

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Service-Oriented Architecture

Concepts, Technology, and Design

Thomas Erl



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

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The authors and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U. S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the U. S., please contact:

International Sales
international@pearsoned.com

Visit us on the Web: www.phptr.com

Library of Congress Number: 2005925019

Copyright © 2005 Pearson Education, Inc. Portions of this work are copyright SOA Systems Inc., and reprinted with permission from SOA Systems Inc. © 2005. Front cover and all photographs by Thomas Erl. Permission to use photographs granted by SOA Systems Inc.

All rights reserved. Printed in the United States of America. This publication is protected by copyright, and permission must be obtained from the copyright holder prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
One Lake Street
Upper Saddle River, NJ 07458

ISBN 0-13-185858-0

Text printed in the United States on recycled paper at R.R. Donnelley in
Crawfordsville, Indiana.

First printing, July 2005

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Contents

Preface

xxvii

Chapter 1

Introduction	1
1.1 Why this book is important	2
1.1.1 The false SOA	2
1.1.2 The ideal SOA	3
1.1.3 The real SOA	4
1.2 Objectives of this book	4
1.2.1 Understanding SOA, service-orientation, and Web services	5
1.2.2 Learning how to build SOA with Web services	5
1.3 Who this book is for	6
1.4 What this book does not cover	6
1.5 How this book is organized	7
1.5.1 Part I: SOA and Web Services Fundamentals	8
1.5.2 Part II: SOA and WS-* Extensions	10
1.5.3 Part III: SOA and Service-Orientation	13
1.5.4 Part IV: Building SOA (Planning and Analysis)	14
1.5.5 Part V: Building SOA (Technology and Design)	16
1.5.6 Conventions	19
1.6 Additional information	19
1.6.1 The XML & Web Services Integration Framework (XWIF)	19
1.6.2 www.serviceoriented.ws	20
1.6.3 Contact the Author	20

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

x

Contents

Chapter 2

Case Studies	21
2.1 How case studies are used	22
2.1.1 Style characteristics	22
2.1.2 Relationship to abstract content	22
2.1.3 Code samples	23
2.2 Case #1 background: RailCo Ltd.	23
2.2.1 History	23
2.2.2 Technical infrastructure	23
2.2.3 Automation solutions	24
2.2.4 Business goals and obstacles	24
2.3 Case #2 background: Transit Line Systems Inc.	25
2.3.1 History	26
2.3.2 Technical infrastructure	26
2.3.3 Automation solutions	27
2.3.4 Business goals and obstacles	27

Part I

SOA and Web Services Fundamentals	29
--	-----------

Chapter 3

Introducing SOA	31
3.1 Fundamental SOA	32
3.1.1 A service-oriented analogy	32
3.1.2 How services encapsulate logic	33
3.1.3 How services relate	35
3.1.4 How services communicate	35
3.1.5 How services are designed	36
3.1.6 How services are built	37
3.1.7 Primitive SOA	38
3.2 Common characteristics of contemporary SOA	40
3.2.1 Contemporary SOA is at the core of the service-oriented computing platform	41
3.2.2 Contemporary SOA increases quality of service	42

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Contents

xi

3.2.3	Contemporary SOA is fundamentally autonomous	42
3.2.4	Contemporary SOA is based on open standards	43
3.2.5	Contemporary SOA supports vendor diversity	43
3.2.6	Contemporary SOA promotes discovery	44
3.2.7	Contemporary SOA fosters intrinsic interoperability	45
3.2.8	Contemporary SOA promotes federation	45
3.2.9	Contemporary SOA promotes architectural composability	46
3.2.10	Contemporary SOA fosters inherent reusability	47
3.2.11	Contemporary SOA emphasizes extensibility	48
3.2.12	Contemporary SOA supports a service-oriented business modeling paradigm	48
3.2.13	Contemporary SOA implements layers of abstraction	49
3.2.14	Contemporary SOA promotes loose coupling throughout the enterprise	50
3.2.15	Contemporary SOA promotes organizational agility	51
3.2.16	Contemporary SOA is a building block	52
3.2.17	Contemporary SOA is an evolution	53
3.2.18	Contemporary SOA is still maturing	53
3.2.19	Contemporary SOA is an achievable ideal	53
3.2.20	Defining SOA	54
3.2.21	Separating concrete characteristics	55
3.3	Common misperceptions about SOA	56
3.3.1	"An application that uses Web services is service-oriented."	56
3.3.2	"SOA is just a marketing term used to re-brand Web services."	57
3.3.3	"SOA is just a marketing term used to re-brand distributed computing with Web services."	57
3.3.4	"SOA simplifies distributed computing."	57
3.3.5	"An application with Web services that uses WS-* extensions is service-oriented."	58
3.3.6	"If you understand Web services you won't have a problem building SOA."	58
3.3.7	"Once you go SOA, everything becomes interoperable."	59
3.4	Common tangible benefits of SOA	59
3.4.1	Improved integration (and intrinsic interoperability)	60
3.4.2	Inherent reuse	60
3.4.3	Streamlined architectures and solutions	61
3.4.4	Leveraging the legacy investment	61
3.4.5	Establishing standardized XML data representation	62

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

3.4.6	Focused investment on communications infrastructure	63
3.4.7	"Best-of-breed" alternatives	63
3.4.8	Organizational agility	63
3.5	Common pitfalls of adopting SOA	64
3.5.1	Building service-oriented architectures like traditional distributed architectures	65
3.5.2	Not standardizing SOA	65
3.5.3	Not creating a transition plan	66
3.5.4	Not starting with an XML foundation architecture	67
3.5.5	Not understanding SOA performance requirements	67
3.5.6	Not understanding Web services security	68
3.5.7	Not keeping in touch with product platforms and standards development	69

Chapter 4

	The Evolution of SOA	71
4.1	An SOA timeline (from XML to Web services to SOA)	72
4.1.1	XML: a brief history	72
4.1.2	Web services: a brief history	73
4.1.3	SOA: a brief history	74
4.1.4	How SOA is re-shaping XML and Web services	76
4.2	The continuing evolution of SOA (standards organizations and contributing vendors)	78
4.2.1	"Standards" vs. "Specifications" vs. "Extensions"	78
4.2.2	Standards organizations that contribute to SOA	79
4.2.3	Major vendors that contribute to SOA	82
4.3	The roots of SOA (comparing SOA to past architectures)	86
4.3.1	What is architecture?	86
4.3.2	SOA vs. client-server architecture	88
4.3.3	SOA vs. distributed Internet architecture	95
4.3.4	SOA vs. hybrid Web service architecture	104
4.3.5	Service-orientation and object-orientation (Part I)	107

