FIFTH EDITION

SERVICE-ORIENTED COMPUTING AND WEB SOFTWARE INTEGRATION

FROM PRINCIPLES TO DEVELOPMENT

YINONG CHEN AND WEI-TEK TSAI

Kendall Hunt
publishing company

Kendall Hunt
publishing company

www.kendallhunt.com
Send all inquiries to:
4050 Westmark Drive
Dubuque, IA 52004-1840

Copyright © 2008, 2010, 2011, 2014, 2015 by Kendall Hunt Publishing Company

ISBN 978-1-4652-5161-9

All rights reserved. No part of this publication may be reproduced,
stored in a retrieval system, or transmitted, in any form or by any means,
electronic, mechanical, photocopying, recording, or otherwise,
without the prior written permission of the copyright owner.

Printed in the United States of America

Table of Contents

Preface ..... **xi**

Part I Distributed Service-Oriented Software Development and Web Data Management ... **1**

Chapter 1 Introduction to Distributed Service-Oriented Computing **3**

1.1 Computer Architecture and Computing Paradigms 3
1.1.1 Computer Architecture 3
1.1.2 Software Architecture 4
1.1.3 Computing Paradigms 4

1.2 Distributed Computing and Distributed Software Architecture 7
1.2.1 Distributed Computing 7
1.2.2 N-Tier Architecture 8
1.2.3 Distributed Object Architecture 10

1.3 Service-Oriented Architecture and Computing 12
1.3.1 Basic Concepts and Terminologies 12
1.3.2 Service-Oriented Computing 15
1.3.3 Object-Oriented Computing versus Service-Oriented Computing 17
1.3.4 Service-Oriented System Engineering 18

1.4 Service-Oriented Software Development and Applications 20
1.4.1 Traditional Software Development Processes 20
1.4.2 Service-Oriented Software Development 20
1.4.3 Applications of Service-Oriented Computing 24
1.4.4 Web Application Composition 25

1.5 Enterprise Software Development 27

1.6 Discussions 30

1.7 Exercises and Projects 35

Chapter 2 Distributed Computing with Multithreading **41**

2.1 Introduction to C# and .Net 41
2.1.1 Getting Started with C# and .Net 41
2.1.2 Comparison between C++ and C# 43
2.1.3 Namespaces and the “using„ Directive 44
2.1.4 Class and Object in C# 46
2.1.5 Parameters: Passing by Reference with ref & out 49
2.1.6 Base Class and Base Calling Class Constructor 49
2.1.7 Constructor, Destructor, and Garbage Collection 50
2.1.8 Pointers in C# 50
2.1.9 C# Unified Type System 51

2.2 Memory Management and Garbage Collection 52
2.2.1 Static Variables and Static Methods 53
2.2.2 Runtime Stack for Local Variables 53
2.2.3 Heap for Dynamic Memory Allocation 55
2.2.4 Scope and Garbage Collection 56

2.3 General Issues in Multitasking and Multithreading 57
2.3.1 Basic Requirements 57
2.3.2 Critical Operations and Synchronization 57
2.3.3 Deadlock and Deadlock Resolving 60
2.3.4 Order of Execution 61
2.3.5 Operating System Support for Multitasking and Multithreading 61

2.4 Multithreading in Java 63
2.4.1 Creating and Starting Threads 63
2.4.2 Thread Synchronization 67
2.4.3 Synchronized Method 67
2.4.4 Synchronized Statements 71

2.5 Multithreading in C# 72
2.5.1 Thread Classes and Properties 72
2.5.2 Monitor 73
2.5.3 Reader and Writer Locks 82
2.5.4 Mutexes 86
2.5.5 Semaphore 86
2.5.6 Coordination Event 87

2.6 Exception Handling and Event-Driven Programming 89
2.6.1 Exception Handling 89
2.6.2 Event-Driven Programming 93
2.6.3 Case Study: An Electronic Commerce Application in Event-Driven Approach 99

