International Journal of Grid Distribution Computing
Vol. 8, No.5, (2015), pp.137-140
http://dx.doi.org/10.14257/ijgdc.2015.8.5.13

Pi-Calculus Based Formal Verification of Web Services Composition

Saurabh Agarwal and Koshel Agarwal

CSE Dept. Shri Ram College of Engineering & Management, Banmore 476444,
(MP) India
(toc.saurabh@gmail.com)

Abstract

One of the advantages of web services is development of the software through composition of web services, which use existing web services as a component to create software on demand, for composition to be useful and valid in development of the software, web services composition need to be validated for correctness. In this paper, we have formally represented web services using Pi-Calculus. It is a kind of process algebra which can be used to model dynamic concurrent systems and behavior of web services composition is very close to it. For verification the tool MWB (Mobility Workbench) has been used.

Keywords*: Web services composition, Pi-Calculus, MWB*

1. Introduction

Web services are autonomous computer programs that can be accessed over internet. They are platform and language independent i.e. they can be installed over any operating system and may be developed in different programming language. The main components of web services architecture are service provider, service requester and service Online book purchase system registry. Service provider implements web services and publishes information required to access those services. Service registry (also called UDDI registry) is used to store this information, any service requester can find information to access a particular web service from this registry.

Reusability is one of the advantages of web services. We can compose two or more web services which already exist; it reduces the efforts of implementing new web services. Sometimes it is necessary to compose two or more web services because a single service is not sufficient for some complex applications. Every web services composition algorithm should be validated before implementation for its correctness. We have used Pi-Calculus [4] for describing and modeling web service composition; for verification process we have used MWB (Mobility Workbench) [6].

2. Related Work

In recent years several methods have been used to model web services composition. CCS [7] is used to model concurrent systems, but it cannot be used for mobility. In [8], a method based on temporal logic of actions (TLA) is proposed. Web services are modeled as automata and described using TLA, then verified using TCL (a model checker of TLA). In [9], FSM (finite state automata) is used to describe the web service composition and it is translated into programs Described by Promela and finally verified using model checking tool SPIN. In [10], Colored Petri-nets (CPN) is used; it is similar to Petri-nets with an extra advantage that has programmable elements. We have used Pi- calculus; it is used to model concurrent systems with mobility. In pi-calculus communication links are transferred as names.

ISSN: 2005-4262 IJGDC

3. Web Service Composition Examples

Online book purchase system

a. Description of web service composition

The services used by this system are client service, Book shop service, Database service and Delivery service (fig. 1). Here x and w are communication links of Client service with Book shop service and Delivery service, respectively. Similarly, y and z are communication links of Book shop service with Database service and delivery service respectively.

First, Client service sends a request to Book Shop service for purchasing a book by specifying required information such as name of book and author's name. Book shop service then sends a lookup message to database service whether the requested book is available or not. Database service sends a message to

Book Shop service based on avail-ability of that book.

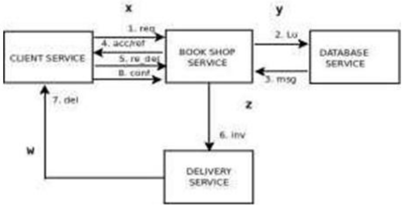

Figure 1. Web Services Composition Model for Online Book Purchase System

Now Book Shop service replies to Client service about the requested book by sending accept or refuse message. If book is available then it sends accept message otherwise sends refuse message. Now Client service confirms its request by sending a request delivery message. This message contains information about client i.e. address and phone number etc. After receiving this message, Book Shop service invokes delivery service with appropriate information. Delivery service then delivers the book to client and client sends a confirmation message to Book Shop service. For simplicity, we assumed that amount for book is paid at the time of delivery i.e. in person.

The various messages used in this system are described as follows:

req : Client service requests for a book.

