ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

Enhancing Change Management in Long Term Composed Services

M.Monicavinodhini [1], C.Kanimozhi [2]

P.G. Student, Department of Computer Engineering, Anna University BIT Campus, Tiruchirapalli, Tamil Nadu, India[1]

Assistant Professor, Department of Computer Engineering, Anna University BIT Campus, Tiruchirapalli, Tamil Nadu, India[2]

Abstract: Composite Web services are an organized structure that cumulate with all other web services and interact according to the users demand. The primary objective of this thesis is to suggest a dynamic selection for web service composition automatically, in which user requirement is used as selection criteria. Composite WS are divided into long and short term. While in long term composed WS all the components and relationship between those components are already defined. Since web services are provided by different service providers there is no guarantee for its functionality and its availability. Hence, change management is an important issue in LCS. In this paper I worked on top down changes that i.e., those changes which are initialized by the owner of a LCS. The owner of LCS will change some functionality of a composed web services in order to attract more customers. This change was taken as a challenging issue and according to the change requirement the LCS will composite itself automatically. In this thesis, web service composition for travel agency system based on functional characteristics is presented.

Keywords—Web service,top down change management component,conflict detection,change verification;

I. INTRODUCTION

Web Services will allow application to integrate on a enormous scale and in a distributed,dynamic environment.The world wide web consortium defines a web service as "a software system designed to support interoperable machine –to-machine interaction over a network".A composed web service will emphasizes compositional properties for its characterization and correct usage.It can be defined as the combination of different web services based on its functionality and demand from user.We can divide the composite web services into two terms namely short and long term.Short term services is served for short period whereas it is contrast to long term services which has no limit on time period.Changes in web services will occur due to some changes in market requirement or due to some introduction of new technology etc,changes in Web service may result in performance improvement,functionality etc.

In a Web Service model, a service provider offers Web Services which provide functions or business operations that can be deployed over the Internet, in the hope that they will be invoked by partners or customers; a Web Service requester describes requirements in order to locate service providers. Publishing, binding, and discovering Web Services are three major tasks in the model. Discovery is the process of finding Web Services provider locations which satisfy specific requirements. Web Services are useless if they cannot be discovered. So, discovery is the most important task in the Web Service model. The service providers offer Web Services which provide functions or business operations. They are created by companies or organizations. In order to be invoked, the Web Services must be described. This will facilitate discovery and composition. WSDL or service profile of semantic Web Service is used to carry out this function. The Web Service requester describes requirements in order to locate service providers. Service requesters usually contain a description of the Web Service, though it is not a Web Service which can run on the Internet. The requirements are usually described by WSDL, service template or service profile. The Web Service discovery or service registry is a broker that provides registry and search functions. The service providers advertise their service information in the discovery system. This information will be stored in the registry and will be searched when there is a request from service requester. UDDI is used as a registry standard for Web Service. The service registry contains a service description for the web service.

A composed Web service is an on-demand and self motivated association between independent Web services that collectively provide a worth service. Each independent service specializes in a core ability, which reduces outlay

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

with increased value and competence for the business entity and its clients. It emphasizes compositional properties for its characterization and correct usage. It can be defined as the combination of different Web Services based on its functionality and demand from the user. We can divide the composite Web Services into two terms namely short and long. Short term services is served for short period whereas it contrast to long term services which has no limit on time period. Changes in Web Service will occur due to some changes in market requirements or due to some introduction of new technology etc.Changes in Web Services may result in performance improvement,functionality,etc.,

