

Chapter 81

Considerations of Adapting Service–Offering Components to RESTful Architectures

Michael Athanasopoulos

National Technical University of Athens, Greece

Kostas Kontogiannis

National Technical University of Athens, Greece

Chris Brealey

IBM Canada, Canada

ABSTRACT

Over the past few years, we have witnessed a paradigm shift on the programming models and on architectural styles, which have been used to design and implement large-scale service-oriented systems. More specifically, the classic message-oriented and remote procedure call paradigm has gradually evolved to the resource-oriented architectural style, inspired by concepts pertinent to the World Wide Web. This shift has been primarily driven by multifaceted functional and non-functional requirements of Web enabled large-scale service offering systems. These requirements include enhanced interoperability, lightweight integration, scalability, enhanced performance, even looser coupling, and less dependence on shifting technology standards. As a consequence, several, and sometimes antagonistic, architectures, design patterns, and programming paradigms have emerged on a quest to overcome the constantly expanding enterprise software needs. In the context of resource-oriented architectures, the Representational State Transfer (REST) architectural style has gained considerable attention due to its simplicity, uniformity, and flexibility. More specifically, the potential for scalability and loose coupling, the uniformity of interfaces, and the efficient bridging of enterprise software systems with the Web are significant factors for software architects and engineers to consider REST when designing, implementing, composing, and deploying service-oriented systems. These issues stir discussion among academics and practitioners about how to properly apply REST constraints both with respect to the development of new enterprise systems and to the migration and adaptation of existing service-oriented systems to RESTful architectures. In this chapter, the authors discuss issues and challenges related to the adaptation of existing service-oriented

DOI: 10.4018/978-1-4666-4153-2.ch081

systems to a RESTful architecture. First, they present the motivation behind such an adaptation need. Second, the authors discuss related adaptation theory, techniques, and challenges that have been recently presented in the research literature. Third, they identify and present several considerations and dimensions that the adaptation to REST entails, and the authors present frameworks to assess resource-oriented designs with regard to compliance to REST. Fourth, the authors introduce an adaptation framework process model in the context of enterprise computing systems and technologies, such as Model Driven Engineering and Service Component Architecture (SCA). Furthermore, they discuss open challenges and considerations on how such an adaptation process to REST can be extended, in order to yield systems that best conform to the REST architectural style and the corresponding REST constraints. Finally, the chapter is concluded with a summary and a discussion on the points raised and on some emerging trends in this area.

INTRODUCTION

During the past decade, service-orientation has become the dominant computing paradigm in the domain of enterprise software systems. More specifically, it has been argued that Service-Oriented Architectures (SOAs) provide significant benefits to organizations, and generally allow for better alignment of business needs and IT solutions. The fundamental principle is that SOAs organize functionality as collections of interoperable services with standardized interface specification and description methods. Furthermore, service communication is independent of implementation and infrastructure allowing thus, for heterogeneous systems to communicate effectively, and for lowering costs related to integration and interoperation. Even though, SOA as a set of architectural principles is not bound to any specific technology, W3C's Web Services technologies and standards have been the primary choice of architects when implementing SOAs, especially for enterprise and B2B systems. For example, it has emerged as de facto standard that Web Service components are described by specifications written in the Web Service Description Language (WSDL), and invoked by utilizing the SOAP family of protocols. These service components usually follow one of two binding styles: Remote Procedure Call (RPC) and document-based message exchange. The RPC binding style explicitly references the service operation to be invoked, while document-based

binding style promotes the usage of messages that include schema-based elements. These schema-based elements are interpreted by the receiver (i.e. server) in order to dispatch and invoke the appropriate operation. An advantage of this approach is that it is easier to validate the requests since there is a direct reference to a schema to which the message is supposed to conform to. As a result, although RPC was considered to be the dominant style when Web Services were first introduced, tools and frameworks started to support a document-based invocation style. However, in practice has been proven that the inclusion of the operation's name in the message is quite significant for the interoperability and consequently, many document-based Web Service frameworks adopted and implemented a special pattern of document-based style called the "wrapped document" pattern. In the "wrapped document" pattern the message that is sent, is essentially a schema element that is named after the operation's name, and the message that is received is also a schema element whose name follows a similar operation-based convention. In this context, a procedure is known to the service consumer, and it is readily addressable and available to be invoked by service messages, regardless of the particular binding or even technology the service may use to expose its functionality. Services that are published following the procedure-oriented Web Services stack of protocols are integrated over the Web and not through the Web—making

26 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/considerations-adapting-service-offering-components/77288

Related Content

eLearning Project Management for Innovation Management: Team Project-Based eLearning and Assessment at the IT Institute

Niki Lambropoulos, Alain Gourdin, Marcella Soamiadana, Sophi Danisand Aneesha Bakharia (2013). *Enterprise Resource Planning: Concepts, Methodologies, Tools, and Applications* (pp. 959-978). www.irma-international.org/chapter/elearning-project-management-innovation-management/77263

The Myth of Integration: A Case Study of an ERP Implementation

Rosio Alvarez (2002). *Enterprise Resource Planning: Global Opportunities and Challenges* (pp. 17-42). www.irma-international.org/chapter/myth-integration-case-study-erp/18462

Knowledge Management and Project Management in 3D: A Virtual World Extension

Steve Russell (2013). *Enterprise Resource Planning: Concepts, Methodologies, Tools, and Applications* (pp. 1452-1467). www.irma-international.org/chapter/knowledge-management-project-management/77284

Adaptive Learning Cycle to Improve the Competence-Building for Enterprise Systems in Higher Education

Dirk Peters, Liane Haakand Jorge Marx Gómez (2013). *Enterprise Resource Planning: Concepts, Methodologies, Tools, and Applications* (pp. 557-580). www.irma-international.org/chapter/adaptive-learning-cycle-improve-competence/77239

Fundamental Building Blocks for Security Interoperability in e-Business

Muhammad Asimand Milan Petkovic (2013). *Enterprise Resource Planning: Concepts, Methodologies, Tools, and Applications* (pp. 118-141). www.irma-international.org/chapter/fundamental-building-blocks-security-interoperability/77215