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Contents

xiii

Chapter 5

Web Services and Primitive SOA	109
5.1 The Web services framework	111
5.2 Services (as Web services)	112
5.2.1 Service roles	114
5.2.2 Service models	126
5.3 Service descriptions (with WSDL)	131
5.3.1 Service endpoints and service descriptions	133
5.3.2 Abstract description	134
5.3.3 Concrete description	135
5.3.4 Metadata and service contracts	136
5.3.5 Semantic descriptions	137
5.3.6 Service description advertisement and discovery	138
5.4 Messaging (with SOAP)	142
5.4.1 Messages	143
5.4.2 Nodes	149
5.4.3 Message paths	152

Part II

SOA and WS-* Extensions	155
What is "WS-*"?	157

Chapter 6

Web Services and Contemporary SOA	
(Part I: Activity Management and Composition)	159
6.1 Message exchange patterns	162
6.1.1 Primitive MEPs	163
6.1.2 MEPs and SOAP	169
6.1.3 MEPs and WSDL	169
6.1.4 MEPs and SOA	171
6.2 Service activity	172
6.2.1 Primitive and complex service activities	174
6.2.2 Service activities and SOA	175

6.3	Coordination	177
6.3.1	Coordinator composition	179
6.3.2	Coordination types and coordination protocols	180
6.3.3	Coordination contexts and coordination participants	180
6.3.5	The activation and registration process	181
6.3.5	The completion process	182
6.3.6	Coordination and SOA	183
6.4	Atomic transactions	186
6.4.1	ACID transactions	187
6.4.2	Atomic transaction protocols	188
6.4.3	The atomic transaction coordinator	188
6.4.4	The atomic transaction process	189
6.4.5	Atomic transactions and SOA	191
6.5	Business activities	193
6.5.1	Business activity protocols	194
6.5.2	The business activity coordinator	195
6.5.3	Business activity states	195
6.5.4	Business activities and atomic transactions	196
6.5.5	Business activities and SOA	197
6.6	Orchestration	200
6.6.1	Business protocols and process definition	203
6.6.2	Process services and partner services	203
6.6.3	Basic activities and structured activities	204
6.6.4	Sequences, flows, and links	204
6.6.5	Orchestrations and activities	205
6.6.6	Orchestration and coordination	205
6.6.7	Orchestration and SOA	205
6.7	Choreography	208
6.7.1	Collaboration	209
6.7.2	Roles and participants	210
6.7.3	Relationships and channels	210
6.7.4	Interactions and work units	210
6.7.5	Reusability, composability, and modularity	210
6.7.6	Orchestrations and choreographies	211
6.7.7	Choreography and SOA	212

Chapter 7

Web Services and Contemporary SOA (Part II: Advanced Messaging, Metadata, and Security)	217
7.1 Addressing	220
7.1.1 Endpoint references	222
7.1.2 Message information headers	223
7.1.3 Addressing and transport protocol independence	225
7.1.4 Addressing and SOA	225
7.2 Reliable messaging	228
7.2.1 RM Source, RM Destination, Application Source, and Application Destination	230
7.2.2 Sequences	230
7.2.3 Acknowledgements	231
7.2.4 Delivery assurances	233
7.2.5 Reliable messaging and addressing	235
7.2.6 Reliable messaging and SOA	235
7.3 Correlation	238
7.3.1 Correlation in abstract	239
7.3.2 Correlation in MEPs and activities	239
7.3.3 Correlation in coordination	240
7.3.4 Correlation in orchestration	240
7.3.5 Correlation in addressing	240
7.3.6 Correlation in reliable messaging	240
7.3.7 Correlation and SOA	241
7.4 Policies	242
7.4.1 The WS-Policy framework	243
7.4.2 Policy assertions and policy alternatives	244
7.4.3 Policy assertion types and policy vocabularies	245
7.4.4 Policy subjects and policy scopes	245
7.4.5 Policy expressions and policy attachments	245
7.4.6 What you really need to know	245
7.4.7 Policies in coordination	246
7.4.8 Policies in orchestration and choreography	246
7.4.9 Policies in reliable messaging	246
7.4.10 Policies and SOA	246

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

7.5	Metadata exchange	248
7.5.1	The WS-MetadataExchange specification	249
7.5.2	Get Metadata request and response messages	250
7.5.3	Get request and response messages	251
7.5.4	Selective retrieval of metadata	252
7.5.5	Metadata exchange and service description discovery	252
7.5.6	Metadata exchange and version control	253
7.5.7	Metadata exchange and SOA	254
7.6	Security	257
7.6.1	Identification, authentication, and authorization	259
7.6.2	Single sign-on	260
7.6.3	Confidentiality and integrity	261
7.6.4	Transport-level security and message-level security	262
7.6.5	Encryption and digital signatures	263
7.6.6	Security and SOA	265
7.7	Notification and eventing	266
7.7.1	Publish-and-subscribe in abstract	267
7.7.2	One concept, two specifications	268
7.7.3	The WS-Notification Framework	268
7.7.4	The WS-Eventing specification	271
7.7.5	WS-Notification and WS-Eventing	274
7.7.6	Notification, eventing, and SOA	274

Part III

SOA and Service-Orientation	277
-----------------------------	-----

Chapter 8

Principles of Service-Orientation	279
8.1 Service-orientation and the enterprise	280
8.2 Anatomy of a service-oriented architecture	284
8.2.1 Logical components of the Web services framework	284
8.2.2 Logical components of automation logic	285
8.2.3 Components of an SOA	288
8.2.4 How components in an SOA inter-relate	289

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Contents

xvii

8.3	Common principles of service-orientation	290
8.3.1	Services are reusable	292
8.3.2	Services share a formal contract	295
8.3.3	Services are loosely coupled	297
8.3.4	Services abstract underlying logic	298
8.3.5	Services are composable	301
8.3.6	Services are autonomous	303
8.3.7	Services are stateless	307
8.3.8	Services are discoverable	309
8.4	How service-orientation principles inter-relate	311
8.4.1	Service reusability	312
8.4.2	Service contract	313
8.4.3	Service loose coupling	315
8.4.4	Service abstraction	316
8.4.5	Service composability	317
8.4.6	Service autonomy	318
8.4.7	Service statelessness	319
8.4.8	Service discoverability	320
8.5	Service-orientation and object-orientation (Part II)	321
8.6	Native Web service support for service-orientation principles	324

Chapter 9

	Service Layers	327
9.1	Service-orientation and contemporary SOA	328
9.1.1	Mapping the origins and supporting sources of concrete SOA characteristics	329
9.1.2	Unsupported SOA characteristics	332
9.2	Service layer abstraction	333
9.2.1	Problems solved by layering services	334
9.3	Application service layer	337
9.4	Business service layer	341
9.5	Orchestration service layer	344
9.6	Agnostic services	346
9.7	Service layer configuration scenarios	347
9.7.1	Scenario #1: Hybrid application services only	348
9.7.2	Scenario #2: Hybrid and utility application services	349

