IJCTA, 9(26) 2016, pp. 109-113

A Web Based Intelligent Orchestration Tool for Smart City

S. Thangam[1], R. Gobi[2], P. Sathya[3] and E. Kirubakaran[1]

ABSTRACT

Web service technologies are becoming a de facto standard to integrate distributed applications and systems using XML-based standards. Developing applications that support web services interfaces will not be enough to provide complete and coordinated business processes. Thus, we need a new approach to compose these web services together in order to form web services orchestration and processes definition. Many new standards have been defined to solve this problem, for example BPEL4WS, and WSCI. The "orchestration tool" is for combine several web services which is technically called as Service Orchestration. Service Orchestration is implemented where Graphical User Interface Provides developers to combine web services in easy manner. GUI makes the developers to do so the process by drag and drop the web service and generates new web service (composite web service). XML file will be generated to store the web service which is inside composite web service. The new web service is available for the developer in the interface which can carry forward for testing in the separate tool. In the Testing environment, The XML file is parsed to know how the web service will be subject for testing in the GUI. The web service which is developed for this project is RESTFul web service, which can communicate over HTTP servers. The request and response may be in any type. The testing interface tests the web service by using a request method over HTTP servers and will get response via the same.

Keywords: Web Services, SOA, WSDL, BPEL4WS, Services Composition, Services Orchestration.

1. INTRODUCTION

Smarter governance is the backbone of smart solutions. Smarter governance is enabled through more informed decision making and participation of disparate opinions and agendas towards overall betterment of cities and communities. World over, technologies are enabling smarter solutions. Technology innovation is helping better collection, processing and analysis of data through conventional and crowd/social media methods. ICTs help cities connect better to their citizens, enabling better feedback and cross fertilization of ideas. Technological solutions help model and analyze urban issues, incorporating multiple factors and generating solutions that have multiples co-benefits. However, pursuing technologies for the sake of technology introduction is never fruitful. Technology is merely the means towards the desired ends and not the other way round. Smarter Urban Management for Smarter Cities and Communities. The key link in enabling smarter solution of cities and communities is smarter governance. With the advances made in the last decade, most cities have a combination. The key features of a Smart City is in the intersect between competitiveness, Capital and Sustainability. The smart cities should be able to provide good infrastructure such as water, sanitation, reliable utility services, health care; attract investments; transparent processes that make it easy to run a commercial activities; simple and on line processes for obtaining approvals, and various citizen centric services to make citizens feel safe and happy. Our work will help in providing an improvement in communication overhead in service composition environment.

[1] Periyar Maniammai University, Thanjavur, *E-mail: thangamradha@gmail.com*
[2] National Institute of Technology, Tiruchirappalli, *E-mail: gobi86@gmail.com*
[3] National Institute of Technology, Tiruchirappalli, *E-mail: vpsathya86@gmail.com*
[4] Bharat Heavy Electricals Limited, Tiruchirappalli, *E-mail: ekiru@bheltry.co.in*

2. EXISTING SYSTEM

Currently, composition of web services is done by orchestration. An orchestration is a workflow that combines invocations of individual operations of the web services involved. It is therefore a composition of individual operations, rather than a composition of entire web services. In this paper we propose a different approach to web service composition, whereby entire services are composed into composite services. The latter are again entire web services that is, they can be further composed using our composition, or they can be used in an orchestration. The service-oriented architecture (SOA) promises a new generation of information systems applications based on a new set of standards for enabling self-describing interoperable Web services. Web service orchestration and choreography are both concerned with the composition of Web services to meet the needs of business processes. There are two important standards for modelling and implementing workflows and business processes based on Web services: BPEL follows the orchestration paradigm, and WS-CDL covers the choreography. From the perspective of composing Web services to execute business processes, the orchestration is more flexible approach compared to the choreography. For that reason we concentrate on some BPEL orchestration aspects using formal models: error, event, and compensation handling using extended version of calculus, correlation mechanisms using process algebra, and control flow constructs using Petri nets. All of the formal models allow the analysis and verification of BPEL processes what gives a good starting point for better understanding and further investigations on selected BPEL issues. This could also contribute towards the improvement of the quality of BPEL specification, the applicability of BPEL itself and the implementation of real orchestration.

3. PROPOSED SYSTEM

Graphical User Interface developed makes the developer or the user to combine web service in no matter of time. Drag and drop facilities make the developer understand the process much clearer than ever. Each and each process in this system makes the developer feel more comfortable while combining web service. XML files improves the reliability of idle data.

