© 2019 IJSRST | Volume 6 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology
DOI : https://doi.org/10.32628/IJSRST196170

QoS Based Discovery and Service Provider System

Ramesh Kumar, Lavanya V, Karthika R, Harshini B
Dhanalakshmi College of Engineering, Erumaiyur, Tamil Nadu, India

ABSTRACT

We propose a two-step, context-based semantic approach to the problem of matching and ranking Web services for possible service composition. Semantic understanding of Web services may provide added value by identifying new possibilities for compositions of services and context-based semantic. The semantic matching ranking approach is unique since it provides the Web service designer with an explicit numeric estimation of the extent to which a possible composition. This process consists of multiple services that can be executed in sequence or in a parallel process. Given a service request, a set of candidates (available services and service patterns) is dynamically generated layer by layer from inputs to outputs of this request. For each layer, the algorithm traverses a priori search space which is a set of service patterns from historical solutions, then it searches available services from the repositories. That means each layer contains all services and service patterns that can be executed with a set of outputs provided by previous layers. The search process terminates until all the outputs of a request are obtained.

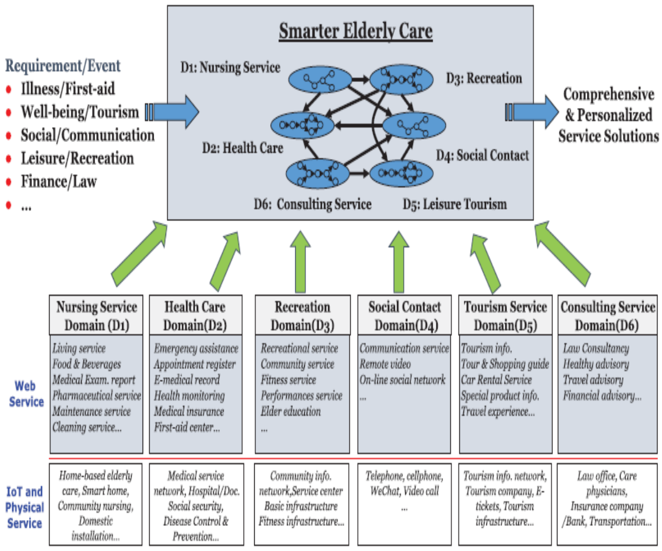
Keywords : QoS Based Discovery, Web service, Service Oriented Architecture, Internet of Things, CDMA, EPC, Bootstrapping Ontologies, CTM, Frequent Subgraph Mining, WSDL, FCA, CBR

I. INTRODUCTION

In Service Oriented Architecture (SOA), Web service as a method of communication between two applications over the Internet has grown up to be an important part of software development. Web services are software components designed to assist interoperable machine-to-machine interactions without considering the development platform or operating environment. Subsequently, semantic Web service approaches give us the ability to describe the capabilities of services in a formal and machine-process able manner and the semantic relationships among services, which are stimulating automatic service discovery and composition.

Over the past decade, the success of Semantic Web turns out to depend on the use of ontology as a means of communication and information sharing.

Ontology is a paramount technology of the Semantic Web, which provides a formal and explicit speciation of knowledge representation, with the advantage that they are reusable and shareable. However, as Web services go through different stages of development, the existing Web services on the Internet utilize various description languages. Not all service providers adopt semantic technology to Web services. Meanwhile, with the development and popularization of intelligent hardware, the booming of Internet of Things (IoT) represents the next most exciting technological revolution. The IoT connects billions of things that include sensors, actuators, services, and other Internet-connected objects which sets up the environment where things can automatically communicate with computers. The implementation of the IoT system will seamlessly integrate the cyber world with our physical world, and computers will be able to learn and gain information and knowledge to solve real-world problems. In order to tackle the previous problems, we propose a Priori Knowledge Based Service Composition (PKBSC) algorithm. For RQ1, we utilize an interoperable approach including an ontology construction method for services without semantic technology and a merging method for multiple ontologies. We also provide a method to calculate the similarity of semantic concepts in a merged ontology for RQ2, we adopt *service pattern* to describe priori knowledge from massive historical solutions, which is a recurrent valuable fragment (or a coarse-grained component) composed of services frequently invoked together in service solutions.



