Efficient Microservices Discovery and Selection Based on QoS Ontology a Data Mining Approach

Neha Singhal, Usha Sakthivel, Pethuru Raj

Abstract— Microservices is emerging as a revolutionary and modern solution in the recent software era. Microservices is also based on web services but it could be implemented as an independent feature to perform a specific task with its own database. In the growing field of e-commerce, when user aim to find microservice meeting to their dynamic business requirement is becoming a big challenge and emerging as a problem. This problem occurs, because the number of related service is increasing day by day in the repository and selection of the appropriate and quality service is a main challenge. In order to solve this problem, in this paper we propose a QoS ontology semantic annotation approach with reducing the difficulty of appropriate service discovery and selection as per user dynamic requirement using association rules and K-means clustering techniques.

Key words: QoS, ontology, Microservices, k-means, Association rules.

1. INTRODUCTION

Now a days Microservices are accepted as a better way to build application .In the few years ago the way of developing the applications was monolithic: - means that a single function can provide the solution for the application. It was a good solution for the small and simple application but as we are switching towards the complex business paradigm there is a need for small, modular chunks, and independent Microservice Architecture [1, 2]. For the complex and dynamic business application the requirements will change very frequently and making all those changes in monolithic application is problematic so the development is preferred using microservices instead of monolithic services. Microservices are fine grained modular and independent services with its own micro database [3, 4]. A microservices architecture takes loosely coupled services approach which can be developed, deployed, and maintained independently. Each of these services is responsible for individual task and can communicate with other services through simple APIs to solve a complex business problem. MSA (microservice architecture) is an architectural style that structures an application as a collection of small Independent services, modeled around a business needs [6]. Micro service architecture [4] provides various features compare to monolithic architecture. Features of Microservices are listed as follows:-Decoupling – Services within a business application are loosely coupled and independent in nature. So the application can be built, flexible and scaled easily. Componentization – Microservices are treated as independent and modular components. Autonomy – each microservice is atomic and singular in nature and can be developed independently of each other, thus development can be speedup. Continuous Delivery – Allows frequent releases of software with independent modules. Responsibility – Microservices are responsible for a specific task and chances of failure are limited. Agility – Microservices support agile development. Any new feature can be quickly added and removed. Fault Isolation – Even if one service of the application fails, the system will still work properly. Platform independent– Different languages and technologies can be used to build different services for the development [3,4].

The MSA organizes the microservice architecture in the following way [4]: -The service requester will request for the service by defining the query. The particular query i.e. the combination of various services will be searched in service repository. In microservice architecture the single service cannot provide the solution for the typical business needs. So the service providers publish similar microservice to the service registry and the best services can be provided by the service provider to the service requester. Microservice architecture addresses the issue of dynamic changes in the today's business paradigm. The concept is represented as figure 1.

Revised Version Manuscript Received on 30 May, 2018.

Neha Singhal, Department of Information Science and Engineering Rajarajeswari College of Engineering, Bangalore, India

Usha Sakthivel, Department of Computer science and Engineering, Rajarajeswari College of Engineering Bangalore, India

Pethuru Raj, Reliance Jio Infocomm. Ltd (RJIL),SARGOD imperial, 23, Residency Road Bangalore, India

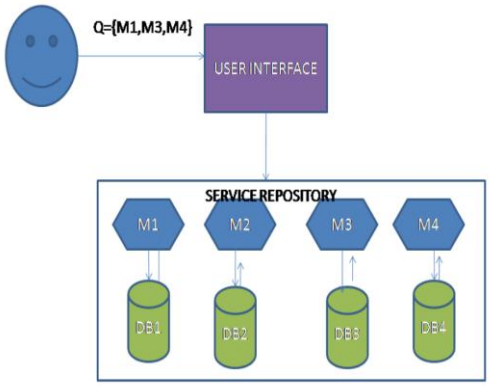


Figure1: Microservice Architecture

As now a days web is considered as a main source of information availability. In general it's a positive aspect for the information users but it has negative side too.