In [5], a framework is proposed for detecting, propagating, and reacting bottom-up changes in an SOE. A bottom-up change starts at the implementation level of an SOE. Change may occur to an individual service without consent of the enterprises that utilize the service. The changes may affect the invocation of other services in the SOE, which affects the entire performance of the SOE. To manage these bottom-up changes, it first presents a taxonomy that classifies bottom-up changes into categories. Changes are distinguished between service level and business level: triggering changes that occurs at the service level and reactive changes that occur at the business level in response to the triggering changes. A set of mapping rules are defined between triggering changes and reactive changes. These rules are used for propagating changes. A petri-net based change model is proposed as a mechanism for automatically reacting changes. Ontologies are used for locating services from an exploratory service space. Agents are employed to assist in detecting and managing changes to the enterprises. In [11], a framework is presented to detecting and reacting to the exceptional changes that can be raised inside workflow-driven Web application is proposed. It first classifies these changes into behavioral (or user-generated), semantic (or application), and system exceptions. The behavior exceptions are driven by improper execution order of process activities. For example, the free user navigation through Web pages may result in the wrong invocation of the expired link, or double-click the link when only one click is respected. The semantic exceptions are driven by unsuccessful logical outcome of activities execution. For example, a user does not keep paying his periodic installments. The system exceptions are driven by the malfunctioning of the workflow-based Web application, such as network failures and system breakdowns. It then proposes a modeling framework that describes the structure of activities inside hypertexts of a Web application. The hypertext belonging to an activity is broken down into pages, where are univocally identified within an activity. It presents a framework to handle these changes. The framework consists of three major components: capturing model, notifying model, and handling model. The capturing model captures events and store the exceptions data in the workflow model. The notifying model propagates the occurred exceptions to the users. The handling model defines a set of recovery policy to resolve the exception. For different types of exceptions, different recovery policies will be used. In [16], a framework is presented to manage the business protocol evolution in service oriented architecture. It uses several features to handling the running instances under the old protocol. These features include impact analysis and data mining based migration analysis. The impact analysis is to analyze how protocol change impacts on the running instances. It will be used to determine whether ongoing conversations are migrateable to the new protocol or not. The data mining based migration analysis is used for cases where the regular impact analysis cannot be performed. Service interaction logs are analyzed using data mining techniques. It then uses the result of the analysis to determine whether a conversion is migrateable or not. In [16], the work mainly deals with dynamic protocol evolution. We focus on automatically modifying the composition of Web services once there is a new requirement introduced by a change.

II. CHANGE MANAGEMENT IN LCS

Changes in composite Web Services may be classified in two, top down and bottom up changes. Top Down Changes may be defined as the changes that are introduced by the owner of the composite Web Service in order to attract the demand of users and those Bottom Up may be defined as the change introduced by the Web Service Providers in order to introduce any new functionalities

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

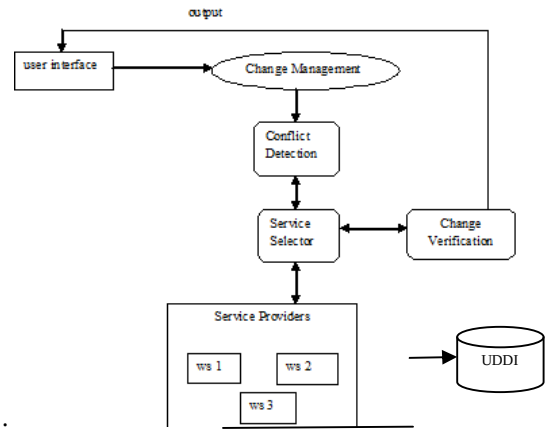

Fig 3.1: Architecture of LCS

A. PROPOSED ARCHITECTURE

In this Chapter, the proposed system architecture for web service composition is presented and shown in Fig. 3.1. The proposed architecture consists of six modules viz. Change Management module, Conflict Detection, Change Verification module, Backup service selector and Universal Description, Discovery and Integration (UDDI) module.

The Change Management module collects the workflow and the user preferences from the End-user and generates the execution plan. The execution plan contains the order of execution of web services. The Change Management module invokes three modules: viz. Conflict Detection, Change Verification, and backup service selector and gets a best selected web service for each activity in the work flow and then it invokes the selected web service.

The Change Management module selects the appropriate service that satisfies the user change from the available services. It scans the Universal Description, Discovery and Integration (UDDI) registry to select the best web service for each activity in the work flow. This module returns the set of web service that satisfies the change provided by the user

The Change Management module then invokes the Service Selector module by passing the set of web service .The Service Selector module selects one best web service that satisfies change constraints and user preference from the set of services passed by Change management module. The selected web service is then passed to the change management module for invocation. It gets the web service and passes it to the backup service selector. The backup service selector selects the best alternative web services for that web service from the UDDI registry and passes that service again to the Change management module.

