Colored Petri-Nets (CPN) based Model for Web Services Composition

Ye Wint Maung Maung, Aung Aung Hein
Computer Science Department
Defence Services Academy (DSA)
linnwai46@gmail.com

Abstract— Technology is, now, looking for distributed applications to develop and increase its domains. Web services are distributed components that provide functionality applications through network. They can be used by applications written in different languages, and performed in different platforms on different systems. The concept of Web services is essentially based on standard internet protocols, such as UDDI (Universal Description, Discovery and Integration), WSDL (Web Service Description Language) and SOAP (Simple Object Access Protocol) which offer solutions for description, publication, discovery, and interoperability of Web services. However, the composition of Web services is not guaranteed by them, therefore, different approaches have been proposed to solve this problem.

Current Web services composition approaches range from practical languages aspiring to become standard like Business Process Execution Language (BPEL) to theoretical model like Petri nets. In this paper, we propose Composite Web service model based on Colored Petri-Net (CPN) approach. A Colored Petri-Net is a high level Petri net that provides a significant increase in the expressiveness and compactness of Petri net models. Colored Petri-Net (CPN) has the capacity of formally modeling and verifying complex systems. This paper can assist Web service composition designers and developers to deliver lasting solutions, in concordance with the technology’s critical needs.

Keywords: Web services, Composite Web service, Petri nets, Colored Petri-Net (CPN).

I. INTRODUCTION

NOWADAYS, with the emergence and the evolution of new technologies, such as e-business, a large number of companies are connected to Internet, and have proposed Web services to trade. Web services are conceptually limited components to relatively simple functionalities. Generally, a single service does not satisfy the user’s needs that are more and more complex. Therefore, services must be made able to be composed to offer added value services.

Web services composition has received much interest to support business-to-business or enterprise application integration. Web services composition is an emerging paradigm for enabling application integration within and across organizational boundaries. And it allows us to combine a number of existing Web services into a new, value-added Web service. Web service has a specific task to perform and may depend on other Web services, hence being composite. Current Web services composition proposals, such as BPEL, cannot provide any kind of mechanisms or tool support for verifying the composition before implementation.

So, in this paper, we introduce an approach to model Composite Web services (CWS) based on Colored Petri-Nets (CPN) before implementing the system.

II. RELATED WORK

Thomas et al. have defined a timed Petri net representation of Web Services Flows; in this case, only the flow of messages and methods are considered, the starting point being WSDL (Web Service Description Language) [6].

Hamadi and Benatallah [1] have proposed a Petri net-based algebra to model Web Services Control flows; hence, constructions such as sequence, choice, iteration, parallelism, discriminator, selection and refinement are studied in that paper, but they omit consideration of timed or prioritized interactions.

Another important issue is verification of a composite services model. Some researches present that CPN (Colored Petri-Net) can be used to model and verify the composite services. Kochut et al. present the approach of design and verification for composite services based on CPN. Hui Kang et al. propose a graphical and formal modeling tool for composite services based on CPN [3].

III. SERVICE COMPOSITION APPROACHES

Composition rules describe how to compose coherent global services. In particular, they specify the order in which, and the conditions under which, services may be invoked. We distinguish syntactic (XML-based) and semantic (ontology-based) service composition. In this paper, we discuss syntactic approaches to service composition. This paper emphasizes the expected advantage of using formal methods, in particular their tool support to verify the correctness of service compositions in order to increase one’s confidence in service compositions. Therefore, we hope to provide a reference for service composition designers and developers willing to use formal methods and tools. [2]

A. Business Process Execution Language (BPEL)

This XML-based language was designed to enable the coordination and composition of a set of services. It is based on the Web Services Description Language (WSDL), which is basically an interface description language for WS providers. BPEL is a behavioral extension of WSDL using a workflow-based approach. Simple Web services can be aggregated to form more complex Web services by using workflow
techniques through the BPEL language. It expresses relationships between multiple invocations by means of control and data flow links, and it employs a distributed concurrent computation model with variables. [5]

B. Web Services Choreography Description Language

This XML-based specification language is targeted at composing interoperable, long running, peer-to-peer collaborations between service participants with different roles, as defined by a choreography description. Its most important element is the interaction activity, which describes an information exchange between parties, with a focus on the receiver. However, WS-CDL has two main drawbacks: first, it does not provide any graphical tool to specify services, and, second, it lacks theoretical foundations to analyze, simulate and verify the services. [5]

IV. FORMAL METHODS FOR SERVICE COMPOSITIONS

In this approach, we tackle the problem of Web services composition using a formalism based on Petri nets. Recently several formal methods, most of them with a semantics based on transition systems (e.g. automata, Petri nets, process algebras) have been used to guarantee correct service compositions. Petri nets theory allows a system to be modeled by Petri nets, a mathematical representation of the system. Further, Petri nets allow not only modeling of systems, but also provide an efficient mean for system analysis. [4]

A. Theory of Petri Nets

Petri nets also called a place/transition net, is a generalized graph introduced by Carl Adam Petri in 1962. The motivation behind Petri nets was the need to address problems of concurrency in systems. A Petri net is a bipartite directed multi-graph, where the two types of nodes are places and transitions. In graphical representations, places are shown as circles or ellipses, and transitions as bars or rectangles.

