Journal of Theoretical and Applied Information Technology
15th July 2016. Vol.89. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FAULT TOLERANTAND RELIABILITY MECHANISM FOR WEB SERVICES BY CHANGE MANAGEMENT AND BROKER

[1]B. MURUGANANTHAM, [2] Dr. K. VIVEKANANDAN

[1] Assistant Professor, Department of Computer Science & Engineering, SRM University, Chennai

[2] Professor, Department of Computer Science & Engineering, Pondicherry Engineering college.,Puducherry

E-mail: [1] bmnantha23373@gmail.com, [2] k.vivekanandan@pec.edu

ABSTRACT

Businesses provide various services that are not offered through one Web Services to users. Component Web Service supply several complex function through a composition of different Web services. Failure of composite services is more vulnerable rather from atomic services. Failure of single Component Services rather unavailable in the implementation of combined Web Services it shows fail of all services. Nowadays World Wide Web consist of a vast quantity of Web Services and many are designed for support mainly kind of business function. Currently, single web services do not support the business enterprises as their necessity is colossal in an environment. Services or Business Source are unable to give the response to the requester. And in that condition entire composite services require to running finally. Thus,the necessity of organization make certain failed services efficiently and rapidly exchanged as well as running procedure is not disturb. For this kind of query to attend modify management problem in LCSs a system is recommended known as Evolution of Long-term Composed Services. Cooperation in mid of autonomous web services and Long-term Composed Services jointly offered value added service. And it results in the long-standing assurances to users.

Keywords: *Web Service, Change Management, Broker, Ev-LCS, Reliability*

1. INTRODUCTION

"A software system designed to support interoperable machine-to-machine interaction over a network" through World Wide Web Consortium is defined as web services [2]. The primary goal of the proposal of web services to supply necessary interoperability level between diverse functions by redefined of web standards. In the process to permit necessary flexibility in a heterogeneous system the integration model of web services is limply joined. Base upon software oriented architecture while different in interface various web services like as Simple Object Access Protocol and Representational State Transfer web services are used.

The 'great leap' of services oriented architecture has been presented by web services introduction while the concept of web services is currently compared to services oriented architecture. Commonly in services oriented architecture web services is established as center element additionally offering essential autonomy with composition, energetic innovation and independence stage. By re-existing system of web services could switch of the information with neglecting the necessary to recognize technical information regarding another computer system. The Architecture of web services, as well as core provision which used to attending every process of services oriented architecture, are express in figure.1, and also an implementation of services oriented architecture concept. Design of web services used three interiors to offered platform for independent messaging over network:

1.1 SOAP (Simple Object Access Protocol): Simple Object Access Protocol (SOAP) is an protocol based on the XML, which allows the exchanging of the data over the HTTP (Hypertext Transfer Protocol). The World

Wide Web Consortium (W3C) [presenting the definition of the Simple Object Access Protocol (SOAP) is a protocol which has the light weight intended for the infrastructure exchanging information in a distributed and decentralized environment [3].

1.2 UDDI (Universal Description Discovery and Integration): Universal Description Discovery and Integration (UDDI) is described as the “a services set for supporting the business discovery and description of the business and organization and other providers of the web services, the web services, which making the availability of the web services and interfacing with the technical aspects of those service by The Organization for the Advancement of Structured Information Standards (OASIS) [5].

1.3 WSDL (Web Service Description Language): The format of XML describing the services of network set as an endpoint of the operating messagecontained the information of either the procedure-oriented or document-oriented [1]. Universal Description Discovery and Integration (UDDI) is initiative for an industry which enables to publish the business and their services and allows potentially the users to discover the services. The registry of the UDDI could be private and public and users could select and search the perfect services from the registry of the UDDI.

