighbours for the target user or service. QoS properties are highly dependent on the user feedback of the services. The location information of both service users and web services are acquired and processed which provide an effective location-aware web service recommendation. The location of the user can be defined based on the IP address of the service user, ASN and Country denotes the Autonomous System (AS) and ID of the country that particular IP belongs. The location information of both service users and web services are easily found since the IP address is known and the entire location information can be found by identifying both the AS and the country in which the particular IP address is located. The processed location information produces a set of similar users and similar services that are closed to each other. The network performance is likely to be poor when the service user and the service are located at different networks. The low performances are mainly due to the transfer delay of the network. Proposed methods make us of location information. As the IP address of the user is known it is much easier to locate AS number and the country where he is located. Similarly the locations of web services are also found. The location information is then processed to produce a set of similar users and similar services that are close to each other and thereby improving the recommendation accuracy.

Our recommendation algorithm is designed as a three-phase process,

1) QoS Recommendation
2) Location based Recommendation
3) Hybrid Recommendation

The first two phases aggregate users and web services into a certain number of clusters based on their respective similarities. QoS Recommendation recommended the web services based the feedback rate of the different quality parameters of the web services. In the case of location based recommendation find the location (ASN) of both users and web services and recommended the services for the user which have the same AS number as the user. In hybrid recommendation combines the results of both QoS and location recommendation. The hybrid recommendations are obtained by combining the user based QoS prediction and the location based recommendation.

The recommendation system [12] can be explained when an active service user searches for high-quality web services in a web service discovery system or the system is recommending high-quality web services to an active user. The process is done by the hybrid recommendation on the basis of QoS similarity and location similarity. The system consists of the following information. They are:

4) User location information: This is used to find the location information of a user which includes the AS number and the country name according to the user's IP address. When a user registering the system their ASN number and country name are entered.
5) Service location information: This is used to find the location information of web services that includes the AS number and the country in which the web services are located.
6) Find similar users: This is used to finds similar service users who are similar to the active user by considering both the user's QoS values and their corresponding locations. Similarity computation is done with the cosine similarity function.
7) User-based QoS Recommendation: After finding similar users from the above step, based on the QoS values they are entered through the feedback mechanism.
8) Location based Recommendation: After finding the location similarity of the users and web services, this system recommended the web services having the ASN which similar to the requested user.
9) Hybrid Recommendation: The final recommendation of web services are obtained by combining the user based QoS recommendation and the location based service recommendation.
10) *Similarity Computation with Cosine Similarity:* Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The cosine of two vectors can be derived by using the Euclidean dot product formula as shown in equation.

$$\text{Similarity} = \cos(\omega) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \; B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

11) *Integrating location of Users and services:* Location information of both service users and web services can be found easily. As the IP address of the user is known it is much easier to locate AS number and the country where he is located. Similarly the locations of web services are also found. The location information is then processed to produce a set of similar users and similar services that are close to each other. The network performance is likely to be poor when the service user and the service are located at different networks. The low performances are mainly due to the transfer delay and the limited bandwidth of the networks.

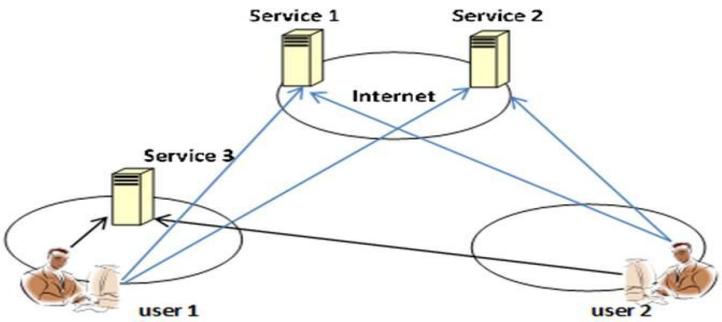


Fig 3: Influence of user location

Performances are found to be high if the user and the service are located in the same network. In fig 4.2, user 1 and 2 are located at two different geographical regions that are far from each other. Both user 1 and user 2 will experience a similar QoS values on accessing service 1 and 2 and they are similar. Service 3 is found to be in the same network of user 1 and therefore it is closer to him and the service is far away from user 2 and the QoS both users will obtain a different QoS values with respect to service 3.