II. LITERATURE REVIEW

2.1 Web service composition

Service composition is a complex task that has inspired a variety of automation techniques trying to overcome some of the complexity.We identified three automation techniques of major importance: synthesis, planning, and model-driven development. Clearly, previous efforts by research and practitioner ´communities have produced promising results that are certainly useful. A commonly overlooked limitation of current systems is that they do not make composition languages accessible to end users (also called knowledge workers).

2.2 A Survey on Issues and Enabling Technologies

We believe emerging techniques such as privacy by design, differential privacy, and light weight public key cryptography will form the building blocks for security in IoT middleware. The results from these systems are still experimental and preliminary. In [46], Ostermaier et al. reported a system, named Dyser, for real-time discovery of things on the Web. limitations of CPU and memory put severe restrictions on the utility of the resource-demanding public-key cryptography as used in the Internet security.

2.3 A survey on enabling technologies, protocols and Applications.

We also reviewed the role of cloud and fog computing in the IoT ecosystem. At the cloud level, we need platforms to support IoT big data, IoT analytics and availability.

Their results show that the expected number of queries for tag identification using the CDMA technique is lower than the EPC Gen-2 protocol. the limitation of the cloud which offers just a specific application protocol is a barrier to its expansion.

2.4 A survey on data-centric Internet of Things

Traffic engineering techniques for managing VNFs in a SDN-based network include for example steering network traffic through an existing sequence of VNFs by using middleboxes that modify packet headers and alter traffic signatures. In conclusion, the lack of a standard definition of a smart city eventually results in various shortcomings in many facets of a smart city. We highlight the achievements in realizing various aspects of smart cities, present the lessons learnt and identify limitations and research challenges.

2.5 Distributed QoS Evaluation for Real-World Web Services

Web services have been emerging in recent years and are by now one of the most popular techniques for building versatile distributed systems. Comprehensive experimental results are presented and reusable datasets are released. The fact that different users will observe quite different QoS of the same Web service limits the applicability of this dataset.

2.6 Automatic Target-driven Ontology Merging

The idea is that these relationships produce different edges in the integrated concept graph and consequently different concepts and relationships in the merged taxonomy. More recent work on schema integration builds on the research results on semiautomatic schema matching and separate matching from merging. Hence, several algorithms have been proposed to merge schemas based on a pre-determined match mapping. It implements a new approach for taxonomy merging which generates a default solution in a fully automatic way that may interactively be adapted by users if needed.

2.7 Bootstrapping Ontologies for Web Services

Our model analyzes the concept results and concept relations and performs stemming on the results. It should be noted that other techniques of clustering could be used to limit the ontology expansion. clusters the results according to textual descriptors, and classifies which set of descriptors identifies the context of the web service. This limitation could be overcome by either recalculating the TF and IDF after each new web service or alternatively collecting an additional set of services and reevaluating the IDF values.

2.8 Leveraging formal concept analysis with topic correlation for service clustering and discovery.

Topic proportions exhibit correlation via the logistic normal distribution. In this paper, we utilized CTM to extract latent factors and the correlation between these topics to propose an efficient Web Service. The results show that the hierarchical clustering method based on CTM performs better than HAC. We also evaluated our service discovery approach by calculating the P recision@n and NDCGn. In our approach, we use the Formal Concept Analysis (FCA) [3] formalism to organize the constructed hierarchical clusters into concept lattices according to their topics. Thus, service discovery may be achieved more easily using the concept lattice.

2.9 Web service interface decomposition using formal concept analysis

A conceptual clustering technique as it also provides intensional descriptions for the abstract concepts or data units it produces. Central to FCA is the notion of a formal context. This practice results in a "hodge-podge" of unrelated operations in a single Web Service interface. Indeed, recent studies found that developers seem to take little care of the structure of their WSDL documents. Limitations of both approaches could be related to the coupling between interfaces which is not considered. Our approach attempts to explicitly address these two drawbacks to improve the interface design quality, taking the advantages of FCA.

2.10 Structural and Role-Oriented Web Service Discovery with Taxonomies in OWL-S

We combine object-based structural matching techniques that are used in the domain of Structural Case-based Reasoning (SCBR) [6], with Description Logic (DL) reasoning [7] over Profile instances, enhancing the discovery with services that cannot be retrieved using only logic-based reasoning. In order to overcome this limitation of the SP paradigm, we enhance our framework with the ability of a role-based Web service functional annotation based on the open-world assumption and the classification capabilities of the DL reasoning paradigm.

