

Semantic Web Services Composition Using Clustering And Parallel Processing

¹A Nagarjuna Reddy , ²Dr. M Chandra Mohan

¹Research Scholar, JNTUH College of Engineering Hyderabad, JNTUH, Kukatpally, Hyderabad – 500085

²Professor of Computer Science & Engineering, JNTUH College of Engineering Hyderabad, JNTUH, Kukatpally, Hyderabad – 500085

ABSTRACT:

Today's Web, Web Services (WS) are developed and upgraded frequently. To meet the demands of the complex users in order to satisfy the demands of complex users, the creation of brand new Web services built upon existing ones is vital. This issue has received a lot of feedback from diverse communities. The issue is referred to as the composition of web-based services. It is a fact that generally accessible Web Services (WS) can not satisfy the diverse requirements of users, and their adaption to changing conditions remains a major issue in developing information system design. The composition of web services is designed to satisfy emerging and complex requirements similar to the processes you will find in many organisations. Its aim is to provide multiple services to meet the needs of customers. Utilizing existing services to develop business process applications can help reduce the time to develop and also costs. In addition, existing atomic services are not adequate to meet user's needs. Satisfaction of users demands a flexible and reusable environment that meets those requirements. But, it's one of the major challenges of recent times in a dynamic and distributed environment. Web service can be created by hand, however it is an extremely time-consuming process. Web services' dynamic structure is among the main characteristics of what's to come from the semantic internet. The various methods for constructing compositions for web services studied by researchers. In this paper, we propose the first design of semantic composition of web services, which makes use of the process of processing services in parallel and clustering models.

Keywords: Dynamic composition, Web Services, Semantic Web Service, Composition Techniques.

1 INTRODUCTION

Web Services are independent apps that are described or published, and then located and accessed via the Internet. Web service descriptions allow Web service descriptions

to be discovered and used in other services. Descriptions are written using the XML standard language. Descriptions are separated into two parts. functional features that are required to initiate the execution of the Web services, while non-functional features include such things as cost, response time and reliability. Webservice composition involves all about the search for services that accomplish an exact task [1]. Webservice composition is a crucial technology within the realm of web service . Its primary goal is to reuse the existing services on web. There are two kinds of composition for web services that are syntactic and semantic. Syntactic compositions are based in syntactic representations. The other one is built on semantic descriptions. Semantic web service composition is based on notions of ontology the purpose of adding semantic description instead of parameters. In recent years, a number of studies have been published on the web service composition.

Semantic web services are an area of research that is aimed to exploit the semantic annotation of descriptions of web services [22]. A majorly researched subjects in this area involves the automated processing of semantic composition of web services that aims at combining multiple services into a single service in order to create more complex workflows. The composition is examined in various settings to ensure the practical application. The composition framework we have developed is employed to automate creating workflows in the context of web services that can be used to create reused services.

2 LITREATURE SURVEY:

A variety of approaches that are available in the literature address the idea of personalization in Web composition of services. In this respect, Shanchen et al [3believe that the concept of context classification is crucial. It is important to distinguish from the context of U (user context) and that of the Web-context (web service context) as well as the contexts like the R-context (context resource). On one hand, the classification of the context can help in establishing more customized and consistency. It also allows you to verify the state of the Web service after it has been customized. However the three classes that are that are classified are interconnected. The user is the most dynamic of the three components: its requirements, preferences, and the conditions constantly change. It is the resource that's steady component whose characteristics and limitations can be identified in advance.

Semantic annotations play a crucial part in the process of identifying semantically aware services as well as recommendation as well as composition. Learned the semantics of an instance are added to the database to verify semantic annotations on other services. Furthermore, a 3-stage optimization process that incorporates local feedback global feedback, as well as global-propagate, is developed to enhance quality of the Qo SA in a gradual manner by correcting incorrect annotations. Experiments with a real-world web services repository have demonstrated that our method is able to improve services' Qo SA for services. It has the advantage of a 78.68 percent improvement in the annotations for input parameters and the identification of 36.47

percent of the output parameters that have incorrect annotations. The method we have proposed can be used in different service repositories to improve recommendations and discovery of services [4].

Knowledge-based service composition differs than knowledge exploration or creation in that both allow users to create sharing, explore, and access their own personal knowledge. Certain recommender systems or recommendations comprise a set of service that are processed by data processing, which blend the services according to preferences of users or the execution environment.