The Universal Description, Discovery and Integration (UDDI) database is a collection of registered web services. The UDDI module registers the web service in the UDDI database. This module allows user to register, modify and delete the service in the UDDI database. Once the individual web service is selected for composition, the change management module will invoke the service. If the invoked service is not available at that instance, then the planner module invoke the alternate service for the composition.

The Change management module invokes the two modules for each activity in the workflow. Once this module gets the web service for all the activities in the workflow, it performs the web service composition.

B. TRAVEL SCENARIO

A typical travel scenario domain is used to express our work. It is used to demonstrate the concepts of the thesis. Consider travel agencies which provide complete travel related package such as, airline, hotel and taxi to assure the users. The owner of a travel agency will integrate these services to a long term composed service.The travel agency outsources its functionality from different types of service providers which may include airline, hotel, car, insurance and payment service. Changes from the user will be fed as an input to the first service which may be airline from whose output is given as an input to other service which may include hotel, car, and weather and at last the cost for all services will be added together and it is redirected to the user as a final output.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

III. TOP DOWN CHANGE MANAGEMENT

Top down change specification should satisfy the following properties namely:

- if a change is specified then it should not be ambiguous
- It should be in formal manner so the machine can process it easily.
- Change should be specified in a such a manner in order to ensure the sufficient details provided to implement the change.

A. CHANGE CATEGORIZATION

An important task to manage the top down changes in long term composed services is to identify the clear division of changes. hence, different procedure can be developed to deal with the change types. Here, we use change requirement as an attribute to classify changes. The change requirement will specify the purpose of introduction of a change. Based on the features of long term composed services changes can be classified using change requirement. The Functionality of a long term composed services are refer to the functional features and those which relate to the quality and perspective are classified under non functional features. Top down changes are likely to modify one or two features of long term composed services.

The changes that are required to change the functionality of a long term composed services are referred as functional changes. The type of services which are outsourced may be changed for the purpose of business regulation or for new policies. The changes may include adding functionality, removing functionality or replacing functional features.

To fulfill a functional requirement the outsourced services of the long term composed services may be changed. In order to extend its functionality a service may be added for example in travel agency, a new functionality may be added to attract more customers. Changes are reacted at two levels namely instance and schema level. To ensure the configuration the changes are then verified in long term composed services. Each change related with both functional and non functional is relay on the change requirement. Changes are classified on the basis of change taxonomy. The associated change requirement is based on the classification. In schema leveling to satisfy the functional requirement, schema is changed according to the change requirement. After change is enacted it is verified to ensure the correctness of the configuration of long term composed services.

B. CHANGE ENDORSEMENT

In this proposal, for each activity in the workflow, web service is selected based on change requirement. When a web service is selected for an activity in the workflow that may or may not executed successfully because of the uncertainty of the web service. If any one or more web services involved in the service composition are failed then there is only one option available that is to compensate all the previously completed activity in the workflow. This approach is best suited for the composition that involves minimum number of web services.

If service composition needs more number of web services to complete the job then this approach of compensating the previously completed web service will not yield good result. In this work instead of compensating the web services to achieve the required reliability, quality of service can be compromised to some extent is considered.

For example consider there is a web service composition of twenty individual web services. In this composition, if the last web service fails then compensating all the previously completed services will not be good idea. Instead of compensating all services, it is possible to provide a backup service for all web services involved in composition. So that any service fails, alternate service can be used. The alternate service selection is similar to the web service selection approach explained above.