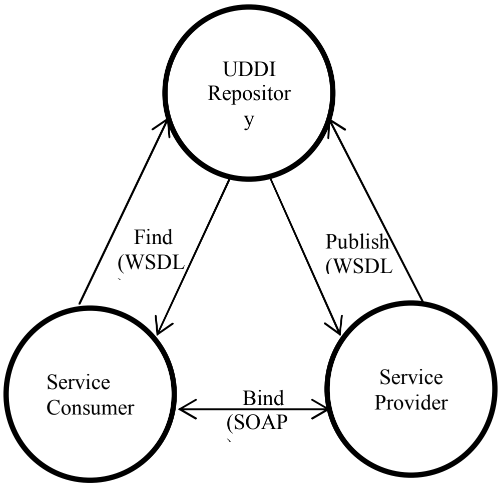

Figure 1: Web Service Architecture

The Quality of Services (QoS) [4] supporting the web services over the hot topics attracting the researchers from industry and academia. In the period of the web services emergence, many researchers have focused on the aspects of the web services interfacing. The management of the Quality of Services (QoS) has extensively studied the web based application and multimedia application based on the network. In the web services context, the very recent issue is researched over it. Nearby the days, the clients and providers of the services are concerning with the guaranteed QoS web services. With the point of view of the client, the QoS selection is having a mechanism of the decision making of the multi-criteria, which requires the knowledge of the service description.

2. RELATED WORK

The paradigm of the web services is a recent idea in web application emerging environment. It’s having the set of protocols, technologies and languages that allow the automatic communication amid of the internet and web application. The use of the web services that exposing the functionality through the description of the interfaces and available publicly for the other programs. The web services are the novel technology, and most focused existing work for the development of their practices. The functionality of the DAML-S providing the description of the semantic ontology of the web services, which includes the function specification and constraints of the QoS [16]. The IBM proposes the WSLA (Web Service Level Agreements) that is a specification of the web services of SLAs and focused on the constraints of the QoS [20]. WSOL (Web Services Offering Language) has developed the specification of the several constraints, classes of services and management statement for web services [10]. QoS supporting the framework makes enable the web services that is proposed in [9], [15], [16] presenting a web service model for the discovery of the web services that includes the non-functional and functional requirements of the web services.

The author of the [16] hasn’t implemented and defined perfectly, and it’s not describing the process of certification details. Additionally, it’s neither validating the content of the WSDL and control of the selected QoS delivery. The paper [17], authors proposing a describe content and the implementation of the architecture based on the broker for QoS web services control.

In the latest survey by the author of [6], presented the web services testing and classify the research, which undertaken in the four categories:

a) Regression,
b) Non-functional testing,
c) Integration and
d) Functional.

The area of the testing hasn't been covering in this research work as the verification has been done formally upon web services and several contemporary techniques. There are certain types of approaches have been discussed in the paper [14], which uses the model for the generation of the test cases. The author has presented a generation of test data criteria based on the semantic WS (Web Services). In this method, the4 generations of the test cases are based on the IOPE and pre-defined model of the fault for the specification of semantic process.

The author of the [8], [13], had used an automata for the WS-TEFSM (Web Service Time Extended Finite State Machine) and IF (Intermediate Format) for the generation of the test cases, which aims to exercise the constraints of the time in compositions of the web services. An intermediate format model that was enabling the time constraints model in BPEL, the immediate effect of the Web Service Time Extended Finite State Machine (WS-TEFSM). For Intermediate Format model of the transformation for BPEL, the author Lallai et al, used a tool, which known as BPEL2IF and for generation of the test cases one another tool has been used that known as TESTGen-IF. The second tool TESTGen-IF have the capability for generating the test cases for intermediate format language.

The author of the [7] had proposed application for the testing techniques of X-machine stream based upon the BPEL. In the paper [18], X-machine stream has been proposed a formal method for controlling the model data of a system and the generation of the test cases.

The author of the paper [19] had proposed an issue over the generation of the message-sequence composition. The approach of the process of BPEL, MSG (Message-Sequence Graph) and the generation of the test cases are used.

