ABSTRACT

The Business Process Execution Language for Web Services (BPEL) is an emerging standard for specifying the behaviour of Web services at different levels of details using business process modeling constructs. It represents a convergence between Web services and business process technology. This chapter introduces the main concepts and constructs of BPEL and illustrates them by means of a comprehensive example. In addition, the chapter reviews some perceived limitations of BPEL and discusses proposals to address these limitations. The chapter also considers the possibility of applying formal methods and Semantic Web technology to support the rigorous development of service-oriented processes using BPEL.
INTRODUCTION

Web services are a standardised technology for building and integrating distributed software systems. Web services are an incarnation of a software development paradigm known as service-oriented architectures (SOAs). Although there is no broad consensus around the definition of SOAs, it can be said that SOAs revolve around at least three major principles: (1) software systems are functionally decomposed into independently developed and maintained software entities (known as “services”); (2) services interact through the exchange of messages containing meta-data; and (3) the interactions in which services can or should engage are explicitly described in the form of interfaces.

At present, the first generation of Web service technology has reached a certain level of maturity and is experiencing increasing levels of adoption, especially in the context of business applications. This first generation relies on XML, SOAP and a number of so-called WS-* specifications for message exchange (Curbera, Duftler, Khalaf, Nagy, Mukhi, & Weerawarana, 2002), and on XML Schema and WSDL for interface description. In the meantime, a second generation of Web services, based on richer service descriptions is gestating. Whereas in first-generation Web services, interface descriptions are usually equated to sets of operations and message types, in the second generation the description of behavioural dependencies between service interactions (e.g., the order in which messages must be exchanged) plays a central role.

The Business Process Execution Language for Web Services (BEA Systems, Microsoft, IBM, & SAP, 2003), known as BPEL4WS or BPEL for short, is emerging as a standard for describing the behaviour of Web services at different levels of abstraction. BPEL is essentially a layer on top of WSDL and XML Schema, with WSDL and XML Schema defining the structural aspects of service interactions, and BPEL defining the behavioural aspects. To capture service behaviour, BPEL adopts principles from business process modeling. Indeed, the central idea of BPEL is to capture the business logic and behavioural interface of services in terms of process models. These models may be expressed at different levels of abstraction, down to the executable level. At the executable level, BPEL can be used to describe the entire behaviour of a new Web service that relies on several other services to deliver its functionality. This practice is known as service composition (Casati & Shan, 2001). An example of a composite service is a travel booking system integrating flight booking, accommodation booking, travel insurance, and car rental Web services.

In this chapter, we introduce BPEL by illustrating its key concepts and the usage of its constructs to define service-oriented processes and to model business protocols between interacting Web services. We also discuss some perceived limitations of BPEL and extensions that have been proposed by industry vendors to address these limitations. Finally, we review some research related to BPEL and conclude with a note on future directions.

WHY BPEL?

BPEL supports the specification of service-oriented processes, that is, processes in which each elementary step is either an internal action performed by a Web service or a communication action performed by a Web service (sending and/or receiving a message). They can be executed to implement a new Web service as a concrete aggregation of existing services to deliver its functionality (i.e., composite Web service). For example, a service-oriented process may specify that when a “Sales” Web service receives a “purchase order” from the “Procurement” Web service of a customer, the Sales service engages in a number of interactions with the “Procurement” Web service as well as several other Web...
34 more pages are available in the full version of this document, which may be purchased using the “Add to Cart” button on the product's webpage:
www.igi-global.com/chapter/service-oriented-processes/28884?camid=4v1

www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

Behaviour-Aware Discovery of Web Service Compositions
www.igi-global.com/article/behaviour-aware-discovery-web-service/3102?camid=4v1a

Automatic Determination of Compatibility in Evolving Services
www.igi-global.com/article/automatic-determination-compatibility-evolving-services/50491?camid=4v1a

A Federated Approach to Information Management in Grids
www.igi-global.com/chapter/federated-approach-information-management-grids/66955?camid=4v1a

Advances in Privacy Preserving Record Linkage
www.igi-global.com/chapter/advances-privacy-preserving-record-linkage/53271?camid=4v1a