Like industry standards such as UML activity diagrams, Petri nets offer a graphical notation for stepwise processes that include choice, iteration, and concurrent execution. Unlike these standards, Petri nets have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process. Usage of Petri nets is widespread and diverse. In this paper, we propose model of Web services composition based on Colored Petri-Nets (CPN).

B. Colored Petri-Nets (CPN)

The abbreviated CPNs or CPN, as well as other types of Petri Nets, are both state and action-oriented. It means that they can represent states (with entities called places) of a system, and also actions (with entities called transitions). [7]

CPN Tool is a special simulation system which uses the language of Petri nets for models’ representation. The system was developed in University of Aarhus in Denmark and is distributed free of charge for non-commercial organizations via web site http://www.daimi.au.dk/CPNTools [8]. The level of service allows the classification of CPN Tools as an enterprise system. It was used in a lot of real-life projects especially in the area of telecommunications. [7]

CPN has great values for modeling, in that:
1) They have a graphical representation.
2) They support concurrency, which is necessary for many non-trivial interactions.
3) They are well researched and understood, and have been applied to many real-world applications.
4) Many tools and techniques exist for the design and analysis of CPN based systems.

The proposed system is the Travel Reservation System which consists of Travel Agent Composite Web service that combines other atomic services, such as Flight, Hotel and Car Rental. Implementation of proposed system will be carried out based on modeling of Web services composition using Colored Petri-Nets (CPN) approach. Our approach is based on Colored Petri-Nets (CPN) model that has the capacity of formally modeling and verifying complex systems. Moreover, an approach to solve these issues we use the Java 2 Enterprise Edition (J2EE) that acts an intermediary between Composite Web service on the one side and atomic services on the other side.

V. PROPOSED SYSTEM ARCHITECTURE

![Figure 1. Overview of Travel Agent Composite Web Service Model](image_url)

Our composition has three atomic services: Flight, Hotel and Car Rental. Overview of Travel Agent Composite Web Service is illustrated in figure 1. This model consists of three parts. They are as follows:

1) Car Rental, Hotel and Flight atomic services, executing specific operation.
2) Travel Agent Composite Web Service developed by JavaCPN application as intermediary.
3) Composite Web Service model based on CPN approach.

The travel agent composite transactions are taken from individual Web services. In the following, we are going to explain how this system works. Atomic services are to be created in the first step.
A. Modeling Travel Agent Composite Web Service

In this part, we have to explain model of Travel Agent composite Web service using JavaCPN approach. In our model, we present the CPN model developed on CPN tool, JavaCPN application and operations of travel agent composite Web service. As an illustration in figure 2, overview of Travel Agent JavaCPN model has been united in one hierarchical model, Travel Agent Composite Web Service.

![Figure 2: Travel Agent Composite Web service CPN model](image)

B. Executing Travel Agent Composite Web Service

The model basically serves as the user interface and accepts three inputs: user information, desired destination, and user’s decision (accept/deny) upon the available options or offers given by the agent. The model returns the booking information as the output.

![Figure 3: Travel Agent Java CPN Model](image)

It is mandatory to build a connection between Java application and CPN model. In the figure 3, this connection is carried out by the transaction, named open JavaCPN connection. Java application serves as the intermediary between CPN model and three atomic Web services. This application retrieves input from CPN model and sends it to atomic Web services. The results which are returned by atomic services are sent back to the CPN model.

Invoking respective operations of particular services is carried out by Travel Agent Composite WS and this process can be divided into three operations: getting information, booking, and giving confirmation. In the figure 3, the blue transaction refers to the Travel Agent Composite Web Service. Its detailed operations are described in hierarchical model as shown in the figure 4.

![Figure 4: Travel Agent Composite Web Service](image)

VI. CONCLUSION

This system presents Colored Petri-Nets (CPN) approach for modeling over Composite Web service. CPN Tool is used in this research. Using Java as intermediary has been emphasized in this system. Actually, we can construct composite Web service model based on CPN approach. From this point, the model cannot connect the actual atomic Web services. So, we use JavaCPN connection as an intermediary connecting to Web services (single, atomic services) from CPN Tools. It is noted that we can present technique connecting Web services from CPN Tool.

ACKNOWLEDGMENT

I would like to thank my supervisor, Mr. Aung Aung Hein, Assistant Lecturer, Computer Science Department, Defence Services Academy (DSA), for his excellent guidance, where the initial scope of the research was defined, and throughout the process of writing this research.

REFERENCES


