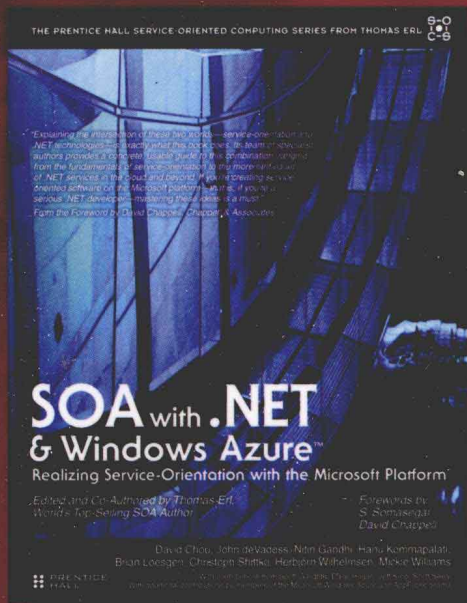


在微软平台上实现SOA

——基于.NET和Windows Azure

(英文版)

David Chou John deVadoss Thomas Erl Nitin Gandhi
 (美) Hanu Kommapalati Brian Loesgen Christoph Shittko 著
 Herbjørn Wilhelmsen Mickie Williams



SOA with .NET and Windows Azure:
 Realizing Service-Oriented Architecture with the Microsoft Platform



科学出版社

服务计算技术丛书

在微软平台上实现 SOA

——基于 .NET 和 Windows Azure

(英文版)

SOA with .NET and Windows Azure:
Realizing Service-Oriented Architecture with the Microsoft Platform

David Chou John deVadoss Thomas Erl Nitin Gandhi

[美] Hanu Kommapalati Brian Loesgen Christoph Shittko 著

Herbjörn Wilhelmsen Mickie Williams

科学出版社

北 京

图字: 01-2012-0462

内 容 简 介

本书主要介绍在 .NET 和 Windows Azure 平台上实现面向服务架构 (SOA) 的方法。首先综合介绍了在 Microsoft 平台上的现代服务技术进展, 并进一步展现了这些技术提高应用和实现面向服务的潜在可能性; 然后深入研究了 Microsoft 企业技术, 如 Windows Azure、WCF、WPF 等, 以及涉及通用 SOA 设计模式和原则的工业技术与模型。书中附有大量的编码实例, 详细分析了与云计算、编制、企业业务总线平台有关的各种技术架构和实现方式。

本书可供 SOA 领域的软件架构师、高级软件工程师、分析师、应用科研人员等参考学习。

Original edition, entitled SOA WITH .NET AND WINDOWS AZURE: REALIZING SERVICE-ORIENTATION WITH THE MICROSOFT PLATFORM, 1E, 9780131582316 by ERL, THOMAS; CHOU, DAVID; DEVADOSS, JOHN; GANDHI, NITIN; KOMMAPALATI, HANU; published by Pearson Education, Inc, publishing as Prentice Hall, Copyright © 2010 SOA Systems Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

China edition published by PEARSON EDUCATION ASIA LTD., and CHINA SCIENCE PUBLISHING & MEDIA LTD. (SCIENCE PRESS) Copyright © 2012.

Authorized for sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR). 本版本仅可在中国地区 (不包括香港、澳门和台湾) 销售发行。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签。无标签者不得销售。

图书在版编目 (CIP) 数据

在微软平台上实现 SOA: 基于 .NET 和 Windows Azure = SOA with .NET and Windows Azure : Realizing Service-Oriented Architecture with the Microsoft Platform : 英文 / (美) 周 (Chou, D.) 等著.
—北京: 科学出版社, 2012

(服务计算技术丛书)

ISBN 978-7-03-033643-9

I. ①在… II. ①周… III. ①互联网络—网络服务器—研究—英文 IV. ①TP368.52

中国版本图书馆 CIP 数据核字 (2012) 第 030027 号

责任编辑: 朱雪玲 / 责任印制: 赵 博 / 封面设计: 迷底书装

科学出版社出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

北京佳信达欣艺术印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2012年3月第 1 版 开本: B5 (720×1000)

2012年3月第一次印刷 印张: 56 1/2

字数: 904 000

定价: 129.00 元

(如有印装质量问题, 我社负责调换)

Foreword by S. Somasegar

Within the last decade, service-oriented architecture has moved from infancy to ubiquity, precipitating and, in many ways, enabling the next paradigm of information technology—cloud computing. As the Software + Services model becomes the norm, businesses that embrace SOA will smoothly transition to the cloud, enabling better scaling, availability, and cost efficacy of their services.

