Chapter II
Web Services Technology:
An Overview

Fani A. Tzima
Aristotle University of Thessaloniki, Greece

Pericles A. Mitkas
Aristotle University of Thessaloniki, Greece

ABSTRACT

This chapter examines the concept of service-oriented architecture (SOA) in conjunction with the Web services technology as an implementation of the former's design principles. Following a brief introduction of SOA and its advantages, a high-level overview of the structure and composition of the Web services platform is provided. This overview covers the core Web services specifications as well as features of the extended architecture stack, which together form a powerful and robust foundation for building distributed systems. The chapter concludes with a discussion of the scope of applicability of SOA and Web services. The overall goal of this chapter is to portray the key assets of the presented technologies and evaluate them as tools for handling adaptability, portability and interoperability issues that arise in modern business environments.

INTRODUCTION

Nowadays, organizations are facing a highly dynamic and challenging environment, characterized by a rising demand for customized, high quality services and products in several segments of business and industry. This environment, combined with the pace of technological innovation and the globalization of economy, has triggered the development of new value-creating economic paradigms, where the concept of the virtual enterprise (VE) has a central position. Sets of economic actors are combining their resources, forming temporary enterprise alliances (VEs), to effectively respond to the shifts of market demand, identify new opportunities and minimize organizational costs through cooperative and dynamic solutions.

The need to realize new forms of collaboration has forced organizations to shift their focus from intra- to inter-enterprise system and process integration. However, most enterprises have made extensive investments in system resources over the course of years and own an enormous amount of data stored in legacy enterprise information systems (EIS). Since it is impractical to discard existing EIS, there is a constant effort to evolve and enhance them. Thus, IT professionals are currently faced with the challenge of capturing and controlling legacy technology in a way that transcends organizational boundaries and heterogeneities, but also promotes system evolution. In this direction, service oriented architecture (SOA) provides a cost-effective solution, with Web services being a promising implementation, intended to enable the construction of interoperable components that can be assembled and deployed in a distributed environment (Estrem, 2003).

SERVICE-ORIENTED ARCHITECTURE (SOA)

Service-Oriented Architecture Overview

A service oriented architecture (SOA) is a design principle intended for the construction of reliable distributed systems that deliver functionality as services, with an additional emphasis on loose coupling between interacting services (Srinivasan & Treadwell, 2005). In this context, services are typically characterized by the following properties (Orchard, Ferris, Newcomer, Haas, Champion, Booth & McCabe, 2004):

- **Logical view:** The service is an abstracted, *logical* view of an actual business-level operation, defined as an implementation-independent interface. Services may be completely self-contained, or they may depend on the availability of other services, or on the existence of specific resources such as a database.
- **Message orientation*:*** A service communicates with its clients by exchanging messages and is formally defined in terms of the message exchange patterns it supports. The internal structure of the provider and requester agents is deliberately abstracted away in the SOA, in order to maintain control of which aspects of an endpoint are revealed to external services.
- **Description orientation:** A service is described by machine-processable metadata. This description only exposes information important for the use of the service, such as its capabilities, interfaces, policies and supported protocols. Further, the description documents, directly or indirectly, the semantics that will govern the interaction between the requester and provider agents.
- **Granularity:** Services tend to use a small number of operations with relatively large and complex messages. However, various levels of granularity are possible, as services may be individually useful, or they can be integrated to provide higher-level services. Among other benefits, this promotes re-use of existing functionality.
- **Network orientation:** Services tend to be oriented toward use over a network. This property emphasizes the need for services to be automatically discoverable.
- **Platform neutrality:** Messages are delivered through the interfaces using a platform-neutral and standardized format, such as XML.

Additionally, services can participate in a workflow, where the order in which messages are exchanged affects the outcome of the operations performed by a service. This notion is defined as a "service choreography" and is actually a model of the sequence of operations, states and conditions that control the interactions involved in the