Chapter 63
Web Services Composition Problem:
Model and Complexity

Fahima Cheikh
Université de Toulouse, France

ABSTRACT

In the approach taken in this chapter, the composition problem is as follows: given a client service, a goal service and a set of available services, determine if there exists a mediator service that enables the communication between the client and the existing services in order to satisfy the client request, represented by the goal service. In this chapter's model, available services that have access control constraints are considered. To formally capture these constraints, the chapter defines Web Services as Conditional Communicating Automata (CCA) in which communication is done through bounded ports. This chapter gives a detailed presentation of said model and gives complexity results of the composition problem.

INTRODUCTION

Service oriented computing (Singh & Huhns, 2005) is a programming paradigm which allow the realization of distributed applications by composing existing services. In particular, in order to realize a client request that is not realizable by the existing services, an eventual solution is to combine/compose the available services.

Several approaches are investigated for the composition problem. A survey can be found in Dustdar and Schreiner (2005) and Hull and Su (2005), using formal methods. Other authors use a planning method (Pistore, Marconi, Bertoli & Traverso, 2005; Pistore, Traverso & Bertoli, 2005), a theorem prover method (Rao, Küngas & Matskin, 2004), and a method based on propositional dynamic logic (Berardi, Calvanese, De Giacomo & Micelle, 2006; Berardi, Calvanese, De Giacomo, Hull & Mecella, 2005). In section 3, we discuss in detail all these approaches. More precisely, for each of them, we answer the following questions:

- How services are defined?
- How the request/goal is defined?
- How services are composed?

DOI: 10.4018/978-1-4666-2919-6.ch063

In our approach, the composition problem is as follows: given a client service, a goal service and a set of available services, to determine if there exists a mediator service which allows the communication between the client and the existing services in order to obtain a system with a behavior equivalent to the goal service. To represent equivalence between two systems, we use one of these relations: trace inclusion, trace equivalence, simulation and bisimulation. Further, in our model, each available service has a security policies. These policies describe the conditions the user of the service should satisfy. For example, conditions can be about the value of credentials/certificate attributes. In our model, services are represented by Conditional Communicating Automata (CCA) in which both communicating actions and internal actions are possible. The communication is done through bounded ports. Moreover, the transitions are augmented by guards that ensure the system security (Cheikh, De Giacomo & Mecella, 2006; Mecella, Ouzzani, Paci & Bertino, 2006).

The chapter is organized as follows. In section 3 we discuss the related works cited above and compare them with our model. In section 4, we give basic definitions on automata. Then, in section 5, we define the notion of conditional communicating automata (CCA). In section 6, we present our model of services and we formally define the composition problem in section 7. Section 8 provides complexity results about the composition problem. We finally conclude in section 9.

RELATED WORKS

Several formal approaches investigated the issue of Web services composition. These approaches differ in several aspects. In fact, they do not use the same formal model to represent services and they do not consider the same definition of the composition problem. In this section, we briefly overview some of these approaches. For each one, we give the answer to the following questions:

- How services are defined?
- How the request/goal is defined?
- How services are composed?

In the approach proposed by Pistore, Traverso and Bertoli (2005), the authors propose a model where Web services are BPEL files. Formally, Web services are defined as State Transition Systems where communicating and internal actions are possible. However, the internal actions of the available services are non-observable. The goal is defined as an EaGLe formula (Dal Lago, U., Pistore, M., & Traverso, 2002) to satisfy and the composition problem is defined as follows: given a set of available services and a goal, synthesize a process that interacts asynchronously with the available services and satisfies the requirements. Formally, the asynchronous product of the available services is the planning domain, the goal represents the formula to satisfy and the composite service is obtained from the solution plan. The authors present their experimental results with different applications. However, they do not give neither an upper bound nor a lower bound complexity of the problem.

In the approach proposed by Rao, Kungas and Matskin (2004), the authors consider semantic Web services. In practice, they are described by the DAML-S language. Formally, Web services are described as axioms of the Linear Logic (LL). Whereas the goal is represented by a sequent of this logic. The composition problem is defined as follow: given a set of available Web services, find a composition of the available services that satisfies the client requirement. Formally, given a set of axioms and a sequent prove the theorem. To solve the problem, the authors use a theorem prover of LL. In particular, if the sequent is proved then a process calculus is extracted from the proof. This process represent the composed services. The proposed method is correct and complete.

In the approach proposed by Mitra, Kumar and Basu (2007), the authors consider Web service interfaces. In practice, interfaces use the WS-

Related Content