Service-oriented architecture is the magic behind many of the Internet-based applications and services that seamlessly integrate information and data from multiple sources into a single experience. The loosely-coupled design of SOA allows developers to take advantage of existing services to build their applications, dynamically adapt to changes in those services, and offer their own services to other application developers. From a developer perspective, it is best to have tools and frameworks that enable you to write **ONLY** the code that you need to write. SOA allows developers to do just that—focus on building the unique parts of their applications by enabling them to reuse existing services others have already written to solve common problems.

Microsoft has long promoted a real-world approach to SOA that focuses on helping organizations create business value by translating business challenges into opportunities. A real-world approach is typically based on rapid, agile iterations of design, implementation, and assessment, resulting in solutions that are able to better track and align to the changing needs of the organization and its business environment.

To developers, service-orientation offers a technology model with the potential for effectively creating and maintaining ‘evolvable’ applications: applications that are better able to change and grow with the business. To the CIO, service-orientation offers strategies and tools for nurturing existing IT assets while simultaneously fostering the development of new capabilities. The ‘rip and replace’ tactic of the past to deal with changing technology and business needs is facing extinction, thanks primarily to the increasing adoption of service-orientation and the pervasive nature of service-oriented architectures. The encapsulation of existing assets behind service-based interfaces provides structured access to legacy systems and applications while facilitating the opportunity for continuous improvement of the underlying business capabilities behind the interface.

However, keep in mind that architecture is a means to an end. The end goal is to create continuing value for your business, and a real-world approach based on proven practices offers a viable map to help get you there. With the emergence of the cloud as an attractive platform for both consumer and business computing, the principles underlying loose-coupling and service-orientation are increasingly relevant beyond the four walls of the data center. Services are more and more being developed and deployed beyond the confines of the firewall.

Microsoft’s innovative new platforms and tools, including Windows Azure and SQL Azure, as well as Windows Azure platform AppFabric, Visual Studio 2010, and .NET Framework 4, enable organizations to extend their service-oriented architectures into the cloud, creating a hybrid Software + Services model. The Microsoft platforms and tools provide businesses with the choice to leverage the ‘right’ technologies, whether on-premises or in the cloud, truly putting the customer in control, and organizations that build on a proven set of service-oriented patterns and practices set themselves up for greater success in a Software + Services world.

This book is the result of mining and collating proven practices from the field. The authors have done an excellent job of explaining the architectural designs and goals behind SOA as well as real-world examples of SOA usage to build elegant IT solutions. It is my hope that this work plays a role in helping you realize loosely coupled, service-oriented solutions, on-premises and in the cloud, using Microsoft platforms and tools.

—S. Somasegar

Senior Vice President, Developer Division, Microsoft

Foreword by David Chappell

What is SOA? In the dozen or so years that the term has been around, service-oriented architecture has meant lots of different things. Some vendors saw it as a way to sell whatever it was they were offering, and so they jumped on the SOA bandwagon with both feet. Other people interpreted SOA in business terms, viewing it as a way to structure how different parts of an organization interact.

Yet from all of this confusion, one clear fact has emerged: The technology of service-orientation has real value. Whether or not it helps vendors sell products or managers organize their business, taking a service-oriented approach to development can make life better for the people who build and maintain applications.

The core reason for this is simple. Since applications rarely live out their lives in isolation, why not design those apps from the start with future connections in mind? Creating software that can expose and consume services marks the end of application silos, and it's fundamental to the value of service-orientation.

Doing this well requires two things. The first is a grasp of how to apply services effectively. Over time, our industry has evolved a set of patterns to help us create and connect applications in a service-oriented style. As with patterns in any other area, those for service-orientation attempt to codify best practices, helping all of us learn from what's gone before rather than reinvent these wheels on our own.

The second thing that's needed is a solid understanding of the technologies used to create service-oriented software. If you're working in the .NET world, there are quite a few things to get your mind around. Windows Communication Foundation provides a unified approach to creating and consuming services, for example, while Windows Workflow Foundation offers a generalized approach to orchestrating them. BizTalk Server takes an explicitly integration-oriented approach to the problem, while the Windows Azure platform brings services to the cloud. And even though the .NET Framework is common to all of these technologies, using them effectively isn't so easy. Each brings its own complexity to the party, and each can be combined with the others in various ways.