2.7 Multithreading Performance 104
2.7.1 Amdahl's Law 104
2.7.2 Multithreading Experiments 105

2.8 Discussions 110

2.9 Exercises and Projects 111

Chapter 3 Essentials in Service-Oriented Software Development 121

3.1 Service-Oriented Software Development Environments 121
3.1.1 Overview 121
3.1.2 Windows Communication Foundation 123
3.1.3 Service Interface: Address, Binding, and Contract 124

3.2 Service Provider: Creating and Hosting Services 125
3.2.1 Getting Started with Developing WCF Services 126
3.2.2 Testing Web Services 129
3.2.3 File Service and Data Management 132
3.2.4 Hosting Your Web Services as a Service Provider 133
3.2.5 Source Code Deployment and Precompiled Code Deployment 135

3.3 Service Brokers: Publishing and Discovering Services 138
3.3.1 An Ideal Service Broker with All Desired Features 138
3.3.2 UDDI Service Registry 141
3.3.3 ebXML Service Registry and Repository 148
3.3.4 Ad Hoc Registry Lists 150

3.4 SOAP and HTTP 150
3.4.1 SOAP Format 150
3.4.2 HTTP 151
3.4.3 SOAP Over HTTP 152

3.5 WSDL: Web Services Description Language 154
3.5.1 Elements of WSDL Documents 154
3.5.2 WSDL Document Example 155

3.6 Service Requesters: Building Applications Using Services 157
3.6.1 Connecting Endpoint and Proxy 157
3.6.2 Creating a Windows Forms Application in ASP.Net 158
3.6.3 Developing a Web Site Application to Consume WCF Services 162
3.6.4 Consuming Services in Different Platform 166

3.7 Web Service Testing and Dynamic Service Invocation 167
3.7.1 Get Web Content from URL 167
3.7.2 Analyze WSDL Document 168
3.7.3 Test Web Service 169
3.7.4 Dynamic Web Services and Web Applications 169

3.8 Java-Based Web Service Development 169
3.8.1 Web Application Building Using AJAX Programming 169
3.8.2 Java-Based Web Service Development and Hosting 171

3.9 Discussions 174

3.10 Exercises and Projects 175

Chapter 4 XML Data Representation and Processing 181

4.1 XML Fundamentals 181
4.1.1 XML versus HTML 182
4.1.2 XML Syntax 183
4.1.3 XML Namespaces 186

4.2 XML Data Processing 187
4.2.1 DOM: Document Object Model 187
4.2.2 SAX: Simple API for XML 189
4.2.3 XML Text Writer 191
4.2.4 XML Processing in Java 192

4.3 XPath 193

4.4 XML Type Definition Languages 196
4.4.1 XML Document Type Definition (DTD) 196
4.4.2 XML Schema 199
4.4.3 Namespace 201
4.4.4 XML Validation 203

4.5 Extensible Stylesheet Language 205

4.6 Other Web Data Formats 209
4.6.1 XHTML 210
4.6.2 RSS 212
4.6.3 Atom 214
4.6.4 JSON 216
4.6.5 Google Protocol Buffers and BigTable 217

4.7 Discussions 218

4.8 Exercises and Projects 219

Chapter 5 Web Application and State Management 225

5.1 Models of Web Applications 225
5.1.1 Pure HTML-Based Web Application 225
5.1.2 Client-Side Scripting 226
5.1.3 Server-Side Scripting and Code-Behind Page Computing 227
5.1.4 Client-Side Out-Of-Browser Computing 228

5.2 Structure of Web Applications 229
5.2.1 Components of Web Applications 229
5.2.2 Server Controls 231
5.2.3 User Controls 235
5.2.4 Web Application Configuration 237
5.2.5 Global Application Class 241
5.2.6 Dynamic Link Library 242