IV. CONFLICT DETECTION

The change to process constraints is not as easy as implementing serviceable change operators. While changing the constraints may cause any impending inconsistency or redundancy.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

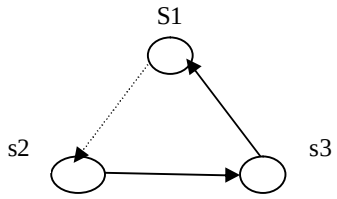

Fig5.1. Conflict detection over edges

The changes of the process constraints may cause conflicts on the invocation orders. As shown in fig 4.1[2], there are three services:s1,s2 and s3.s2 is invoked before s3 is invoked.s3 is invoked before s1.if a sequential process constraint is added to s1 and s2,where s1 is required to be invoked before s2 is invoked. This implies that s1 will be invoked before s3 is invoked. In this case, it will conflict the previous sequential invocation order between s2 and s3 and cause an invocation deadlock among these three services.

Algorithm Conflicts Detection (Change Request cr, Services S)

//Input: Change Request

//Output: Detecting Conflicts

```
for all cr!null
if (function ∣ attribute)in cr is !complete then
        if(attribute)in cr !computable then
        Discard request
        Restore status
        break
        else
                Add cr to the cs
        //cs is the Change Specification
        end if
  else
        if cr conflicts services Sr  then
  //Sr is the Service which the cr conflicts
                Alert as "Conflict Detection"
                Discard changes
                Restore State
                else
                        Add Cr to the Service
        end if
  end for
  end
```

In order to avoid above situation the service should be checked before the change is implemented, if any conflict is detected then the service should be removed and new service should be added.

V. CHANGE VERIFICATION

A. VERIFY CHANGES

Once a long term composition service is changed, it should be verified for its correctness. After a change has been implemented, the correct configuration of schema has to be maintained. If it is not in proper order then it should be corrected.

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

B. VERIFICATION ORDER

We first check the correctness of a schema in two aspects namely structural and semantic. The structural correctness refers to the incoming and outgoing edges of schema. The composition should not have any isolated services. The semantics correctness will ensure the correctness of the composition properly.

The Semantics correctness criteria will ensure that all the services in the LCS is needed for the complete business package. During the process of change verification, if an error is detected it should be modified by adding or removing any services. Once modification is carried, it should be verified for further conflict detection and redundancy detection. If a structural incorrectness is detected, the respected services will be removed from the composition. Hence the verification process should be performed carefully.

If a data transfer between the services is verified, the consistency between the invocation order and the data transfer needs to be checked. This verification is to ensure the services are composed correctly.

VI. PERFORMANCE EVALUATION

To assess the proposed methodology a web based travel scenario was developed. Performance of the composed travel services is analyzed by the following criteria:

1. Adding a Service
2. Removing a Service

The Performance is evaluated based on the response time with respect to the size of a services that are added/removed. The response time is calculated using the equation mentioned below.

$$T(\sigma)=W_n(\sigma)+S_n(\sigma) \qquad (1)$$

Where $T(\sigma)$ is the response time that are calculated

The response time which is analyzed over the existing system is tabulate and shown in table1**[**11**].**

Number of Test	Execution Time in Seconds	
	Existing System	Proposed System
1	27.33	25.44
2	23.66	22
3	17.33	16.55
4	20.16	19.75

Table 1 illustrates that execution time of proposed system is better than existing system.The pictorial representation of performance evaluation is given in Figure 3[8] below.

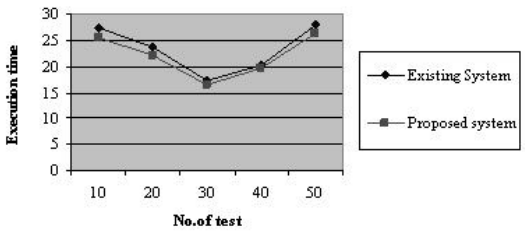

Figure 6.1: Execution Time comparison for proposed technique with existing technique

ISSN: 2319-8753

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2014

VII CONCLUSION

With the emerging role of web services in business processes, the requirement of composing and executing them have begun to draw high attention, and today the need to find the optimal web services composition for the business processes is a challenging issue. The proposed technique addresses the issues of conflict detection and check the correctness of the web service composition by selecting an optimal and reliable component web services according to the change requirement. Workflow for the web service composition is generated based on user's request and makes a better web service selection for the composition. The proposed web service selection technique selects a best component web service for the each activity in composition workflow and backup module provides a dynamic backup of a web service by one of its alternative.