Explaining the intersection of these two worlds—service-orientation and .NET technologies—is exactly what this book does. Its team of specialist authors provides a concrete, usable guide to this combination, ranging from the fundamentals of service-orientation to the more rarified air of .NET services in the cloud and beyond. If you're creating service-oriented software on the Microsoft platform—that is, if you're a serious .NET developer—mastering these ideas is a must.

—David Chappell

Chappell & Associates (San Francisco, CA, USA)

Contents

Foreword by S. Somasegar	xxiii
---	--------------

Foreword by David Chappell	xxv
---	------------

CHAPTER 1: Introduction	1
--	----------

1.1 About this Book	2
1.2 Objectives of this Book	3
1.3 Who this Book is For	4
1.4 What this Book Does Not Cover	4
1.5 Prerequisite Reading	4
1.6 How this Book is Organized	6
Part I: Fundamentals	7
Chapter 3: SOA Fundamentals	7
Chapter 4: A Brief History of Legacy .NET Distributed Technologies	7
Chapter 5: WCF Services	7
Chapter 6: WCF Extensions	7
Chapter 7: .NET Enterprise Services Technologies	7
Chapter 8: Cloud Services with Windows Azure	8
Part II: Services and Service Composition	8
Chapter 9: Service-Orientation with .NET Part I: Service Contracts and Interoperability	8
Chapter 10: Service-Orientation with .NET Part II: Coupling, Abstraction, and Discoverability	8
Chapter 11: Service-Orientation with .NET Part III: Reusability and Agnostic Service Models	8
Chapter 12: Service-Orientation with .NET Part IV: Service Composition and Orchestration Basics	9
Chapter 13: Orchestration Patterns with WF	9
Chapter 14: Orchestration Patterns with BizTalk Server	9