5.3 State Management 243
5.3.1 Overview of State Management Techniques 244
5.3.2 View State 244
5.3.3 Session State 246
5.3.4 Cookies 250
5.3.5 Session State and Cookies 251
5.3.6 Application State and Service Execution Model 252

5.4 Save User Data to Server File System 253
5.4.1 File System and Standard File Operations 253
5.4.2 Reading and Writing XML Files 254

5.5 Caching and Recommendation 258
5.5.1 Output Caching 259
5.5.2 Fragment Caching 260
5.5.3 Data Caching 261

5.6 Dynamic Graphics Design and Generation 268
5.6.1 Dynamic Graphics 268
5.6.2 Getting Started with Dynamic Graphics Generation 269
5.6.3 Generating Dynamic Graphics in User Control 272

5.7 Animation and Phone Application Development 276
5.7.1 Silverlight Applications 277
5.7.2 Developing Windows Phone Apps Using Silverlight 279
5.7.3 Android Phone App Development with App Inventor 287

5.8 Architecture Design 293
5.8.1 ASP .Net Forms-Centric Architecture 294
5.8.2 ASP .Net MVC Architecture 294

5.9 Discussions 296

5.10 Exercises and Projects 297

Chapter 6 Dependability of Service-Oriented Software 305

6.1 Basic Concepts 305
6.1.1 Dependability 305
6.1.2 Dependability Attributes and Quality of Service 306
6.1.3 Security Issues in SOA Software 307

6.2 Access Control in Web Applications 309

6.2.1 IIS and Windows-Based Security Mechanisms 310
6.2.2 Forms-Based Security 311
6.2.3 User Registration and Account Management 315

6.3 Encryption and Decryption 318
6.3.1 Developing Encryption and Decryption Service in WCF 318
6.3.2 Developing a Secure Hashing Service in WCF 320
6.3.3 WCF Service Client 322
6.3.4 Error Control Code 324

6.4 Dependable Computing in Windows Communication Foundation 326
6.4.1 WS-Security 326
6.4.2 WS-Reliability 327
6.4.3 Transactions 329

6.5 Discussions 331

6.6 Exercises and Projects 333

Part II Advanced Service-Oriented Computing and System Integration 337

Chapter 7 Advanced Services and Architecture-Driven Application Development 338

7.1 Self-Hosting Services 338
7.1.1 Developing a Service and a Hosting Service 338
7.1.2 Developing Clients to Consume Self-Hosting Services 342

7.2 Advanced WCF Services 346
7.2.1 Advanced Communication Models 346
7.2.2 Advanced Bindings 349
7.2.3 Service Behavior and State Management 350

7.3 REST Concept and RESTful Services 355
7.3.1 REST Concept and REST Architecture 355
7.3.2 RESTful Services 356
7.3.3 Developing RESTful Service 360
7.3.4 Developing an Image Verifier in RESTful Service 362
7.3.5 Consuming Simple RESTful Service 366
7.3.6 Google RESTful Services: Map Services 367
7.3.7 Making Asynchronous Calls to RESTful Services 370
7.3.8 RESTful Service-Based Web Application Description Language 371

7.4 Workflow Foundation 373
7.4.1 Architecture and Components 373
7.4.2 Creating a WF Flowchart Application 375
7.4.3 Creating Custom Code Activity for Workflow Application 377
7.4.4 Implementing Finite State Machine in Workflow 380
7.4.5 Creating Workflow Service 381

7.5 Discussions 390

7.6 Exercises and Projects 391

Chapter 8 Enterprise Software Development and Integration 397

8.1 Overview of Composition Languages 397

8.2 BPEL 400
8.2.1 BPEL Activities and Constructs 400
8.2.2 BPEL Process 400

8.2.3 WSDL Interface Definition of BPEL Process 403
8.2.4 BPEL Process 405
8.2.5 An Example Invoking Real Web Services 407

8.3 Stateless versus Stateful Web Services 413
8.3.1 BizTalk's Singleton Object Approach 414
8.3.2 BPEL's Correlation Approach 414