As numbers of services are increasing, for a single problem a huge number of solutions are possible. If a user wants to search any particular services he will get n number of services which can provide the solution to the same problem. In this case it's a big challenge which one to select out of number of similar microservices, In these scenario QoS parameters pays a major role in order to satisfy the user requirements. Semantic web services have been proposed to automate MS- discovery. Sematic microservice languages relay on ontologies. Ontology is a language in which content can be defined as tags. In this paper we are using the following two approaches for the related service discovery and selection.

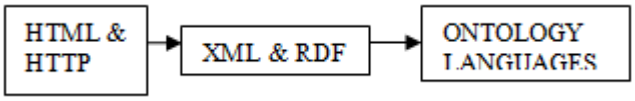
Association analysis:-A implies B, which means that if anyone interested in service a then they can also be interested in the associated service B also. Association rules can be helpful for the discovery of associated services and to provide the solution in the quickly manner.

Cluster analysis: - cluster analysis is a technique which helps to frame the group of services based on the measures of similarity. Clustering means group of similar services based on some common criteria of similarity.

A. Conceptual Foundations

The Semantic Web: - The semantic web is an emerging and upcoming version of the web technology. Web is a rich source of information that contains various reusable contents and services. On the web most of the services are solely designed only for human understanding initially. Web semantic development is a concept well underway with the goal that machines should be able to understand the contents on the web rather than just only to display for the human interpretation. There is a need for well-structured way of representation so the machines can understand the contents and process the contents for the future reuse. The purpose of semantic web is to provide the various capabilities to the computer to understand a) structured representation of Information b) Understand the Meaning of this Information c) implement the sets of inference logic. The inference logic provides the direction for the automation [7].

EVOLUTION OF SEMANTIC WEB:-
1) HTML AND HTTP
2) XML AND RDF
3) ONTOLOGY LANGUAGES



The Web Ontology: -Ontology is a technology that can be used to develop the semantic web concepts. Web ontology is a semantic web language that is used to describe the complex things on the web. Ontology can define the rich concepts and knowledge about the information interpretation. Web ontology is a way to create inference logic for the web things. OWL is part of the W3C's Semantic Web technology stack, which includes RDF, RDFS, SPARQL, etc.

Protégé:-In the web ontology field the protégé is the upcoming API for OWL and RDF. Protégé is the open source java library. Protégé API provides classes and methods for loading and saving the OWL files. With the protégé library the manipulation, query processing and data reasoning is efficient and easy way to go.

The rest of the paper is organized as follows:-section 2 discuss about the approaches in QoS aware service composition. Section 3 focus on the proposed work. Section 4 deals the case scenario. Section 5 describes about the implementation and result analysis of the proposed work. Section 6 concludes the paper and extends for the future work.

II. APPROACHES IN QOS-AWARE SERVICE COMPOSITION

QoS (quality of service) parameters cover a huge range of techniques on which basis selection of appropriate service can be done effectively. QoS Parameters cover various nonfunctional aspects for the service selection [7, 11].

As in today's world the most expensive thing is good quality of information because information retrieval is playing huge role for the e-commerce success. In this aspect the web is the biggest source of information and referred as the information world. So when the user wanted to search some particular service on the web the biggest problem and challenge is to provide the user with the most effective and appropriate service. This problem is growing day by day as for the same user demanded service there is lots of services is available on the web repository and when user search for some service he will be provided with the number of various available options. It's the big ambiguity for the user to select the effective one for his requirements. In this scenario the QoS parameters plays the important role to make the search more effective and efficient [12].

QoS parameters can be considered as follows:-

Response time: - Response time is a factor of microservice which tells us how fast the service can be respond.

Response time= total time taken –Requested time

Availability: - Availability is the probable factor of service performance is up when it's invoked and requested. Availability is a major criterion for the user satisfaction level.

Availability=1-(failure time/ requested time)

Throughput: - Throughput represents the number of success in per unit time for a requested web service. Throughput represents in the per unit time how many requests is treated successfully.

Throughput= no of success Request/ unit time

Success-ability: - success ability is the ratio of number of successful response out of total response.