3. PROPOSED SYSTEM

3.1 Overview

In this research proposed work, being proposed a novel idea Ev-LCS (Evolution of Long-term Composed Services), which present the problem in LCS (Long-Term Composed Services). In autonomous web services, Long-Term Composed Services is dynamically collaborating, which provide the value-added services. Firstly, the presented formal method is for providing the support of grounding semantic for the change management automation. This work is presenting the techniques of Evolution of Long-term Composed Services (Ev-LCS), a framework that specifies in end-to-end and validates the changes in Long-Term Composed Services on top-down approach including the ontology of web services and Long-Term Composed Services (LCS) Schema. In this research work, the module of change management confirming the changes on two phases: Instance phase and schema phase. Then after define, the changes set for operating the top-down approach changes from formal model. Here, proposing a novel idea to make changes automatically in mid of the enactment process of change. Try to prove the presented work practically and did it with an expected output.

3.2 Broker

The technique has been presented to execute the present issues in the web services. A medium of the technique for fix the issues (Broker) has been utilized to interact with the service requester and provider. The proposed techniques are introducing the user's requirement for the service of the desired component to include in the intricate services. The brokers match the parameter and try to execute the UDDI in web services based on the request of the users and represent the composite web services (Business Process) through the web searched services and after that prepare for the web services component execution in composite services of web. If During the execution of a Composite Web Service, one of the Component Service fails or becomes unavailable, the whole Composite Web Service fails.

3.3 Overall Architecture

3.3.1 Service Requester: The users are the actual requester for them developed the system. The users are entering the services of the desired component with functional requirement for including the complex web services.

3.3.2 Service Provider: This functionality responds the user's request with the real and suitable services which are demanding by the requester. These functions are providing the core functionality of the business.

3.3.3 UDDI Registry: In this feature, the server storing all the information of the services. Then the broker looks for the entire component which is using the service provider in the registry of the UDDI for matching with the requirements of the user's request.

3.3.4 Web service Finder: This function act as a searching media for responding to the user's need or requirement.

3.3.5 Web Service Composer: This function act as the web services composer behalf of the demands of the users. It read the input from the WSF (Web Service Finder) and composes the service of the component and then sends it to the Web service Execution Monitor for the purpose of the result.
3.3.6 Web service Execution Monitor: This function controlling and monitoring the services execution. If any of the service being fails then the execution monitor will detect the failed services and forward the notification to the recovery manager about the failed service. If there is no occurrence of a failure in the system, then it sends to the service requester.

3.3.7 Web Service Recovery Manager: It fix the failed service in the registry of the UDDI of the web services and complete the execution process after finding the secondary or alternative way. If recovery manager is not able to find out the service node for replacing the failed service, then it calculates all the failed services subsets.

