Chapter XII
Web Service Orchestration and Choreography:
Enabling Business Processes on the Web

Florian Daniel
Politecnico di Milano, Italy

Barbara Pernici
Politecnico di Milano, Italy

ABSTRACT

The Web service domain is a fast growing and fast changing environment. From a business perspective, the trend over the last few years in the Web services area firmly points toward seamless business logic integration and inter-enterprise collaboration. However, in order to accomplish such goals, both technological and conceptual advances are required. Some already have proven their viability, others still have to be made. Among them, Web service orchestration and choreography are of crucial importance, but still lack a widely agreed on development framework comprising both technological and conceptual aspects. In this chapter, we try to provide a critical snapshot of current standards for Web service development and particularly we focus on Web service orchestration and choreography. We discuss problems and solutions from a conceptual point of view, exemplify the illustrated ideas by means of real-world technologies and standards, and highlight the mutual dependencies that exist among orchestration and choreography of Web services.[1]

INTRODUCTION

When analyzing the current literature on Web services and the main problems the authors focus on, it is possible to identify one main trend toward the adoption of novel and emerging Web service technologies as basis for the next generation of (Web) applications and composite Web services. In this context, especially the need for flexible solutions for composing Web services into composite

applications or services is manifest. Composite applications or services leverage the functionalities provided by their individual component services by combining them in a value adding manner.

Web services are driven by the paradigm of the so-called *service-oriented architecture* (SOA), which describes the relationships that exist among service *providers*, service *consumers,* and service *brokers* and thereby provides an abstract execution environment for Web services. The research area of *service-oriented computing* (SOC) endorses the SOA paradigm and aims at producing technologies and solutions that address the efficient development, flexible composition, and execution of (composite) Web services. From their first appearance, SOA and SOC have emerged as key factors for the success of the world of Web services.

Just as the advent of *object-oriented programming* (OOP) was based on the notion of *objects* as means to modularize programming functionality, SOC could be defined as a paradigm that looks at *services* as basic functional modules that can be composed or newly defined. OOP per se did not suddenly provide revolutionary new programming capabilities with respect to conventional procedural techniques, it rather proved to be an efficient means for abstraction and isolation and thus fostered reuse, robustness, and scalability. These factors encouraged the emergence of higher-level concepts like object brokers, Java Beans, object containers, which finally enhanced interoperability.

Analogously, current specification proposals for Web services can be interpreted as a transition toward a robust SOC framework. Several Web service standardization bodies are currently addressing issues that can be interpreted as definition of a proper new programming framework. For example, even if we are already speaking about service composition and seamless inter-enterprise integration, there is still discussion over standardization of other system aspects (e.g., reliable messaging or transaction support) that have already been solved or are under study in other research areas. Past experiences taught us, however, that as long as there are no robust and commonly agreed on standards, real interoperation, and composition problems cannot be addressed adequately.

In this chapter, we will introduce the reader to the orchestration and choreography of Web services, which are becoming the cornerstones for the execution of business processes on the Web, and we will discuss the state of current research and open issues. More precisely, we will first try to clarify the main terminology in use, and then we will give an explanation for the actual need for coordination protocols and composition technologies. We will exemplify such a discussion by means of a possible protocol stack for Web service composition, and we also discuss some advanced issues. Finally, we will provide an outlook over expected future trends and draw our conclusions.

USING THE RIGHT TERMINOLOGY

Specifications and technologies for Web service composition in many cases still have to reach stable definitions and usage scenarios. Accordingly, also authors writing about service composition are far from using a commonly agreed on terminology. Peltz (2003a) defines *orchestration* as executable business process that interacts with both internal and external Web services, and *choreography* "…tracks the message sequences among multiple parties and sources—typically the public message exchanges that occur between Web services—rather than a specific business process that a single party executes…" (Peltz, 2003b).

Alonso, Casati, Kuno, and Machiraju (2004) prefer the terms coordination (protocol) and composition rather than choreography and orchestration. Literally, they clarify "…we will use the term *conversation* to refer to the sequences of operations (i.e., message exchanges) that could