3 CLUSTER BASED COMPOSITION FRAMEWORK FOR PARALLEL PROCESSING OF WEBSERVICES:

Composition of Web Services is among the key activities of the field of service-oriented architecture. The aim of this paper is to provide a collection of web services that have the ability to combine in order to meet the varied requirements of users. We've studied a variety of approaches to compose web services that however, there are issues such as a reduction in precision, an increase in response time and a failure to select the optimal composition. We developed an algorithm for resolving these issues. Based on the time to respond, the accuracy and optimality of composition parameters were found in section 4, we present the proposed algorithm to ensure get the best results. * Search for all solutions. We search for all solutions and pick the most suitable according to the QoS. * Optimize the preprocessing. We carry out a large amount of computation in the preprocessing stage and create the data structure to allow prompt response to users' request. * Parallel processing. We designed our framework so that it efficiently utilizes the capabilities that multiprocessor systems offer.

Clustering: The clustering of web services within our algorithm is designed in the course of design and can the process more precise and less time since connected web service are placed into an identified group. This increases the chances of finding the initial web services that are in the best compatible with the requests of the requester. As you will observe in Fig 1 the dots Route is the one that was discovered making use of this clustering method. Based on our research the amount of comparisons is reduced due to the clustering of web-based services. This is due to the reality that only those services are included in the composition that have input information. They are accessible in conjunction with the query to the user , or as outputs of previously used services. This is why we label the web services in order to determine whether they are able to be used in the composition process or not. In the beginning, we are not sure which services could be classified as. So, we begin by marking every service as indeterminate. If we determine there is access to necessary input data for the service and it's usable then we will mark the item. If, however, we discover that there isn't any resource that could supply the necessary input data required for a particular service, we mark it as unusable. This is the reason why our method is a combination of three distinct operations running within identical data structures 1.) Finding usable services 2.) Finding services that aren't usable.3) backward chaining.

Locating services that can be utilized (see Algorithm 1) is based on forward chaining, beginning with the services we are able to input from the query made by the user. The aim of this method is to determine which services are useful, and then construct the structure of services using the input information available.

The goal of identifying unusable services is to reduce the amount of services to be taken into consideration when creating. The procedure is based on the premise that by eliminating services that are not used can improve the efficiency of both other processes, since they will not have to spend time processing these types of services. The process is performed for each service on the user-specific list of services for which inputs are not part of the query made by the user.

The purpose of forward chaining procedure is to construct the structure backwards from the original goal. First, the services that offer the information needed in the query of the user are chosen. This is why the querying mechanism of relational databases can be used to rapidly select the needed services.

Algorithm: Usable Services:

Input: request_service

For-all-successor in services do

if _is_Unusable (successor) == true_then

continue;

End_if

set connected_Input of _successor as _provided;

Chain_(service_, successor);

if are_AllInputs_Provided_(successor) == true_then

Mark_successor as _usable;

Find_Usable_Services(_successor);

End_if

End_for

Algorithm: Unusable Services:

Input: request_service

For_all_successor_in_service_do

if_is_Only_Input_Source (connected_Input,_service)_==_true_then

mark_successor_as_unusable;

Find Unusable Services (successor);

end_if

end_for

Algorithm_3_Backward Chaining:_

Input_service

for_all_input_of_service_do

if_isProvided (input)_==_true_then

continue;

end_if

$S \leftarrow$ _get Providers (input)

for_all_provider_in_S_do

if_is Unusable (provider)_==_true_then

continue;

end_if

chain (provider,_service);

if_is Usable (provider)_==_false_then

Backward Chaining (provider);

end_if

end_for

end_for

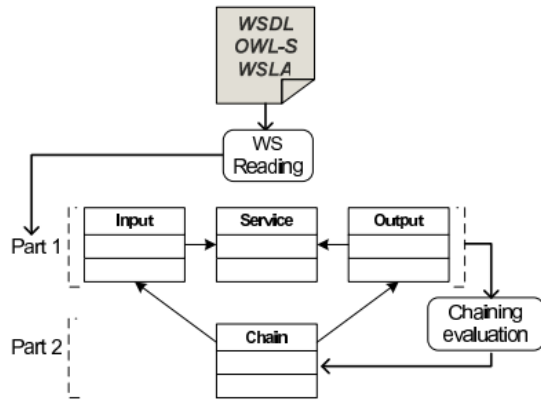


Fig 1: Structure of Data creation in a unit.

