IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

User Preferences Based Automatic Web Service Composition

K.Purshothaman[1], S.K.Ram Prasath[2], Ramya Ramprasad[3]

[1, 2, 3]**Department of Information Technology, Anand Institute of Higher Technology, Chennai, Tamil Nadu, India**

Abstract

With the growth of internet in the modern scenario, the major components of internet needs to be more and more user friendly in order to occupy a permanent position in the web market. This is possible by the effective use of web services and the manner in which these web services are provided to the user. For this, having web service composition as the basis, the user preferences play an important role in the composition process. With the effective use of user request as input, all possible web services among a wide range of services can be composed by using specific structural conditions. The existing sequential structure of web service composition process have few drawbacks with that of the proposed conditional and parallel branch structure with respect to meeting the user preferences. With the help of branch and parallel structure for web service composition, the user is satisfied with the quality of service that meets his/her preferences. Since there are only limited constraints regarding web service composition and greater necessity in the field of healthcare, we prefer to implement this concept in healthcare services.

Keywords: *Web services, User Preferences, Automatic, Conditional Structure*

1. Introduction

Over the last two decades the use of computers in peoples' daily life was unmatched. The inventions of mankind lead to a future where a person doesn't work physically to complete a task, but mentally. The one who invented computer would have never been aware of its future. Meanwhile, computers found their future in internet. Internet, serving the people worldwide to connect and share has led to a new era where the entire universe was brought into human hands. Internet that provides everything on a key press has web services as its backbone in order for serving people. The web services are based on a small concept but with large, distinct algorithms that helps in communicating. The user, client machine residing on one side gains information from the server at the other end by communicating to it with the help of HTTP, HTML, SOAP, XML, so and so. As of everything performed by a software package in a computer system, web services serve everything over the internet. A simple web service is nothing but a set of codes written to perform an action. A website built to serve people has numerous web services enclosed in it to perform different tasks. But this would be ambiguous when people have to use their entire knowledge in selecting the required service from the whole bunch. Hence the concept of web service composition was introduced to compose only the required web services as requested by the user and was provided as a single service. Web service composition could be either automatic or semiautomatic. An automatic service composition plays a major role in simplifying the task further. Hence in this paper, the concept of automatic web service composition that is wholly based on the user preferences was used to build an algorithm and its future implementation was started to make it possible.

2. Related Work

The web service composition concept was already in practice in various domains and is functioning to its best in people oriented domains like shopping, e-services, healthcare etc. Yet those services failed to attain complete success on meeting the user preferences at ease. Consider a shopping scenario where the user enters a keyword to search for a product and filter the product based on his preferences and selects a product, then opt for next service i.e. payment process and the he chooses the mode of payment, finally finishes payment and request for receipt of payment and exits. In this process each and every service is executed only on command of the user as per his wish. This might be helpful in processes like payment since one must be careful with what he pays for the product but in case of processes where there is no need of diverse user interaction with the services could implement automatic service composition. Take an example of healthcare service system where the user who specifies his input and preferences must be provided with entire information contained in various services composed together to get a final resulting service. To make the health services

domain effective, we have considered implementing our concept over that platform.

Some of the earlier health systems that served people in fulfilling their health related problems are stated as follows:

Google Health-Personal health information centralized service by Google in 2008. It contains Google Health Profile with user health records.

Health vault-A service by Microsoft started in 2007. Here, users can collect store and share health information. It also provides user to choose third party applications to access health information.

Dossia- It was started by a group of employees in US. It is based on open source software. Here users can access and download health records.

SMART- It was started by UK Engineering and Physical Science Research Council. This system is limited to three specific diseases: Chronic pain, Stroke, Congestive heart failure.

Though these services were started mainly to fulfil user needs for that particular period and by the rise of the need they lack in fulfilling current generation user requirements. Some of the major drawbacks of such services are-

- Some projects like Google Health, Dossia are limited to people of particular country.
- Existing services are limited to some diseases & activities of users.
- Contains only limited information of all the ailments.
- Third party restriction in joining the service system.

There are also few works that implemented the service composition concept-

Berardi et al. [1][2][3] present a formal framework where services are characterized in terms of their behavioural descriptions and then provide techniques for computing service composition in the case where the behavioural description is expressed as finite state machines. Their basic idea consists in reducing the problem of composition into satisfiability of a suitable formula of deterministic propositional dynamic logic.

McIlraith et al. [5] adopt and extend the Golog language for automatic construction of Web services. They address the Web service composition (WSC) problem through the provision of high-level generic procedures and by customizing users' constraints.

Medjahed et al. [14] present a technique to generate composite services from high-level declarative description. They define formal safeguards for meaningful composition through the use of composability rules.

Sirin et al. [15][16] present a semiautomatic method for WSC. Each time a user has to select a service, all possible services, which match with the selected service, are presented to the user.

Most existing methods for automatic service composition are AI planning based, and the resulting composite services are of sequential structures only. In such models, component services are composed together in a 1-D linear fashion. No matter how diverse the application needs are and how the environment changes, it calls services in accordance with an established order. Hence, they are applicable to only simple cases with a deterministic environment, and cannot meet the diverse needs of users and adapt to the dynamically changing environment. In addition, user preferences are very important for service composition but largely ignored by these existing methods. In particular, in the field of medical and health care, user and domain preferences are very common and have a significant impact on service composition.

In Wang et al. [4][6], the branch structures appearing in the process model of a composite service and have presented the corresponding solutions for two simple types of user preferences.

So here we adopt the idea of SOA and Web service technologies to design and implement the platform. Also several key techniques for service composition are integrated in the platform, which include WSC techniques supporting branch structures and parallel structures. WSC techniques supporting parallel structures are completely new.

3. System Architecture

Based on the proposed system that is derived out of existing systems with further improvements in the composition techniques, the system architecture (Fig. 1) that explains how the system works in a nutshell.

Here is how the system architecture is described:

Obtain user request as input to the system. Also create individual web services depending on the executing platform.

A user request contains the Input from the user, Expected output by the user, Preferences of the user and quality of system expected by the user.

Based on these inputs the web services are tested to make sure that they meet the user expectation, preferences and quality.

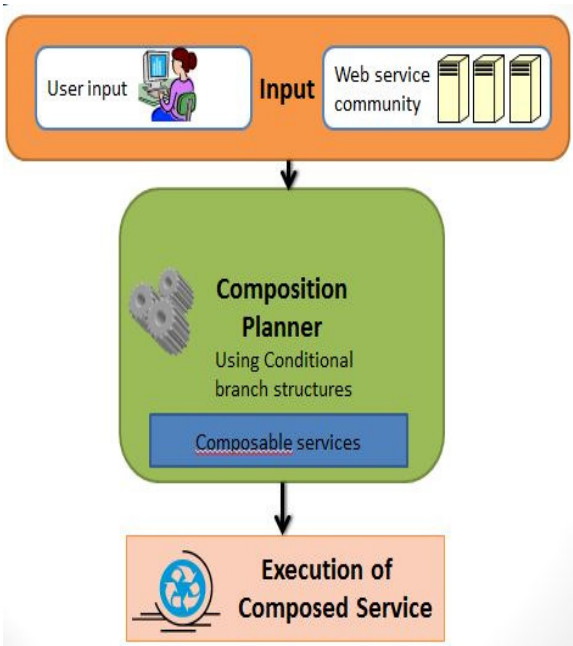

Fig. 1 System Architecture

The above process can be performed by using conditional branch and parallel structures.

Finally, an activity diagram showing how the services should be composed based on user preference is generated which was implemented to obtain the resultant service output.

4. Preliminaries

This part includes several key techniques for service composition.

4.1 Basic concepts

A Web Service can be represented by the following,

- Input parameters
- Output parameters
- Quality parameters

The user request can be represented by the following,

- Input provided by the user
- Output expected by the user
- Preferences of the user
- Quality parameters specified by user

✓ There are many semantic models also called service ontologies(a software system or set of functions) are in practice in the medical field to follow a particular standard for input/output of the services in web service composition.

✓ SNOMED CT: Among the many medical terminologies and classification systems exist the highly recommended is SNOMED CT due to the vast usage in different medical services.

✓ To extend our service usage with the third party service providers who develop with different ontologies, the different services can be merged by using the technique called ontology mapping.

4.2 Activity Diagram

UML Activity diagrams are used to show dependencies and control flow among various services. Here the actions are denoted by round rectangles and diamonds are used for decision and merge operation.

4.3 User preference

The main component considered in this is the user preferences where the user has to specify the properties of solution to choose a desirable service from a group of services. But user preferences can harm the composition process hence preferences over services can be considered than preferences over the solutions.
The three types of user preferences:

1) ccondition?WS1 : WS2;
Here if condition is true select ws1, else select ws2
2) switch (condition){case C1 : WS1; case C2 : WS2; . . . ; case Cn : WSn; };

Here according to the condition a service is selected among n number of services.

3) WS1 >> WS2 >> . . . >> WSn

- Here the services are assigned with priorities such as ws1 with highest priority and then ws2 and so on.
- A service that is selected from a group of services on the basis is that it must accept the user input, its quality must be to the level of user expectation, and the final composite must reflect user preference.
- Such a way the services are selected and merged from the web service community.

5. Key Techniques

The key technique involved is the conditional branch structure used in filtering the web services. The conditional branch structure plays the main role in making decisions based on a condition, this or that. With this structure, using if and else conditions the web

services are compared with the input, preference, expected output and the right web service is selected to be composed with other such services. For this all the tasks to be performed are made as web services and the web service community is formed from which each service is sent through the conditional structure and corresponding state of the service is saved on every pass so that it can be sent to another branch using the saved state. The web services here are considered to be the nodes of an activity diagram. After every node passes the condition, the nodes are properly connected with previous node and the process goes on. At the end of all passes we get all the nodes composed depending on the user preferences.

An algorithm that explains the basic conditional structure used in selecting web services is provided below:

Algorithm: Composition Planner

Step 1: Create a web service community

Step 2: Group the services that help in input gathering

Step 3: Invoke the web services sequentially or in parallel to obtain inputs from user

Step 4: Save the states of all services as each finishes invoking

Step 5: Gather all the states (output) of services to form a final state

Step 6: Use the final state to identify the primary service that matches expected output completely

Step 7: Using the input values and primary service characteristics, calculate domain preference

Step 8: Based on the preference, do conditional branching and flow through any one direction

Step 9: Invoke all the services required for the corresponding preference

Step 10: Combine their states (output) to obtain final state

Step 11: Provide the resultant state as a service to the user

---

The above algorithm is only structural and explains only the basic concept behind the composition and it does not give total information about how it is composed using the conditional structures. The semantic matching of web services among the WSC also plays a major role in filtering out the necessary web services from others. It actually works by matching semantic terms of the web services with the user's input. The next process involves the conditional structured filtering of web services followed by calculation of domain preference which plays a major role in the field of healthcare by providing the health condition of the user by warning along with the immediate action to be taken for the problem. This case differs the healthcare platform from others that use the composition technique.

6. Implementation

Using Service Oriented Architecture as the base for composing web services we have started our implementation in .NET platform that was found to be a great tool for effective design and implementation of web service related works. The first phase of our implementation was the requirement analysis where the day-to-day user requirements are identified and listed. Then going through the design and coding where the design is based on the above specified system architecture. The coding part is the detailed work flow of the algorithm and it is carried out in the health services domain. Hence the result will be a full-fledged service that well serves the people about their requirements in the field of Medicare. The outcome of this concept will be of providing required information for people meanwhile warning them about their health condition using their input to the web services.

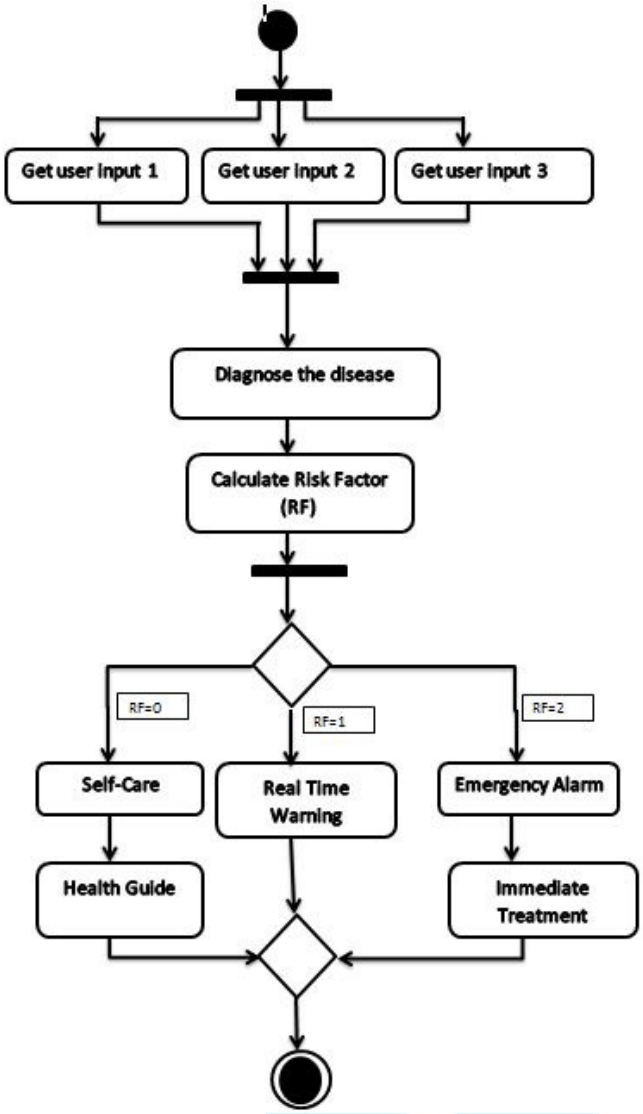

Fig. 2 Implementation Process

In Fig. 2, The implementation process was explained in a nutshell based on the Algorithm specified above. The final product will be the developed system of this implementation work.

7. Conclusion and Future Work

With the help of information and communication technologies, especially the service-oriented technologies, we have designed and implemented the health care platform based on SOA and Web service technologies, which makes our developed system have higher reusability, flexibility, and extensibility. In particular, in order to realize the personalized customization and active recommendation of health care services for individuals and to provide them with intelligent health information management, several key techniques for service composition are integrated in the platform. Their use can significantly help meet the users' diverse and personalized needs in changing application environments. Implementing the same concepts in depth of the platform will be more useful to get a well sustained healthcare system. Further by using other types of composition techniques combined with this concept may help in achieving more user friendly system.

References

[1]D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella,
"Automatic service composition based on behavioral descriptions," Int. J.
Coop. Inf. Syst., vol. 14, no. 4, pp. 333–376, 2005.

[2]D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella,"Automatic composition of e-services that export their behavior," in Proc.ICSOC, 2003, pp. 43–58.

[3]D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Synthesis of underspecified composite e-services based on automated reasoning," in Proc. ICSOC, 2004, pp. 105–114.

[4]P. W. Wang, Z. J. Ding, C. J. Jiang, and M. C. Zhou, "Automated Web
service composition supporting conditional branch structures," Enterp.
Inf. Syst., to be published. [Online]. Available: http://dx.doi.org/10.1080/
17517575.2011.584132

[5]S. Narayanan and S. McIlraith, "Simulation, verification and automated
composition of Web service," in Proc. WWW, 2002, pp. 77–88.

[6] P. W. Wang, Z. J. Ding, C. J. Jiang, and M. C. Zhou, "Web service composition techniques in a health care service platform," in Proc. ICWS,
Jul. 2011, pp. 355–362.

[7]S. C. Oh, D. Lee, and S. R. T. Kumara, "Effective Web service composition
in diverse and large-scale service networks," IEEE Trans. Serv.Comput., vol. 1, no. 1, pp. 15–32, Jan.–Mar. 2008.

[8] S. R. Ponnekanti and A. Fox, "SWORD: A developer toolkit for
Web service composition," in Proc. WWW, Honolulu, HI, May 2002,
pp. 83–107.

[9] S. V. Hashemian and F. Mavaddat, "A graph-based approach
to Web services composition," in Proc. SAINT, Feb. 2005,
pp. 183–189.

[10]Q. A. Lang and S. Y. W. Su, "AND/OR graph and search algorithm for
discovering compositeWeb services," Int. J. Web. Serv. Res., vol. 2, no. 4,
pp. 48–67, Oct. 2005.

[11] S. C. Oh, B. W. On, E. J. Larson, and D. Lee, "BF∗: Web services
discovery and composition as graph search problem," in Proc. EEE, Mar.
2005, pp. 784–786.

[12] S. C. Oh, D. Lee, and S. R. T. Kumara, "Web service planner (WSPR):
An effective and scalableWeb service composition algorithm," Int. J. Web
Serv. Res., vol. 4, no. 1, pp. 1–23, 2007.

[13] J. Rao, P. Küngas, and M. Matskin, "Composition of semantic Web
services using linear logic theorem proving," Inf. Syst., vol. 31, no. 4/5,
pp. 340–360, Jun. 2006.

[14]B. Medjahed, A. Bouguettaya, and A. K. Elmagarmid, "Composing Web services on the semantic Web," VLDB J., vol. 12, no. 4, pp. 333–351, Nov. 2003.

[15]E. Sirin, J. Hendler, and B. Parsia, "Semi-automatic composition of
Web services using semantic descriptions," in Proc. Web Services:
Model., Arch. Infrastruct. Workshop ICEIS, Angers, France, Apr. 2003,
pp. 17–24.

[16]E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, "HTN planning for Web Service composition using SHOP2," J. Web Semantics, vol. 1, no. 4, pp. 377–396, Oct. 2004.

[17]D. Basin, Y. Deville, P. Flener, A. Hamfelt, and J. F. Nilsson, "Synthesis of programs in computational logic," in Program Development in Computational Logic, vol. 3049, LNCS, M. Bruynooghe and K.-K. Lau, Eds.New York: Springer-Verlag, 2004, pp. 30–65.

[18]Y. Liu, A. H. Ngu, and L. Z. Zeng, "QoS computation and policing
in dynamic Web service selection," in Proc. WWW, May 2004,
pp. 66–73.

[19]W3C Recommendation, Semantic Annotations for WSDL and XMLSchema (SAWSDL), Aug. 28, 2007. [Online]. Available: http://www.w3. org/TR/sawsdl/