

Simplified Syntactic Web Service Composition for flaw less propagation

Uday Pratap Singh

Student, KNIT Sultanpur, India

Abhay Agrawal

Assistant Professor, KNIT Sultanpur, India

Abstract - Until the end of twentieth century the only repository of information considered was World Wide Web. But recently Web services due to their potential in facilitating seamless machine to machine or enterprise application integration are dominating the current web scenario. Web service standardization has significantly contributed towards improving functionality of Service Oriented Architecture (SOA). Web services have provided an independent and autonomous platform to perform a specific task. When a single Web service is unable to fulfill the given request, the solution is provided by Web service composition. However there are a number of unresolved issues for which a few solutions have been proposed already for the execution of web services composition. The paper addresses the problem of web service composition by providing a data flow between web services by taking the output of one web service as the input to another web service. The paper addresses the method to overcome the limitations of UDDI by introducing a substitute named portal. Here we use a target oriented XML based language for establishing the coordination and composition between the set of services to facilitate the reuse of existing services. Further efforts are made to make the composition platform independent by using different platforms like java and .net.

Keywords – Web service, Web service composition, WSDL UDDI

I. INTRODUCTION

Web services are software components developed to simplify interaction between machines over the Web. This value-added service and process to integrate various pre-existing service is referred as composite Web service and Web service composition respectively. Web service composition is facing various challenges in different directions. The most common problem is to automate the process of composition. This is the most challenging problem to analyze the web services required using artificial planning. The description rich languages like OWL (Web Ontology Languages) belonging to semantic web is also gaining attention. However WSDL (Web Service Description Language) is widely accepted to provide an abstract definition of the syntactic description of the web services.

In this paper we propose a method to manually register the web services in a portal which is managed by an administrator. The administrator is allowed access by an authentication procedure. The controller can select the web services from the available list after testing them. By integrating these web services we perform the composition by passing input and output parameters and using only the WSDL description to provide a peer-to-peer interaction among the web services involved in the composition.

It follows the composition rules facilitating the reuse mechanism of web service for new applications. We face the issue of managing the XML message at various levels. This issue is dealt by selecting and interconnecting the selected web services which decide the control flow or data flow required achieving the goal. We have reduced our search space of web services required by only providing the best solution thus eliminating the need for any search algorithm. Here we provide a demonstration for the chosen problem domain where on providing the symptoms as input we get the disease, medicine and the price of the medicine as output. Furthermore the web application also traces the IP address of user thus provide the price in the currency of the country of the user. The remainder of the paper is organized as follows. Section 2 represents the related work. Proposed work is elaborated in section 3 followed by conclusion and future work in section 4.

II. RELATED WORK

As the concept of web service is emerging, researchers are putting a lot of effort in composing them so as to make more complex services. Various researches have been made in this direction and problems faced during composition have been discussed, for example in [1, 2, 12]. The aim is to find the best-fit web service. Various semantic approaches are explored as in [5] for matching of web services and enabling to choose the most desirable web service. These days efforts are being made to make a high degree automated process by using AI planning. A Hierarchical Task Network (HTN) planner SHOP2 is especially well-suited for this task for process model like DAML-S or with OWL-S Web Service descriptions as discussed in [3,4].

While concentrating on the composition process methodologies are being discovered to maintain the reusability property of the web services. Reusability saves our need to rewrite a code again and again when required at several places in any web application. The process of abstraction is used to promote the reusability, as explored in [6].

Other approaches for semantic, syntactic and automated web service composition has been explored in [7, 8, 9]. semantic web service composition is ontology based while the syntactic web service composition is XML based.

Since the requirement of users is increasing day by day, the need for automatic selection of web service is increasing. Recent work is done in the direction to make the composition dynamic, for example in [11, 12].

III. PROPOSED WORK

In this section strategy followed is discussed. The earlier automatic web service composition used are time consuming and expensive. So as a solution to this we use Syntactic web service composition. We have also tried to overcome the shortcomings of UDDI. The public Web Service Registry implementation like UDDI have several issues like centralized architecture, complex APIs for registration and deregistration processes, limited accuracy of registered services and lack of management, monitoring and governance of registered entries. The proposed model consists of three components – Service Providers that develops and publishes web services, Service Consumers that uses these web services to develop Web Applications, and Service registry (instead of UDDI we are using our own portal) where these web services are registered. The proposed model is a light-weight service registration based on manual registration and WSDL analyzer. In manual registration Service Providers are provided a form to register Web services.