The orchestration tool developed combines several web services which are technically called as Service Orchestration. Graphical User Interface Provides developers to combine web services in an easy manner. GUI makes the developers to do the process by drag and drop the web service and generates new web service (composite web service). XML file will be generated to store the web service which is inside composite web service. The new web service is available for the developer in the interface which can carry forward for testing in the separate tool. There are two major disadvantages in case of orchestration. One is the communication overhead and the other is single point failure. In existing service orchestration platforms, data and control dependencies between services are managed centrally by Orchestrator. The centralized orchestrator is a bottleneck and may cause performance degradation and availability issues. It also causes additional traffic of messages, since every activity execution involves a back-and-forth message exchange between services, which may be located arbitrarily apart and in a different organizational domain. The Figure 1.1 depicts the centralized orchestration.

An alternative improved method is to execute the orchestration script by using regional concept shown in Figure 1.2.

An Orchestration tool is developed travel planning system is implemented with the concept of regional orchestration. A scenario of travel system where customer approaches the travel agent with travel plan is implemented as in Figure 1.3.

The data exchanged between the services in a traditional orchestration is given in Figure 6.6.The Transport Service (TS), Hotel reservation service (HR), Railway Reservation service (RR), Inspector service (INS) and Banking service (BS) are implemented. The orchestrator takes care of coordinating the services.D1, D2, D3, D4, D5and D6 represents the data exchanged between the services.