IV. RESULTS AND ANALYSIS

The experimental results of the proposed system a hybrid parameter based composite web service discovery with underlying semantics profile is discussed in this section. In the system here using the operating system of version windows 8 and platform using is c# .net. And the database created is SQL server. Proposed system using synthetic data and real data for result evaluation. Synthetic data are data which are created. Synthetic data are created for obtaining specific requirements or certain criteria. Synthetic data are very useful for designing system of any type because this data can be used as a simulation. Real time data is some of the weather web services. The proposed system works both synthetic dataset and real dataset.

In the proposed system implemented by using many modules and sub modules. The input of the system is a weather request which contains a place and weather parameter. In the case of real dataset, select the count of the parameters and the corresponding web services. Then store this values in a matrix format. Compare the values in the matrix and select the web service which has the maximum values. Here user can rate web services. In the case synthetic dataset, if a user requested the synthetic web service first check the availability of the web service (availability from date and availability to date). Then select the count values of the place, parameter and the corresponding web services. Then creates the synthetic web service that access the service which has the highest count value. Here user can also rate the services. Admin can analyse the ratings of the web service in both real and synthetic datasets. Here the web service discovery process is done. Then the quality parameters and location parameters are used for the weather recommendation. Finally combines the results of both recommendation and generate a hybrid recommendation. The proposed system efficiently discovered a web service and recommend some similar web services for the user based on the users request and rating values.

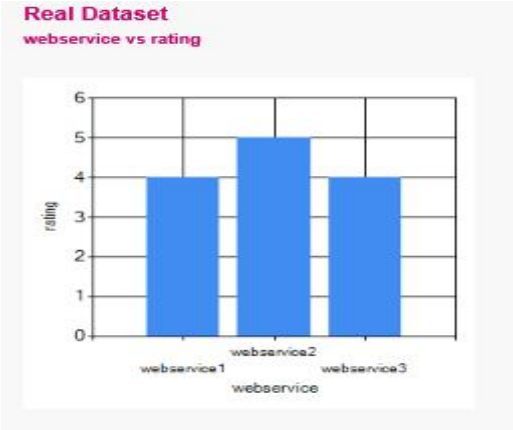


Fig 4: Analysis of the real web services based on the user rating

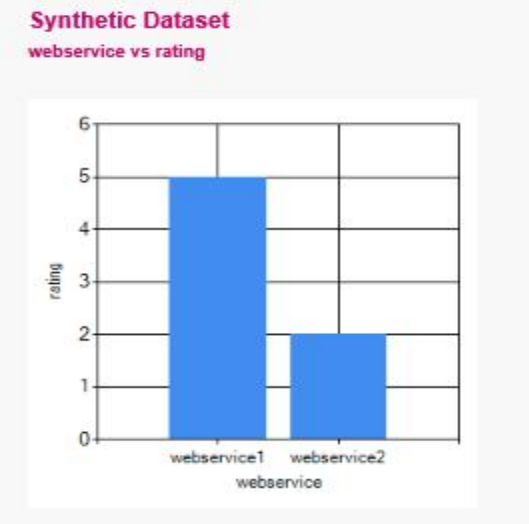


Fig 5: Analysis of the synthetic web services based on the user rating.