Success-ability=no of success response/no of total requested in unit time

Reliability: - Reliability corresponds to the ability of a microservice to execute its function under the given constraints.

A. Classification of QoS Composition Approaches:-

QoS Based service discovery approach is classified in to three categories Non heuristic algorithm, heuristic algorithm, and meta- heuristic algorithm [11].

Table 1:- Classification of QoS Approaches

CLASSIFICATION OF QoS APPROACHES		
QoS Algorithm	**CLASSIFICATION TYPE**	**QoS Parameters considered**
Dynamic Approach [13]	Non heuristic	Response time, throughput
Greedy Approach[16]	Non heuristic	Cost, Availability, Reliability, Reputation, Response time
Divide and Conquer [14]	Non heuristic	Cost, Response time ,Reliability, Availability
Knapsack problem[15]	Non heuristic	Cost, Availability, Reputation, Response time
Hill-climbing[18]	Heuristic	Availability, Response time, Throughput , Security
A* Algorithm[17]	Heuristic	Availability, Cost, Response time , Reliability
Win-Win strategy [23]	Heuristic	Availability, Response time , Reliability
Hill climbing[22]	Heuristic	Availability, Cost, Response time , Reliability
Genetic algorithm [19]	Meta-heuristic	Availability, Cost, Response time , Reliability
Ant colony optimization [21]	Meta –heuristic	Availability, Cost, Response time , Reliability

The classification of QoS is represented in the table 1.

III. PROPOSED WORK

In this section, we introduce the proposed architecture and approach which can facilitate service discovery and automatic annotation of microservices. The research architecture is proposed for effective microservices are shown in figure 2.

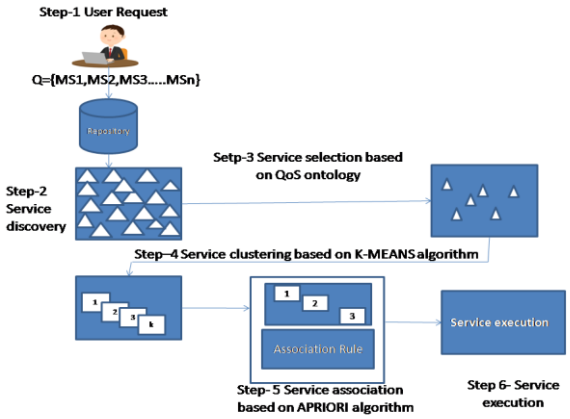


Figure: 2 System Architecture of the Proposed Work

Figure: 2 is the system architecture of the proposed work. The system process consists of following six steps:-

Step 1:-User Request for the Service:-

Web is emerging as biggest source of information. User can get almost all of his dynamic requirement solution on the web .users every small requirement can be considered as the combination of various microservices. For the solution of user simple query there is a need for composing several related microservices together.

If MS1,MS2.....MSn is microservices.

Q={MS1,MS2,MS3.....MSn}

Step 2:- Service discovery in Service Repository:-

Service Repository is the repository of the web software modules. Service contains the various microservices on the repository. User requested services will be searched in the repository and various similar services will be extracted from the repository based on the user specific requirements.

Step 3:-Service selection based on QoS ontology:-

Service repository contains various similar services which provide the same functionality so now to select the appropriate service as per the user requirement is a challenge. In this scenario QoS (quality of service) parameters plays a mark able role for the selection of user requested microservices. Ontology is a semantic web language that is used to describe the microservices with QoS parameters. In our research work we have used progete web ontology API with OWL for the discovery for web service based on the user requirements.

Step 4:-Service clustering based on K-MEANS algorithm

Clustering is a data mining technique which is used to make the group of similar services. K-MEANS algorithm basically works on the concept of centroids. for the microservices clustering K-MEANS work as follows:-

1. Select K- microservices as the initial centre points.
2. Repeat
3. Form K-microservices by assigning each microservice to the closest centric service.
4. Recompute the centred microservice of each cluster.
5. untill centred microservice do not change.