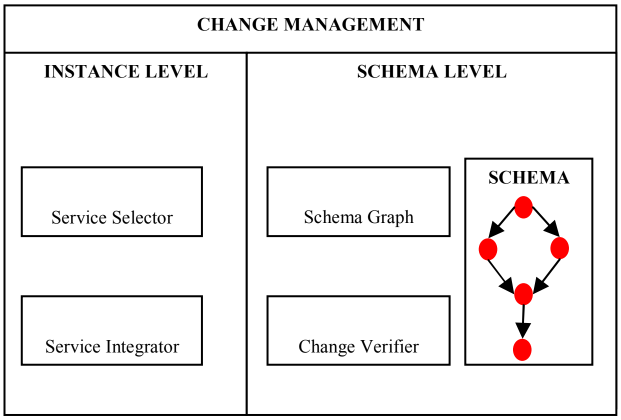

Figure 2 : Overall System Architecture

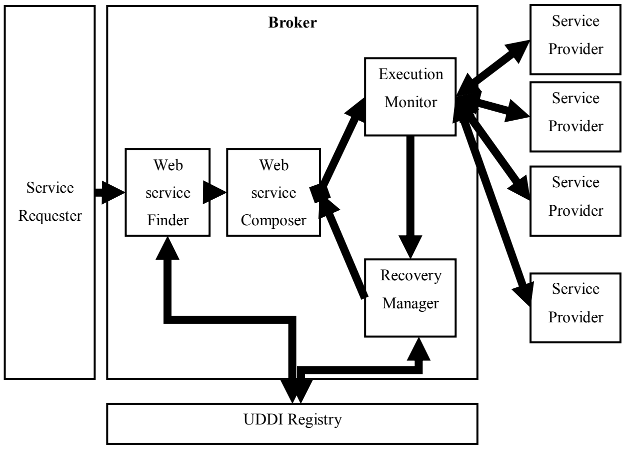

Figure 3 : Broker function Architecture

3.4 Detection of Fault Subset: In computer services, there are several errors and faults are occurring in it, some of the typical artificial errors and faults could be injected and errors will be observed for characterizing the faults presents in the computer system, it not mean that the real faults and injected faults will be exact equal, but may they have similar errors. An aspect that is most important in the injection of the fault is fault trigger, which describing the condition for exposing the fault.

3.5 Subset Replacement Policy: This system replacing the failed subset and executing the equivalent subset only after verifying the given rules:

1. The subset of the web services must have to provide the similar functionality that is provided by the Failure subset.
2. The subset of the web services must have to follow the similar constraints of the user which allow to follows the failed subset of the web services.
3. The cost must have to be same for the failed subset.
4. The subset of the web service must have the highest parameter of QoS which is selected for the replacement for the failed subset.

3.6 Change Management Module: There are two steps or level in the change schema that are: Instance level and Schema level. The schema level focuses on the change reaction for the requirement of the change, which is related to the LCS functional features.
Ev-LCS (Evolution of Long-term Composed Services): This system is providing the practical study of the proposed method. The travel agencies are running on the Long-term Composed Services (LCS) scenario. The implementation of the Ev-LCS

is based on the twofold. Firstly, the implementation of the proposed algorithm for expressing the practical benefit then the system have to provide a perfect graphic user interface (GUI) for a Long-term Composed Services (LCS) owner for specified change submission. It generates automatically the enactment of the changes and secondly, this research work is conducting the experiment set for assessing the system performance. Once, the owner of the LCS specified the changes, and then the system will generate the new schema for LCS with a suitable time range.

