THE PRENTICE HALL SERVICE-ORIENTED COMPUTING SERIES FROM THOMAS ERL

Web Service Contract Design & Versioning for SOA

Thomas Erl, Anish Karmarkar, Priscilla Walmsley,
Hugo Haas, Umit Yalcinalp, Canyang (Kevin) Liu,
David Orchard, Andre Tost, James Pasley

PRENTICE HALL

Erl_00FM.qxd 8/9/08 11:14 AM Page v

Web Service Contract Design and Versioning for SOA

Thomas Erl, Anish Karmarkar, Priscilla Walmsley,
Hugo Haas, Umit Yalcinalp, Canyang Kevin Liu,
David Orchard, Andre Tost, James Pasley

PRENTICE HALL
UPPER SADDLE RIVER, NJ • BOSTON • INDIANAPOLIS • SAN FRANCISCO
NEW YORK • TORONTO • MONTREAL • LONDON • MUNICH • PARIS • MADRID
CAPETOWN • SYDNEY • TOKYO • SINGAPORE • MEXICO CITY

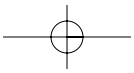
Erl_01.qxd 7/24/08 12:21 PM Page 1

Chapter 1

Introduction

1.1 About this Book
1.2 Objectives of this Book
1.3 Who this Book Is For
1.4 What this Book Does Not Cover
1.5 Prerequisite Reading
1.6 Supplementary Reading
1.7 How this Book Is Organized
1.8 Symbols, Figures, and Style Conventions
1.9 Additional Information

Erl_01.qxd 7/24/08 12:21 PM Page 2



1.1 About this Book

The notion of a contract is something most of us are very familiar with. It's an agreement between two parties on a set of terms usually to govern the exchange of something. Without contracts, the world (especially the business world) could not operate the way it does today. Each contract that two parties adhere to guarantees that an exchange is carried out predictably, as expected.

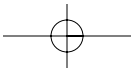
It is this type of predictability that we are very interested in when establishing contracts within technical environments. When we design two software programs to exchange information, we want to ensure that both follow the terms established by the technical contract every time. In fact, this makes predictability not just a preference but a requirement for software design.

But it's not just the terms of exchange that we want to have documented in some form; we also need to establish the interface used to carry out the exchange. As with contracts, interfaces are also part of everyday life.

A remote control, an automobile dashboard, a calculator, a computer keyboard—each of these items represents an interface into something. They are interfaces designed to be used by humans to issue commands into a machine or a device. As such, they establish a pre-defined, standardized means of issuing these commands.

They also represent and express the contract that we must comply to in order to get what we want from whatever it is we are interfacing with. If you don't press the right buttons on the remote control, you won't get to view what you're looking for. Similarly, if you don't turn the steering wheel in the right direction, the car won't go where you want it to.

Interfaces and the contracts they express are essential to just about everything we need to do in order to function in society. They are just as fundamental to software program design in the service-oriented computing world. Whereas older systems may have been designed as self-contained, monolithic programs with no need to expose technical interfaces, service-oriented solutions are naturally partitioned into sets of software programs (called "services") that must interface with each other.



Erl_01.qxd 7/24/08 12:21 PM Page 3

But, there's more to it than that. When applying service-orientation to design software programs as services, most service contracts will need to facilitate interaction with not just one, but a range of potential programs. To realistically enable this level of flexibility within a technical interface means that its design will demand extra attention and care.

Requirements like this are nothing new. The aforementioned remote control and automobile dashboard are almost always built to accommodate numerous types of human users. The better designed these interfaces are, the more people will be able to use them effectively, and the more consistent and predictable the usage experience will be.

When building Web services for SOA, this is exactly what we need to accomplish. The services must have balanced, effective technical contracts that will allow numerous programs to use (and reuse) the functionality the services have to offer.

But, there's still more to it than that. Delivering a well-designed service contract is the first major step in a service lifecycle that can easily span years. During that period, change is inevitable. Although common SOA methodologies will help produce service contracts with increased longevity, their lifespans are, more often than not, limited.

We therefore need to be prepared for this eventuality by having solid governance practices in place that can help evolve a service through times of change. This will extend the lifespan of the service itself beyond the lifespans of its contracts.