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

xviii**Contents**

9.7.3	Scenario #3: Task-centric business services and utility application services	349
9.7.4	Scenario #4: Task-centric business services, entity-centric business services, and utility application services	350
9.7.5	Scenario #5: Process services, hybrid application services, and utility application services	350
9.7.6	Scenario #6: Process services, task-centric business services, and utility application services	351
9.7.7	Scenario #7: Process services, task-centric business services, entity-centric business services, and utility application services	352
9.7.8	Scenario #8: Process services, entity-centric business services, and utility application services	352

Part IV

Building SOA (Planning and Analysis)	355
---	------------

Chapter 10

SOA Delivery Strategies	357
--------------------------------	------------

10.1	SOA delivery lifecycle phases	358
10.1.1	Basic phases of the SOA delivery lifecycle	358
10.1.2	Service-oriented analysis	359
10.1.3	Service-oriented design	359
10.1.4	Service development	360
10.1.5	Service testing	360
10.1.6	Service deployment	361
10.1.7	Service administration	361
10.1.8	SOA delivery strategies	362
10.2	The top-down strategy	363
10.2.1	Process	363
10.2.2	Pros and cons	365
10.3	The bottom-up strategy	366
10.3.1	Process	367
10.3.2	Pros and cons	368
10.4	The agile strategy	370
10.4.1	Process	370
10.4.2	Pros and cons	373

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Contents

xix

Chapter 11

Service-Oriented Analysis (Part I: Introduction) 375

- 11.1 Introduction to service-oriented analysis 377
 - 11.1.1 Objectives of service-oriented analysis 377
 - 11.1.2 The service-oriented analysis process 377
- 11.2 Benefits of a business-centric SOA 382
 - 11.2.1 Business services build agility into business models 383
 - 11.2.2 Business services prepare a process for orchestration 384
 - 11.2.3 Business services enable reuse 384
 - 11.2.4 Only business services can realize the
service-oriented enterprise 385
- 11.3 Deriving business services 386
 - 11.3.1 Sources from which business services can be derived 387
 - 11.3.2 Types of derived business services 392
 - 11.3.3 Business services and orchestration 395

Chapter 12

Service-Oriented Analysis (Part II: Service Modeling) 397

- 12.1 Service modeling (a step-by-step process) 398
 - 12.1.1 "Services" versus "Service Candidates" 398
 - 12.1.2 Process description 399
- 12.2 Service modeling guidelines 416
 - 12.2.1 Take into account potential cross-process reusability
of logic being encapsulated (task-centric business
service candidates) 416
 - 12.2.2 Consider potential intra-process reusability of logic being
encapsulated
(task-centric business service candidates) 417
 - 12.2.3 Factor in process-related dependencies (task-centric
business service candidates) 417
 - 12.2.4 Model for cross-application reuse (application
service candidates) 418
 - 12.2.5 Speculate on further decomposition requirements 418
 - 12.2.6 Identify logical units of work with explicit boundaries 419
 - 12.2.7 Prevent logic boundary creep 419
 - 12.2.8 Emulate process services when not using orchestration
(task-centric business service candidates) 420

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

xx

Contents

12.2.9	Target a balanced model	421
12.2.10	Classify service modeling logic	422
12.2.11	Allocate appropriate modeling resources	422
12.2.12	Create and publish business service modeling standards	422
12.3	Classifying service model logic	423
12.3.1	The SOE model	424
12.3.2	The enterprise business model	426
12.3.3	"Building Blocks" versus "Service Models"	426
12.3.4	Basic modeling building blocks	426
12.4	Contrasting service modeling approaches (an example)	430

Part V

Building SOA (Technology and Design)	445
--------------------------------------	-----

Chapter 13

Service-Oriented Design (Part I: Introduction)	447
13.1 Introduction to service-oriented design	448
13.1.1 Objectives of service-oriented design	448
13.1.2 "Design standards" versus "Industry standards"	449
13.1.3 The service-oriented design process	449
13.1.4 Prerequisites	451
13.2 WSDL-related XML Schema language basics	453
13.2.1 The <code>schema</code> element	454
13.2.2 The <code>element</code> element	455
13.2.3 The <code>complexType</code> and <code>simpleType</code> elements	455
13.2.4 The <code>import</code> and <code>include</code> elements	456
13.2.5 Other important elements	456
13.3 WSDL language basics	457
13.3.1 The <code>definitions</code> element	458
13.3.2 The <code>types</code> element	459
13.3.3 The <code>message</code> and <code>part</code> elements	461
13.3.4 The <code>portType</code> , <code>interface</code> , and <code>operation</code> elements	462
13.3.5 The <code>input</code> and <code>output</code> elements (when used with <code>operation</code>)	462

Contents

xxi

13.3.6	The binding element	463
13.3.7	The input and output elements (when used with binding) ..	464
13.3.8	The service, port, and endpoint elements	465
13.3.9	The import element	465
13.3.10	The documentation element	466
13.4	SOAP language basics	466
13.4.1	The Envelope element	468
13.4.2	The Header element	468
13.4.3	The Body element	468
13.4.4	The Fault element	470
13.5	Service interface design tools	471
13.5.1	Auto-generation	471
13.5.2	Design tools	472
13.5.3	Hand coding	473

Chapter 14

Service-Oriented Design (Part II: SOA Composition Guidelines) 475

14.1	Steps to composing SOA	476
14.1.1	Step 1: Choose service layers	478
14.1.2	Step 2: Position core standards	478
14.1.3	Step 3: Choose SOA extensions	478
14.2	Considerations for choosing service layers	478
14.3	Considerations for positioning core SOA standards	481
14.3.1	Industry standards and SOA	481
14.3.2	XML and SOA	482
14.3.3	The WS-I Basic Profile	483
14.3.4	WSDL and SOA	485
14.3.5	XML Schema and SOA	485
14.3.6	SOAP and SOA	486
14.3.7	Namespaces and SOA	487
14.3.8	UDDI and SOA	488
14.4	Considerations for choosing SOA extensions	490
14.4.1	Choosing SOA characteristics	490
14.4.2	Choosing WS-* specifications	491
14.4.3	WS-BPEL and SOA	492

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Chapter 15

Service-Oriented Design (Part III: Service Design)	495
15.1 Service design overview	497
15.1.1 Design standards	498
15.1.2 About the process descriptions	498
15.1.3 Prerequisites	499
15.2 Entity-centric business service design (a step-by-step process)	501
15.2.1 Process description	502
15.3 Application service design (a step-by-step process)	522
15.3.1 Process description	523
15.4 Task-centric business service design (a step-by-step process)	540
15.4.1 Process description	540
15.5 Service design guidelines	555
15.5.1 Apply naming standards	555
15.5.2 Apply a suitable level of interface granularity	556
15.5.3 Design service operations to be inherently extensible	558
15.5.4 Identify known and potential service requestors	559
15.5.5 Consider using modular WSDL documents	559
15.5.6 Use namespaces carefully	560
15.5.7 Use the SOAP document and literal attribute values	561
15.5.8 Use WS-I Profiles even if WS-I compliance isn't required	563
15.5.9 Document services with metadata	563