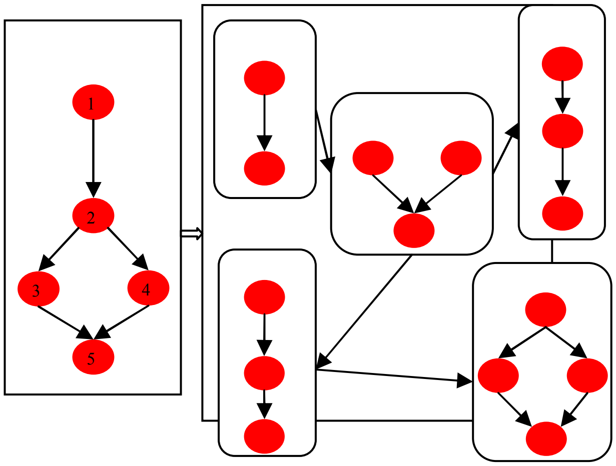

Figure 4. Change Management Module Architecture

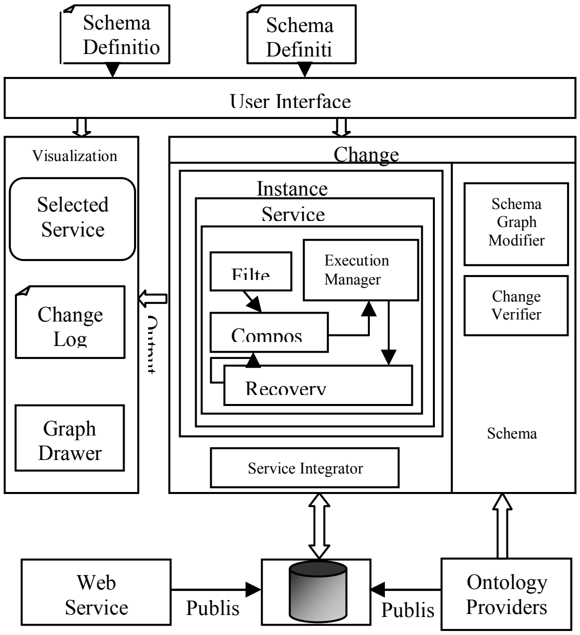

Fig.5. Global and Local View of Schema

3.7 ALGORITHM

3.7.1 Algorithm for Alternative Subset Replacement

This algorithm defines and detects the failed services ad replace those failed services with a specific working services in similar behavior or characteristics. Firstly, this algorithm finds and identifies the faulty subset and replace with the equivalent one.

Input: FS: failed subset in which failed service occur
CSS: Composite service set

Output: NCS new composite service

```
Step.1. i=1 //First take the failed Service
Step.2. While (all sub set are not finished or failed subset successfully replaced)
Step.3. Select subset in which number of service=i
Step.4. Search for alternative service in UDDI
Step.5. Make the possible subset
Step.6. Identify the equivalent subset
Step.7. If equivalent subset is not found
then i=i+1
Step.8. Else
Rank that equivalent subset based on cost and QoS
Select best subset
Replace failed subset to the equivalent subset (ES) in composite service set
Take the set difference (SD) of CSS and FS
SD = CSS-FS
Take the Union of the SD and ES
New Composite Service (NCS) = SD U ES
Step.9. End of while
```

3.7.2 Ev-LCS Algorithm

This project is using change operator algorithm that is classified into types such as functional requirements and non-functional requirements

The functionality of an LCS is determined by the outsourced abstract services and their composition, which is defined in the LCS's schema. In this research work M has been used to denote an LCS schema. A change of M consists of the change of the outsourced services and the change of the service composition.

A change of the outsourced services can be adding, removing, or replacing a set of abstract services. The selection of these abstract services is always associated with the change requirement, such as finding an abstract service that fulfills the functional requirement specified in a change.

Algorithm

```
Construct the set M using Algorithm Pre. Let Mi= {(i; j) ∈ M, 1≤ J ≤n }, 1≤ I ≤ n.
globalLCS.Instance = ∈
globalLCS.Value = 0
for i= 0 to n {Index 0 is needed for the computation of T. value [I, j], j = 1}do
    S[i]: Value = 0 {Initialize the temporary array S}
    S[i].Instance = ∈
end for
for i= 1 to ndo
    H = S {Update H for the next row}
    Preprocess H. value for Range Maxima Query
    for each (I, j) ∈Mido
        maxindex= RMQH(1, j - 1)fRange Maxima Query on Array H}
        T. value[i, j] = H[maxindex]. value + 1
        T. Prev[I, j] = H[maxindex].Instance
        S[j]. value = T. value[i, j]
        S[j].Instance = (i, j)
        if globalLCS.Value< T .value[i, j] then
            globalLCS.Value = T .value[i, j]
            globalLCS.Instance = (i, j)
        end if
    end for
end for
return globalLCS
```

4. Results and discussion

4.1 Experimental setup

In order to measure the performance of our proposed approach, a sequence of experiments on extracted dataset was conducted. We need a laboratory with some systems with high-end configuration that supports latest applications along with LAN setup to do these experiments efficiently.