The portal has been a great advantage to the controller service also. It enables him to see the list of web services available as shown in Fig.1.

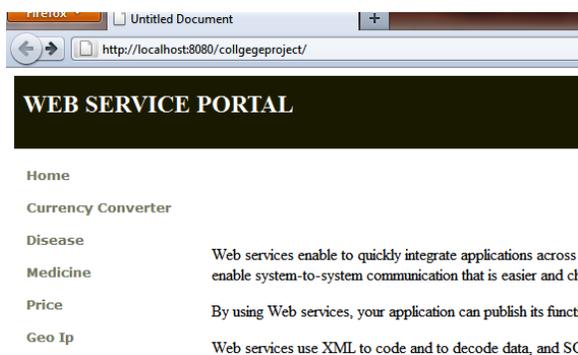


Figure 1. List of available web services in Portal

Another advantage of portal is that it enables the controller service to test the web services as shown in fig. 2, before he can integrate them to perform the composition.

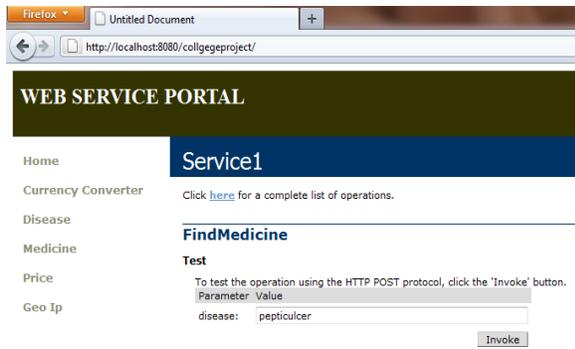


Figure 2. Demonstration and Testing of Web services on Portal

In the implementation only an abstract view is created for the user through which it is able to get an input which generates a required output without giving the user the knowledge of the web services used to get the output. The composition has used two different work flow patterns like sequential and parallel as shown in fig.3.

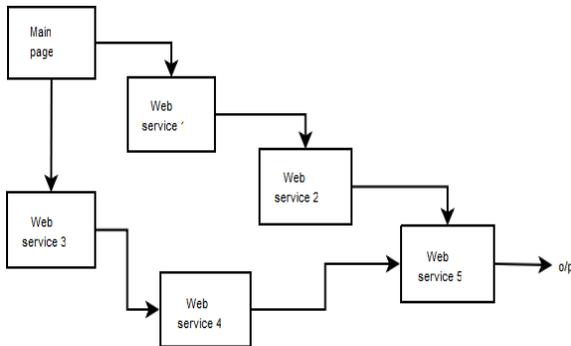


Figure 3. Work Flow Pattern of Web Services

In the demonstration we have tried to make our composition platform independent by using java and .net as platforms. Fig.4 demonstrates the frontend web application called “Medicinal Shop” visible to the user, where he specifies three symptoms as input to the web service named “Disease” designed using java as a platform. Also the IP address of the user is traced and passed as an input automatically to another web service named “Geographiclocation” designed using java as a platform. Here the work flow pattern is parallel where two input parameters from one web application pass to two web services simultaneously.

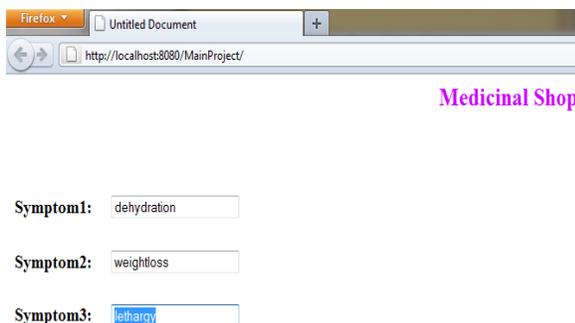


Figure 4. Front-end Application Medicinal Shop

Now the web application “Medicinal shop” passes automatically symptoms as input to the web service “Disease” which analyze these inputs to predict the disease as output as shown in fig.5.

