Chapter 11

Applying Formal Methods to Verify Web Services Orchestration and Choreography

Zohra Sbai

Prince Sattam Bin Abdulaziz University, Saudi Arabia & Tunis El Manar University, Tunisia

ABSTRACT

The contribution presented in this chapter is to provide a formal framework ensuring the model checking based verification of the Composition of Web Services (WSC). For this, the authors propose first to model the web services composition by an interaction of open workflow nets: a special class of Petri nets. Then, they detail how to check behavioral properties specified in temporal logic using the model checker NuSMV. A WSC is with added value only if the involved services are compatible. So, in this context, across the translation proposed, the authors develop a verification layer of the WSC compatibility. This work is developed in a framework named D&A4WSC which allows to model the WSC by oWF-nets and to check their compatibility by invoking the model checker NuSMV. Furthermore, the authors extended D&A4WSC so that it permits a web services choreography described in a WS-CDL specification. For this they developed a translation from WS-CDL to a composition of oWF-nets, so that one can verify this choreography by the aforementioned approach.

INTRODUCTION

Composition of reusable services often provided by different service providers is the key of service oriented software development. This involves the composition of software systems from reusable components and the ability to dynamically locate necessary services at run-time. These services may be provided by other parts as Web services, but the essential element is the dynamic nature of the connection between the service users and the service providers. Web service technology provides a flexible and universal approach to the interoperability of heterogeneous systems. We distinguish two categories of coordination
Applying Formal Methods to Verify Web Services Orchestration and Choreography

processes: processes which implement the choreography of services and processes implementing their orchestration. While the orchestration is centralized around a composition engine and there is a static service composition which follows a predefined pattern, the choreography form a composition approach based on decentralized cooperation between a collection of services. Dealing with the one or the other, the formal verification of such a composition is a very interested research subject.

In the SOA framework, process automation is a major focus, with the concepts of orchestration and composition of services. This is to centralize the logic of a process in a dedicated component, which supports the Web and business rules associated. This approach tends to reduce the impacts related to fluctuations in the process.

The generalization of the orchestration approach leads to the choreography which consists in designing a decentralized coordination of applications, in which there is no privileged machine but a network of interconnected machines that exchange messages.

Several languages have been proposed to ensure the composition of Web services namely BPEL (Ma, C., Xu, Q., and J.W. Sanders, 2009) and WS-CDL (Hongli, Y., Xiangpeng, Z., Zongyan, Q., Geguang, P., & Shuling, W., 2006) which are based on the XML syntax. BPEL is used to define the abstract and executable business processes as a set of Web services interaction coordinated recursively. WS-CDL is a language that describes peer-to-peer collaborations of participants by defining, from a global viewpoint, their common and complementary observable behavior; where ordered message exchanges result in accomplishing a common business goal.

For the best achievement of this common goal, the composed services should be compatible. This property ensures to obtain a composite service that does not suffer from problems of deadlock. It is therefore necessary to make sure that these services will be able to interact properly, which requires a study, a clear understanding and a verification of the concept of compatibility. In this context, we propose to use formal methods to model a given composition and thus to be able to verify it. In particular, we propose to use open workflow nets (oWF-nets), a subclass of Petri nets, which are characterized by their interface places that ensure the communication between the different services. In addition to this, given that the choreography refers to the description of coordinated interactions between two or more parties, we propose to translate a WS-CDL description into a composition of oWF-nets so that we will be able to formally verify the given choreography description.

To implement our approach, we propose D&A4WSC as a framework having as a main goal the verification of WSC using formal methods. The verified WSC can be generated to a valid BPEL process so that it can be executed by an orchestration engine. More precisely, D&A4WSC allows first the modeling of a WSC using a composition of oWF-nets or the generation of oWF-nets composition from a WS-CDL description. Secondly, it allows the compatibility verification. For this, we first check the compatibility at the level of exchanged messages between the different services, that we call syntactical verification. Then, we move to the semantic verification based on the principle of model checking, and more precisely, on the model checker NuSMV (Cimatti, A., Clarke, E.M., Giunchiglia, F., & Roveri, M., 1999). In fact, model checking is one of the most used formal verification methods because of its effectiveness, its ease of use and the generation of a counter example in case of error. If the compatibility is valid, then, we generate an owfn file of the composition and we invoke the compiler oWFN2BPEL (Lohmann, 2008) which takes as input an owfn file and generates a BPEL file as output.

This chapter is organized as follows: Section 2 presents preliminaries on oWF-nets, compatibility, BPEL, WS-CDL and the model checking technique used in this proposal. Section 3 describes the approach of formal checking of the compatibility of the composition of Web services and explains the translation