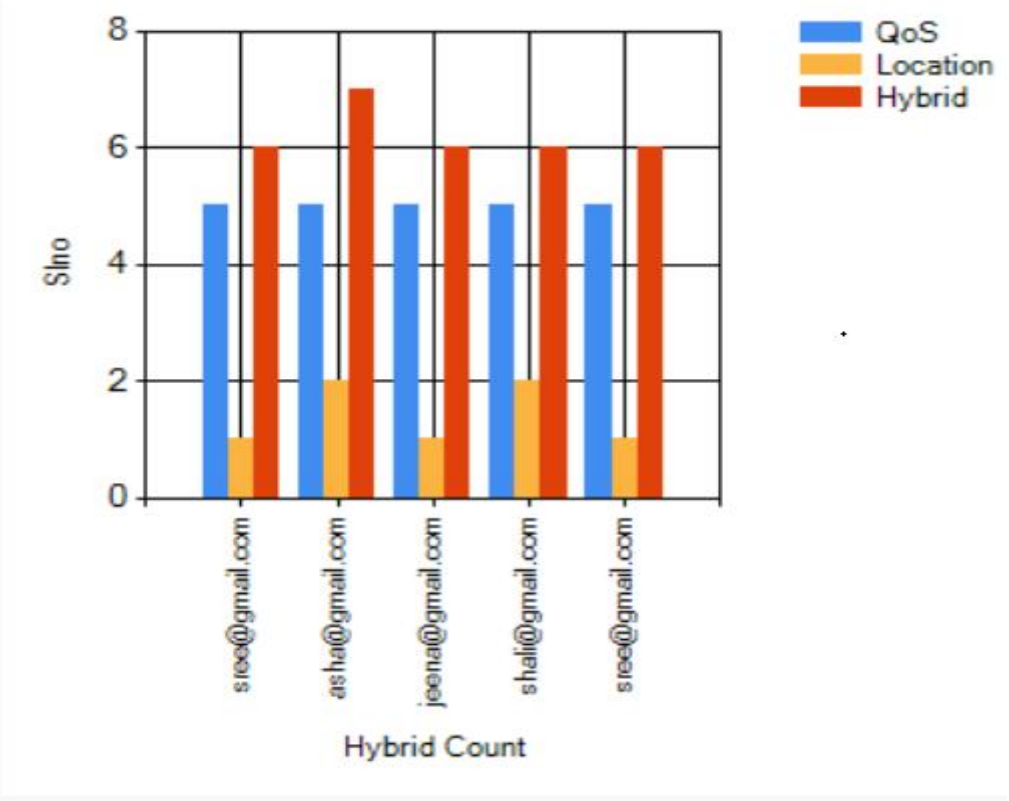


Fig 6: Analysis of QoS, Location and hybrid recommendations count

V. CONCLUSIONS AND FUTURE WORK

The proposed system is a web service discovery and recommendation system based on user preferences and rating. The main goal of this system is to provide accurate recommendation based on user's interest. The system discovered the accurate web services from both real and synthetic dataset on the basis of user requested parameter and place. Here the user can also rate the services. The proposed method uses both the location of users and web services on selecting similar neighbours and recommend the web services. Location information of both service users and web services can be found easily. As the AS number of the user is known it is much easier to locate the country where he is located. Similarly the locations of web services are also found. In the case of QoS recommendation find the cosine similarity of the user ratings. Integrating both the methods for QoS recommendation along with the location information will provide a better recommendation than the individual recommendations. The system can be improved by integrating different non-functional properties into consideration which helps in providing a better results.

This also opens new possibilities for future work, including: improve the accuracy of recommendation. Also improves the precision and recall rate.

REFERENCES

[1] G. Cassar, P. Barnaghi, and K. P. Moessner (2014) Probabilistic matchmaking methods for automated service discovery, IEEE Trans. Serv. Comput., vol. 7, no. 4, pp. 654–666.

[2] Y. Wu, C. G. Yan, L. Liu, Z. J. Ding, and C. J. Jiang, (2015) An adaptive multilevel indexing method for disaster service discovery, IEEE Trans. Comput., vol. 64, no. 9, pp. 2447–2459.

[3] Z. Zhou, M Sellami., W. Gaaloul, M. Barhamgi, and B. Defude, (2013) Data providing services clustering and management for facilitating service discovery and replacement, IEEE Trans. Autom. Sci. Eng., vol. 10, no. 4, pp. 1131–1146.

[4] S. Sellami and O. Boucelma, (2011) Web services discovery and composition: A schema matching approach, IEEE Int. Conf. Web Services, pp. 706–707.

[5] J. Zhang, J. Li, S. Wang, and J. Bian (2014) A neural network based schema matching method for web service matching, in Proc. IEEE Int. Conf. Services Comput., pp. 448–455.

[6] J. I. Fernandez-Villamor, C. A. Iglesias, and M. Garijo (2014) A framework for goal-oriented discovery of resources in the RESTful architecture, IEEE Trans. Syst., Man, Cybern., vol. 44, no. 6, pp. 796–803.

[7] T. A. Farrag, A. I. Saleh, and H. A. Ali (2013) Toward SWSs discovery: Mapping from WSDL to OWL-S based on ontology search and standardization engine, IEEE Trans. Knowl. Data Eng., vol. 25, no. 5, pp. 1135–1147.

[8] I. Paik, W. Chen, and M. N. Huhns (2014) A scalable architecture for automatic service composition, IEEE Trans. Serv. Comput., vol. 7, no. 1, pp. 82–95.

[9] O. Hatzi, D. Vrakas, M. Nikolaidou, N. Bassiliades, D. Anagnostopoulos, and L. Vlahavas (2012) An integrated approach to automated Semantic Web service composition through planning, IEEE Trans. Serv. Comput., vol. 5, no. 3, pp. 319–332.

