Chapter 9

User Preference-Based Web Service Composition and Execution Framework

Bassam Al-Shargabi
Isra University, Jordan

Omar Sabri
Isra University, Jordan

ABSTRACT

The motivation behind this chapter is that Service Oriented architecture issued to compose an application as a set of services that are language and platform independent, communicate with each other. Therefore, user preferences rules in web service composition process plays crucial role and has opened a wide spectrum of challenge. In this chapter, an agent for composing web services based on user preferences was introduced to fulfill a certain process, where the user preferences are essential for determining which web service are to be selected. In other word, the agent designed to maintain the following function: an intelligent web services selection and planning based on user preferences (such as price or availability), along with web services execution, tracking and adaptation.

INTRODUCTION

Service-Oriented Architecture (SOA) is paradigm to build a distributed systems that bring application functionality as services to end-user applications (Booth, Hass, Mccabe, Newcomer, Champion, & Ferris, 2005). The basic idea of SOA is to compose an application as a set of services that are language and platform independent, communicate with each other using standardized messages like XML. Web services is a technology that realize the SOA.

A web service is a software system identified by a URL, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems (Booth, Hass, Mccabe, Newcomer, Champion, & Ferris, 2005). As individual web services are limited in their capability, which created the need for composing existing services to create new functionality in the form of composite service. However, the process of creating composite service is achieved by combin-
User Preference-Based Web Service Composition and Execution Framework

ing existing elementary or complex services, possibly offered by different providers. For example, a travel plan service can be developed by combining several elementary services such as hotel reservation, ticket booking, car rental, sightseeing package, etc. In carrying out this composition task, one should be concerned with the efficiency and the QoS that the composed process will exhibit upon its execution (Chandrasekaran, Miller, Silver, Arpinar, & Sheth, 2003).

Some proposals are being made to enable dynamic composition of web services and execution monitoring frameworks (Al-Shargabi, El Shiekh, & Sabri, 2010). Few of these proposals address user QoS constraints: whether these constraints are locally on every individual web service or globally for the whole composition process according to (Al-Shargabi, El Shiekh, & Sabri, 2010). These constraints must be addressed to satisfy client requirements, such as price, availability, so it is necessary to represent required QoS in the selected and composed web services (Bakhshi, & Hashem, 2012). Moreover, Evaluation of composition process: when the composer selects a web service, it is quite common that many web services have the same functionalities. So it is possible that the composer generates more than one composite service fulfilling the requirements. In that case, the composed web services are evaluated by their overall utilities using the information provided from the non-functional attributes. The most commonly used method is utility functions as in WSCE framework. The requester should specify weights to each QoS attribute and the best composite service is the one that is ranked on top. During the execution of composed web service, some web services may update their QoS properties others may become unavailable. A dynamic composition approach is needed, in which runtime changes in the QoS of the component services are taken into account (Chen, Ha, & Zhang, 2013). It is imperative to design a Web Service Composition and Execution (WSCE) framework that adapts to failure of web services or changes in their QoS offerings to satisfy user preferences or constraints, these issues already have been discussed in previous work by (Al-Shargabi, El Shiekh, & Sabri, 2010).

The remained of this chapter organized as follows: section 2 the User Preferences WSCE framework is presented. Section 3 describes User Preferences WSCE agent functions. Section 4 presents domain registries. Section 5 the QoS certifier is presented. Section 6 analysis and validation, and finally conclusion and future work in section 7.

WSCE FRAMEWORK ARCHITECTURE

The User Preferences WSCE framework is a agent-based framework for the dynamic composition of web services as illustrated in Figure 1. The main motivation behind this framework is to build a User Preferences WSCE agent to make intelligent service selection decisions for composite service which fits with user preferences in his/her web process. The main functions of User Preferences WSCE agent include: execution tracking: User Preferences WSCE agent has a composition repository to record all feasible composition plans of composed services it is aware of, which QoS information of these composition plans are optimal or closer to user constraints. Dynamic service selection: This is the key function of User Preferences WSCE agent, when the WSCE agent selects web services to execute a web process according to the user-defined utility function, and user’s QoS requirements. Dynamic service adaptation: In case of individual web service failure during execution of composite service, the WSCE agent either replaces the failed services or replaces the composition plan with an alternative plan. The WSCE agent either way can create a new composition plan from scratch. In this framework a QoS certifier is which is controlled by the UDDI registry to verify the claimed QoS attributes for the registration requests of web service provider.