Chapter 8

WEB SERVICES COMPOSITION

Daniela Barreiro Claro\(^1,2\) and Patrick Albers\(^1\) and Jin-Kao Hao\(^2\)
\(^1\)ESEO, 4 rue Merlet de la Boulaye, BP 30926 49009 Angers cedex 01 France. 
daniela.claro@eseo.fr, patrick.albers@eseo.fr

\(^2\)LERIA, University of Angers, 2 Boulevard Lavoisier, 49045, Angers cedex 01 France. 
jin-kao.hao@univ-angers.fr

1. INTRODUCTION

Nowadays many enterprises publish their applications functionalities on the Internet. This new generation of applications allows greater efficiency and availability for business. In fact, more and more applications make functionalities available using a web service format.

However there are many services around the web, each one, taken alone, has a limited functionality. In many cases, a single service is not sufficient to respond to the user's request and often services should be combined through services composition to achieve a specific goal. For example, if a user wants to travel, it is not sufficient to book a flight, but she should also take care of reserving a hotel, renting a car, getting entertained, and so on. Such composition is carried out manually today, it means that the user needs to execute all these services one by one and these tasks can be time and effort consuming.

For that reason, the notion of composite services is starting to be used as a collection of services combined to achieve a user's request. In other words, from a user perspective, this composition will continue to be considered as a simple service, even though it is composed of several web services.

Nevertheless, prior to composing web services, candidate services should first be discovered and then selected. One difficulty is that many functionally
similar services are available and thus, the number of discovered services by search mechanisms increases as a consequence. The discovery process returns a set of candidate services from which the subset of those belonging to the composition should be extracted according to non-functional criteria (i.e. cost, availability, reputation). In fact, discovery is a prerequisite for selection, but selection is the main problem (Sreenath and Singh 2004). The non-functional criteria are here characterized by the QoS model presented in each web service. The QoS model has more than one criterion to be evaluated. Thus, services composition can be considered as a multiobjective optimization problem.

As depicted in Figure 8-1, we propose SPOC (Semantic based Planning for Optimal web services Composition), an architecture to compose web services. In our point of view, the problem of composing web services can be reduced into four fundamental phases: the first one is planning, which determines the execution order of the tasks, we consider here a task as being a service functionality or a service activity. The second one is discovery that aims at finding candidate services for each task in the plan. The third phase aims at optimizing services composition and is the point treated in this chapter, and, finally, the fourth concerns execution. This fourth phase is characterized as a problem because, even during the execution process, the services may not be found and another tradeoff composition needs to be used or other plan needs to be envisioned.

The composition of web services starts by creating the initial plan based on tasks definition. All the definitions of existing tasks should be located in a repository that the planner can consult for obtaining tasks interfaces. This repository can be represented as an ontology and for us, it can be an improvement over UDDI registries. Hence, we propose a UDDI (Universal Description, Discovery and Integration) that is actually an ontology which describes the services and their providers in an unambiguous way. The name