Lu : Book Shop service sends this message to Database service to check the availability of that book.

msg : Database service sends availability information to Book Shop service.

acc : Book Shop service sends this message to client service if book is available. ref : Book Shop service sends this message to client service if book is not available.

req del : Client confirms its request by sending its information such as address and phone number etc.

inv : Book Shop Service invokes delivery service.

del : Delivery service delivers the required book to client.

conf : Client service sends confirmation to Book Shop Service.

The description of different agents in pi-calculus is given as follows:

Client service: CS(x,w) = xreq.x(msg). ([msg=acc] xdel book. w(del).x conf.CS(x,w) + [msg=ref]0)

BSS(x,y,z) + [msg=not found]z
ref.BSS(x,y,z))

Delivery service: DS(z,w) = z(inv):wdel.DS(z,w) Database service : DBS(y) = y(lu):ymsg.DBS(y) Composite Web service = (CSjBSSjDSjDBS)

b. Verification of web service composition

The MWB code for different agents is given as follows:

agent CS(x; req; msg; acc; del book; del; conf; ref) =^{0} x < req > :x(msg):([msg = acc]^{0} x <

del book > :w(del):x < conf > :CS < x; req; msg; acc; Del book; del; conf; ref > +[msg = ref]0)

agent BSS(x; req; y; lu; msg; found; acc; req del; z; inv; conf; not found; ref) = x(req):

^{0} y < lu > :y(msg):([msg = found]x < acc > :x(req$_{d}$el):^{0} z < inv > :x(conf):BSS < x; req; y; lu; msg; found; acc; req del; z; inv; conf; not found; ref > +[msg = not found]

^{0} x < ref > :BSS < x; req; y; lu; msg; found; acc; req del; z; inv; conf; not found; ref >)

agent DS(z; inv; w; del) = z(inv):^{0} w < del > :DS < z; inv; w; del >

agent DBS(y; lu; msg) = y(lu):^{0} y < msg > :DBS < y; lu; msg >

In verification process, we have checked deadlocks for different agents through MWB, it ensures the correctness of web services composition algorithm.

4. Conclusion

Formal description and verification ensures the correct composition of web services. It also ensures that system works as it is intended such as it can be checked for deadlock freedom. Online book purchase system example has been used to illustrate the modeling of web services composition as well as its verification. This paper formally describes web services composition through Pi-calculus and checks it's correctness with the help of MWB.

References

[1] P. W. P. Chan and M. R. Lyu, "Dynamic Web service composition, A new approach in building reliable web services", 22nd international conference on Advance information networking and applications, **(2008)**.

[2] S. Prabhu, "Towards distributed dynamic web service composition", eighth international symposium on autonomous decentralized systems (ISADS'07), **(2007)**.

[3] R. Miler, "The polyadic pi-calculus, a tutorial, research report ECS-LFCS-91-180", University of Edinburgh, October **(1991)**.

[4] J. Parrow, "An introduction to the pi-calculus, Handbook of Process Algebra", ed. Bergstra, Ponse, Smolka, **(2001)**, pp. 479-543, Elsevier.

[5] R. Milner, J. Parrow and D. Walker, "A calculus of mobile process (part 1 and 2), Journal of information and computing", vol. 100, **(1992)** September, pp. 1-77.

[6] B. Victor and F. Moller, “The Mobility Workbench, A tool for the pi-calculus, Uppsala University and University of Edinburgh”, **(1994)** February.

[7] L. Bao, W. Zhang and X. Zhang, “Describing and verifying web services using CCS”, proceedings of the seventh international conference on parallel and distributed computing, applications and technologies, **(2006)**.

[8] H. Wang, C. Wang and Y. Liu, “logic based approach to web service composition and verification”, Second world conference on services, **(2009)**.

[9] Z. Wei, R. Dong, X. Luo and F. Liu, “Model checking Airline tickets reservation system based on BPEL”, third international conference on genetic and evolutionary computing, **(2009)**.

[10] A. Mukherjee, Z. Tari and P. Bertok, “Modeling of BPEL Composite service using clustered colored Petri-nets”, Second world conference on services, **(2009)**.