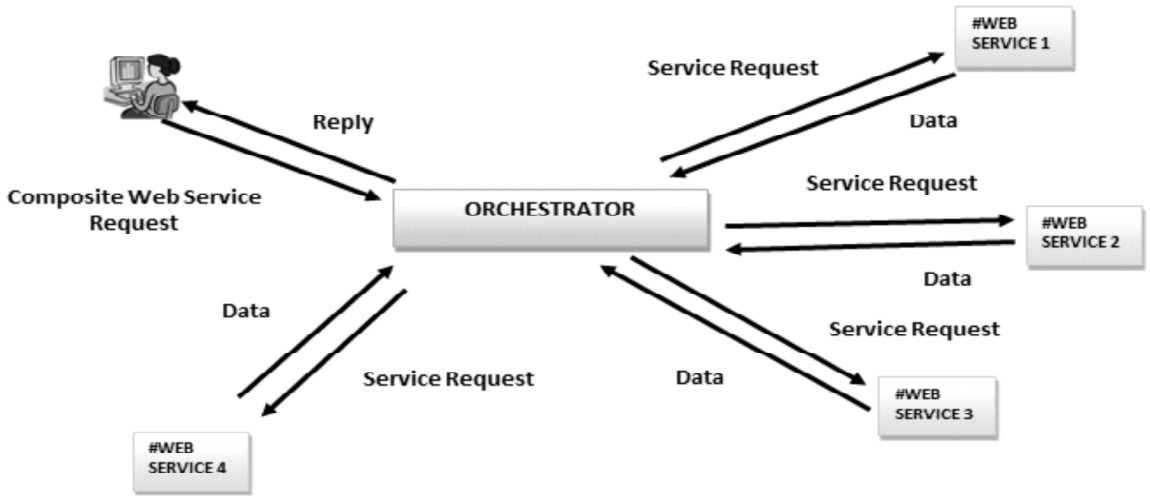

Figure 1.1: Centralized Orchestration

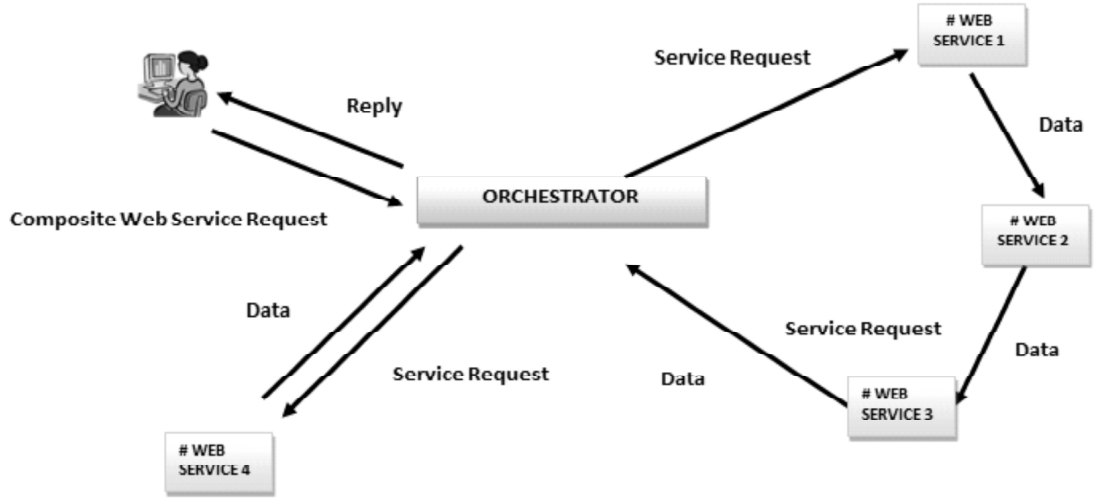

Figure 1.2: Regional Orchestration

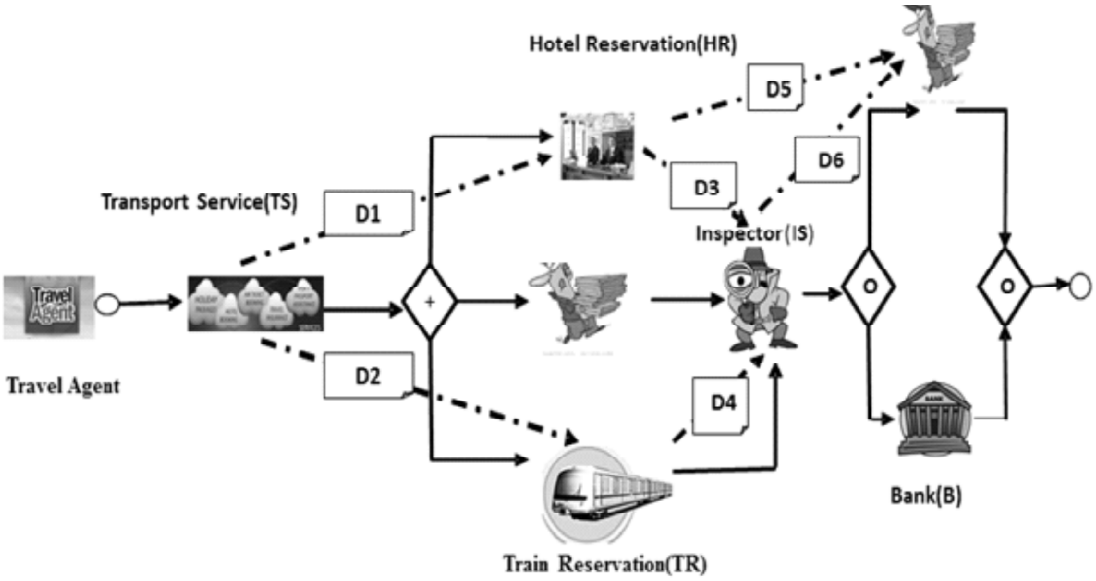

Figure 1.3: Travel Planning system with Regional Orchestration

Communication Overhead is calculated for regional orchestration as follow

COH(A1,A2)=CONS(A1,A2)*NE(A1)*PA(A1,A2)+”NE(A1)*SIZE(D)

WHERE

A1, A2 – activity

D- Data exchanged

COH (A1, A2) – communication overhead of activities A1, A2

CONS (A1, A2) –consecutive activity.

4. IMPLEMENTATION

Graphical User Interface makes the developer/user to combine web service in no matter of time. Drag and drop facilities makes the developer to understand the process much clear than ever. Each and each process in this system makes the developer to feel more comfort while combining web service. XML files improves the reliability of idle data.

- Less Time for orchestration.
- Less employees for process.
- More profit via reducing efforts.
- Less risks by automating the process works.

Input design is a part of overall design system that needs careful attention and it includes specifying the means by which actions are taken. A user interacting through screen must be able to tell the system whether to accept input, forward it into actions for end processing. The input design is done in a manner that would be adaptable to SQL back end storage and its data security and validity is maintained throughout. The input format for the entries that are done predefined. The input design differ from each and every module depending on the application for the module. Output is the most important direct source of information to the user. Efficient and intelligent output should improve the system relationship with the user and the system. All output made is specified in a way that would be easy to understood to the user. In the output design the confirmation message is generated by the application server and sends back as response. Also whenever the stock gets empty a report is thrown as the notification. The design displayed in the screen is specified to give the appropriate output to the end user. Thus input and output design of this system will make the end user the job easily.

Currently, composition of web services is done by orchestration. An orchestration is a workflow that combines invocations of individual operations of the web services involved. It is therefore a composition of individual operations, rather than a composition of entire web services. In this paper we propose a different approach to web service composition, whereby entire services are composed into composite services. The latter are again entire web services, that is, they can be further composed using our composition, or they can be used in an orchestration.