This book is exclusively focused on the design and evolution of Web service contracts. The upcoming chapters have been authored by some of the top experts in the fields of SOA, Web service technologies, service contract design, and service versioning and governance.

1.2 Objectives of this Book

Collectively, these chapters were written with the following primary goals in mind:

- to document Web service contract-related technologies within the context of SOA
- to highlight proven contract-related techniques and patterns for contract design and versioning
- to demonstrate how first-generation Web service technologies (WSDL, SOAP, XML Schema) work together with WS-* technologies, such as WS-Addressing and WS-Policy
- to highlight how the application of various Web service technologies can be influenced by SOA design principles and patterns

Erl_01.qxd 7/24/08 12:21 PM Page 4

1.3 Who this Book Is For

This book can be used as a tutorial and a reference text and is intended for the following types of readers:

- developers who want to learn how to work with Web service technologies as part of SOA projects
- architects who want to learn how to design Web service contracts in support of SOA projects
- governance specialists who want to learn proven practices for service contract versioning
- SOA practitioners who want to better understand how to build Web service contracts in support of service-orientation
- IT professionals who want to gain a better insight into the concepts and mechanics that underlie modern Web service contracts and messages

1.4 What this Book Does Not Cover

This is a book about Web service contracts only. It explores a wide range of technologies and techniques that pertain to the development, design, and versioning of Web service contracts and related message design topics. However, this book does not get into the development or implementation of Web service programs. Therefore, wire-level topics, such as reliable messaging, security, and transactions, are not covered.

Similarly, while the overarching context of this book is SOA, there is only one chapter dedicated to explaining fundamental terms and concepts. If you are new to SOA, be sure to read the resources recommended in the following *Prerequisite Reading* section.

1.5 Prerequisite Reading

This book assumes you have a basic knowledge of fundamental XML concepts. If you have not yet worked with XML, you can begin catching up by reading some of the brief tutorials published at www.xmlenterprise.com.

If you are new to SOA, you can get a basic understanding of service-oriented computing, service-orientation, and related design patterns by studying the content at the following Web sites:

Erl_01.qxd 7/24/08 12:21 PM Page 5

- www.whatissoa.com
- www.soaprinciples.com
- www.soapatterns.org

To further ensure that you have a clear understanding of key terms used and referenced in the upcoming chapters, you can also visit the online master glossary for this book series at www.soaglossary.com to look up definitions for terms that may not be fully described in this book.

Even if you are an experienced SOA practitioner, we suggest you take the time to have a look at these online resources. A great deal of ambiguity has surrounded SOA and service-oriented computing and these explanations and definitions will ensure that you fully understand key terms and concepts in relation to this book and the book series as a whole.

1.6 Supplementary Reading

Here are some recommendations for additional books that elaborate on key topics covered by this title:

- *SOA Principles of Service Design* – A comprehensive documentation of the service-orientation design paradigm with full descriptions of all of the principles referenced in this book.
- *SOA Design Patterns* – This book provides a comprehensive catalog of design patterns, many of which are specific to Web service contract design and versioning. You can look up concise descriptions for these patterns at www.soapatterns.org.
- *Definitive XML Schema* – This classic title provides complete coverage of the XML Schema language, including several advanced topics outside of the scope of the upcoming chapters dedicated to XML Schema-based message design.
- *Service-Oriented Architecture: Concepts, Technology, and Design* – The coverage of service-oriented analysis and design processes in this title supplements the technology-centric focus of this book with methodology-related topics.

The following titles are currently in development as part of the *Prentice Hall Service-Oriented Computing Series from Thomas Erl*:

Erl_01.qxd 7/24/08 12:21 PM Page 6

- *SOA with .NET* – A book dedicated to building services and service-oriented solutions with .NET development tools and technologies, with an emphasis on Web services and REST services.
- *SOA with Java* – As with the previously listed title, this book is about developing service-oriented solutions with Web services and REST services, except the focus is on the use of Java technologies and platforms.
- *SOA Governance* – This book will explore a wide range of organizational and technological governance topics, including Web service contract versioning and evolution.
- *ESB Architecture for SOA* – Several of the policy-related topics covered in the upcoming chapters will be further explored in relation to ESB architectures in this title.

