Design and Development of Communities of Web Services

Zakaria Maamar
Zayed University, UAE

INTRODUCTION

In the field of Web services (Benatallah, Sheng, & Dumas, 2003; Bentahar, Maamar, Benslimane, & Thiran, 2007; Medjahed & Bouguettaya, 2005), a community gathers Web services that offer similar functionalities. Hotel booking and car rental are samples of functionalities. This gathering takes place regardless of who developed the Web services, where the Web services are located, and how the Web services function to satisfy their functionalities. A Web service is an accessible application that can be discovered according to its functionality and then invoked in order to satisfy users' needs. In addition, Web services can be composed in a way that permits modeling and executing complex business processes. Composition is one of Web services' strengths as it targets user needs that cannot be satisfied by any single available Web service. A composite Web service obtained by combining available Web services may be used (Figure 1). The use of communities in composition scenarios offers two immediate benefits. The first benefit is the possibility of accelerating the search of Web services required to satisfy user needs by looking for communities rather than screening UDDI (universal description, discovery, and integration) and ebXML registries. The second benefit is the late execution binding of the required Web services once the appropriate communities are identified. Both benefits stress the need of examining Web services in a different way.

Current practices in the field of Web services assume that a community is static and Web services in a community always exhibit a cooperative attitude. These practices need to be revisited as per the following arguments. A community is dynamic: New Web services enter, other Web services leave, some Web services become temporarily unavailable, and some Web services resume operation after suspension. All these events need to be closely monitored so that inconsistent situations are avoided. Moreover, Web services in a community can compete on nonshareable computing resources, which may delay their performance scheduling. Web services can also announce misleading information (e.g., nonfunctional details) in order to boost their participation opportunities in composition scenarios. Finally, Web services can be malicious in that they can try to alter other Web services' data or operations.

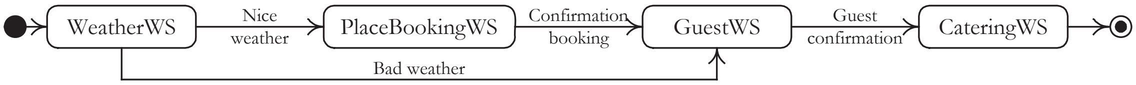
To look into ways of making Web services communities active, we describe in this article some mechanisms that would enable Web services among other things to enter a community, to leave a community after awhile, to reenter the same community if some opportunities loom, and to be rewarded for being part of a community. These mechanisms would be developed along three perspectives, which we refer to as the following.

- Community management: How do we establish or dismantle a new or existing community of Web services?
- Web services attraction and retention: How do we invite and convince new Web services to join a community? How do we retain existing Web services in a community?
- Interaction management: How are interactions between Web services regulated in a community? How do we deal with conflicts in a community?

BACKGROUND

The term *community* means different things to different people. In *Longman Dictionary*, community is "a group

Figure 1. Example of composition scenario



Web Service: It is a software application identified by a URI whose interfaces and binding are capable of being defined, described, and discovered by XML (extensible markup language) artifacts, and that supports direct interactions with other software applications using XML-based messages via Internet-based applications.