Chapter 26
An Event-Based Middleware for the Management of Choreographed Services

Liliana Ardissono
Università di Torino, Italy

Roberto Furnari
Università di Torino, Italy

Giovanna Petrone
Università di Torino, Italy

Marino Segnan
Università di Torino, Italy

ABSTRACT

A critical issue in the choreographed services development is the conciliation of interaction protocol mismatches that affect the cooperation among the services to be integrated in the composite service. This chapter attempts to answer such an issue by presenting a mediation framework that supports the development and management of choreographed services by offering a loosely-coupled integration model and a flexible management of protocol mismatches. Such features are obtained by introducing (i) an action-based representation of the functions offered by a Web service; (ii) an explicit management of the state of the choreographed service, which determines the operations to be carried out in order to contribute to the service completion; (iii) an event-driven Web service execution model, which makes it possible to abandon the direct invocation of Web service operations in favour of an autonomous execution of actions, based on the available context information.

DOI: 10.4018/978-1-4666-0146-8.ch026
INTRODUCTION

Service Oriented Computing (SOC) (Papazoglou & Georgakopoulos 2003) is a new computing paradigm that utilizes services as the basic constructs to support the development of rapid, low-cost and easy composition of distributed applications in possibly heterogeneous environments. As stated in (Papazoglou et al. 2008), the vision of Service Oriented Computing is a world of cooperating services where application components are assembled into a network of loosely-coupled services to create flexible and dynamic business processes and agile applications that may span over organizations and computing platforms.

In the Service Oriented Architecture (SOA) research, two main service composition models have been defined: as described in (Peltz 2003a), web service orchestration is used to specify executable business processes that may interact with both internal and external web services at the message level, including the business logic and execution logic of the interactions. In this model, the process is controlled from the perspective of one of the business partners, which plays the orchestrator role. Differently, web service choreography has a collaborative nature and does not presuppose a centralized view of the business process: each party involved in the process describes the part it plays in the interaction and choreography is associated to the public message exchanges that occur between multiple web services. These two models answer rather different business integration needs: on the one hand, orchestration is suitable for composing services in Enterprise Application Integration, where the composite service provider has its own business logic and presupposes a stable relation with the invoked web services; however, it introduces a central bottleneck, as the orchestrator is responsible for the coordination of the activities to be carried out by all the integrated web services. On the other hand, choreography is suitable for defining composite services which cannot be controlled by a single party (e.g., for privacy concerns) and to address the bottleneck issue. In that case, the overall service does not specify business logic details, which are delegated to the individual service providers. The execution of the composite service is decentralized and only prescribes the information to be exchanged in order to achieve the expected results, abstracting from how such information is obtained. Thus, services coordinate their activities by adhering to the choreography specifications.

The idea of reusing and composing existing services highlights the need of interface adaptation: services that aim at collaborating are hampered by differences in their interfaces. This happens when services deliver their functions by means of atomic operations, but the problem becomes challenging when services adopt complex forms of interaction based on conversations. Furthermore, the problem is even more evident when composite services exploit several web services that collaborate following a complex, multi-peer conversation, such as in the case of web service choreographies (Peltz 2003b; Dijkman & Dumas 2004; Barros et al. 2005). In such a scenario a web service that aims at participating in a given choreography has to adhere to the prescribed model by implementing the required behaviour. Therefore existing services, that already implement their own observable behaviour, have to adapt to the behaviour imposed by the choreography. This hampers a real-world exploitation of choreographed services.

The middleware described in this chapter aims at providing an execution environment where web services can collaborate in a seamless way, overcoming the differences between their observable behaviour and the one expected by the choreography.

The idea is that a general run-time model for composite service management can be developed by abstracting from the flow details imposed by the message-oriented coordination and by proposing a new choreography execution model based on events. In this novel approach, data and synchro-