For the latest information regarding the release of these new books, visit www.soabooks.com.

1.7 How this Book Is Organized

This book begins with Chapters 1 and 2 providing introductory content and case study background information, respectively. All subsequent chapters are grouped into the following parts:

- Part I: Fundamental Service Contract Design
- Part II: Advanced Service Contract Design
- Part III: Service Contract Versioning
- Part IV: Appendices

Part I: Fundamental Service Contract Design

Chapter 3: SOA Fundamentals and Web Service Contracts

We begin with an overview of key terms and concepts associated with SOA, service-orientation, and service-oriented computing in general as they pertain to Web service contract design.

Erl_01.qxd 7/24/08 12:21 PM Page 7

Chapter 4: Anatomy of a Web Service Contract

This introductory chapter provides a visual exploration of Web service contract structures from both logical and physical perspectives. The focus is on establishing a conceptual understanding of the various parts that can comprise a Web service contract and the mechanics that make these parts work together.

Concepts pertaining to abstract and concrete descriptions are discussed in relation to message, operation, and port type (interface) definitions, along with different message parts, such as the body, header, and headerfault. Also covered are the binding, service, and port (endpoint) sections and how they relate to message parts (body, header, headerfault), envelope structure, and each other.

This chapter contains no code examples and only begins discussing the actual technologies used to build Web service contracts after the basics have been covered in abstract. The final section concludes the chapter with a series of guidelines.

Chapter 5: A Plain English Guide to Namespaces

As an integral part of the XML-based Web services framework, both custom and predefined namespaces are used to ensure that all of the technologies covered in this book can harmoniously work together. Therefore, before we get into the actual technology languages used to build Web service contracts, we first provide an informal but in-depth tutorial about namespaces.

Because the topic of namespaces represents a common point of confusion for many IT professionals, this chapter takes extra time to explain the concepts and applications of namespaces in plain English. A range of examples and perspectives are provided so that you are fully prepared for any namespace-related issues discussed in subsequent chapters.

Chapter 6: Fundamental XML Schema: Types and Message Structure Basics

We now begin to get into the details of contract design with a chapter dedicated to describing basic XML Schema topics, as they pertain to message design for Web service contracts. The emphasis is on different types of constraints for complex and simple types. Also provided is coverage of how XML Schema types can be customized and extended using type inheritance features.

Erl_01.qxd 7/24/08 12:21 PM Page 8

This chapter also officially kicks off the case study by introducing a set of problems that are solved via the application of XML Schema features. The examples explored ultimately lead to the creation of a set of XML Schema types that will be used by message definitions in the following chapters.

Chapter 7: Fundamental WSDL Part I: Abstract Description Design

In this chapter we discuss all aspects of the Web Services Description Language (WSDL) 1.1 that relate to designing and building the parts that comprise the abstract service description. This includes the association of XML Schema types with message definitions, the organization of messages into operation definitions, and the grouping of operations within port type definitions. Also covered is the creation of message headers and the usage of the various namespaces required for different technologies to co-exist within a WSDL document.

Through a series of informal case study examples a complete abstract description is created, incorporating the XML Schema types that were defined in Chapter 6.

Chapter 8: Fundamental WSDL Part II: Concrete Description Design

We now move on to the design and development of the concrete description, the part of the WSDL document that binds the abstract description to actual transport and messaging protocols and also assigns it one or more physical network addresses.

This chapter covers all of the primary WSDL 1.1 language elements used to establish this binding, including the service and port definitions, as well as the important binding construct itself. A great deal of content is dedicated to exploring how these native WSDL elements are further supplemented via the use of extensibility elements from other languages, such as SOAP. How header and headerfault message parts are bound to actual SOAP message areas is also described.

The case study examples continue and culminate in the creation of a complete WSDL definition that includes both the abstract and concrete descriptions.

Chapter 9: Fundamental WSDL 2.0: New Features and Design Options

This book also covers the WSDL 2.0 standard, which introduces new features and imposes a series of syntactical changes to the WSDL 1.1 language. Through a series of examples, this chapter documents the differences between WSDL 1.1 and 2.0 as they pertain to both abstract and concrete descriptions, and then concludes with a complete WSDL 2.0 version of the WSDL document from Chapter 8.