Chapter 16

Service-Oriented Design (Part IV: Business Process Design)	565
16.1 WS-BPEL language basics	566
16.1.1 A brief history of BPEL4WS and WS-BPEL	567
16.1.2 Prerequisites	568
16.1.3 The process element	568
16.1.4 The partnerLinks and partnerLink elements	569
16.1.5 The partnerLinkType element	570
16.1.6 The variables element	571
16.1.7 The getVariableProperty and getVariableData functions	572

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Contents

xxiii

16.1.8	The sequence element	573
16.1.9	The invoke element	574
16.1.10	The receive element	575
16.1.11	The reply element	576
16.1.12	The switch, case, and otherwise elements	577
16.1.13	The assign, copy, from, and to elements	577
16.1.14	faultHandlers, catch, and catchAll elements	578
16.1.15	Other WS-BPEL elements	579
16.2	WS-Coordination overview	581
16.2.1	The CoordinationContext element	582
16.2.2	The Identifier and Expires elements	583
16.2.3	The CoordinationType element	583
16.2.4	The RegistrationService element	583
16.2.5	Designating the WS-BusinessActivity coordination type	584
16.2.6	Designating the WS-AtomicTransaction coordination type	584
16.3	Service-oriented business process design (a step-by-step process)	585
16.3.1	Process description	586

Chapter 17

Fundamental WS-* Extensions	613
You mustUnderstand this	614
17.1 WS-Addressing language basics	615
17.1.1 The EndpointReference element	616
17.1.2 Message information header elements	617
17.1.3 WS-Addressing reusability	620
17.2 WS-ReliableMessaging language basics	622
17.2.1 The Sequence, MessageNumber, and LastMessage elements	623
17.2.2 The SequenceAcknowledgement and AcknowledgementRange elements	625
17.2.3 The Nack element	626
17.2.4 The AckRequested element	627
17.2.5 Other WS-ReliableMessaging elements	628
17.3 WS-Policy language basics	629
17.3.1 The Policy element and common policy assertions	630
17.3.2 The ExactlyOne element	631

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

xxiv**Contents**

17.3.3	The All element	632
17.3.4	The Usage attribute	633
17.3.5	The Preference attribute	633
17.3.6	The PolicyReference element	633
17.3.7	The PolicyURIs attribute	634
17.3.8	The PolicyAttachment element	635
17.3.9	Additional types of policy assertions	635
17.4	WS-MetadataExchange language basics	636
17.4.1	The GetMetadata element	637
17.4.2	The Dialect element	638
17.4.3	The Identifier element	639
17.4.4	The Metadata, MetadataSection, and MetadataReference elements	640
17.4.5	The Get message	641
17.5	WS-Security language basics	642
17.5.1	The Security element (WS-Security)	644
17.5.2	The UsernameToken, Username, and Password elements (WS-Security)	644
17.5.3	The BinarySecurityToken element (WS-Security)	644
17.5.4	The SecurityTokenReference element (WS-Security)	644
17.5.5	Composing Security element contents (WS-Security)	645
17.5.6	The EncryptedData element (XML-Encryption)	646
17.5.7	The CipherData, CipherValue, and CipherReference elements (XML-Encryption)	647
17.5.8	XML-Signature elements	648

Chapter 18

SOA Platforms	651
18.1 SOA platform basics	652
18.1.1 Basic platform building blocks	653
18.1.2 Common SOA platform layers	654
18.1.3 Relationship between SOA layers and technologies	655
18.1.4 Fundamental service technology architecture	656
18.1.5 Vendor platforms	667
18.2 SOA support in J2EE	668
18.2.1 Platform overview	668
18.2.2 Primitive SOA support	681

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Contents

xxv

18.2.3	Support for service-orientation principles	682
18.2.4	Contemporary SOA support	683
18.3	SOA support in .NET	688
18.3.1	Platform overview	688
18.3.2	Primitive SOA support	697
18.3.3	Support for service-orientation principles	698
18.3.4	Contemporary SOA support	700
18.4	Integration considerations	703

Appendix A

Case Studies: Conclusion	707
A.1 RailCo Ltd.	708
A.2 Transit Line Systems Inc.	711
A.3 The Oasis Car Wash	715

Appendix B

Service Models Reference	717
--------------------------	-----

About the Author	721
------------------	-----

About SOA Systems	723
-------------------	-----

About the Photographs	725
-----------------------	-----

Index	727
-------	-----

Preface

Authoring this book involved nearly a year of writing, research, and staying on top of a subject matter that is constantly expanding its reach and importance. Although the majority of the chapters focus on service-oriented architecture from a vendor-neutral perspective, achieving an accurate representation of this perspective required that I spend a great deal of time evaluating SOA support in all primary vendor platforms. As part of this research stage I spoke with more than a hundred senior IT professionals, either through interviews or through my work as an awards judge evaluating platform submissions.

One of the most interesting facets of this project has been in defining service-orientation within the context of Web services. While studying the individual parts of what constitutes service-orientation as a paradigm, I came to realize just how many of its roots lie in past innovations. Yet at the same time, it is distinct, blending traditional and new concepts in support of a unique architectural model.

Despite its apparent “newness,” SOA, on a fundamental level, is based on a very old and established school of thought. Service-orientation, as a means of separating things into independent and logical units, is a very common concept. As I progressed through these chapters, I began to notice this more often in everyday life. Items, people, organizations we come into contact with either offer some form of service or participate in performing a service. Once applied to technology architecture, though, service-orientation is concerned with a specific part of our service-oriented world: business automation.

Competitive business climates demand that corporations minimize redundant effort and maximize the expediency with which strategic goals can be achieved. Inefficient organizations that consistently waste resources are bound to fall behind. The manner in

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

xxviii

Acknowledgments

which an organization automates its business is a critical factor in determining the level of efficiency at which it operates and, ultimately, the extent of success it attains in its ventures.

This is what makes SOA so valuable. By shaping automation logic through service-orientation, existing investments can be leveraged, business intelligence can be accurately expressed, and inherent automation agility can be achieved. When coupled with the Web services technology platform, SOA offers a significant and real benefit potential that can transform the technology and outlook of an organization. My goal for this book is to help you explore, understand, and realize this potential.

Acknowledgments

While writing this book I was blessed with a strong team of technical reviewers and superior editorial, production, and marketing professionals. My thanks to all of you for your tireless efforts. A special thanks to my family for their patience and unwavering support.

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

Chapter 1



Introduction

- 1.1 Why this book is important
- 1.2 Objectives of this book
- 1.3 Who this book is for
- 1.4 What this book does not cover
- 1.5 How this book is organized
- 1.6 Additional information

1.1 Why this book is important

One of my favorite quotes came from an exchange I overheard while preparing to speak at a conference. Two IT professionals were discussing their respective environments, when one asked the other if his team was building a service-oriented architecture. The individual responded by saying “My architect thinks it’s service-oriented, my developers insist it’s object-oriented, and my analysts wish it would be more business-oriented. All I can tell you is that it isn’t what it was before we started building Web services.”