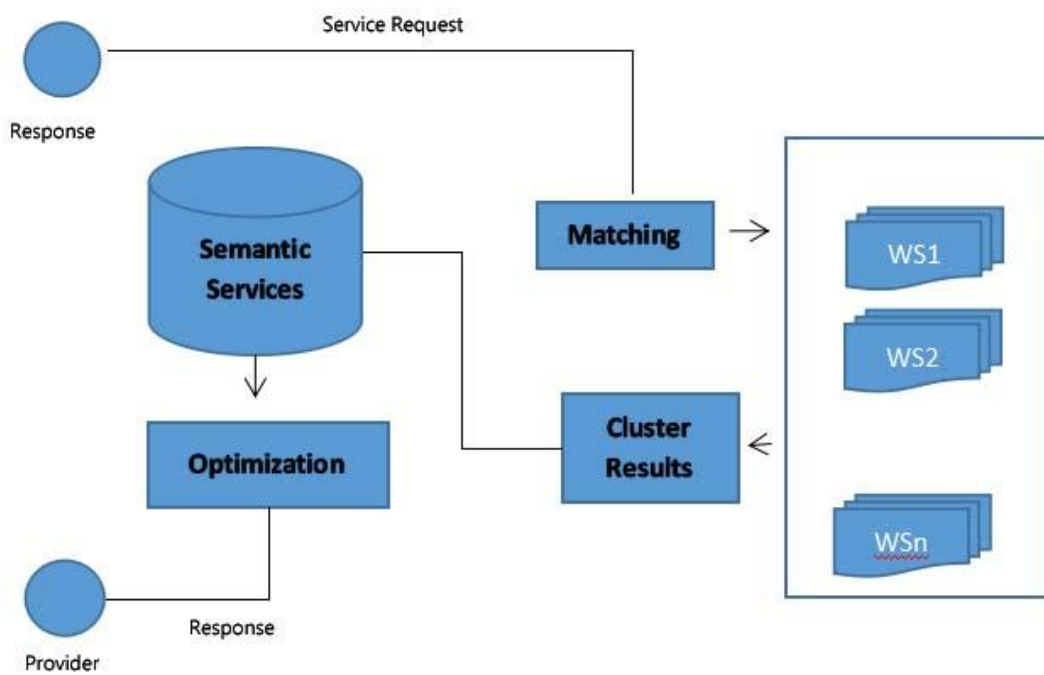


Fig 2: The Proposed framework of Web service Composition

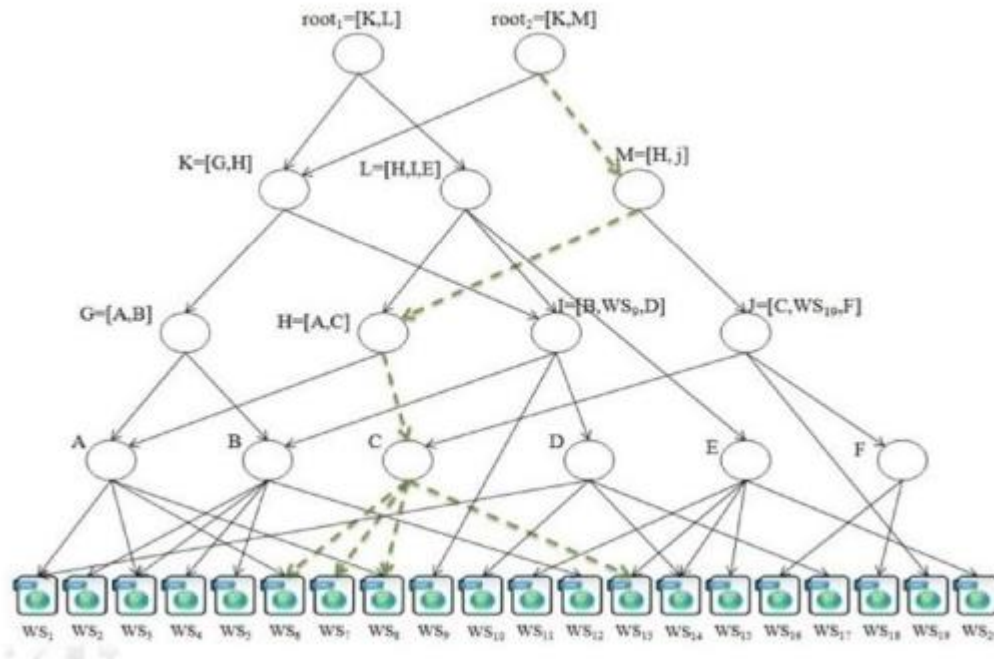


Fig 3: A case study of a finding of the first Web services in the tree structure of Clustering