Table 1.1
Comparison of Orchestration Models

Model	*M1*	*M2*	*M3*	*M4*
Centralized Orchestration	0.795	0.499	0.587	0.469
Regional Orchestration	0.103	0.472	0.475	0.112

The comparison of Orchestration model is shown in Table 1.1 and the graph in Figure 1.4 shows the improvement of Regional Orchestration over centralized Orchestration.

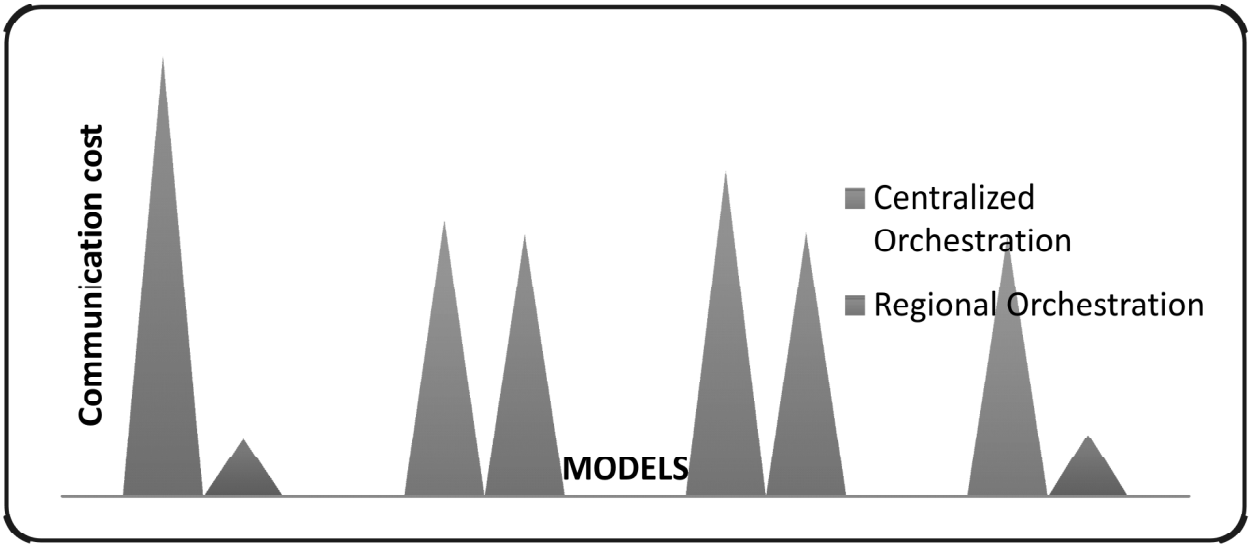

Figure 1.4: Comparisons of Orchestration Models

6. CONCLUSIONS

Service orchestration is done via graphical user interface in this project effectively. Developers can easily combine several web service according to their requirement and can test the combined web service in the testing environment. Graphical user interface makes the user much more confident and comfortable on testing. Drag and drop facilities improved the interface more user friendly for the developer. Restful web services are very effective for implementing business logics anywhere. By this way, monolithic applications can be shift away and have instead a set of reusable services that can be composed to build applications. Enhance rich interface and more user friendly. Communication between web services be improved via special frameworks. Customized frameworks implemented for developers requirements. IDE much more enhanced to do the job of complex works in some clicks. WADL, WSDL, SOAP request and response replaced with some simple standard implementations. This research work proposed the design and implementation of a system that can scale up and down the number of application instances automatically based on demand and developed a color set extensible framework for managing clusters of virtual machines.

REFERENCES

[1] AartiKarande, MilindKarande and B.B.Meshram,'Choreography and Orchestration using Business ProcessExecution Language for SOA with Web Services', *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 2, March 2011.

[2] SaswatAnand, Edmund Burke, John Clark, 'An Orchestrated Survey on Automated Software Test Case Generation', *Journal of Systems and Software*, February 2013.

[3] Florian Lautenbacher, Bernhard Bauer,'A Survey on Workflow Annotation & Composition Approaches', ISSN 1613-0073, online CEUR-WS.org/Vol-251/2013.

[4] GirishChafle,SunilChandra,VijayMann,MangalaGowri Nanda,' Decentralized Orchestration of Composite Web Services',WWW2004,.ACM 1581139128/04/0005.

[5] Ward Jaradat, Alan Dearle, and Adam Barker,'An Architecture for Decentralised Orchestration of Web Service Workflows', arXiv:1305.1842v1 [cs.DC] 8 May 2013.

[6] Dadape Jinendra et al,'Smart Travel Guide: Application for Android Mobile', *International Journal of electronics, Communication & Soft Computing Science*, ISSN: 2277-9477.