Chapter II
A Retrospective on the Development of Web Service Specifications

Shrideep Pallickara  
Community Grids Lab, Indiana University, USA

Mehmet S. Aktas  
Community Grids Lab, Indiana University, USA

Beytullah Yildiz  
Community Grids Lab, Indiana University, USA

Sima Patel  
Community Grids Lab, Indiana University, USA

Damodar Yemme  
Wheaton Van Lines Inc., USA

Geoffrey Fox  
Community Grids Lab, Indiana University, USA

Harshwardhan Gadgil  
Community Grids Lab, Indiana University, USA, & Amazon.com, USA

Marlon E. Pierce  
Community Grids Lab, Indiana University, USA

Sangyoon Oh  
SK Telecom, Korea

ABSTRACT

In this chapter, we present a discussion on our experiences with the development of Web service specifications. Web services, and the service oriented architecture model engendered therein, have gained significant traction in recent years with deployments in ever increasing domains. In this chapter, we describe our experiences with several Web service specifications. In general lessons learned, and design decisions made, during these implementations would be applicable to several other specifications. The authors hope that their insights and experiences with the development of Web service specifications would be beneficial to other researchers in this area in formulating a strategy for the development of systems based on Web services.
INTRODUCTION

Web services have gained considerable traction over the past several years, and are being increasingly leveraged within the academic, business, and research communities. The service oriented architecture (SOA) model engendered within Web services provides a simple and flexible framework for building sophisticated applications. A slew of specifications addressing several core areas, such as reliable messaging, addressing, security, and so forth, within distributed systems have emerged recently. The term WS-* is used as an umbrella term to collectively refer to these specifications. The use of XML throughout the Web services stack of specifications facilitates interactions between services implemented in different languages, running on different platforms, and over multiple transports. This use of XML distinguishes Web services from previous efforts such as CORBA (common object resource broker architecture) to simply building distributed systems.

In this chapter, we describe our experiences with several Web service specifications. In general lessons learned, and design decisions made, during these implementations would be applicable to several other specifications. We begin this chapter with some observations regarding Web service specifications. This includes a discussion on the SOAP-centric (simple object access protocol) (Gudgin, 2003) nature of the specifications, their reliance on one-way asynchronous message exchanges, how a specification can itself leverage other specifications, and, finally, how these specifications are intended to be stackable to facilitate use in tandem with each other. Several specifications leverage the WS-addressing (Box, 2004a) specification, and this specification has, in recent years, become the de facto standard to target Web service instances; we also include a brief description of this specification.

WS-* specifications are XML-based and have schemas associated with them. In the section on Processing XML Schemas we describe the choices available to designers for processing these schemas. In subsequent sections, we describe the implementation strategy for various specifications that we have implemented. These include WS-reliable messaging (Bilorusets, 2004) (hereafter WSRM), WS-reliability (Oasis-WSR, 2004) (hereafter WSR), WS-eventing (Box, 2004b) (hereafter WSE), WS-Context [7], the universal description, discovery and integration (UDDI) (Bunting, 2003), WS-management (Arora, 2005) and WS-transfer (Alexander, 2004a). Depending on the interactions and exchanges that are part of these specifications, the complexity of the implementation and corresponding deployments varies.

The WSRM and WSR specifications pertain to providing support for reliable messaging between Web service endpoints. These aforementioned specifications guarantee delivery of messages in the presence of failures and disconnects; endpoints can also retrieve lost messages after a failure. The WSE specification provides a mechanism for routing notifications from the producers to the registered consumers. Consumers can register their interest in specific messages using XPath queries; only messages that satisfy the previously specified constraint are routed to a consumer. Implementation of the WSRM, WSR, and WSE specifications are outlined in a separate section.

WS-management facilitates the efficient management of distributed systems; this specification identifies a core set of Web service specifications and usage requirements to expose a common set of operations central to all systems management. Our implementation is described in the section on management within distributed systems.

WS-context models session metadata as an external entity where more than two services can
Related Content

A New Encryption Algorithm based on Chaotic Map for Wireless Sensor Network
www.igi-global.com/chapter/new-encryption-algorithm-based-chaotic/78868?camid=4v1a

A Pragmatic Approach to Intrusion Response Metrics
www.igi-global.com/chapter/pragmatic-approach-intrusion-response-metrics/65760?camid=4v1a

Supply Risk Management Process: Modeling Enablers to Develop a Structural Framework
www.igi-global.com/article/supply-risk-management-process/120555?camid=4v1a

Vulnerability Analysis and Defenses in Wireless Networks
www.igi-global.com/chapter/vulnerability-analysis-defenses-wireless-networks/22045?camid=4v1a