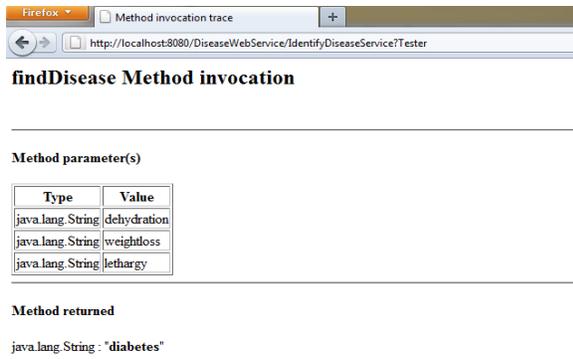


Figure 5. Output of testing of Disease Web Service

This output further act as input for the web service “Medicine”, designed using .net as a platform, which take disease as input and predict the specified medicine as output. Meanwhile the IP address from the main web application is passed as input to the web service named “Geographiclocation” which identifies the country of the user according to its IP address. The country name serves as input to the web service named “Currency Converter” which according to the country finds its conversion rate. Now the output from two web services “medicine” and “currency converter” are passed as input to the third web service named “ price” designed using java as a platform, which gives the price of the medicine in the currency of the user

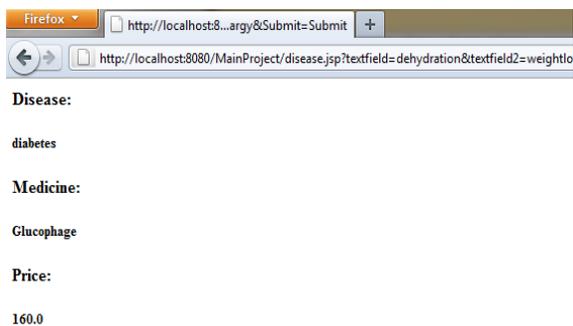


Figure 6. Final Output of Composition

Thus we get disease, medicine and price in the currency of the user as final output shown in Fig.6.

IV. CONCLUSION AND FUTURE WORK

We have been successful to create a demonstration for syntactic web service composition. The workflow patterns like parallel and sequential along with WSDL description help in generating the composition. Here the output from web service acts as an input for another web service thus creating a view for the syntactic web service composition.

The composition thus achieved is a platform independent allowing to work on web services on two different platform named java and .net. Also the portal works as a substitute of UDDI overcoming its shortcomings.

Further we are planning to make our composition more mobile by using other platforms like GSoap, Linux following the composition rules. We are also trying to further reduce its time complexity.

REFERENCES

- [1] Rajendra Pandey, Shrabani Mallick, Sanjeev Neupane et.al, A Simplified Web Service Discovery & Composition Mechanism. In Int. J. on Recent Trends in Engineering & Technology, Vol. 05, No. 01, Mar 2011.
- [2] Ruey-Kai Sheu, Win-Tsung Lo, Chia-Feng Lin, Shyan-Ming Yuan, Design and Implementation of A Relaxable Web Service Composition System. 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery
- [3] Evren Sirin, Bijan Parsia, Dan Wu, James Hendler, Dana Nau, HTN Planning for Web Service Composition Using SHOP2. In June 2004.

- [4] Dan Wu, Bijan Parsia, Evren Sirin, James Hendler, and Dana Nau, Automating DAML-S Web Services Composition Using SHOP2. In 2004.
- [5] Aviv Segev, Eran Toch, Context-Based Matching and Ranking of Web Services for Composition. In 2009.
- [6] Carlos Granell, Michael Gould, Roy Grønmo, David Skogan, Improving Reuse of Web Service Compositions. To appear in EC-Web 2005. Copenhagen (Denmark), August 2005.
- [7] Ken Pu, Vagelis Hristidis, Nick Koudas, A Syntactic Rule Based Approach To Web Service Composition.
- [8] Freddy Lecue, Samir Salibi, Philippe Bron, Aurélien Moreau, Semantic and Syntactic Data Flow in Web Service Composition, (2006).
- [9] Jinghai Rao and Xiaomeng Su, A Survey of Automated Web Service Composition Methods (2004).
- [10] Chan, P.P.W.; Lyu, M.R., Dynamic Web Service Composition: A New Approach in Building Reliable Web Service. March 2008.
- [11] Mustafa, Faisal and McCluskey, T.L., Dynamic web services composition: current issues (2008).
- [12] Biplav Srivastava, Jana Koehler, Web Service Composition - Current Solutions and Open Problems, [2004]