4 RESULTS AND EVALUATION

To illustrate the process of discovering Consider two services specifically P1, P2 from the list of Provider services for the T1 user sub-task. The calculation of the similarity between the two services is carried out to calculate the score of functional similarity. The same is true for scores calculated for all other providers in the list of services, compiled with ontology. The computations of similarity are described in this manner Consider T1 as one of the IVR (Interactive Voice Response System) subtask that handles the user input request. In this subtask, a variety of web services offered by different service providers are pulled from the provider repository and are available on the list of provider services. To determine the semantic similarities two services, P1 and P2 that are listed in the list of provider services are looked at. This WSDL depiction of P1 and P2 services can be seen on Figures 4 and 5 respectively.

```
<?xml version = "1.0" encoding = "UTF-8"?>
<wSDL:definitions
  xmlns:soap = http://schemas.xmlsoap.org/wSDL/soap/
  xmlns:wSDL = "http://schemas.xmlsoap.org/wSDL/"
  xmlns:xsd = http://www.w3.org/2001/XMLSchema>
  <wSDL:portTypename = "obds_port">
    <wSDL:operationname = "outbound_dialled_services">
      <wSDL:inputmessage = "DialledNumberIn"/>
      <wSDL:outputmessage = "ResponseMessageOut"/>
    <wSDL:operation>
  </wSDL:portType>
```

Fig 4: WSDL Representation of Web service P1

```

<?xml version = "1.0" encoding = "UTF-8"?>
<wsdl:definitions
  xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <wsdl:portTypename = "dialservice_port">
    <wsdl:operationname = "dial_out_service">
      <wsdl:inputmessage = "GetDialIn"/>
      <wsdl:outputmessage = "GetDialOut"/>
    </wsdl:operation>
  </wsdl:portType>

```

Fig 5 : WSDL Representation of Web service P2

The files are broken down after WSDL parsing. When parsing, remove the metadata fields such as names of ports the operation, input as well as output message. On Table 1, the results from WSDL parsing is presented for two services, P1 and P2. The decomposition process is where each term is extracted by removing all numeric values, any special characters, and then converting the upper and lower case, when applicable, and then the data is stored inside the database. The terms that are decomposed for P1 and P2 both are summarized in Table 2.

Table 1 : Metadata Values obtained after WSDL Parsing

METADATA	P1	P2
PORT TYPE	obds_port	Dial_service_port
OPERATION	Out bound_dialled_services	dial_out_service
INPUT MESSAGE	Dialled Number In	Get Dial In
OUTPUT MESSAGE	Response Message Out	Get Dial Out

Table 2 : Decomposed Metadata Terms of services P1 and P2

METADATA	P1	P2
PORT TYPE	Obds, port	Dial, service, port
OPERATION	outbound, dialled, services	dial, out, service
INPUT MESSAGE	Dialled Number In	Get Dial In
OUTPUT MESSAGE	Response Message Out	Get Dial Out

The main activity of making the service classes from the provider service lists is built on semantic distances determined between metadata port types for the services P1 and P2. using the similarity API that is provided via Amazon Web Services (AWS) these distances are calculated in a semantic way. The semantic distance between the terms P1 and P2 services used for Port Name metadata are listed in Table 3. The bi-graph representation of the same is illustrated in Figure 4.

Table 3 Semantic distance - Output Message of services P1 and P2

OUTPUT MESSAGE		P2		
		get	dial	out
P1	response	0.9657	0.8452	0.8524
	message	0.9845	1.1875	0.4867
	out	0.6214	0.8741	0.9587

The words with the highest scores and are considered to be semantic measurements are,,response" and,,get" messages" as well as dial" out" and,,out" with scores of 0.9657, 1.1875 and 0.9587 respectively.

Table 4: result of the composition system proposed by completing each set of challenges of WSC 08

Challenge set	Number of services	Number of given input	Number of given output concept	Minimum composition Length (Proposed Algorithm)	Min Execution Time (ms)	Optimal composition length (reported in WSC 2008)
Set01	100	4	4	9	31	10
Set02	500	3	2	6	76	5
Set03	500	6	4	35	987	40
Set04	1000	5	5	10	1567	10
Set05	1000	8	8	19	70932	20
Set06	2000	12	11	36	1807116	40