Erl_01.qxd 7/24/08 12:21 PM Page 9

Chapter 10: Fundamental WS-Policy: Expression, Assertion, and Attachment

Through the creation of policy assertions and policy expressions, the WS-Policy language provides various means of extending the technical interface that you can create with WSDL and XML Schema.

This chapter introduces the basics of how policies can be defined and expressed and then associated (attached) to different parts of a WSDL document. Topics include policy subjects, policy alternatives, composite policies, and embedded and external attachment options.

The examples again lead to a revised WSDL definition that includes a simple, attached policy expression.

Chapter 11: Fundamental Message Design: SOAP Envelope Structure and Header Block Processing

We now turn our attention to the SOAP technology language that is used to express the structure of messages as they exist "on the wire," when they are transmitted to and from Web services. Although this book is dedicated to the Web service contract only, it is important to understand how messages will be processed in order for us to design them effectively.

A primary topic in this chapter is the runtime usage of SOAP header blocks and the various roles and options that exist to carry out "targeted processing" where header blocks are designed to be accessed and used only by certain programs along a message path.

Note that this chapter does not explain the SOAP communications framework or any other topics that do not pertain directly to Web service contract design.

Part II: Advanced Service Contract Design

Chapter 12: Advanced XML Schema Part I: Message Flexibility and Type Inheritance and Composition

There are a variety of messages that a Web service will need to exchange, each with its own unique requirements. This, the first of two chapters dedicated to advanced XML Schema topics, explores the usage of wildcards, extension buckets, and content model groups and documents a variety of techniques that use these XML Schema features to accommodate different types of Web service message designs.

The chapter then continues with an explanation of type inheritance and composition and further discusses how cross-schema inheritance, as a concept, may have different

Erl_01.qxd 7/24/08 12:21 PM Page 10

implications than cross-service inheritance. Specifically, this part of the chapter provides examples that detail how abstract schemas can be created and how schemas in general can be extended.

Finally, a discussion of message design in support of CRUD-style Web service operations is provided. Individual Add, Get, Delete, and Update message designs are described, along with common granularity levels.

Chapter 13: Advanced XML Schema Part II: Reusability, Derived Types, Relational Design

The ability to share schemas across Web services is fundamental to establishing separate schema and contract architecture layers. This chapter delves into the various options by which schemas can be partitioned into reusable modules and then included or imported by different schemas (and, ultimately, different WSDL definitions). The creation of a common schema library is furthermore discussed and demonstrated.

Next are sections dedicated to explaining how relational data structures can be simulated using XML Schema features and also how narrative content can be added to supplement base schema content. Finally, the usage and incorporation of pre-defined industry schemas is explored along with several examples and techniques.

Chapter 14: Advanced WSDL Part I: Modularization, Extensibility, MEPs, and Asynchrony

Both WSDL 1.1 and 2.0 modularization features and techniques are covered and demonstrated in this chapter, along with issues pertaining to non-transitive limitations that also relate the reuse of XML Schema definitions across WSDL definitions. The use of WSDL import and include are compared, along with WSDL 2.0 interface inheritance design options.

This is followed by a comprehensive exploration of various WSDL extensibility elements and how they can be used and positioned within the WSDL definition structure, with an emphasis on the WSDL binding construct.

Subsequent sections discuss special message exchange patterns (including WSDL 1.1 outbound MEPs and the WSDL 2.0 Robust In-Only MEP) and various advanced techniques for asynchronous message exchanges involving HTTP, SOAP, and SMTP. Finally, a section dedicated to WS-BPEL related extensibility elements is provided along with recommendations as to how to best prepare WSDL definitions for eventual composition by WS-BPEL process definitions.

Erl_01.qxd 7/24/08 12:21 PM Page 11

Chapter 15: Advanced WSDL Part II: Message Dispatch, Service Instance Identification, and Non-SOAP HTTP Binding

Various advanced WSDL techniques are explored in this chapter, including the use of new WSDL 2.0 features that allow for broad message coupling. A section dedicated to message dispatch design issues is also provided, along with guidelines as to how to leverage the SOAP Action value to support interpretation of WSDL definitions for dispatch purposes.