Part III: Infrastructure and Architecture 9

 Chapter 15: Enterprise Service Bus with BizTalk Server and
 Windows Azure 9

 Chapter 16: Windows Azure Platform AppFabric Service Bus 10

 Chapter 17: SOA Security with .NET and Windows Azure 10

 Chapter 18: Service-Oriented Presentation Layers with .NET 10

 Chapter 19: Service Performance Optimization 10

 Chapter 20: SOA Metrics with BAM 10

Part IV: Appendices 10

 Appendix A: Case Study Conclusion 10

 Appendix B: Industry Standards Reference 11

 Appendix C: Service-Oriented Principles Reference 11

 Appendix D: SOA Design Patterns Reference 11

 Appendix E: The Annotated SOA Manifesto 11

 Appendix F: Additional Resources 11

1.7 How Principles and Patterns are Used in this Book 11

 Sources 11

 Reference Notation 12

1.8 Symbols, Figures, and Style Conventions. 13

 Symbol Legend 13

 How Color is Used 13

 Additional Information 13

 Updates, Errata, and Resources (www.soabooks.com) 13

 Master Glossary (www.soaglossary.com) 13

 Referenced Specifications (www.soaspecs.com) 13

 SOASchool.com™ SOA Certified Professional (SOACP) 14

 The SOA Magazine (www.soamag.com) 14

 Notification Service 14

CHAPTER 2: Case Study Background 15

2.1 How Case Studies Are Used. 16

2.2 Case Study Background #1: Standard Mold 16

 History 16

 Technical Infrastructure 16

 Business Goals and Obstacles 17

2.3 Case Study Background #2: Superior Stamping 18

 History 18

 Technical Infrastructure 18

 Business Goals and Obstacles 19

PART I: FUNDAMENTALS

CHAPTER 3: SOA Fundamentals 23

3.1 Basic SOA Terminology. 24

 Service-Oriented Computing 25

 Service-Oriented Architecture (SOA) 27

 Services 28

Services as Components 29

Services as Web Services 30

Services as REST Services 31

 Service Models 31

Agnostic Logic and Non-Agnostic Logic 32

 Service Composition 33

 Service Inventory 34

 Service-Oriented Analysis 34

 Service Candidate 35

 Service-Oriented Design 35

 Service Contract 36

 Service-Related Granularity 37

 SOA Design Patterns 38

3.2 Service-Oriented Computing Goals 40

 Increased Intrinsic Interoperability 40

 Increased Federation 40

 Increased Vendor Diversification Options 40

 Increased Business and Technology Domain Alignment 41

3.3 Further Reading 41

CHAPTER 4: A Brief History of Legacy .NET**Distributed Technologies 43**

4.1 Distributed Computing 101	44
Client-Server	44
Distributed Architecture	45
Service-Oriented Architecture	47
4.2 .NET Enterprise Services	48
It All Began with COM (and DCOM)	48
COM+ Services	49
.NET Assemblies	51
Distributed Transaction Coordinator	51
.NET Enterprise Services and Service-Orientation	53
4.3 .NET Remoting	54
.NET Remoting Architecture	54
<i>Serializable Classes</i>	56
<i>Remotable Classes</i>	56
<i>Ordinary Classes</i>	56
Hosting .NET Remoting Components	56
<i>Windows Service</i>	56
<i>IIS Hosting Under ASP.NET</i>	57
<i>Hosting a .NET Remoting Component in a Console Application</i>	57
<i>.NET COM+ Services</i>	57
.NET Remoting Configurations	57
<i>Activation Types</i>	58
<i>Message Formats</i>	60
<i>Communication Protocols</i>	60
<i>Object Lifetime Management</i>	61
.NET Remoting and Service-Orientation	61
4.4 Microsoft Messaging Queue (MSMQ)	63
The Queues	64
Sending and Receiving Messages	65
MSMQ and Service-Orientation	66
4.5 System Transactions	67
Distributed Resource Transactions	67
Explicit and Implicit Programming Models	68
Ambient Transactions	69

- 4.6 Web Services (ASMX and WSE) 70
 - XML Web Services (ASMX) 71
 - The WebService Attribute* 71
 - The WebMethod Attribute* 72
 - Web Service Enhancements (WSE) 73
- 4.7 REST Service Processing with IHttpHandler 74

CHAPTER 5: WCF Services 75

- 5.1 Overview 76
- 5.2 Service Contracts with WCF 78
 - WCF Terminology 78
 - WCF Service Contract* 78
 - Interface Contract* 78
 - Operation Contract* 78
 - Data Contract* 78
 - Message Contract* 79
 - Service Endpoint* 79
 - The ServiceContract and OperationContract Attributes 79
 - Data Models and the DataContract Attribute 82
 - Messaging and the MessageContract Attribute 83
 - Service Endpoints and the endpoint Element 86
 - Address* 88
 - Bindings* 89
 - Contract* 92
 - REST Service Classes and Attributes 92
 - The WebGet Attribute* 93
 - The WebInvoke Attribute* 95
 - WCF UriTemplate Attribute* 96
 - Faults and the FaultContract Attribute 98
 - MEX Endpoints 100
 - Versioning Considerations 102
- 5.3 Service Implementation with WCF 104
 - Behaviors 104
 - Instanting 105
 - A Sample Implementation 106

5.4 Service Hosting with WCF	108
Self-Hosted Services	110
Managed Windows Services	112
IIS Process Boundary	113
Windows Activation Services (WAS)	114
Hosting REST Services	115
5.5 Service Consumers with WCF	116
Using the Service Metadata Tool	117
Writing the Proxy Class for a Service	118
Using the ChannelFactory Class	119

CHAPTER 6: WCF Extensions 121

6.1 WCF Security	122
Security Modes	123
Authorization	125
Federated Identity	126
6.2 WCF Transactions	127
Operation Attributes for Transactions	127
<i>TransactionScopeRequired</i>	128
<i>TransactionAutoComplete</i>	128
<i>TransactionFlow</i>	128
Service Attributes for Transactions	129
<i>TransactionIsolationLevel</i>	129
<i>TransactionAutoCompleteOnSessionClose</i>	130
<i>TransactionTimeout</i>	130
Durable Services	131
6.3 WCF Router	132
The RoutingService Class	133
Routing Contracts	134
Routing Configuration	135
<i>Step 1: Define Endpoints</i>	135
<i>Step 2: Configure Service Behavior</i>	136
<i>Step 3: Enumerate Target Endpoints</i>	136
<i>Step 4: Define Message Filters</i>	137
<i>Step 5: Create a Filter Table</i>	138
Fault Tolerance	139

- 6.4 WCF Discovery 140
 - Discovery Modes 141
 - Locating a Service Ad Hoc 143
 - Sending and Receiving Service Announcements 144
 - Discovery Proxies for Managed Discovery 146
 - Discovering from a Discovery Proxy* 146
 - Implicit Service Discovery 147
- 6.5 WCF Extensibility 148
 - WCF Layers 149
 - Layered Extensibility 149
 - Channel Layer Extensibility 150
- 6.6 WCF Management Tools 151
 - Administration 151
 - Troubleshooting 151
 - Logging Messages 153