This candid statement is a sign of the times. Service-oriented architecture (SOA) has become the focal point of the IT industry, yet few fully understand it. This book aims to fill this knowledge gap by helping you accomplish the following goals:

- understand SOA, service-orientation, and Web services
- learn how to build SOA with Web services

Let’s begin by identifying the most common obstacle to adopting SOA.

1.1.1 The false SOA

I cannot recall any one term causing as much confusion as “service-oriented.” Its apparent ambiguity has led vendors, IT professionals, and the media to claim their own interpretations. This, of course, makes grasping the meaning of a technical architecture labeled as “service-oriented” all the more difficult.

SOA, as an abstract paradigm, has traditionally represented a baseline distributed architecture with no reference to implementation. While relevant to us, this model represents only a subset of SOA in its most common and contemporary form.

Coupled with the Web services platform and a set of commonly accepted service-orientation principles, SOA has emerged as an architectural platform explicitly distinct from its predecessors. It introduces new concepts supported by select technologies that significantly augment characteristics of traditional distributed computing platforms—so much so that service-oriented environments often end up redefining IT infrastructure.

Why this book is important

3

This contemporary variety of SOA has received its share of attention. It has been promoted as a platform capable of revolutionizing enterprise environments by leveraging advancements in Web services technology and injecting organizations with hopes of federation, agility, and cross-platform harmony.

Many have been led to the notion that a technical architecture deemed service-oriented is simply one comprised of Web services. This is a common but dangerous assumption that leads to the number one mistake made by organizations intending to adopt SOA—the perception that the benefits promised by current mainstream SOA are attainable solely through a deeper investment in the Web services platform.

The reason this is happening is understandable. It is difficult for an organization to measure the extent of service-orientation possessed by its automation solutions when it is not clear what it actually means for automation logic to be truly service-oriented. What is needed is an ideal organizations can use as a target model.

1.1.2 The ideal SOA

We all have ideals that we aspire to attain. Ideals represent a state of excellence that motivate us to accomplish things beyond what we may have been able to without the ideal to look up to.

Service-orientation presents an ideal vision of a world in which resources are cleanly partitioned and consistently represented. When applied to IT architecture, service-orientation establishes a universal model in which automation logic and even business logic conform to this vision. This model applies equally to a task, a solution, an enterprise, a community, and beyond.

By adhering to this vision, past technical and philosophical disparities are blanketed by layers of abstraction that introduce a globally accepted standard for representing logic and information. This level of standardization offers an enormous benefit potential for organizations, as many of the traditional challenges faced by ever-changing IT environments can be directly addressed through the application of these standardized layers.

The service-orientation ideal has sparked a movement that has positioned SOA as the next phase in the evolution of business automation. In the same manner in which main-frame systems were succeeded by client-server applications, and client-server environments then evolved into distributed solutions based on Web technologies, the contemporary, Web services-driven SOA is succeeding traditional distributed architecture on a global scale.

All major software manufacturers and vendors are promoting support for SOA—some even through direct involvement in the development of open standards. As a result, every major development platform now officially supports the creation of service-oriented solutions. It would appear as though the realization of the SOA ideal is well underway. Why, then, is the false SOA so common?

1.1.3 The real SOA

The reality is that the rise of the false SOA has distorted the vision of the ideal SOA. Not only is the false SOA divergent from the “true path of service-orientation,” it reinforces SOA anti-patterns by extending and further entrenching the traditional distributed computing model to which SOA offers an alternative. The eventual realization that initial expectations will not be fulfilled can be further compounded once the costs, effort, and overall ugliness of a retro-fitting effort are calculated.

All of this can be avoided. What is required is an understanding of service-orientation, how it shapes technical architecture into SOA, and concrete, step-by-step processes for realizing SOA in a contemporary form.

Be forewarned, though, that SOA makes some impositions. A change in mindset is required, as business logic now needs to be viewed within a service-oriented context. Applying this context also requires a change in automation logic, as solutions now need to be built in support of service-orientation. Finally, a technical architecture capable of hosting service-oriented automation logic further introduces new technology and infrastructure requirements.

Real SOAs demand that an organization undergo this form of top-down transformation. However, the ideal an organization works toward during this process is not necessarily part of a universal vision of global service-orientation. It is an ideal based on how the concept of service-orientation, the architectural model provided by contemporary SOA, and the feature set offered by supporting technologies can benefit the vision and goals of your organization.

A real SOA requires real change, real foresight, and real commitment. Most of all, though, it requires guidance. This last requirement is what this book intends to assist you with.

1.2 Objectives of this book

Let's revisit the two primary goals we established earlier and elaborate on each.

1.2.1 Understanding SOA, service-orientation, and Web services

This book is not solely focused on architecture. Service-oriented architecture is a core part of the service-oriented computing platform that brings with it new concepts, technologies, and challenges. This book explores key parts of this platform to provide well-rounded coverage of the multi-faceted world of building service-oriented automation solutions.

Specifically, the following aspects of the SOA platform are explained:

- Primitive and contemporary variations of SOA are described and defined, establishing a set of nearly 20 common characteristics that can be fulfilled by current Web services technologies and design techniques explained in the step-by-step “how to” processes.
- Fundamental Web services theory is covered, along with a study of how the emergence of XML and Web services, coupled with the dynamics between standards organizations and software vendors, have all influenced and contributed to the evolution of SOA.
- The principles of service-orientation are described in detail. Their influence on Web service design is explained, and they are further incorporated into the step-by-step design processes.
- Over 10 WS-* specifications are described in detail. Separate parts of this book are dedicated to explaining concepts in plain English and then covering the technical details with code samples.
- Advanced SOA concepts and design issues are discussed, including the creation of specialized service layers. These allow for the abstraction of business and technology domains within the enterprise and form the basis for business and application-centric service designs.

1.2.2 Learning how to build SOA with Web services

A large portion of this book is dedicated to providing step-by-step instructions on how to accomplish the following tasks:

- perform a service-oriented analysis
- model service candidates derived from existing business documentation
- design the composition of an SOA

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

6

Chapter 1: Introduction

- design application services for technology abstraction
- design business services for business logic abstraction
- design service-oriented business processes
- assess SOA support provided by J2EE and .NET platforms

1.3 Who this book is for

SOA is a broad subject matter. It represents a new generation architectural platform that encompasses a series of contemporary technologies (both proprietary and vendor-neutral).

This book will therefore be useful to various IT professionals who are interested in learning more about the following:

- how to build SOA
- service-orientation principles
- designing different types of services for SOA
- service-oriented business modeling
- features provided by key WS-* specifications
- orchestration with WS-BPEL
- SOA support in J2EE and .NET platforms
- modeling business-centric services
- creating design standards for SOA-based solutions
- Web services technology within the context of SOA

1.4 What this book does not cover

While issues relating to integration and interoperability are referenced and discussed throughout this book, service-oriented integration as a specific topic is not covered. This is to prevent overlap with *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, this book's companion guide. The *Field Guide* is dedicated to matters of integration and explores numerous service-oriented integration architectures, strategies, and best practices.