REFERENCES

[1] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard.Web Services Architecture. Technical report,W3C, w.w3.org/TR/2004/NOTE-ws-arch-20040211/, February 2004.

[2] Xumin Liu, Athman Bouguettaya, Jemma Wu, and Li Zhou, "Ev-LCS: A System for the Evolution of Long-Term Composed Services", IEEE Transactions on services computing, vol. 6, No. 1, 2013

[3] C. Bussler. " The role of semantic web technology in enterprise applicatin integration.Data Engineering Bulletin", 26(4):62–68, December 2003.

[4] F. Casati, S. Ceri, B. Pernici, and G. Pozzi.Workflow evolution. Data Knowl. Eng., 24(3):211-238, 1998

[5] F. Casati, E. Shan, U. Dayal, and M-C. Shan." Business-Oriented Management of Web Services". ACM Communications, October 2003

[6] S. Abiteboul. "Querying semi-structured data", In ICDT, pages 1–18, 1997.

[7] G. Cobena, S. Abiteboul, and A. Marian. "Detecting changes in xml documents", Proceedings of the 18th International Conference on Data Engineering, pages 41-52, San Diego, USA, March 2002

[8] M. Conti, M. Kumar, S. K. Das, and B. A. Shirazi. "Quality of Service Issues in Internet Web Services", IEEE Transactions on Computers, 51(6):593 – 594, 2002.

[9] R. Joseph Manoj, Dr.A. Chandra sekhar, M. D. Anto Praveena "An Approach to Detect and Prevent Tautology Type SQL Injection in Web Service Based on XSchema validation", International Journal Of Engineering And Computer Science,Vol.1,Issue 1, Page No. 3695-3699,January 2014.

[10] F. Curbera, M. Duftler, R. Khalaf,W. Nagy, N.Mukhi, and S.Weerawarana. "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI", IEEE Internet Computing, 6(2):86–93, 2002.

[11] D. Fensel and C. Bussler, "The Web Service Modeling Framework WSMF", Electronic Commerce: Research and Applications, 2002.

[12] K. Gottschalk, S. Graham, H. Kreger, and J. Snell. "Introduction to Web Services Architecture", IBM system journal, 41(2):170–177, 2002.

[13] Indrani Balasundaram, Dr. E. Ramaraj "An Approach to Detect and Prevent SQL Injection Attacks in Database Using Web Service", IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.1, January 2011.

[14] Marco Brambilla, Stefano Ceri, Sara Comai, and Christina Tziviskou. "Exception handling in workflow-driven web applications.", WWW '05: Proceedings of the 14th international conference on World Wide Web, pages 170–179, New York, NY, USA, 2005.ACM Press.

[15] Farookh Khadeer Hussain, Elizabeth Chang, and Tharam S. Dillon. "Defining reputation in service oriented environment", AICT-ICIW '06:Proceedings of the Advanced Int'lConference on Telecommunications and Int'l Conference on Internet and Web Applications and Services, page 177, Washington, DC, USA, 2006. IEEE Computer Society.

[16] Setrag Khoshafian. "Service oriented enterprises", Auerbach Publications, Boston, MA,USA, 2006.

[17] Xumin Liu and Athman Bouguettaya. "Managing top-down changes in service-oriented enterprises", In IEEE International Conference on Web Services 2007, UTAH, USA,2007.

[18] B. Medjahed, A. Bouguettaya, and A. Elmagarmid. "Composing Web Services on the Semantic Web". The VLDB Journal, Special Issue on the Semantic Web, 12(4), November 2003.

[19] Seung Hwan Ryu, Fabio Casati, Halvard Skogsrud, Boualem Benatallah, and Régis Saint-Paul. Supporting the dynamic evolution of web service protocols in serviceoriented architectures. ACM Trans. Web, 2(2):1–46, 2008.

BIOGRAPHY

Monicavinodhini.M, received her BE degree from Anjalai Ammal Mahalingam Engineering College affiliated to Anna University,Chennai,India. and Pursuing M.E(SE) degree from Anna University BIT Campus Trichy, India.Her research area includes web services ,Data Mining.