CHAPTER 7: .NET Enterprise Services Technologies 155

- 7.1 SQL Server 156
 - Native XML Web Services Support 157
 - Service Broker (SSB) 160
 - Query Notification 165
 - XML Support in SQL Server 165
- 7.2 Windows Workflow Foundation (WF) 166
 - WF Architecture 167
 - Workflows 168
 - Sequential Workflows* 169
 - State Machine Workflows* 169
 - Workflow Designer 169
 - Workflow Persistence (with WF) 170
 - Communicating with the Host Container 171
 - Activities 172
 - Workflow Runtime Environment 175
 - WF Programming Model 176
 - Passing Parameters into a Workflow Instance 178

Returning Parameters from a Workflow Instance	178
Workflow-Enabled Services.	179
Versioning Orchestrations	180
WF Extensibility	180
Business Rules	180
7.3 Application Blocks and Software Factories	181
Application Blocks	182
Software Factories.	184
<i>Guidance Toolkits</i>	184
<i>Web Services Software Factory</i>	184
7.4 Windows Server AppFabric	187
Configurable Hosting Environment	188
Workflow Persistence (with AppFabric)	189
In-Memory Application Cache Platform	190
Manageability Extensions	192
Application Server Event Collector	192
7.5 BizTalk Server	193
BizTalk Server Architecture	194
Messaging	196
<i>Pipelines</i>	197
<i>Pipeline Components</i>	198
<i>Ports and Locations</i>	199
Adapters	199
Context Properties.	200
Itineraries	201
Unified Exception Management	202

CHAPTER 8: Cloud Services with Windows Azure 205

8.1 Cloud Computing 101.	206
Cloud Deployment Models	208
<i>Public Cloud</i>	208
<i>Private Cloud</i>	208
<i>Community Cloud</i>	209
<i>Other Deployment Models</i>	209
<i>The Intercloud (Cloud of Clouds)</i>	209
<i>Deployment Models and Windows Azure.</i>	210

Service Delivery Models	210
<i>Infrastructure-as-a-Service (IaaS)</i>	210
<i>Platform-as-a-Service (PaaS)</i>	211
<i>Software-as-a-Service (SaaS)</i>	211
<i>Other Delivery Models</i>	211
<i>IaaS vs. PaaS</i>	211
8.2 Windows Azure Platform Overview	213
Windows Azure (Application Container)	216
SQL Azure	217
Windows Azure Platform AppFabric	218
8.3 Windows Azure Roles	219
<i>Web Roles and Worker Roles</i>	220
<i>Virtual Machines</i>	220
<i>Input Endpoints</i>	221
<i>Inter-Role Communication</i>	222
8.4 Hello World in Windows Azure	223
1. Create a Cloud Service Project	224
2. Choose an ASP.NET Web Role	224
3. Create the Solution	225
4. Instantiate the Service	226
8.5 A Web Service in Windows Azure	227
1. Create a Host Service and Storage Service	233
2. Create and Deploy a Service Package	233
3. Promote the Service to Production	234
8.6 A REST Service in Windows Azure	235
REST Service Addressing	235
Creating a Windows Azure REST Service	236
8.7 Windows Azure Storage	239
Tables	240
<i>Entities and Properties</i>	240
<i>Data Access</i>	241
Queues	241
Blobs	242
<i>Block Blobs</i>	242
<i>Page Blobs</i>	243
Windows Azure Drive	243

PART II: SERVICES AND SERVICE COMPOSITION

**CHAPTER 9: Service-Orientation
with .NET Part I: Service Contracts
and Interoperability 247**

- 9.1 Standardized Service Contract 250
 - Contract-First 250
 - 1. Create or Reuse Data Contract 251
 - 2. Create Message Contract 251
 - 3. Create Interface Contract 252
 - Standardized Service Contract and Patterns 252
- 9.2 Canonical Schema 253
 - Creating Schemas with Visual Studio 254
 - Generating .NET Types 258
 - Using the DataContract Library 264
- 9.3 Data Model Transformation 267
 - Object-to-Object 269
 - LINQ-to-XML 271
 - XSLT Transformation 272
- 9.4 Canonical Protocol 274
 - Web Service 275
 - REST Service 277
 - Component 278
 - Another WCF Option: Named Pipes 279
 - Dual Protocols with WCF 279
- 9.5 Canonical Expression 280
 - Service Naming Conventions 280
 - Service Capability Naming Conventions 281

**CHAPTER 10: Service-Orientation
with .NET Part II: Coupling, Abstraction,
and Discoverability. 283**

- 10.1 Service Loose Coupling 285
 - Service Loose Coupling and Patterns 286