8.4 Frameworks Supporting BPEL Composition 417
8.4.1 Oracle SOA Suite 417
8.4.2 ActiveBPEL 419
8.4.3 BizTalk 420
8.4.4 Simple Workflow Service 421

8.5 Message-Based Integration 422
8.5.1 Java Message Service 423
8.5.2 Microsoft Message Queue 424
8.5.3 Database-Supported Messaging 426
8.5.4 Email Supported Messaging 427
8.5.5 Enterprise Service Bus 428

8.6 Mashup for Web Application Integration 428

8.7 Other Composition Languages 432
8.7.1 SCA/SDO 432
8.7.2 WSFL: Web Services Flow Language 433

8.8 Discussions 434

8.9 Exercises and Projects 435

Chapter 9 Internet of Things and Robot as a Service 441

9.1 Introduction 441
9.1.1 Internet of Things 441
9.1.2 Service-Oriented Robotics Computing 443
9.1.3 Event-Driven Robotics Applications 444

9.2 Robot as a Service 447
9.2.1 Robot as a Service Definition 447
9.2.2 Robot as a Service Design 448
9.2.3 Robot as a Service Implementation 450

9.3 Robotics Developer Studio and Visual Programming Language 452

9.4 Simulating an ALU Using VPL 455
9.4.1 Logic Design of an Arithmetic and Logic Unit 456
9.4.2 Creating VPL Activities 456
9.4.3 VPL Diagram and Testing 458

9.5 Finite State Machine and VPL Diagram 461

9.6 Developing Service-Oriented Robotics Applications 466
9.6.1 Service Repository in Robotics Developer Studio 466
9.6.2 Sensor Service 467
9.6.3 Maze Navigation Algorithms 468
9.6.4 Implementing a Heuristic Maze Navigation Algorithm 468
9.6.5 Implementing a Wall-Following Maze Navigation Algorithm 475

9.7 Using Visual Programming Language in Other Platforms 477

9.8 ASU-VPL: A New Visual Programming Language 479

9.8.1 ASU-VPL versus Microsoft VPL 480
9.8.2 Drive-by-wire in ASU-VPL 482
9.8.2 Line-follower in ASU-VPL 485
9.8.3 Wall-following maze navigation in ASU-VPL 486
9.8.4 Other Platforms supported by ASU-VPL 487
9.9 Discussions 491
9.10 Exercises and Projects 493
Chapter 10 Interfacing Service-Oriented Software with Databases 497
10.1 Databases in Service-Oriented Software 497
10.2 Relational Databases in Service-Oriented Software 498
10.2.1 Interface between Database and Software 498
10.2.2 SQL Database in ADO .Net 500
10.2.3 DataAdapter and DataSet in ADO .Net 506
10.2.4 WCF RIA Service 508
10.3 LINQ, Language Integrated Query 509
10.3.1 What is the Purpose of LINQ? 510
10.3.2 Lambda Expressions 511
10.3.3 LINQ to Object 514
10.3.4 LINQ to SQL 516
10.3.5 LINQ to XML 519
10.4 XML-Based Database and Query Language XQuery 522
10.4.1 Expressing Queries 522
10.4.2 Transforming XML Document 524
10.4.3 XQuery Discussions 525
10.5 Discussions 526
10.6 Exercises and Projects 527
Chapter 11 Big Data and Ontology Systems 531
11.1 Big Data Systems 531
11.2 Big Data Management 533
11.3 Big Data Processing and Analytics 534
11.3.1 Map and Reduce 534
11.3.2 Hadoop and Cassandra 535
11.3.3 Apache Pig 540
11.3.4 Big Data Analytics 540
11.4 Big Data Applications and Case Studies 542
11.4.1 Big Data Applications in different domains 542
11.4.2 Network traffic monitoring 543
11.4.3 Recommender System 545
11.5 Semantic Web and Ontology 549
11.5.1 Ontology Language RDF 550
11.5.2 RDF Schema 552
11.5.3 Reasoning and Verification in Ontology 558
11.5.4 Web Ontology Language OWL 559
11.5.5 OWL-S and Semantic Web Services 563
11.5.6 Linked Data 564
11.5.7 Ontology Development Environments 565