Step 5:-Service association based on APRIORI algorithm:-

Association analysis is also a data mining concept which can be used to find the association between microservices to satisfy the user complex requirements. Association rule is that can be used to find association among services in repository .due to association rules we are able to find the relationship between the related microservices. Apriori algorithm is used to find the relationship. Association analysis can be useful to find out the sequence of services in order to execute user query. For any two services A and B (A is not equal to B), Then support and confidence count can be used to find the association.

Support count = (Number Of services containing both A and B) / total num of services

Confidence count = (Number Of services containing both A and B) / num of services containing A

Support count is the percentage of satisfaction.

Confidence count is a measure of certainty

Step 6: Service execution

In the final step after extracting the clusters of similar services and the sequence of associated services the service composition is executed in order to fulfil user's requirement. Service composition is nothing but executing various services together to provide the solution for the users complex queries.

IV. CASE SCENARIO

In this section, a health care application scenario will be described in detail as per the proposed methodology. In this healthcare scenario various microservices is considered:-

M= {M1, M2, M3.....Mn}

M1= Patient Appointment

M2=Patient Registration

M3=Doctor Assignment

M4= Claim Insurance

M5=Billing

M6=Feedback

We are assumed that the above six microservices used in health care domain.

For Patient Appointments booking various microservices are available in the repository named Book Appointment, Cancel Appointment, Get Appointment and Search Appointment. Similarly there are various services related to Patient Registration, Doctor Assignment , Claim Insurance, Billing, and Feedback as given below:-

Patient Appointment= {Book Appointment, Cancel Appointment, Get Appointment, Search Appointment}

Patient Registration= {New Registration, Update Registration, Renew Registration}

Doctor = {Physician, Cardiologist, ENT, Neurologist}

Claim Insurance= {Cashless, Claim letter}

Billing= {Cash, Cashless}

Feedback = {Patient Feedback}

If the patient is requesting for an appointment for the particular doctor ex-cardiologist.the patient request will be send to the repository of services .there are various services available as per the user request. Based on PROTEGE ontology and QoS parameters the patient will be provided with above mentioned microservices which are related to the user requirement. In the further steps the concept of k-means clustering techniques is used i.e. formation of clusters based on measures of similarity described in the above given algorithm.

C1= {Book Appointment, Cancel Appointment, Get Appointment, Search Appointment}

C2= {New Registration, Update Registration, Renew Registration}

C3= {Physician, Cardiologist, ENT, Neurologist}

C4= {Cashless, Claim letter}

C5= {Cash, Cashless}

C6 = {Patient Feedback}

On the above given clusters data set k-means algorithm is applied and based on the similarity index as per user requirement suitable microservice will be provided to the patient.

The patient requirement cannot be fulfilled by the single or standalone microservice.

There is a need to merge various microservices in respect to fulfil user requirement. For dynamic user requirement there is a need for the concept of association analysis to find the relationship between the microservices.

We are considering the example of market basket transaction set for health care domain given below:-

Table2:-Market Basket Data for Health Care Domain

TID	Items
1	{Appointment, Patient Registration}
2	{Appointment, Doctor, Claim Insurance, Billing}
3	{Patient Registration, Doctor, Claim Insurance, Feedback}
4	{Appointment ,Patient Registration, Doctor, Insurance}
5	{Appointment, Patient Registration, Doctor, Feedback}

Association rules suggest that there is a strong relationship exists.

ex:-{Appointment}->{Doctor}

The rule suggests that the patient who will book the appointment he will meet with doctor as well.

Item set and support count:-let I={i1,i2,i3....in} is the set of all the services service set and T={t1,t2,t3.....tn} be the set of all the transductions.

Support count = (Number Of services containing both A and B) / total num of services

Confidence count = (Number Of services containing both A and B) / num of services containing A

Support is an important measure because a rule with the very minimum support count may occur simply and hence it can be ignored. Confidence count on the other hand measures the reliability of inference made by that particular rule.

The Apriori Algorithm mainly consists of the following steps:-

- Enter the data set.
- Find the support count.
- Generate frequent item set.
- Generate the association.