III. EXISTING SYSTEM

To promote the efficiency and accuracy of constructing a new service solution, considering the historical solutions could be a promising approach, which contains a set of already developed service solutions to satisfy customer's request. Often, there are lots of valuable sub processes frequently appeared in service solutions. Such a large number of sub processes could serve as a knowledge base for providing the guidance for solution construction effort. Therefore, the discovery and presentation of these valuable sub processes (a.k.a. service patterns) have become of great importance. A service solution could be abstracted to a Directed Acyclic Graph (DAG), the valuable sub processes mining can be considered as Frequent Subgraph Mining (FSM).

3.1 DISADVANTAGES OF EXISTING SYSTEM

They just create clients for giving false information. Executed in sequence or in a parallel process. Only contain one service .Domain is created for only particular service.

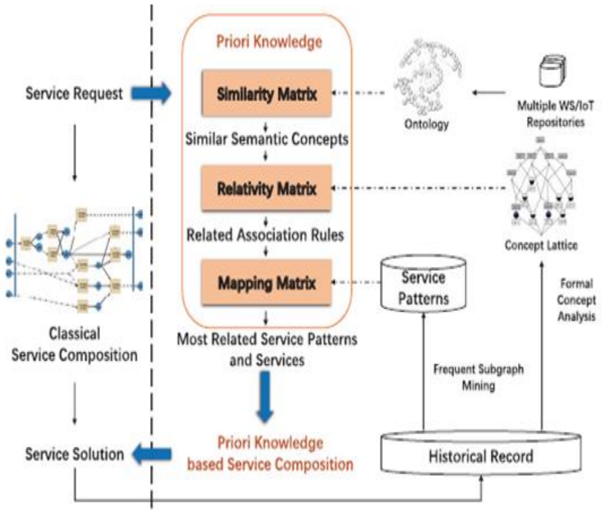
IV. PROPOSED SYSTEM

The Service Management System is a new self-contained software product which will be produced by the project team in order to overcome the problems that have occurred due to the current manual system. The newly introduced system will provide an easy access to the system and it will contain user friendly functions with attractive interfaces. The system will give better options for the problem of handling large scale of physical file system, for the errors occurring in calculations and all the other required tasks that has been specified by the client. The final outcome of this project will increase the efficiency of almost all the tasks done at the Hotel in a much convenient manner. We conduct two experiments to verify the effectiveness of our proposed approach. First, we compare the performance of Priori Knowledge Based Service Composition algorithm (PKBSC) with the A* based traditional.Graph searching service composition algorithm

4.1ADVANTAGES OF PROPOSED SYSTEM

Work will focus on how to integrate QoS into service patterns. The System is design and developed in such way that it tries to overcome all the prescribe problem. The system being an online system will give accurate information regarding the property which helps to view all the stuff information directly from anywhere. The voting system will also be included in this system.

4.2 SYSTEM ARCHITECTURE



V. ALGORITHM

Algorithm 5.1 Apriori-Based Service Pattern Mining Algorithm

Input: Historical service solutions *G*, minimum support

Output: *F*1; *F*2; _ _ _ ; *Fk* a set of frequent patterns of cardinality 1 to *k*

```
begin
  F1 Identify all frequent k=1 pattern in G
  k 2
  while Fk1 6D ; do
    Fk ;
    Ck candidate generation(Fk1) /*Candidate generation by a self-join of (Fk1)*/
    foreach candidate c 2 Ck do
      c.count 0
      foreach g 2 G do
        if subgraph-ismorphism(g; c) then
          c.count c.count C 1 /*Counting candidates*/
        end
      end
      if c.count _ 'jGj ^ c =2 Fk then
        Fk Fk [ c /*Pruning based on support*/
      end
    end
    k kC 1
  end
end
```

Algorithm 5.2 Priori Knowledge Based Forward Search Algorithm

Input: Service request *r* D (*Inr* ; *Outr*), all services *S*, all service patterns *SP*, similarity matrix *SimM*, relativity matrix *RelM*, mapping matrix *MapM*