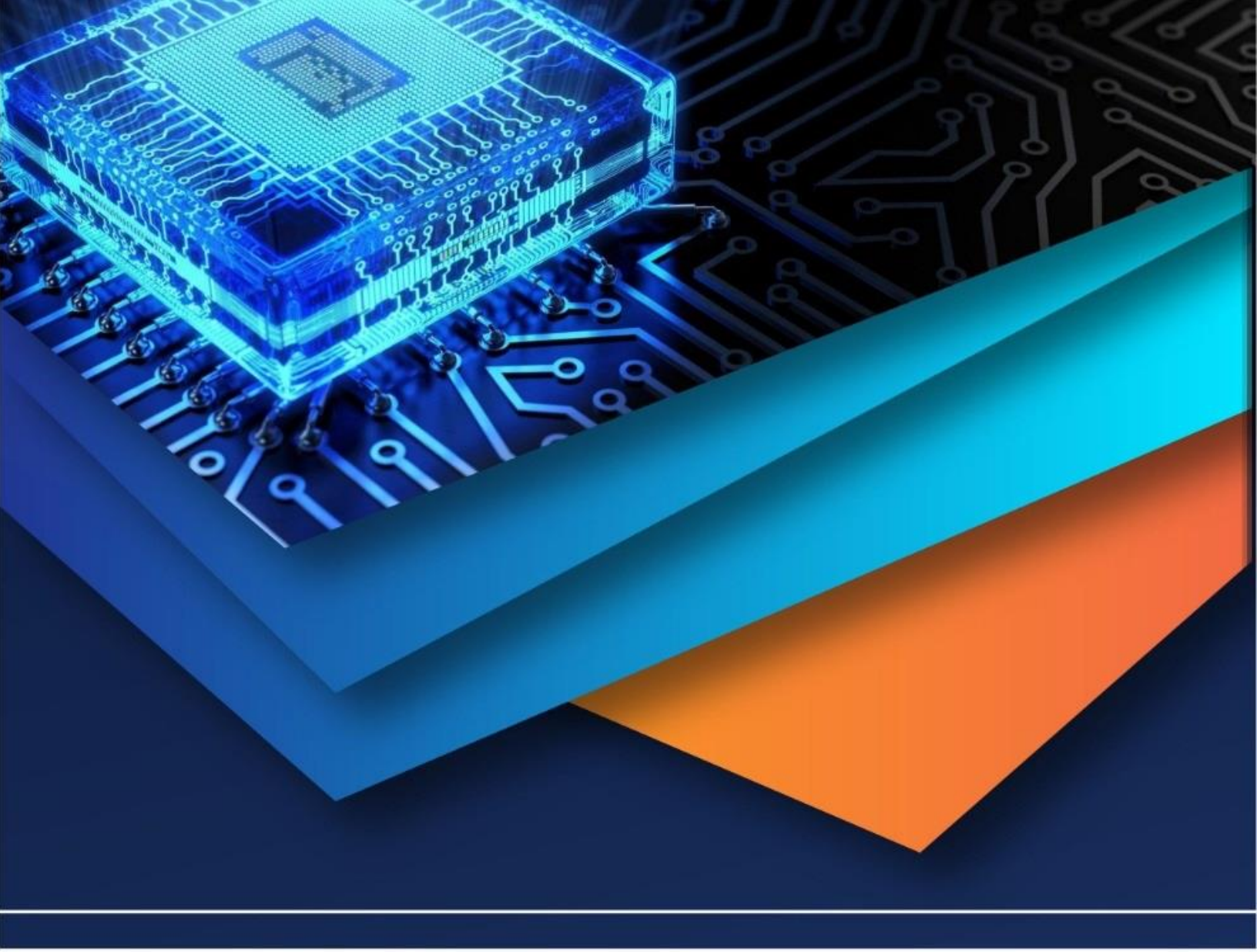
[10] D. Roman, H. Lausen, and U. Keller (2005) Web service modeling ontology - standard (WSMO - standard), WSMO deliverable D2 version 1.1, [Online]. Available: http://www.w3.org/ Submission/WSMO.

[11] Bo Cheng, Shuai, Changbao and j.Chen (2017) A web service discovery approach based on mining underlying interface semantics, IEEE Trans. Knowl. Data Eng., vol.29, no. 5, pp. 950-962.

[12] G. Adomavicius, and A. Tuzhilin (2005) Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowledge and Data Engineering, pp.734 –749.

[13] M. Alrifai, and T. Rise (2009) Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition, in Proc. of the International World Wide Web Conference, pp. 881-890.

[14] S.S.Yau and Y. Yin (2011) QoS-based service ranking and selection for service based systems, in Proc. of the International conference on Services Computing, Washington DC, USA, pp. 56 - 63.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY