A Prototype for Composing SOAP, RESTful and Mobile Web Services

P.Thangaraj[1], P.Balasubramanie[2], D.Jugisathvi[3]
S.A.Bhuvanasoundarya[4]
1 Assistant Professor, Kongu Engineering College and thangarajtkvalasu@gmail.com
2 Professor, Kongu Engineering College and p_balu@kongu.ac.in
3 PG Scholar, Kongu Engineering College and jugisathvidc@gmail.com
4 PG Scholar, Kongu Engineering College and bhuvanasoundarya004@gmail.com

ABSTRACT

Mobile web services play a major role in recent years due to its flexibility in access at different location and sites. A method is required for retrieval of appropriate web services using mobile web services from various sources such as Simple Object Access Protocol (SOAP) services, Representation State Transfer service (RESTful) and Mobile Web Services. The main drawbacks faced by mobile web service composition are
i) Communication media used by the mobile device to connect internet ii) operating system used in the mobile device iii) protocol used for communication.

Considering the above issues a mechanism using Quality of Service (QoS) based approach is required for this type of service composition. The challenges are the availability of heterogeneous web services, android application IOS application and other services offering the same service. Due to the vast availability, certain factors are considered in selecting the suitable service amongst all available services. An approach using composite web service composition is proposed in this paper with quality constraints to satisfy the needs of the user request.

Key Words: Service Composition, protocol, Quality of Service, web service.

1. INTRODUCTION

The trend in developing a mobile web service application is in a larger extent due to the use of mobile devices for each and every need of the service by the user. There is a rapid growth in the use of mobile application by the user. So there is in need of a mechanism to integrate SOAP, RESTful and mobile web services through a common platform to provide an appropriate service for the user. Based on service oriented architecture techniques, a composite web service composition method is to be proposed. The input parameter is to be designed in a general manner based on the needs of the user and availability of the similar services on demand. The output is the exact service that fulfils the need of the user. A quality constraint is to be enforced to leverage appropriate service for the user. There are variety of approaches proposed by different authors using heuristics and non-heuristics approach but still there exist a drawback in providing appropriate service to the user due to the availability of plenty of service for the same request. The communication medium used such as air or wired also influences the rate of transmission of data. The protocol such as SOAP, HTTP and others used for interpreting the data also causes delay in selecting the service. The device used for making the request varies in wide rage as Apple iphone, Microsoft, Google and others which understands and exchanges data in different forms. To resolve all these issues a method is to be adopted for providing an efficient transfer of service to the end user without delay and effectively by satisfying the consumers' needs in all aspects.

2. RELATED WORK

The Combination of multiple web services into a composite service is more beneficial to users, instead of finding a complex atomic service that satisfy a special request [17], [28], [27]. The resulted composite services can be used by themselves or in other service compositions to satisfy client requests. BPEL4WS [6] provides the combination of block-structured and graph-structured process models and variables in relevance with the message types that can specified as input or output variables to request or invoke, receive and reply web services. BPEL has

extended to support simulation of composition of heterogeneous web services, such as RESTful and OSGi services. REST [29] is the software architecture to distribute web applications as hypermedia systems such as the World Wide Web. REST defines a set of architectural principles [29] with GET and POST method by which the design of web services focuses on a system's resources, including how resource states are received and transferred over HTTP by all the client applications written in different languages. In [8], F. Curbera et al. define a model for composing RESTful service using service-composite model. The messages returned from a service are stored as BPEL process variables during the invocation of RESTful web service. However, the transformation of the messages from different content types for follow-up service invocations is unclear.

In [11], S. Farokhi. et al. proposed a framework to support dynamic composition for both SOAP-based and RESTful Web services simultaneously. He also discussed the composite services with three different views such as data, process, and component view. However, the message transformation between the SOAP and RESTful is not discussed as in other approaches.

In [33], J. Nitzsche et al. composed a method that extends BPEL 2.0 with WSDL-less interaction model by coupling business logic and Web service technology including WSDL. A new single type of interaction activity resuming all BPEL interaction activities is introduced. This paper discusses about enabling modeling processes or process fragments that can be reused and bound to specific service interfaces in any interface description language. The approach in this paper focuses more on how to design and implement the binding relationships among various heterogeneous web services.

In [31][32], C. Pautasso described a process-based composition language for composing RESTful and traditional WSDL-based services based on BPEL. The local adapters such as XSLT and JavaScript are used to accept, process the data and transform it in to a compatible format so that the other service requests can access. In the proposed approach SOAP, RESTful and OSGi services are supported for invocation of a service by executing a composite process. The entire service request message returned in the form of SOAP, JSON, JQUERY, YAML, Protocol Buffer and Java object is transformed into Javaobjects and these objects are converted and used as variables of a BPEL process. The subsequent service invocations or followup services can use the same variables of different content type can be used after the first transformation.

3. WEB SERVICE COMPOSITION TECHNIQUES

The different web service composition techniques available are discussed with respect to their drawbacks.

3.1. SOAP Web Service Composition

The web services are registered in public repository. The service request is made using URI (uniform Resource Identifier) or WSDL (Web Service Description Language) and are retrieved with the help of UDDI (Universal Description Discovery and Integration) using the protocol SOAP[22]. This suffers with the lack of locating the appropriate service due to the availability of numerous services offering the same users request. When the request is enabled through the mobile device the input parameters are transformed in to BPEL [27] variables and are processed to retrieve the response. Once a response is retrieved it can be maintained and can be served for the subsequent requests. The retrieved service is an atomic service to the user but the time, availability, reliability and interoperability are the factors to be considered during retrieval process based on the protocol such as SOAP and thus the methodology is proposed.

3.2. RESTful Service Composition

Most of the web services which are owned by a single corporate are developed using RESTful service [15]. This uses the protocol HTTP to transfer the message across message exchange. This offers the methods as GET and POST to receive and send messages respectively. Even though it uses only XML, other design tools, javascript and other scripting languages it is heavier to parse when it involves large amount of images and Activex controls [24]. The parsing is done majorly on server side and thus it leads to time delay in processing the request and delivering the service to the user.

3.2. Mobile Web Service Composition

In Mobile service composition each application is created with src, lib, res and manifest file. When a mobile app is deployed it

integrates source code, resource file, manifest with importing include files and modifies the flexi files to deploy the new android application on OSGi Extensions framework [30][32]. The application is add to the OSGi bundle to generate manifest and .dex file to execute as OSGi application.

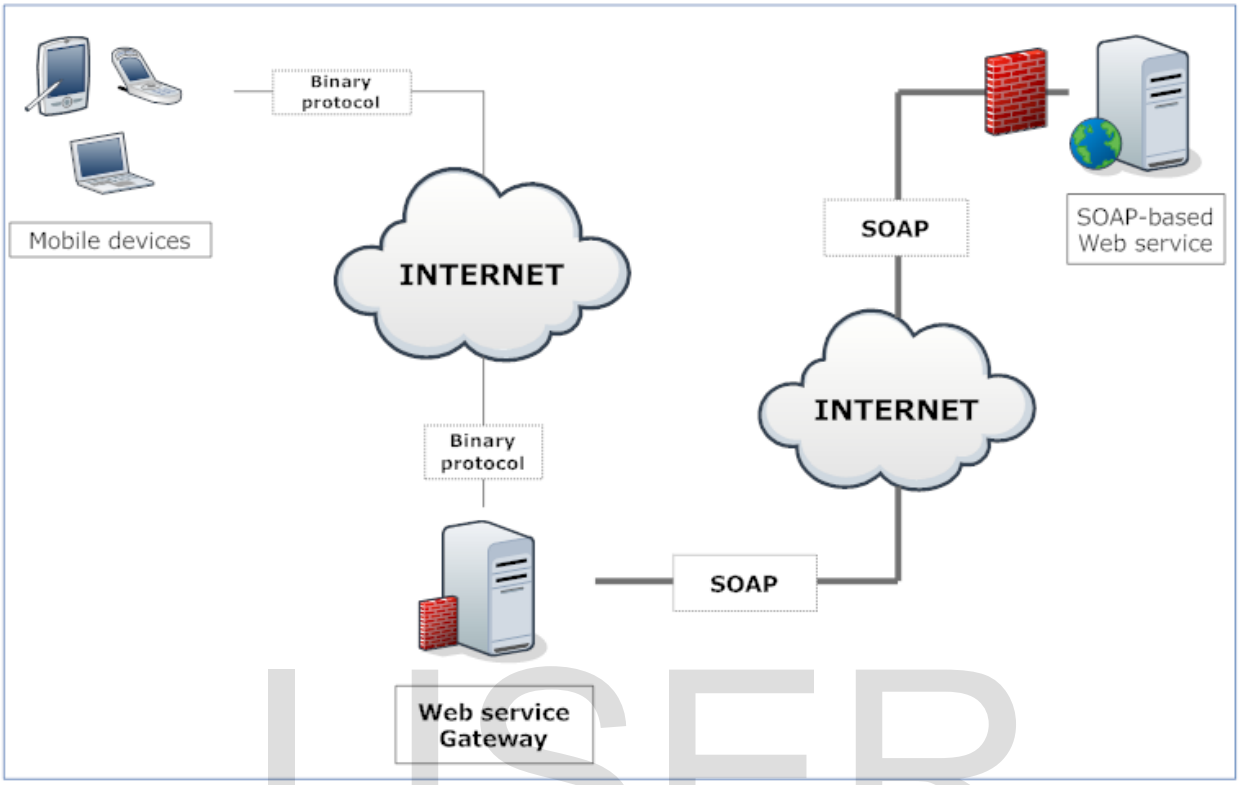

Fig:3.1 Communication Between Mobile, SOAP and RESTful Web Services

3.3 PROPOSED METHODOLOGY

The methodology proposed uses a composite service composition mechanism which integrates the three different types of services in to single framework [12]. The SOAP, RESTful and OSGi web services[25] uses its own composition engine to locate the service requested by the consumer. The request is received by the framework and based on the availability and response time and the end user needs the service is selected and delivered to the user by this frame work.

The development environment includes eclipse using ADT bundle with java [25]. The developed service is tested on the emulator by giving minimal service request with fewer parameters.

3.3 CONCLUSION AND FUTURE WORK

The proposed method gives an significant improvement in composing a web service with SOAP, RESTful and Mobile web services. The QoS constraint considered are response time and availability of resources for only limited number of sites.

The future research may focus on implementation on a real time environment with inclusion of other QoS Attributes such as responsiveness, reliability, interoperability.

REFERENCES

[1] Itinerary-based Mobile Agent as a Basis for Distributed OSGi Services, J. Lee, S.-J. Lee, H.-M. Chen, and K.-H. Hsu., IEEE Transactions on Computers, 62(10):1988-2000. 2013

[2] Service-oriented architecture for mobile applications, Y. Natchetoi, V. Kaufman, and A. Shapiro, Proceedings of the 1st international workshop on Software architectures and mobility. ACM, 2008.

[3] Design and implementation of smart home energy management systems based on zigbee, D.-M. Han and J. Lim., IEEE Transactionson Consumer Electronics, 56(3): 1417-1425, 2010.

[4] Smart home mobile RFID-based Internet-of-Things systems and services, M. Darianian, and M. P. Michael., Advanced Computer Theory and Engineering, 2008. IEEE International Conference on ICACTE, 2008.

[5] The internet of things: A survey, L. Atzori, A. Iera, and G. Morabito., Computer Networks, 54(15): 2787-2805, 2010.

[6] Business Process Execution Language for Web Service (BPEL4WS) 1.1. Technical Report, T. Andrews, F. C, et al.., BEA Systems and International Business Machines Corporation and Microsoft Corporation and SAP AG and Siebel Systems, 2003.

[7] OWL-S: Semantic Markup for Web Services, W3C Member Submission, http://www.w3.org/Submission/OWL-S/. 2004.

[8] Workflow composition for the web. F. Curbera, M. Duftler, R. Khalaf, and D. Lovell. Bite, ICSOC (2007) 94–106.

[9] Enhancing Residential Gateways:OSGi Service Composition, R. P. Diaz Redondo, A. Fernandez Vilas, M. Ramos Cabrer, J. J.Pazos Arias and M. R. Lopez. IEEE Transactions on Consumer Electronics, Vol 53, No 1, February 2007, 87-95.

[10] Developing web services choreography standards V the case of REST vs.SOAP, M. zur Muehlen, J.V. Nickerson,and K.D. Swenson, Decision Support System 37 (2004)

[11] MDCHeS:Model-Driven Dynamic Composition of Heterogeneous Service, S. Farokhi, A. Ghaffari, H. Haghighi, and F. Shams., International Journal of Communications, Network and System Sciences 5 (2012).

[12] Towards a Service Composition Language for Heterogeneous Service Environments., J. Niemoller, K. Vandikas, R. Levenshteyn, D. Schleicher, and F. Leymann. 2011 15th IEEE International Conference on Intelligence in Next Generation Networks, 2011.

[13] Dynamic Service Composition: A Discovery-Based Approach, J. Lee, S.P. Ma, S.J. Lee, Y.C. Wang, and Y.Y. Lin. , International Journal of Software Engineering and Knowledge Engineering, March 2008, 18(2):199-222.

[14] Composing web services enacted by autonomous agents through agent-centric contract net protocol, J. Lee, S.J. Lee, H.M. Chen, and C.L. Wu., Information and Software Technology, 54 (2012) 951V967

[15] REST2SOAP: a Framework to Integrate SOAP Services and RESTful Services, Y.Y. Peng, S.P. Ma, and J. Lee., IEEE International Conference on Service-Oriented Computing and Applications (SOCA). 2009

[16] Integration and Orchestration of Heterogeneous Services, K. He., Proceedings of the IEEE Joint Conferences Pervasive Computing (JCPC), 3-5 December 2009, pp. 467-470.

[17] A survey on web services composition, S. Dustdar and W. Schreiner, Technical report, Distributed Systems Group, Technical University of Vienna, 2004.

[18] Service Oriented Concepts and Technologies in the book Service-Oriented Software System Engineering: Challenges and Practices, H. Cervantes and R. S. Hall., Idea Group Publishing, 2005.

[19] Enhancing residential gateways: A semantic osgi platform., R. P. D. Redondo, A. F. Vilas, M. R. Cabrer, J. J. P. Arias, J.G. Duque, and A. GilSolla., IEEE Intelligent Systems, 23(1):32V40, 2008.

[20] RESTful Web Services., L. Richardson and S. Ruby., O'Reilly (May 2007)

[21] Head First Design Patterns. By Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra, O'Reilly Media(October 2004)

[22] Simple Object Access Protocol (SOAP) 1.2. , M. Gudgin, M. Hadley, N. Mendelsohn, J.J. Moreau and H.F.Nielsen, W3C, April 2007, http://www.w3.org/TR/soap12.

[23] Web services description language (WSDL) version 1.1., E. Christensen, F. Curbera, G. Meredith and S. Weerawarana. W3C, March 2001, http://www.w3.org/TR/wsdl.

[24] Knopflerfish 3, OSGi R4., http://www.knopflerfish.org/.

[25] Eclipse, http://www.eclipse.org/.

[26] The next step in web services, F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana. Communications of the ACM, 46(10):29-34, 2003.

[27] Service-oriented computing: introduction, M.P. Papazoglou and D. Georgakopoulos, Communications of the ACM, 46(10):24-28, 2003.

[28] Service Life Cycle Tools and Technologies: Methods, Trends and Advances, J. Lee, S.-P. Ma, and A. Liu., IGI Grobal, 2011.

[29] Architectural styles and the design of network-based software architectures, R.

Fielding., Ph.D. thesis, University of California, Irvine, 2000.

[30] OSGi service platform, core specification, release4, OSGi Alliance, http://www.osgi.org/.

[31] A Flexible System for Visual Service Composition, C. Pautasso, Diss. ETH No. 15608, July 2004.

[32] From Web Service Composition to Megaprogramming, C. Pautasso, and G. Alonso, In Proceedings of the 5th VLDB Workshop on Technologies for E-Services (TES-04), Toronto, Canada, August 29-30, 2004.

[33] BPELLight. Business process management, J. Nitzsche, T. van Lessen, D. Karastoyanova, and F. Leymann, Springer, 4714(214-229), 2007.