4.2 Differentiated class of web service

Web Services Class→ QoS Parameters↓	Class 1	Class 2	Class 3	………	Class n
Response Time	0.8 m/s	0.65 m/s	0.52 m/s	……	0.05 m/s
Latency	N/A	N/A	0.15 m/s	……	0.01 m/s
Availability	N/A	N/A	0.85	……	1 (100%)
Reputation	N/A	N/A	N/A	……	5/5
Service charge	0.29$	0.21$	0.10$	……	0.51$

Table.1. Differentiated class of web service

4.3 Web Services Availability

In the above Table.1 we describe the function of the differentiated classes of the web services based on the QoS for the several clients or users profile. Every class is defining the QoS set of the attributes of the web services that offers the distinct value, which presenting in the above table based on the table. Where, RT (Response Time) is the service taken time for responding the user or clients. The costs of the requesting services are known as the service charge. Time for requesting the service and responding the request is latency. The reliability of the services is known as the reputation.

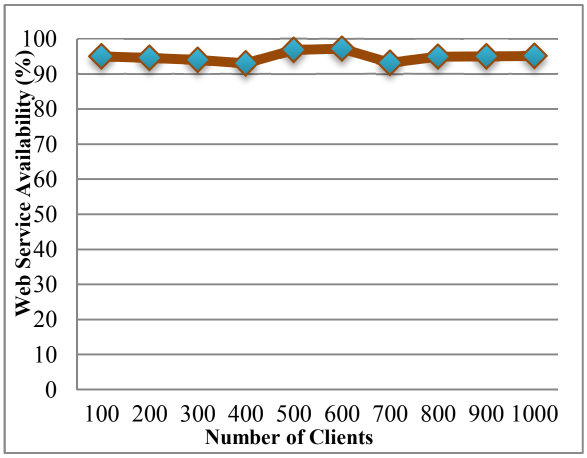

Figure 6: Web Services Availability

The above graph, Fig.6 presenting the availability of the services in any situation, even the clients or crowd is increasing. The services availability is high with the proposed system.

4.4 Web Service Reliability

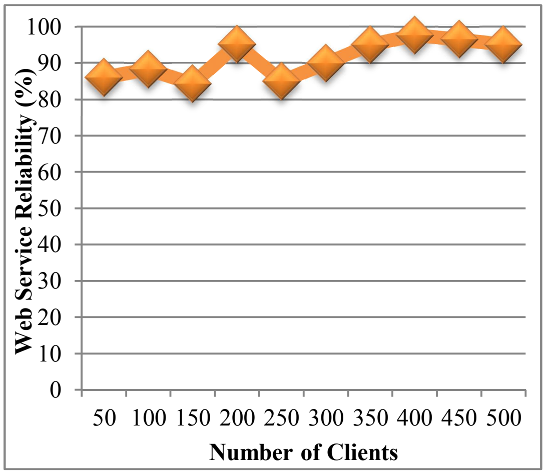

Figure 7: Web Service Reliability

The fig.6, presenting the Reliability of the services over the system is high. The number of clients is increasing, and it's showing the crowd on the system is more and thus representing the reliability of the services. The more number of the user representing that the service is more reliable and user interface.

4.5 Web Services Response Time

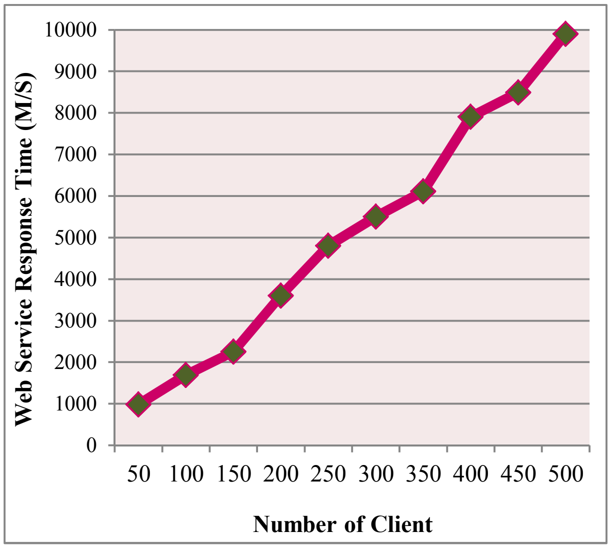

Figure 8 : Web Services Response Time

In the above fig.7 is presenting the situation of response time for the services over system. The response time is simultaneously moving upward with increasing number of the clients. The response time is satisfactory for the user's response and request. The response time is in M/S.