How this book is organized

7

Also though this book will be useful to developers who want to understand how to build services for SOA and how different technology platforms support the SOA model, this is not a book that explains how to program Web services using any particular programming language. The step-by-step instructions provided focus on building and orchestrating service endpoints—not the underlying component logic. We therefore supply tutorials and/or code examples for the following open Web services languages: WSDL, SOAP, XML Schema, WS-BPEL, WS-Coordination, WS-Policy, WS-Metadata-Exchange, WS-Security, WS-Addressing, and WS-ReliableMessaging.

NOTE

A knowledge of XML is recommended prior to reading this book. Suggested reading materials are listed at www.serviceoriented.ws, and a collection of introductory papers can be found at www.xmltechnologyexpert.com.

1.5 How this book is organized

The next 17 chapters contain a mountain of information. Some serious thought was given to organization so that this book would be easy to read, while maintaining a logical structure.

Content was finally divided into the following primary parts:

- Part I: SOA and Web Services Fundamentals
- Part II: SOA and WS-* Extensions
- Part III: SOA and Service-Orientation
- Part IV: Building SOA (Planning and Analysis)
- Part V: Building SOA (Technology and Design)

Essentially, Parts I, II, and III cover basic and advanced SOA concepts and theory that prepare you for Parts IV and V, which supply a series of step-by-step “how to” instructions for building SOA. Part V further contains coverage of WS-* technologies and SOA platform support provided by J2EE and .NET.

A common thread across all parts of the book is the consistent use of case studies. Over 125 individual case study examples are interspersed throughout the chapters to provide constant real-life reference points that further demonstrate key topics. Case studies are introduced in Chapter 2, which establishes background information for two fictional organizations.

Let’s now take a closer look at what’s covered in the remaining chapters.

1.5.1 Part I: SOA and Web Services Fundamentals

Key SOA concepts are explained, a look at how SOA has evolved from past platforms follows, and then a description of the Web services framework wraps up this first part of the book.

Introducing SOA (Chapter 3)

We start off with a chapter dedicated to nailing down a clear definition of what SOA actually is and is not. We accomplish this by first studying the core characteristics of what constitutes a fundamental or “primitive SOA.” We supplement this by introducing the principles of service-orientation, and then look at the many influences that are elevating the primitive service-oriented architecture into a broader, enterprise-level platform.

As part of this exercise, we identify and explain 20 key characteristics associated with what we term as “contemporary SOA.” After we progress through individual descriptions of these characteristics, we provide detailed and generic definitions of what constitutes an SOA.

To further clarify what SOA is *not*, we proceed to address a series of common myths and misperceptions. These sections help cut through some of the confusion surrounding SOA and set the groundwork for the many SOA-related topics we discuss in subsequent chapters.

We then move on to identifying and explaining the key benefits behind adopting SOA. Although these benefits are discussed throughout this book, it is important to separate them ahead of time so that we can form a clear vision of what it is we are accomplishing by transitioning to this architectural model.

Finally, we conclude this chapter with a look at the most common pitfalls facing any organization on the path toward SOA. Understanding these “worst practices” is important not only to avoiding a whole lot of problems, but also to better appreciate the reasoning behind some of the analysis and design processes provided in later chapters.

The Evolution of SOA (Chapter 4)

This chapter continues with an exploration of how SOA came to be. Specifically, we follow a timeline that looks at the following:

- Past architectural platforms from which SOA has evolved and inherited traits and qualities.

How this book is organized

9

- Current influences (as fueled by XML and Web services technology platforms) that have shaped SOA into what it is now.
- The ongoing activity of standards organizations and contributing vendors that are further extending the breadth of the SOA platform.

We begin with a brief historical account of XML and Web services and discuss how these now established technologies have shaped SOA and are, to a large extent, responsible for its success. Subsequently, we turn the tables and discuss how the resulting popularity of SOA has changed the manner in which some XML and Web services technologies have been traditionally positioned and utilized.

We then dive into the current world of SOA as we discuss who and what is making SOA happen. Organizations and software vendors involved with developing contemporary SOA specifications and products are discussed. Most notably, the roles played by the following organizations are explained:

- World Wide Web Consortium (W3C)
- Web Services Interoperability Organization (WS-I)
- Organization for the Advancement of Structured Information Standards (OASIS)

The unique dynamics between standards organizations and software manufacturers are explored, with an emphasis on how vendors have influenced and contributed to the development of Web services specifications.

Next, we provide definitions for the terms “application architecture” and “enterprise architecture” and then define the architectural scope of SOA as it relates to these types of architectures. We subsequently dig up the roots of service-orientation by describing each of the following traditional platforms for the purpose of contrasting them with SOA:

- client-service architecture
- distributed Internet architecture
- distributed architectures that use Web services peripherally

For each of these architectural models, we explore the following aspects: application logic, application processing, technology, security, and administration. This part of the chapter is rounded out by a preliminary comparison of service-orientation and object-orientation.

Web Services and Primitive SOA (Chapter 5)

In Chapter 3 we formally defined the characteristics of primitive SOA. This chapter now explains how these characteristics are fulfilled by first-generation Web services technologies.

Note that this chapter introduces a new feature of the book called *In Plain English*. Even though all sections in this chapter are supplemented with examples that are part of our continuing case studies, they are further outfitted with these intentionally simplistic, non-technical analogies.

We begin with a review of the fundamental mechanics behind the Web services communications framework.

Topics covered include the following:

- basic Web services messaging
- service providers and service requestors
- service descriptions, service contracts, and metadata
- active and passive intermediaries
- message paths and service compositions
- common service models

Next, we move on to primitive SOA specifics, as we describe how service descriptions accomplish the core quality of loose coupling that contractually binds services within an SOA. Concepts specific to abstract and concrete WSDL definitions are explained and then supplemented with an introduction to UDDI and service registries.

We follow this section with an explanation of how SOAP is being used to address the messaging needs of SOA. The standardized messaging format provided by SOAP is discussed, along with a look at the SOAP message structure and the runtime roles played by SOAP processing nodes.

1.5.2 Part II: SOA and WS-* Extensions

The next set of chapters tackle advanced issues related to a multitude of contemporary SOA extensions.

Web Services and Contemporary SOA—Part I: Activity Management and Composition (Chapter 6)

This chapter picks up the tempo by venturing into the WS-* landscape. This is the first of two chapters dedicated to exploring how SOA can be extended using features provided by WS-* specifications.

The following parts of contemporary SOA are explored:

- Message exchange patterns
- Activities
- Coordination
- Atomic transactions
- Business activities
- Orchestration
- Choreography

The sequence of these topics is intentional, as each establishes a layer of functionality upon which the next builds.