Additional topics include different approaches for Web service instance identification and creating header blocks, including the use of the header fault extension. Finally, the binding of WSDL to HTTP without SOAP is explored and demonstrated for both WSDL 1.1 and 2.0.

Chapter 16: Advanced WS-Policy Part I: Policy Centralization and Nested, Parameterized, and Ignorable Assertions

Continuing the coverage of the WS-Policy framework that began in Chapter 10, we now delve into various architectural issues, including the application of the Policy Centralization design pattern and different potential approaches for associating separate policy definition documents with WSDL definitions.

Subsequent parts of this chapter demonstrate the use of nested and parameterized policy expressions and then provide detailed coverage of ignorable policy assertions, including a comparison with optional policy assertions. The chapter concludes with an overview of concurrent policy-enabled Web service contracts.

Chapter 17: Advanced WS-Policy Part II: Custom Policy Assertion Design, Runtime Representation, and Compatibility

This chapter describes the design process and implications behind creating custom policy assertions. Various examples demonstrate different types of assertions and the pros and cons of customizing policies are further explained. A separate section is provided, dedicated to issues pertaining to the maintenance of custom policy assertion vocabularies.

Later in the chapter, the runtime representation or policy syntax is explored to provide insight as to how policy expressions and alternatives are streamlined into normalized representations. Finally, an intersection algorithm is explained to identify compatibility between service provider and consumer policies.

Erl_01.qxd 7/24/08 12:21 PM Page 12

Chapter 18: Advanced Message Design Part I: WS-Addressing Vocabularies

WS-Addressing provides an industry-standard means of extending the base SOAP message design to accommodate a wide range of complex message exchange requirements.

This chapter formally introduces the WS-Addressing standard by individually describing the language elements and SOAP headers established by the Endpoint References (EPR) and Message Addressing Properties (MAP) vocabularies. The chapter provides numerous code samples and concludes with a case study example comprised of a complete set of WS-Addressing headers.

Chapter 19: Advanced Message Design Part II: WS-Addressing Rules and Design Techniques

Various, more specialized topics in relation to WS-Addressing are covered in this chapter, including EPR parameter mapping techniques, design issues relating to the binding of Endpoint References with WSDL definitions, and a detailed comparison of WS-Addressing Action values to alternative SOAP Action expressions. The chapter concludes with descriptions of policy assertions provided and standardized by the WS-Addressing specification.

Part III: Service Contract Versioning

Chapter 20: Versioning Fundamentals

Basic concepts and terminology associated with Web service contract versioning are established in this chapter, together with a detailed explanation of forward and backward compatibility from both technical and governance perspectives.

Version identification is then described and demonstrated through the use of annotated version numbers and namespace values. The chapter ends with sections that describe three primary versioning strategies (Strict, Flexible, Loose) that each support forward and backward compatibility to different extents. These strategies will be used for reference purposes throughout subsequent chapters.

Chapter 21: Versioning WSDL Definitions

This chapter explores different techniques for versioning different parts of a WSDL definition with an emphasis on the versioning of WSDL operations. Separate sections provide versioning guidance and examples for adding, renaming, and removing operations as well as modifying operation message exchange patterns and associated fault messages.

Erl_01.qxd 7/24/08 12:21 PM Page 13

Also provided are techniques for versioning port type definitions and various parts of the concrete description. The last section describes what parts of the WSDL language can be used to potentially support forward compatibility.

Chapter 22: Versioning Message Schemas

The complex topic of XML Schema-based message versioning is tackled in this chapter, which groups techniques as per the three versioning strategies established in Chapter 20. Versioning approaches to adding, removing, renaming, and modifying existing XML Schema components are documented separately in relation to each versioning strategy.

Throughout these sections various XML Schema language features are explored, including the use of wildcards for forward compatibility and optional elements to accommodate backward compatibility. Another primary topic addressed throughout this chapter is the use of XML Schema namespaces for versioning purposes.