4.6 Accuracy of System

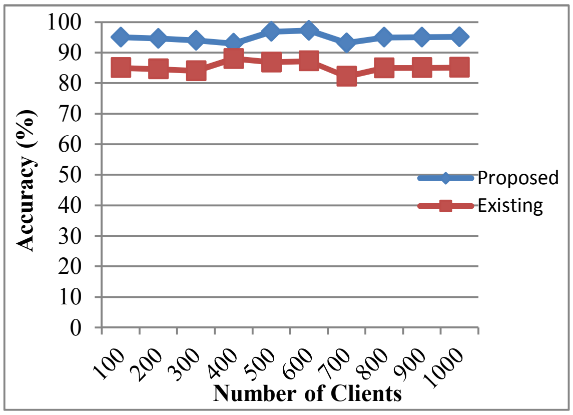

Figure 9 : Comparison of System Accuracy over Existing and Proposed Techniques

In this research work, the assumption result has been produced by the experiment, which is showing the efficiency or accuracy of the system. The proposed work is presenting the high accuracy of the user's request for the web services.

5. Conclusion

In our proposed work, we have presented a novel technique Ev-LCS and Broker system that are end-to-end framework and specify, validate, and reacts to the changes over top-down approaches in Long-term Composite Service. Firstly, we proposed a formal model to provide the semantic of groundings for supporting the change management automation, which include the ontology of the web services and a schema of LCS. Then after, we define the operators set for top-down approach changes based on the formal model. The Reliability and accuracy of the Ev-LCS assure the change enactment for the change management. In our proposed work, the mechanism of the fault tolerance has been used for the quality purpose which detects the fault of the services and makes it to replace with the individual working services system. This method based on the replacement of the fault services and providing the accuracy in the service. The proposed techniques has enables the users to be available the complex services that meets the requirement of the users, even if the system or services being failed the system will hide the failure services and replace it with the broker or agent. The proposed techniques are introducing the user's requirement for the service of the desired component to include in the intricate services. The brokers match the parameter and try to execute the UDDI in web services based on the request of the users and represent the composite web services (Business Process) through the web searched services and after that prepare for the web services component execution in composite services of the web.

We have proposed novel techniques to improve the web services with the new idea of the Broker and Change management. Our proposed techniques are more efficient to cover the existed problem in future also. We hope that more number of researchers will be attracting with these techniques to do the researches and for improving the functionality of the web services in future.

REFRENCES:

[1]. Web Services Description Language (WSDL 1.1), [Online] http://www.w3.org/TR/wsdl.

[2]. Web Services Glossary, [Online] http://www.w3.org/TR/ws-gloss/.

[3]. SOAP Version 1.2, [Online] http://www.w3.org/TR/soap12-part1/.

[4]. ApacheWeb Services Project - Apache Axis, [Online] http://ws.apache.org/axis/

[5]. UDDI Spec Technical Committee Draft, [Online]http://www.oasisopen.org/committees/uddispec/doc/spec/v3/uddi-v3.0.2-20041019.htm

[6]. G.Canfora and M. Di Penta, "Service-oriented architectures testing: A survey," in Proceedings of the 31st International Spring Seminar on Electronics Technology (ISSSE 2008), pp. 78–105, Budapest, Hungary, 2008.

[7]. C. Ma, J. Wu, T. Zhang, Y. Zhang, and X. Cai, "Testing BPEL with Stream X-Machine," in ISISE '08: Proceedings of the 2008 International Symposium on Information Science and Engineering, pp. 578–582, Shanghai, China, Dec. 2008, IEEE Computer Society.