Output: Layered candidates *LCS* D f*L*1; *L*2; _ _ _ g

```
begin
  Calculate related services S0 and service patterns SP0 by SimM; RelM and MapM
  C ;; i 0; Li ;; Outi ; /*i indicates current layer */
  Create a dummy service SIn D (;; Inr )
  Li Li [ SIn /*Li indicates candidate set in layer i*/
  Outi Outi [ Inr /*Outi indicates obtained semantic concepts by layer i*/
  C C [ Outi /*C indicates obtained semantic concepts so far*/
  while C _ Outr 6D Outr ^ Li 6D ; do
    i iC 1
    Li ;; Outi ;
    foreach spj 2 SP0 /*Search for available service patterns*/ do
      if C _ Inspj D Inspj then
        Li Li [ spj
```

```
            Outi   Outi [ Outspj
            SP0   SP0 n spj
        end
    end
    foreach sk 2 S0 /*Search for available services*/ do
        if C _ Insk D Insk then
            Li   Li [ sk
            Outi   Outi [ Outsk
            S0   S0 n sk
        end
    end
    C   C [ Outi
end
if C _ Outr D Outr then
    Create a dummy service SOut D (Outr ; ;)
    i   i C 1
    Li   ;
    Li   Li [ SOut
else
    Execute A*-TSC algorithm
end
end
```

VI. SERVICES

1) Login/Registration Module
2) Service Analysis
    a. Tourism Service Domain
    b. Home Tuitions
    c. Recreation Domain
    d. Consulting Domain
    e. Nursing Domain
    f. Furnishing Service
3) Physical Service
4) Quality of Service

VII. SERVICE ANALYSIS

1. Tourism Service Domain
    a. Tourism Information
    b. Guide Information
    c. Car Rental Service
2. Home Tuitions
    a. Tuition Information
    b. Criteria
    c. Fees Description
    d. Experience
3. Recreation Domain
    a. Fitness Service
    b. Beauty Parlour Service
    c. Performance Service
4. Consulting Domain
    a. Law Consulting
    b. Health advisory
    c. Financial Advisory
5. Nursing Domain
    a. First aid service
    b. Maintenance Service
    c. Elder service
    d. Pharmaceutical Service
6. Furnishing Service
    a. Painting Service
    b. False Ceiling
    c. Carpenting
    d. Plumbing

VIII. PHYSICAL SERVICE

In this module, physical service can act as a home based service. User can directly get above mentioned services at home.

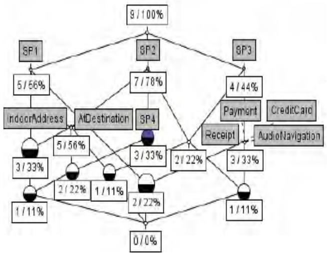
IX. QUALITY OF SERVICE

In this module, Web provided services based on quality. In this system, quality means user reviews and ratings.

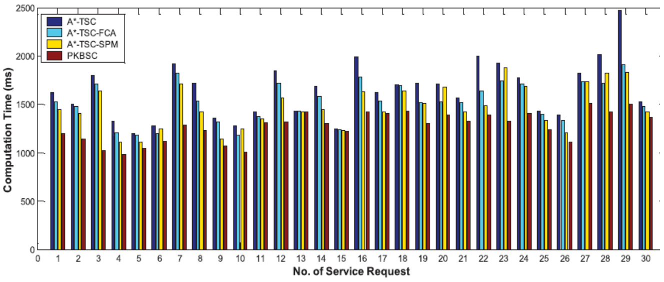
	Service Solution		Semantic Concept / Service Pattern									
	Original Solution	Annotated with Service Pattern	SP1	At Destination	Indoor Address	SP2	Audio Navigation	Credit Card	Receipt	Payment	SP3	SP4
Sol 1	W1-W2-W3-W5-W6	**SP1**-W3-**SP2**	X	X	X	X						
Sol 2	W1-W2-W4-W5-W6	**SP1**-W4-**SP2**	X	X	X	X						
Sol 3	W1-W2-W7-W8-W10	**SP1**-W7-W8-**SP3**	X				X	X	X	X	X	
Sol 4	W1-W2-W7-W9-W10	**SP1**-W7-W9-**SP3**	X				X	X	X	X	X	
Sol 5	W1-W2-W4-W5-W6-W11-W12	**SP1**-W4-**SP2-SP4**	X	X	X	X						X
Sol 6	W5-W6-W7-W8-W10	**SP2**-W7-W8-**SP3**				X	X	X	X	X	X	
Sol 7	W3-W5-W6-W7-W9-W10	W3-**SP1**-W7-W9-**SP3**		X	X	X					X	
Sol 8	W5-W6-W11-W12	**SP2-SP4**				X						X
Sol 9	W4-W5-W6-W11-W12	W4- **SP2- SP4**		X	X	X						X