Frequent item set generation objective is to find all the items that satisfy the minimum support count threshold. Illustration of frequent item set generation using the Apriori algorithm:-

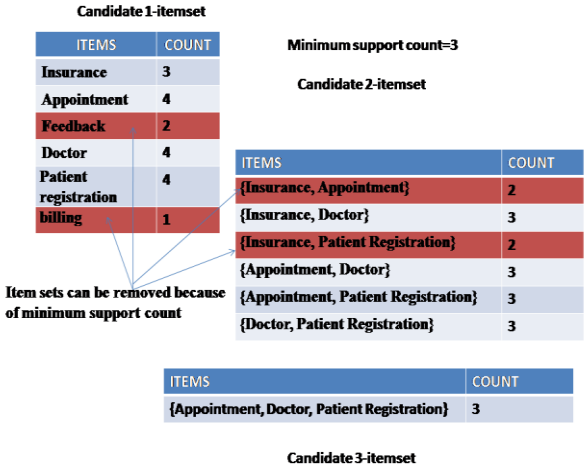


Figure 3: Frequent Service Set Generation

Consider the user interest for using the service is as follows:-

Assume that the threshold of input and output matching is 60%. If the affinity is greater than the 60 %, so there is high probability to combine them together. Based on input and output matching the following rules can be described. So that if any patient comes for the appointment booking so they may be interested for hospital registration and doctor. So based on the association mining M1 and M2 and M3 can be combined together.

Hence association analysis technique is helpful to find the related services and the user query can be answered in the minimum time.

V. IMPLEMENTATION AND RESULT ANALYSIS

This proposed research work is implemented by JAVA with eclipse and for the implementation of web semantic and ontology we installed protégé 3.4 with the eclipse IDE. The protégé OWL editor is provided with the standard installation of protégé. Here we have implemented the health care application. Microservices are implemented related to appointment booking, doctor's selection, and bill payments. The selection of microservices is performed based on QoS parameters for the user query execution. QoS attributes plays the important role for the filtration of appropriate services. Later we applied K-means clustering algorithm and association analysis method to improve the performance of the searching. With the implementation of these algorithms the results is improved and the services execution is performed with the improved time performance.

Header files used for protégé owl:-

```
import edu.stanford.smi.protegex.owl.model.OWLModel;
import
edu.stanford.smi.protegex.owl.model.OWLNamedClass;
import edu.stanford.smi.protegex.owl.ProtegeOWL;
```

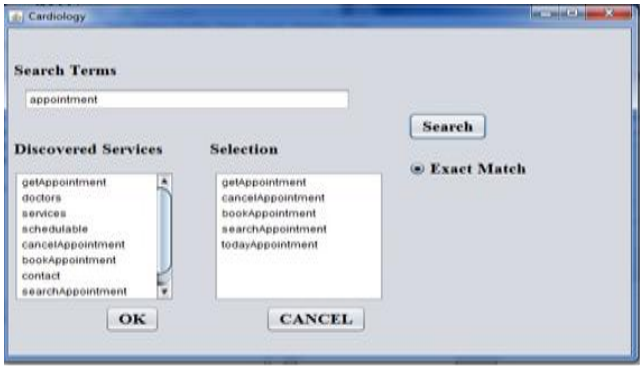


Figure:-4 JAVA Snapshot for Service Selection

The figure 5 shown as the graphical representation of service execution Response time performance during service orchestration. From this graph it is clearly noticed the service execution can be performed in the lesser time.