11.6 Discussions 566
11.7 Exercises and Projects 569

Chapter 12 Service-Oriented Application Architecture 573

12.1 Introduction 573
12.2 Application Architectures 575
12.2.1 Dynamic Architecture via Dynamic Composition 576
12.2.2 Dynamic Recomposition 577
12.2.3 Lifecycle Management Embedded in Operation Infrastructure 578
12.3 Examples of Service-Oriented Application Architectures 580
12.3.1 IBM WebSphere Architecture 580
12.3.2 Enterprise Service Bus 581
12.3.3 SAP NetWeaver 582
12.3.4 User-Centric Service Oriented Architecture 584
12.4 Discussions 585
12.5 Exercises and Projects 587

Chapter 13 A Mini Walkthrough of Service-Oriented Software Development 591

13.1 Introduction 591
13.2 Sample Domain Model 595
13.2.1 Ontology Systems 595
13.2.2 Published Services 599
13.2.3 Published Workflows 602
13.2.4 Shipping Domain Collaboration Templates 603
13.3 Specific Requirements for a Project 605
13.4 A Working Example 607
13.5 Discussions 614
13.6 Exercises and Projects 617

Chapter 14 Cloud Computing and Software as a Service 619

14.1 Introduction 619
14.2 SaaS Maturity Model 622
14.3 Database Design for Multi-Tenancy SaaS 624
14.3.1 Resource Isolation Patterns 625
14.3.2 Security 628
14.3.3 Scalability 630
14.4 Google App Engine 630
14.4.1 Services 631
14.4.2 Datastore 632
14.4.3 Hints for Developing Scalable Applications 632
14.4.4 Development and Tooling 633
14.4.5 Other Constraints 633
14.5 Google File System 633
14.5.1 GFS System Structure and Operations 634
14.5.2 Lessons Learned from Developing GFS 635
14.5.3 Other Similar Projects 636

14.6 BigTable 636
14.6.1 Major Components 637
14.6.2 BigTable Overview 637
14.7 MapReduce 643
14.7.1 MapReduce Programming Model 643
14.7.2 Example 644
14.7.3 Applications 644
14.7.4 Execution Overview 645
14.7.5 Task Granularity and Fault-Tolerant Computing 645
14.8 Hadoop 646
14.8.1 Hadoop Ecosystem 646
14.8.2 Hadoop HDFS and MapReduce 646
14.8.3 Example 647
14.9 Microsoft Azure 648
14.9.1 Azure Architecture 649
14.9.2 Azure Elements 650
14.9.3 Creating Your First Application Azure Cloud 651
14.10 Salesforce.com 652
14.11 Cloud Computing Algorithms 658
14.12 Applications of Data Differencer 669
14.13 Discussions 670
14.14 Exercises and Projects 671

Part III Appendices: Tutorials on Service-Oriented Software Development 673

Appendix A Web Application Development 675

A.1 Design of Graphical User Interface 675
A.2 Discovering Web Services Available Online 680
A.3 Access Web Services in Your Program: ImageService 682
A.4 Access Web services in Your Program: Weather Forecasting Service 686
A.5 Access Web Services in Your Program: USZip Service 688
A.6 Creating ASP .Net Web Services 690
A.7 Exercises and Projects 693

Appendix B Service-Oriented Robotics Applications 697

B.1 Getting Started with Robotics Developer Studio and VPL 697
B.2 Programming Conditions in VPL 699
B.3 Programming Loop in VPL 700
B.4 Programming a Robot in a Simulation Environment 700
B.5 Programming the Robot to Traverse a Maze 705
B.6 Deploying the Program to a Real Robot 706