SP1: W1-W2 **SP2:** W5-W6 **SP3:** W10 **SP4:** W11-W12

(a)



1. { } => SP2; [100%, 78%]
2. SP2 => AtDestination IndoorAddress; [78%, 71%]
3. AtDestination => IndoorAddress SP2; [56%, 100%]
4. IndoorAddress => AtDestination SP2; [56%, 100%]
5. AtDestination IndoorAddress SP2 => SP1; [56%, 60%]
6. SP1 => AtDestination IndoorAddress SP2; [56%, 60%]
7. SP3 => AudioNavigation CreditCard Receipt Payment; [44%, 75%]
8. SP1 SP2 => AtDestination IndoorAddress; [33%, 100%]
9. AudioNavigation => CreditCard Receipt Payment SP3; [33%, 100%]
10. CreditCard => AudioNavigation Receipt Payment SP3; [33%, 100%]
11. Receipt => AudioNavigation CreditCard Payment SP3; [33%, 100%]
12. Payment => AudioNavigation CreditCard Receipt SP3; [33%, 100%]
13. SP4 => SP2; [33%, 100%]



IGURE 7. Performance with different approaches (color for better reading result).

LE 1. Results of performance.

	Avg. computation time (ms)	Avg. number of layer	Avg. size of solution	Improvement
A*-TSC	1644.2	8.04	**8.96**	-
A*-TSC-FCA	1526.167	8.03	**8.96**	7.18%
A*-TSC-SPM	1475.367	**4.57**	**8.96**	10.27%
PKBSC	[illegible]	[illegible]	[illegible]	[illegible]

X. CONCLUSION AND FUTURE WORK

In this paper, we present priori knowledge based service composition (PKBSC) that assists the discovering and composition of services which best match user needs. We solve the heterogeneous service interoperability caused by different service repositories. We obtain prior knowledge by mining service patterns from historical records and quickly retrieve prior knowledge by establishing the relationship between requests and service patterns. We conduct a series of experiments using both real and synthetic data to evaluate the efficiency and effectiveness of our approach. We improve the efficiency of A* based traditional service composition algorithm (A*-TSC) by 22.44%. We conduct a series of experiments analyzing system performance in terms of thetime-aware frequency of service patterns. Our future work will focus on how to integrate QoS into service patterns and build service solutions through functional and non-functional requests.

XI. REFERENCES

[1]. A. L. Lemos, F. Daniel, and B. Benatallah, "Web service composition: A survey of techniques and tools," ACM Comput. Surv., vol. 48, no. 3, p. 33, 2016.

[2]. E. Christensen et al. (2001). Web Services Description Language (WSDL) 1.1.

[3]. M. J. Hadley, "Web application description language (WADL)," Sun Microsyst. Inc., Mountain View, CA, USA, Tech. Rep., 2006. [Online].

[4]. D. Martin et al., "OWL-S: Semantic markup for Web services," W3C Member Submission, vol. 22, no. 4, 2004.

[5]. R. Akkiraju, J. Farrell, J. A. Miller, M. Nagarajan, A. P. Sheth, and K. Verma. (2005). Web Service Semantics-WSDL-S.

[6]. D. Roman et al., "Web service modeling ontology," Appl. Ontol., vol. 1, no. 1, pp. 77_106, 2005.

[7]. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," Comput. Netw., vol. 54, no. 15, pp. 2787_2805, Oct. 2010.

[8]. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2347_2376, 4th Quart., 2015.

[9]. A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT Middleware:A survey on issues and enabling technologies," IEEE

Internet Things J., vol. 4, no. 1, pp. 1_20, Feb. 2017.

[10]. Y. Qin, Q. Z. Sheng, N. J. G. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric Internet of Things," J. Netw. Comput. Appl., vol. 64, pp. 137_153, Apr. 2016.

Cite this article as :

Ramesh Kumar, Lavanya V, Karthika R, Harshini B, "QoS Based Discovery and Service Provider System", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 6 Issue 2, pp. 197-204, March-April 2019. Available at doi : https://doi.org/10.32628/IJSRST196170
Journal URL : http://ijsrst.com/IJSRST196170