Chapter 23: Advanced Versioning

This final chapter provides a mixed bag of versioning guidelines, techniques, and design considerations. It starts off with versioning strategies specifically for WS-Policy expressions and assertions, and then continues with design methods that use policy assertions to express termination information and explore the utilization of custom attributes to express non-ignorable unknown elements (when using wildcards).

Various other versioning approaches are covered, several based on actual contract versioning design patterns, such as Partial Understanding. The chapter concludes with a series of tips for creating and customizing your own versioning strategy.

Part IV: Appendices

Appendix A: Case Study Conclusion

This appendix provides a brief summary and conclusion of the case study storyline, as it pertains to the Web service contracts designed within the ActionCon IT enterprise environment.

Appendix B: How Technology Standards Are Developed

A concise overview of how the W3C organizes its technical committees is provided, along with descriptions of the processes and stages through which submitted specifications are developed into ratified standards.

Erl_01.qxd 7/24/08 12:21 PM Page 14

Appendix C: Alphabetical Pseudo Schema Reference

Skeleton pseudo Schemas are supplied for all of the language elements explained in this book. These Schemas are listed in alphabetical order for quick reference purposes.

Appendix D: Namespaces and Prefixes Used in this Book

A reference list of the namespaces and associated prefixes used in previous chapters.

Appendix E: SOA Design Patterns Related to This Book

Concise descriptions of each of the design patterns referenced and discused throughout this book. Note that these pattern descriptions are also available online at www.soapatterns.org.

1.8 Symbols, Figures, and Style Conventions

Symbol Legend

This book contains a series of diagrams that are referred to as *figures*. The primary symbols used throughout all figures are individually described in the symbol legend located on the inside of the front cover.

How Color Is Used

The color red is occasionally used to highlight text, especially within code samples. Generally, the highlighted code will be related to the current topic being discussed.

1.9 Additional Information

The following sections provide supplementary information and resources for the *Prentice Hall Service-Oriented Computing Series from Thomas Erl*.

Official Book Series Site (www.soabooks.com)

The latest details about all books in this series and various supporting resources can be found at soabooks.com. Be sure to also check for updates, errata, and resources, such as supplementary posters.

Erl_01.qxd 7/24/08 12:21 PM Page 15

Visio Stencil (www.soabooks.com)

Prentice Hall has produced a Visio stencil containing the color symbols used by the books in this series. This stencil can be downloaded at www.soabooks.com.

Community Patterns Site (www.soapatterns.org)

All of the pattern profile summary tables documented in this book are also published online at soapatterns.org, as part of an open site for the SOA community dedicated to SOA design patterns. This site allows you to provide feedback regarding any of the design patterns and to further submit your own pattern candidates.

Master Glossary (www.soaglossary.com)

To ensure constant content currency, a dedicated Web site at soaglossary.com provides a master online glossary for all series titles. This site continues to grow and expand with new glossary definitions as new series titles are developed and released.

Referenced Specifications (www.soaspecs.com)

The chapters throughout this book reference XML and Web services specifications and standards. The www.soaspecs.com Web site provides a central portal to the original specification documents created and maintained by the primary standards organizations.

Supplementary Posters (www.soaposters.com)

Color reference posters are available for free download at www.soaposters.com. The poster for this book is primarily comprised of the symbols and diagrams in Chapter 4 to provide an overview of how Web service contracts are structured.

The SOA Magazine (www.soamag.com)

The SOA Magazine is a regular publication provided by SOA Systems Inc. and Prentice Hall/PearsonPTR and is officially associated with the *Prentice Hall Service-Oriented Computing Series from Thomas Erl.* The SOA Magazine is dedicated to publishing specialized SOA articles, case studies, and papers by industry experts and professionals. The common criteria for contributions is that each explore a distinct aspect of service-oriented computing.

Erl_01.qxd 7/24/08 12:21 PM Page 16

Referenced Specifications (www.soaspecs.com)

Various series titles reference or provide tutorials and examples of open XML and Web services specifications and standards. The www.soaspecs.com Web site provides a central portal to the original specification documents created and maintained by the primary standards organizations.

Notification Service (www.soabooks.com)

If you'd like to be automatically notified of new book releases in this series, new supplementary content for this title, or key changes to the previously listed Web sites, use the notification form at www.soabooks.com.