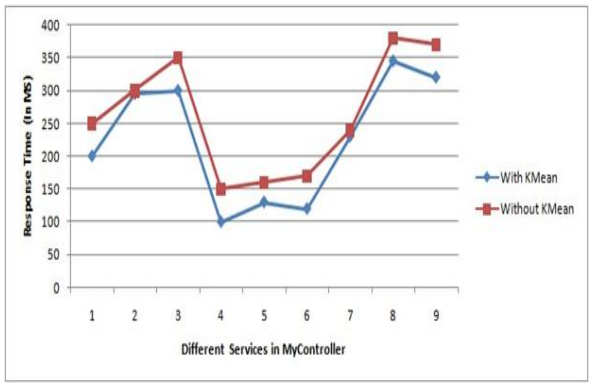


Figure 5:-Service Execution Graph

VI. CONCLUSION & FUTURE WORK

In this paper we have applied protégé –OWL API that is an open source java library for the web ontology language (OWL) and RDF. Protégé provides various features for the semantic annotation of web services. Semantic web service annotation is a complex task and it's very difficult to select the desired service dynamically in fewer amounts of time and with the high user satisfaction. In our research we proposed the idea of automatically generate the semantic annotation of services based on semantic knowledge. As the contribution of our research we have used data mining concepts i.e . Association analysis and k-means clustering techniques to improve the service selection automation with the improved performance. Association analysis is used to find the relation between associated services on the basis of support count and the confidence count. clustering algorithm is used to frame the groups of similar services .by using these data mining techniques the searching performance is improved with time efficiency. QoS parameters are used to provide the most appropriate solution to the user. It could help to improve the efficiency in service matching and the service execution.

In future the proposed work could be extended to apply the approach in the cloud service environment in order to examine its full potentials in the complex environment.

REFERENCES

1. Eyhab Al-Masri" QoS-Aware IIoT Microservices Architecture" 2018 IEEE International Conference on Industrial Internet (ICII), DOI 10.1109/ICII.2018.00030,2018
2. Mohammad Javad Kargar, Alireza Hanifizade" Automation of Regression test in Microservice Architecture" 2018 4th International Conference on Web Research (ICWR) 978-1-5386-5364-7
3. Fernando H. L. Buzato, Alfredo Goldman, Daniel Batista" Efficient Resources Utilization by Different Microservices Deployment Models" 978-1-5386-7659-2/18 ©2018 IEEE
4. Wilhelm Hasselbring, Guido Steinacker" Microservice Architectures for Scalability, Agility and Reliability in E-Commerce" 2017 IEEE International Conference on Software Architecture Workshops DOI 10.1109/ICSAW.2017.11
5. Sara Hassan, Nour Ali, Rami Bahsoon" Microservice Ambients: An Architectural Meta-modelling Approach for Microservice Granularity" 2017 IEEE International Conference on Software Architecture DOI 10.1109/ICSA.2017.32
6. Genc Mazlami, J¨urgen Cito, Philipp Leitner" Extraction of Microservices from Monolithic Software Architectures" 2017 IEEE 24th International Conference on Web Services DOI 10.1109/ICWS.2017.61
7. Shanshan Li" Understanding Quality Attributes in Microservice Architecture" 2017 24th Asia-Pacific Software Engineering Conference Workshops DOI 10.1109/APSECW.2017.33
8. XiaoDong Liu, SongHao Jiang, XiaoFang Zhao, Yan Jin" A Shortest-Response-Time Assured Microservices Selection Framework" DOI 10.1109/ISPA/IUCC.2017.00192
9. Christof Fetzer" Building Critical Applications Using Microservices" 1540-7993 © 2016 IEEE
10. Neha Singhal, Usha Sakthivel, Pethuru Raj" RESTful Web Services Composition & Performance Evaluation with Different Databases" 2017 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT) 978-1-5386-2361-9,vol 191,new york,USA:springer,2009,pp:1-14
11. X-s yang," harmony search as a metaheuristic algorithm"in music –inspired harmony search algorithm
12. Eyhab Al-Masri" QoS-Aware IIoT Microservices Architecture" 2018 IEEE International Conference on Industrial Internet (ICII) DOI 10.1109/ICII.2018.00030
13. Y. Gao,J.Na, B.Zhang,L.Yang and Q. Gong"optimal web service selection using dymaic programming"in proc.11th IEEE sym.comput.commun,jun 2006,pp365-370.
14. C,Wan,C.Ullrich,L. Chen,R. HUANG,J. Luo and Z.Shi,"on solving QoS aware service selection problem with service composition." In proc7th int conf. grid cooperative computing,oct 2008pp.467-474.
15. M.Jaeger,G.Muhl,and S. Golze,"QoS –Aware composition of web services: a look at selection algorithm,"in proc IEEEint conf web services,jul 2005,pp807-808
16. Y.Yang,S.Tang,Y.Xu,W.Zhang,and L.Fang"an approach to QoS aware service selection in dynamic web service composition" in proc 3rd int conf,jun 2007,pp18-23.
17. P.Rodriguez-Mier,M.Mucientes,M.Luma,"automatic web service composition with aheuristic based search algorithm," in proc IEEEint conf web service ,2011,pp 81-88.
18. A.Klein, F. ishikawa,and S. honiden"efficient heuristic approach with improved time complexity for QoS aware service composition ," in proc IEEE int conf. web service ,jul2011,pp436-443.
19. Y.Yu,H.Ma,and M.zhang"a genetic programming approach to distributed QoS-aware web service composition"in proc.IEEEcongr. evol,comput, jul 2014,pp.1840-1846.
20. W.Yunwm,"application of chaos ant colony algorithm in web service composition based on QoS",in proc. Int.forum inf technol.appl.,vol2,may 2009,pp 225-227.
21. Q.Wu and Q. zhu,"transectional and QoS awaredynamic service composition based on ant colony optimization,"future gener comput syst,vol29, no 5,pp 1112-1119,2013.
22. A.Klein,f.ishikawa,and s honiden,"efficient huristic approach with improved time complexity for QoS aware service composition",in proc IEEE int conf, web service ,jul2011,pp 436-443
23. Y,Li and T.wen,"an approach for QoS guaranteed web service composition based on a win win strategy,"in proc,IEEE 19 th int conf on web service ,jun 2012,pp 628-630