B.7 Programming the Arm of the Robot 707
B.8 Autonomous Robot in an Obstacle Course 709
B.9 Exercises and Projects 715
Appendix C ASU Repository of Services and Applications 717
References 723
Index 731

Preface

Web software development and cloud computing, which are based on Service-Oriented Architecture (SOA), Service-Oriented Computing (SOC), and Service-Oriented Development (SOD), represent the modern software engineering theory, practices, and technologies. The availability and unification of computing capacities and data resources are reshaping the world. The latest wonder of technology is the Big Data. The amount of the data is not the key. The relationship among all data is the key. Finding the connections of all related data and making use of the connections to make decisions become possible after the maturity of cloud computing and related data management technologies.

The text takes a comprehensive and coherent approach to study the latest distributed and service-oriented architecture, distributed computing paradigm, and distributed software development and integration technologies. The goal is to learn the concepts, principles, methods, development frameworks, and their applications. The methodology is learning by developing examples. In the service development part, we assume that students have good knowledge in object-oriented computing. Students learn to build services through class definition, interface specification, the association between class methods and service operations, service deployment and service hosting. In the software integration part, we assume that students have basic understanding of software architecture. We take an architecture-driven approach to help students creating the working solution step by step from their architecture design. The first step is to design the architecture, which includes the major components and the interconnection. The next step is to define the interfaces among the components using the standard data types. Finally, the behavior of each component is linked to remote service or local object. The elaborated architecture is automatically translated into the executable.

The text consists of 14 chapters and 3 appendices, which are organized into three parts.

Part I: Distributed Service-Oriented Software Development and Web Data Management

Chapter 1	Introduction to Distributed Service-Oriented Computing
Chapter 2	Distributed Computing with Multithreading
Chapter 3	Essentials in Service-Oriented Software Development
Chapter 4	XML Data Representation and Processing
Chapter 5	Web Application and State Management
Chapter 6	Dependability of Service-Oriented Software

Part II: Advanced Service-Oriented Computing and System Integration

Chapter 7	Advanced Services and Architecture-Driven Application Development
Chapter 8	Enterprise Software Development and Integration
Chapter 9	Internet of Things and Robot as a Service
Chapter 10	Interfacing Service-Oriented Software with Databases

Chapter 11	Big Data and Ontology Systems
Chapter 12	Service-Oriented Application Architecture
Chapter 13	A Mini Walkthrough of Service-Oriented Software Development
Chapter 14	Cloud Computing and Software as a Service

Part III: Appendices: Tutorials on Service-Oriented Software Development

Appendix A	Web Application Development
Appendix B	Service-Oriented Robotics Applications
Appendix C	ASU Repository of Services and Applications

Part I includes the first six chapters, which can be used for a service-oriented computing and distributed software development course at the senior level or graduate level of universities. This part emphasizes the computing paradigm, data representation, state management, and programming language-based SOC software development. It introduces fundamental concepts and principles, in addition to technologies and tools, which are not being taught in traditional software engineering courses.

Chapter 1 gives an overview and explains fundamental concepts of distributed software architecture, design patterns, distributed computing, service-oriented architecture, and enterprise software architecture. The connections and distinctions between object orientation and service orientation are discussed in detail.

Chapter 2 studies parallel computing in multithreading. It discusses threading, synchronization, coordination, event-driven programming, and performance of parallel computing under multicore computers.

Chapter 3 introduces the essential concepts and techniques in service-oriented architecture, particularly the three-party model of service-oriented software development: Service providers, service brokers, and service consumers. Service interfaces, service communication protocols, and service hosting are keys for supporting this new computing paradigm.

Chapter 4 discusses XML and related technologies, which are used in almost everywhere in service oriented software development. XML is used for representing data, interface, standards, protocols, and even the execution process definition. This chapter studies not only XML representations, but also XML processing and transforming.

