Chapter VIII

Transactional Composite Applications

Frederic Montagut
SAP Labs France, France

Refik Molva
Institut Eurecom, France

Silvan Tecumseh Golega
Hasso-Plattner-Institut, Germany

ABSTRACT

Composite applications leveraging the functionalities offered by Web services are today the underpinnings of enterprise computing. However, current Web services composition systems make use of only functional requirements in the selection process of component Web services while transactional consistency is a crucial parameter of most business applications. The transactional challenges raised by the composition of Web services are twofold: integrating relaxed atomicity constraints at both design and composition time and coping with the dynamicity introduced by the service-oriented computing paradigm. In this chapter, we present a new procedure towards automating the composition of transactional Web services. This composition procedure does not take into account functional requirements only but also transactional ones based on the acceptable termination states model. The resulting composite Web service is compliant with the consistency requirements expressed by business application designers and its execution can easily be coordinated using the coordination rules provided as an outcome of our approach. An implementation of our theoretical results based on OWL-S and business process execution language (BPEL) technologies is further detailed as a proof of concept.
INTRODUCTION

Web services composition has been gaining momentum over the last years as it leverages the capabilities of simple operations to offer value-added services. These complex services such as airline booking systems result from interactions between Web services that can span over organizational boundaries. Considering the lack of reliability akin to distributed environments, assuring data and transactional consistency of the outcome of cross-organizational workflow-based applications, such as composite applications, is necessary. The requirements that are relevant to assuring consistency within the execution of Web services composite applications are mainly twofold:

- **Relaxed atomicity**: Atomicity of the execution can be relaxed as intermediate results produced by a workflow-based application may be kept despite the failure of a service. The specification process of transactional requirements associated with workflows has to be flexible enough to support coordination scenarios more complex than the coordination rule “all or nothing” specified within the two phase commit protocol (ISO, n.d.).

- **Dynamic assignment of business partners**: Composite applications are dynamic in that the workflow partners or component services offering different characteristics can be assigned to tasks depending on the resources available at runtime. Business partners’ characteristics have thus to be combined or composed in a way such that the transactional requirements specified for the workflow are met.

Existing transactional protocols (Elmagarmid, 1992; Greenfield, Fekete, Jang, & Kuo, 2003) are not adapted to meet these two requirements as they do not offer sufficient flexibility to cope, for instance, with the runtime assignment of computational tasks. In addition, existing solutions to combine or compose service providers based on the characteristics they offer appear to be limited when it comes to integrating at the composition phase the consistency requirements defined by workflow designers. These solutions indeed only offer means to validate transactional requirements once the workflow business partners have been selected but no solution to integrate these requirements as part of the composite application building process. The next sections present our approach to overcome these limitations.

Chapter Contributions

In this chapter, we present an adaptive transactional protocol to support the execution of composite applications. The execution of this protocol takes place in two phases. First, business partners are assigned to tasks using an algorithm whereby workflow partners are selected based on functional and transactional requirements. Given an abstract representation of a process wherein business partners are not yet assigned to workflow tasks, this algorithm enables the selection of service providers not only according to functional requirements but also based on transactional ones. In our approach, these transactional requirements are defined at the workflow design stage using the acceptable termination states (ATS) model. The resulting workflow instance is compliant with the defined consistency requirements and its execution can be easily coordinated as our algorithm also provides coordination rules. The workflow execution further proceeds through a coordination protocol that leverages the coordination rules computed as an outcome of the partner assignment procedure.