[8]. M. Lallali, F. Zaidi, and A. Cavalli, "Timed modeling of web services composition for automatic testing," in SITIS '07: Proceedings of the 2007 International IEEE Conference on Signal-Image Technologies and Internet-Based System, pp. 417–426, Shanghai, China, Dec. 2007, IEEE Computer Society.

[9]. Shuping Ran, "A Framework for discovering web services with Desired Quality of Services Attributes", IEEE International Conference on Web Services, Las Vegas, Nevada, USA, June 2003.

[10]. V. Tosic, B. Pagurek, K. Patel, "WSOL A Language for the Formal Specification of Classes of Service for Web Services", International Conference on Web Services, Las Vegas, Nevada, USA, June 2003.

[11]. D. Pilone and N. Pitman, UML 2.0 in a Nutshell (In a Nutshell (O'Reilly)). O'Reilly Media, Inc., 2005.

[12]. K. Conroy, M. Grechanik, M. Hellige, E. Liongosari, and Q. Xie, "Automatic test generation from GUI applications for testing web services," in ICSM: Proceedings of the 23rd IEEE International Conference on Software Maintenance (ICSM 2007), pp. 345–354, Paris, France, Oct. 2007, IEEE Computer Society

[13]. M. Lallali, F. Zaidi, A. Cavalli, and I. Hwang, "Automatic timed test case generation for web services composition," in ECOWS '08: Proceedings of the 2008 6th European Conference onWeb Services, pp.53–62, Dublin, Ireland, Nov. 2008, IEEE Computer Society.

[14]. A. Paradkar, A. Sinha, C. Williams, R. Johnson, S. Outterson, C. Shriver, and C. Liang, "Automated functional conformance test generation for semantic web services," in ICWS '07: Proceedings of the 2007 IEEE International Conference on Web Services, pp. 110–117, Salt Lake City, UT, USA, July 2007.

[15]. M. Tian, A. Gramm, T. Naumowicz, H. Ritter, J. Schiller, "A Concept for QoS Integration in Web Services", 4th International Conference on Web Information Systems Engineering, Rome, Italy, December 2003

[16]. DAML-S Coalition, DAMLS-S, "Web Service Description for the Semantic Web", In Proceeding of the International Semantic Web Conference, June 2002.

[17]. Hongan Chen, Tao Yu, Kwei-Jay Lin, "QCWS: an implementation of QoS-capable multimedia web services", IEEE Fifth International Symposium on Multimedia Software Engineering, December 2003.

[18]. G.Laycock, The Theory and Practice of Specification Based Software Testing. Ph.D thesis, University of Shefield, 2003.

[19]. S. S. Hou, L. Zhang, Q. Lan, H. Mei, and J. S. Sun, "Generating effective test sequences for BPEL testing," in QSIC 2009: Proceedings of the 9th International Conference on Quality Software, Jeju, Korea, August 2009, IEEE Computer Society Press.

[20]. A. Keller and H. Ludwing, "The WSLA framework: Specifying and Monitoring Service Level Agreements for Web Services", IBM Research Report, May 2002.

[21]. Z. Guangquan, R. Mei, and Z. Jun, "A business process of web services testing method based on uml2.0 activity diagram," in IITA'07: Proceedings of the Workshop on Intelligent Information Technology Application, pp. 59–65, Nanchang, China, Dec. 2007, IEEE Computer Society.

[22]. A. T. Endo, A. S. Sim˜ao, S. R. S. Souza, and P. S. L. Souza, "Web services composition testing: A strategy based on structural testing of parallel programs," in TAIC-PART '08: Proceedings of the Testing: Academic & Industrial Conference - Practice and Research Techniques, pp. 3–12, Windsor, UK, Aug. 2008, IEEE Computer Society.