AUTHOR PROFILE

Mrs. Neha Singhal

Neha Singhal is having 9.5 years of teaching experience, presently working as an Assistant Professor, Dept. of Information Science and Engineering since 2010. She obtained her M.Tech from banasthali university, Rajasthan. She is pursuing her Ph.D under VTU in the area of web services. Her teaching and research interests are in the field of web services.

Dr.S. Usha

S. Usha, is working as a professor and head , CSE, RRCE. Graduated from Manonmanium Sundaranar University, in CSE during the year 1998. She obtained her Master degree in CSE and PhD degree from sathyabama university in the area of Mobile Ad Hoc Networks in the year 2013. She has 54 publications in International and National conferences, 22 publication in national journal and international journals in the area of Mobile Ad hoc Networks and wireless security. Most of the publications are having impact factor citied in google scholar (h index and i10index), Microsoft etc.. Received fund from AICTE under NCP scheme and SERB(DST). Received best teacher award from Lions club in the year 2010&2012.Dveloped Centre of Excellence lab in IoT with industry collobration.Organized many conferences ,FDPs and Technical Talks. Associated with ISTE, CSI,IEEE,IAENG ,IDES and IACSIT.Reviewed papers in IJCs and CiiT jouranals. Acted as a TPC member in MIRA'14 IoTBDS '17 and IoTBDS'18 Portugal.. Chaired sessions in FCS'14, ICISC'13 & ICCCT'15., ICCCT'17and IoTBDS'18.

Dr. Pethuru Raj Chelliah

.

Finished the CSIR-sponsored PhD degree at Anna University, Chennai and continued with the UGC-sponsored postdoctoral research in the Department of Computer Science and Automation, Indian Institute of Science, Bangalore. Thereafter, I was granted a couple of international research fellowships (JSPS and JST) to work as a research scientist for 3.5 years in two leading Japanese universities. Published more than 30 research papers in peer-reviewed journals such as IEEE, ACM, Springer-Verlag, Inderscience, etc. Have authored 8 books thus far and focus on some of the emerging technologies such as IoT, Cognitive Analytics, Blockchain, Digital Twin, Docker Containerization, Data Science, Microservices Architecture, fog / edge computing, etc. Have contributed 30 book chapters thus far for various technology books edited by highly acclaimed and accomplished professors and professionals.