Set07	4000	18	14	21	4446021	20
Set08	8000	27	21	22	1892817	30



Fig 6 : Model Performance

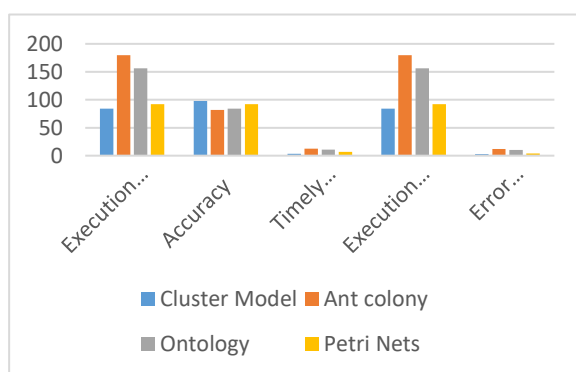


Fig 7 : Models comparison

5 CONCLUSION AND DISCUSSION

This paper created semantic web service composition system. To explain semantics, we employed description of Web services. In this paper, we employed clustering to classify web services in order to enable consumers of web services to connections between services quickly and a parallel model that allows web services that can function more efficiently. Clusters are that is used to determine the most effective collection of web services with an excellent ability for combining. The study showed that the proposed system could determine the best length for web service composition based on the various challenges set for various number of services.

6 REFERENCES:

- [1] R. Studer, S. Grimm, and A. Abecker. Semantic Web Services: Concepts, Technologies, and Applications. Springer, 2007.

- [2] Fekih H, Mtibaa S, Bouamama S. «An Efficient User-Centric Web Service Composition Based on Harmony Particle Swarm Optimization». *International Journal of Web Services Research* · January Volume 16 . Issue 1 . (2019)
- [3] Shanchen P, Qian G, Ting L.H, Guanguan X, Kaitali L. « A Behavior Based Trustworthy Service Composition Discovery Approach in Cloud Environment ». *IEEE Access* (Volume 7), (2019)
- [4] Keman Huang; Jia Zhang; Wei Tan; Zhiyong Feng; Shizhan Chen, "Optimizing Semantic Annotations for Web Service Invocation", *IEEE* 2019
- [5] S. Kalasapur, M. Kumar, and B. A. Shirazi, "Dynamic service composition in pervasive computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 907–918, 2017
- [6] Cotfas Liviu Adrian, Diosteanu Andreea, and Smeureanu Ion, "Fractal Web Service Composition Framework," in *Communications (COMM), 2010 8th International Conference on*, Bucharest, 2010, pp. 405 - 408.
- [7] Pat. P. W. Chan and Michael R. Lyu, "Dynamic Web Service Composition: A New Approach in Building Reliable Web Service," in *22nd International Conference on Advanced Information Networking and Applications*, Okinawa, 2008, pp. 20 - 25.
- [8] Muhammad Adeel Talib and Zongkai Yang, "SEMI-AUTOMATIC CODE GENERATION OF STATIC WEB SERVICES," in *Engineering, Sciences and Technology*, 2004, pp. 132 - 137.
- [9] Rui Wang et al., "Web Service Composition using Service Suggestions," in *2011 IEEE World Congress on Services*, Washington, DC, 2011, pp. 482 - 489.
- [10] Kostadinov D. Personalization of information a profile management approach and reformulation of queries. Thesis in Computer Science. University of Versailles Saint-Quentin-En-Yvelines. (2005)
- [11] Serge Garlatti and Yannick Pri, *Adaptation and personalization in the Semantic Web*, *Revue I3 Information-Interaction - Intelligence*, 24 pp. (2004)
- [12] Fekih, H., Mtibaa, S. and Bouamama, S. (2016), "User-centric web services composition approach based on swarm intelligence", *IEEE 18th International Conference on High Performance Computing and Communications*.(2016)
- [13]. Mahmoud Zakaria Maamar. Qusay H. Soraya Kouadri Mostéfaoui. Context for personalized web services, *Proceedings of the 38th Hawaii International Conference on System Science*. (2005)
- [14] T. Stuzel and H. H. Hoos , "MAX-MIN Ant System" , *Future Generation Computer Systems* , vol. 16 , no. 8, 2000,pp. 889-914

[15] Agre, Marina , “An INFRAWEBs Approach to Dynamic Composition of Semantic Web Services” , Cybernetics and Information Technologies , vol. 7, no. 1,2007,pp. 45-61.

[16] Yogesh Hole et al 2019 J. Phys.: Conf. Ser. 1362 012121

[17] Roman, D., U. Keller, H. Lausen (Eds.) , Web Service Modeling Ontology, 2006, WSMO Final Draft.