Characterizing Compatibility of Timed Choreography

Nawal Guermouche, CNRS, Université de Toulouse, LORIA-Nancy University, France
Claude Godart, LORIA-Nancy University, France

ABSTRACT

Web services are the main pillar of the Service Oriented Computing (SOC) paradigm which enables application integration within and across business organizations. One of the most important features of Web services is the idea of choreography which captures collaborative processes involving multiple services. In this context, compatibility analysis of choreography is a central point to investigate. The compatibility of a choreography means the capability of a set of Web services to interact by exchanging messages in a safe way. Whether a set of services are compatible depends not only on their sequences of messages but also on quantitative properties like timed properties. In this paper, the authors investigate a model checking based approach that checks the compatibility of a choreography in which Web services support asynchronous timed communications.

Keywords: Asynchronous Web Services, Choreography Analysis, Compatibility Analysis, Service Oriented Computing (SOC), Timed Properties

INTRODUCTION

The evolution of computer science technologies has given life to many paradigms such as the Service Oriented Computing (SOC) paradigm (Alonso et al., 2004; Benatallah, Hamid, & Nezhad, 2007; Dijkman & Dumas, 2004). In the latter, Web services are the main pillar. Based on standard interfaces, Web services facilitate application-to-application interactions. This advantageous property of Web services gives rise to several important concepts such as the notion of choreography. Such a feature offers the possibility to capture collaborative processes involving multiple services where the interactions between these services are seen from a global perspective. In this context, one of the important elements is the compatibility analysis. By compatibility we mean the capability of a set of services of actually fulfilling successful interactions by exchanging messages.

In the last few years, some works have investigated the compatibility problem of two Web services: a client and a provider service (Bordeaux et al., 2004; Benatallah, Casati, Ponge, & Toumani, 2005; Benatallah, Casati, Toumani, & Ponge, 2005; Ponge et al., 2007; Guermouche, Perrin, & Ringeissen, 2008). In all these works, the authors deal with services that support synchronous communications. In that case, to characterize the compatibility class of two services, the authors check if each input (resp. output) message of a service corresponds...
to an output (resp. input) message of the other service in the same order (i.e., the services are synchronized over messages). However, the nature of distributed systems and particularly of Web services can be asynchronous, hence the problem of the applicability of these approaches which are very restrictive in real application scenarios is still open. To overcome such a limitation, in this paper we tackle the problem of analyzing the compatibility of a choreography in which Web services support asynchronous communications. In an asynchronous communication, when a message is sent, it is inserted into a bounded message queue, and the receiver consumes (i.e., receives) the message while it is available in the queue.

On the other side, it is commonly agreed that in general the interaction of Web services and in particular the compatibility of Web services depends not only on the supported sequences of messages but there are other crucial quantitative properties such as timed properties (Benatallah, Casati, Ponge, et al., 2005; Benatallah, Casati, Toumani, et al., 2005; Kazhamiakin et al., 2006a, 2006b; Ponge et al., 2007; Guermouche, Perrin, et al., 2008). We mean by timed properties the required delays to exchange messages (e.g., in an e-government application, a prefecture can send its final decision to grant an handicapped pension to a requester after 7 days and within 14 days). When services are interacting, their timed properties can be conflicting. The existing works cannot discover all the eventual timed conflicts since the authors rely on the principle of synchronizing the services over messages (Bordeaux et al., 2004; Benatallah, Casati, Ponge, et al., 2005; Benatallah, Casati, Toumani, et al., 2005; Ponge et al., 2007; Guermouche, Perrin, et al., 2008). In this paper, we propose a framework for analyzing a choreography compatibility in the context of asynchronous communicating services. In this framework we take into account data flow that can be involved when exchanging messages. Furthermore, we consider constraints over data and timed properties that specify delays concerning message exchanges. By studying the possible impacts of timed properties on a choreography, we remarked that when Web services are interacting together, implicit timed dependencies can be derived from different timed properties of the different services. Such dependencies can give rise to implicit timed conflicts. To discover deadlocks due to timed conflicts, we first study the possibility to apply the existing compatibility approaches of synchronous services (Benatallah, Casati, Ponge, et al., 2005; Ponge et al., 2007; Guermouche, Perrin, et al., 2008; Guermouche & Godart, 2008b), but we concluded that the existing approaches are inadequate to discover all the eventual timed deadlocks since the authors rely on synchronizing the services over messages. In order to catch all the possible timed deadlocks, we propose a set of model checking based primitives.

One of the important ingredients we need in a compatibility framework is the Web services description behavior. The behavior of a Web service specifies the sequences of messages the service supports, the involved data types, and the associated timed requirements. The timed behavior of a Web service specifies the timed conversational protocol (for short we say conversational protocol). For compatibility analysis, we have chosen to model a conversational protocol as a finite state machine (FSM) specification. This kind of formal representation has been already used in a series of work (Bultan et al., 2003; Benatallah, Casati, Ponge, et al., 2005; Berardi, 2005; Ponge et al., 2007; Anca & Igor, 2007; Guermouche, Perrin, et al., 2008) and seems adequate. In fact, a state machine based model is suitable to describe reactive behaviors (Benatallah, Casati, Toumani, et al., 2005), it is fairly easy to understand, and at the same time it is expressive enough to model the properties we consider. In addition, we rely on clocks as defined in standard timed automata (Alur & Dill, 1994).

To summarize, in this paper we make the following contributions: (1) We propose an asynchronous model of Web services that gathers messages, data types, data constraints, and timed requirements. (2) We propose an abstraction process that allows applying a
Web Services Enabled E-Market Access Control Model
[www.igi-global.com/article/web-services-enabled-market-access/3035?camid=4v1a](www.igi-global.com/article/web-services-enabled-market-access/3035?camid=4v1a)

Web Services Compositions Modelling and Choreographies Analysis
[www.igi-global.com/chapter/web-services-compositions-modelling-choreographies/66952?camid=4v1a](www.igi-global.com/chapter/web-services-compositions-modelling-choreographies/66952?camid=4v1a)