Concepts relating to the latter five items in the above list are derived from the following WS-* specifications:

- WS-Coordination
- WS-AtomicTransaction
- WS-BusinessActivity
- WS-BPEL (formerly known as BPEL4WS)
- WS-CDL (formerly known as WS-Choreography)

Because this book intentionally separates concepts from technology, the actual language and syntax-level details for these WS-* extensions are covered in *Part V: Building SOA (Technology and Design)*.

Further, this chapter explains how these specifications and their associated concepts inter-relate, as well as how they individually tie into and fulfill the predefined characteristics of contemporary SOA. Finally, it is also worth mentioning that this chapter continues providing *In Plain English* sections to help clarify concepts using non-technical analogies.

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

12

Chapter 1: Introduction

Web Services and Contemporary SOA—Part II: Advanced Messaging, Metadata, and Security (Chapter 7)

This chapter dives even more deeply into the world of SOA extensions, as we study and explain another series of concepts related to additional WS-* specifications.

The following topics are covered:

- Addressing
- Reliable Messaging
- Correlation
- Policies
- Metadata Exchange
- Security
- Notification and Eventing

The concepts behind each of these topics are derived from the following WS-* specifications:

- WS-Addressing
- WS-ReliableMessaging
- WS-Policy Framework (including WS-PolicyAttachments and WS-PolicyAssertions)
- WS-MetadataExchange
- WS-Security (including XML-Encryption and XML-Signature)
- WS-Notification Framework (including WS-BaseNotification, WS-Topics, and WS-BrokeredNotification)
- WS-Eventing

As with Chapter 6, only concepts are discussed at this stage. The respective languages of the first five specifications in the above list are explained later in Chapter 17.

Also as with the previous chapter, how the individual extensions inter-relate and address specific characteristics of contemporary SOA is explained and supplemented with additional *In Plain English* sections.

1.5.3 Part III: SOA and Service-Orientation

Service-orientation, as a design paradigm, is explored in this part of the book. These chapters establish crucial concepts that form the basis for service and architecture-level design approaches described in later chapters.

Principles of Service-Orientation (Chapter 8)

We now turn our attention to the fundamental principles of service-orientation. It is these principles that form the basis for SOA and shape and standardize the individual services from which SOA is comprised.

This chapter introduces a view of the enterprise referenced throughout subsequent parts of this book, establishing a model that separates an enterprise into business and application logic domains. We then go on to discuss how SOA promotes service-orientation throughout both of these domains.

Next, we dissect a logical SOA and study its most fundamental parts. We begin this process with an examination of the core components of the Web services framework and then illustrate how these are positioned and augmented within SOA. We continue this exercise by examining how the components of an SOA inter-relate.

We then move on to a detailed review of the eight most common principles of service-orientation. Each is explained individually and each is accompanied by a case study example. These principles are then revisited as we explore how they inter-relate. In these sections we discover how some principles support or rely on others.

The subsequent section provides a mini-study of how common object-orientation (OO) principles relate to or influence the service-orientation principles we just discussed. This is an interesting analysis for those familiar with object-orientation.

The chapter concludes with an important revelation. After explaining the principles of service-orientation, we compare them with the feature set supplied by the first-generation Web services platform. This then tells us which of the service-orientation principles are provided automatically by the mere use of Web services and which require explicit effort to realize. This is an important piece of knowledge, as it gives us a checklist of design issues that we later incorporate in the step-by-step design processes.

Service Layers (Chapter 9)

Advanced service-orientation issues are addressed in this chapter, as we set our sights on realizing further characteristics of *contemporary* SOA. We first examine the primary influences that are shaping and broadening the reach of SOA. We then study which of

these influences are responsible for realizing certain characteristics. Through deduction we end up with a list of contemporary SOA characteristics that require explicit design effort. Key among these are service-oriented business modeling and organizational agility.

We then move on to defining an approach to designing SOA in support of these characteristics wherein a series of specialized service layers abstract key parts of generic and application-specific logic. This fosters SOA throughout the enterprise and facilitates the realization of many important benefits.

The following three service layers are defined at this stage:

- Application service layer
- Business service layer
- Orchestration service layer

These layers establish the basis for a series of standardized services that are discussed and further explained in subsequent chapters. Next, we raise some issues in relation to the creation of solution-agnostic services and then conclude this chapter with an exploration of eight different service layer configuration scenarios that illustrate a range of possible SOA designs.

1.5.4 Part IV: Building SOA (Planning and Analysis)

All of the previous chapters provide a knowledge of concepts and theory that can now be applied to the real world. These next two chapters structure an SOA delivery project around the creation of a contemporary SOA and then supply detailed guidance as to how business and application logic can be defined and modeled into service candidates.

SOA Delivery Strategies (Chapter 10)

SOA is now viewed from a project delivery perspective by identifying and describing the typical phases that comprise an SOA delivery lifecycle. These phases are then assembled into the following three SOA delivery strategies:

- Top-down strategy
- Bottom-up strategy
- Agile strategy

How this book is organized

15

The pros and cons of each are contrasted, and an emphasis is placed on the agile strategy, which attempts to blend the benefits and requirements of the top-down and bottom-up approaches.

Service-Oriented Analysis—Part I: Introduction (Chapter 11)

At this point we have covered fundamental and advanced concepts relating to SOA, service-orientation, and the many facets of the supporting Web services framework. This chapter now takes the first step of applying this knowledge by establishing the service-oriented analysis phase.

The overall objectives of service-oriented analysis are defined, followed by a list of steps required to complete this stage. The last of these steps is a service modeling process that is described in Chapter 12.

This chapter then examines what constitutes a business-centric SOA. It explains the key benefits of investing in the creation of business service layers and highlights the various ways business services can be derived from existing business models.

For the most part, the sections in this chapter assist you in preparing for the step-by-step service modeling process described in the following chapter.

Service-Oriented Analysis—Part II: Service Modeling (Chapter 12)

We now embark on a twelve-step analysis process wherein we apply service-orientation to an existing business workflow and derive business and application service candidates.

This important part of building SOA allows us to create service candidates that become a primary input for the ultimate SOA design we finalize as part of the service-oriented design processes described in upcoming chapters. Our service modeling process is supplemented with detailed case study examples that demonstrate the execution of individual process steps.

Following the process description are a dozen service modeling guidelines, providing advice, recommended analysis standards, and further insights into how best to approach and complete an SOA analysis.

Next, we provide an optional classification system that can be applied to further enhance the analysis process. This approach breaks down and labels units of logic, which can improve the clarity of documentation and the identification of potential reuse opportunities.

Finally, we complete this chapter with another detailed case study example wherein the second of our two fictional companies takes us through the service modeling process again, this time applying the aforementioned classification system. Additionally, this example results in the creation of three different service candidate combinations for the purpose of contrasting approaches.

1.5.5 Part V: Building SOA (Technology and Design)

This, the largest part in the book, provides step-by-step processes for designing specialized SOA services and creating a service-oriented business process. Numerous technology tutorials are supplied to help understand the code examples used throughout these chapters. This part concludes with an overview of what constitutes an SOA technology platform, including a review of current SOA support provided by the .NET framework and the J2EE platform.

Service-Oriented Design—Part I: Introduction (Chapter 13)

This chapter continues where we left off when we completed the service-oriented analysis phase. We now prepare to move our service candidates into service-oriented design.

The first step is an SOA composition exercise that helps identify the architectural boundary of our planned solution (this step is detailed in Chapter 14). The remaining steps consist of the following individual design processes:

- Entity-centric business service design
- Application service design
- Task-centric business service design
- Service-oriented business process design

Step-by-step descriptions for each of these design processes are provided in Chapters 15 and 16.

These exercises will result in the creation of WSDL definitions that implement service candidates (which originated from the service-oriented analysis process). Therefore, this chapter helps us further prepare by providing short tutorials for the following key language elements:

- WSDL
- WSDL-related XML Schema elements
- SOAP message structure elements

How this book is organized

17

Note that the language elements described are limited to those used in the case study code samples.

This chapter ends with a discussion of service interface modeling approaches, during which modeling tools are contrasted with hand coding techniques.

Service-Oriented Design—Part II: SOA Composition Guidelines (Chapter 14)

Chapter 14 kicks off the service-oriented design process by providing guidance for composing a service-oriented architecture based on known functional requirements and technical limitations. As part of this procedure, we provide guidelines for choosing service layers and positioning identified standards and SOA extensions.

Specifically, we raise design issues related to incorporating XML, WSDL, XML Schema, SOAP, UDDI, and the WS-I Basic Profile into SOA. We then conclude this chapter with a set of considerations and guidelines for choosing WS-* specifications, with an emphasis on the use of WS-BPEL.

Service-Oriented Design—Part III: Service Design (Chapter 15)

This chapter, the longest in this book, contains three detailed, step-by-step process descriptions for designing services that correspond to two of the three service layers we established in Chapter 9.

The following design processes are described:

- Entity-centric business service design
- Application service design
- Task-centric business service design

Each process description is supplemented with extensive case study examples that demonstrate the application of individual process steps in real-world scenarios. This important chapter is then concluded with a set of service design guidelines applicable to the previously described processes.

Service-Oriented Design—Part IV: Business Process Design (Chapter 16)

Step-by-step instructions for building a service-oriented business process are provided in this chapter. A WS-BPEL process definition is created as part of the case study examples to orchestrate services that were modeled and designed in previous chapters.

Before we get into the service-oriented business process design, we provide a tutorial describing key WS-BPEL language elements used in the detailed examples that supplement the process description steps. (A brief look at the contents of WS-Coordination SOAP headers is also included.)

Fundamental WS- Extensions (Chapter 17)*

Our SOA so far consists of a set of services that establish up to three levels of abstraction, along with a service-oriented business process responsible for orchestrating them. This next chapter provides technical insight into how the feature set of SOA can be extended with the WS-* specifications we introduced in Chapter 7.

Key elements and constructs for the following specifications are covered:

- WS-Addressing
- WS-ReliableMessaging
- WS-Policy
- WS-MetadataExchange
- WS-Security

Each language description is supplemented with case study examples containing the code used to implement the corresponding conceptual examples provided in Chapter 7.

SOA Platforms (Chapter 18)

Our final chapter takes a close look at what constitutes an implementation platform for SOA. The individual parts that comprise the development and runtime environments required to build and host a service-oriented solution are explained, along with an “under the hood” look at the implementation logic behind a typical Web service.

This is followed by two identically structured sections that supply an overview of SOA support provided by the J2EE and .NET platforms. Each of these sections begins with a high-level introduction of the respective platforms and then continues to revisit the following aspects of SOA established in earlier chapters:

- Characteristics of primitive SOA
- Principles of service-orientation
- Characteristics of contemporary SOA

Case Studies: Conclusion (Appendix A)

Appendix A acts as a bookend to the case study storylines that began in Chapter 2. The progress of each organization is reviewed, and the resulting solution environments are studied. The original objectives established at the beginning of the book are revisited to ensure that all have been met.

Service Models Reference (Appendix B)

This appendix provides a quick reference table for all of the service models described in this book.

1.5.6 Conventions

- *Summary of Key Points*—Each primary section within a chapter ends with a summary that recaps and highlights the main topics or conclusions covered. These summaries are provided to allow readers to confirm that a given subject matter was fully understood.
- *Figures*—This book contains over 300 diagrams, which are referred to as “figures.” A legend for the symbols used in these figures is provided by the book’s Web site at www.serviceoriented.ws.
- *Code examples*—On several occasions code used in this book contains examples with long lines. This happens most often when some of the larger URLs are used for namespace references. To avoid these lines from wrapping, hard line breaks are sometimes inserted. While this is done intentionally to improve clarity, it can also result in invalid XML. If you are trying out any of these examples, rejoin these lines before validating the code.

1.6 Additional information

1.6.1 The XML & Web Services Integration Framework (XWIF)

Some of the contents in this book originated from research I performed for SOA Systems Inc. (formerly XMLTC Consulting Inc.), as part of the XML & Web Services Integration Framework (XWIF) project. For more information, visit www.soasystems.com or www.xwif.com.

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

20

Chapter 1: Introduction

1.6.2 www.serviceoriented.ws

Updates, source code, and various other supporting resources can be found at www.serviceoriented.ws. I am interested in your feedback. Any experiences you'd like to share or suggestions you may have as to how I can continue to improve this book would be much appreciated.

1.6.3 Contact the Author

To contact me directly, visit my bio site at www.thomaserl.com/technology.

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.



About the Author

Thomas Erl is an independent consultant with XMLTC Consulting in Vancouver, Canada. His previous book, *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, became the top-selling book of 2004 in both Web Services and SOA categories. This guide addresses numerous integration issues and provides strategies and best practices for transitioning toward SOA.

Thomas is a member of OASIS and is active in related research efforts, such as the XML & Web Services Integration Framework (XWIF). He is a speaker and instructor for private and public events and conferences, and has published numerous papers, including articles for the *Web Services Journal*, *WLDJ*, and *Application Development Trends*.

For more information, visit <http://www.thomaserl.com/technology/>.

Sample Chapter 1 from "Service-Oriented Architecture: Concepts, Technology, and Design" by Thomas Erl
For more information visit www.serviceoriented.ws.

About SOA Systems

SOA Systems Inc. is a consulting firm actively involved in the research and development of service-oriented architecture, service-orientation, XML, and Web services standards and technology. Through its research and enterprise solution projects SOA Systems has developed a recognized methodology for integrating and realizing service-oriented concepts, technology, and architecture.

For more information, visit www.soasystems.com.

One of the consulting services provided by SOA Systems is comprehensive SOA transition planning and the objective assessment of vendor technology products.

For more information, visit www.soaplanning.com.

The content in this book is the basis for a series of SOA seminars and workshops developed and offered by SOA Systems.

For more information, visit www.soatraining.com.