Chapter 5 is a long chapter and comprises application logic, data and state management, and presentation design. It involves application building based on architecture design using existing services and components, stateful Web application development using different state management techniques, including view state, session state, application state, file management, and Web caching. At the presentation layer, it discusses dynamic graphics generation, animation, and phone app development.

Chapter 6 deals with the dependability issues in Web-based applications, including access control through Forms security, encryption and decryption applications, and Secure Sockets Layer in Web communication. The chapter also discusses the reliability issues in Web application design and particularly in Web communication.

Part II includes the next eight chapters. These chapters are built on the basic concepts and principles discussed in Part I, yet they do not rely on the details of the first six chapters. This part emphasizes software composition and integration using existing services and components. The approach is based on architecture-driven approach, higher-level of data management, and message-based integration.

Chapter 7 starts with reviewing service-oriented computing and service development covered in Part I. Then the chapter moves to discuss more advanced service development that supports asynchronous communications. It also presents more detail in RESTful service development that has been briefly

discussed in Part I. Finally, the chapter moves to Workflow Foundation that supports architecture-driven software development. It uses examples and case studies to demonstrate software development by drawing the flowchart consisting of blocks of services and local components, adding inputs/outputs to the blocks, and then compiling the flowchart into executable.

Chapter 8 further discusses flowchart-based and architecture-driven software development by using other process languages and developments environments. It starts with BPEL (Business Process Execution Language) and BPEL-based development environments. Then the discussion is extended into message-based software integration, Enterprise Service Bus, and Mashup tools for integrating Web contents.

Chapter 9 extends Web-based computing to Internet of Things and Robot as a Service (RaaS). As an example, robotics applications are studied in detail, which use the service-oriented Visual Programming Language (VPL) from Microsoft to develop robotics services and applications. A new visual programming environment, ASU-VPL, is introduced, which can be used as a follow-up language of Microsoft VPL. Full programming examples in ASU-VPL and hardware platforms supported are given.

Chapter 10 is on service-oriented database management, which focuses the interface between service-oriented software and relational database, XML database, and LINQ (Language Integrated Query) and using LINQ to access object, relational database, and XML database. LINQ is the focus of the chapter.

Chapter 11 studies the major issues in big data, including big data infrastructure, big data management, big data analytics, and big data applications. The relationship between big data and cloud computing is discussed. Ontology is presented as a part of big data representation and big data processing techniques.

Chapter 12 focuses on the application layer of service-oriented computing systems. It compares and contrasts the architectures and features of a few major enterprise software systems.

Chapter 13 offers a full case study on service-oriented software development.

Chapter 14 dedicates to cloud computing, which introduces the most recent trend in cloud computing and data centers. It covers multi-tenancy architecture, Web databases and file systems, scheduling, and fault-tolerant techniques in cloud computing and in real-time computing. It also presents cloud computing platforms and development environments from Google, Microsoft, and Salesforce.com. Software as a Service is the focus of the chapter.

Part III has three appendices that supplement and support the contents on Web application development and service-oriented robotics application development.

Appendices A and B contain tutorial-based materials that provide stepwise instructions, without missing pieces, to build working applications from scratch. These tutorials and exercises can help students to learn concepts by examples. This part, in conjunction with Chapter 9, can also be used for a freshman level course to introduce computing concepts through robotics programming and Web application programming.

Appendix C is the entrance to ASU Repository of Services and Applications. It lists and explains some of the deployed examples and URLs of SOAP services, RESTful services, Web applications, and other resources used in this text. Free services found on Internet come and go without any guarantee on quality of service. The repository provides a stable resource for teaching from this book without worrying about the availability and performance of the free services found on Internet. ASU Repository of Services and Applications is open to public and can be accessed at the link: http://venus.eas.asu.edu/WSRepository/

The book embraces extensive contents. It can be used in multiple courses. At Arizona State University, we use the book as the text for two major courses. The first course is CSE445/598 (Distributed Software Development), where the CSE445 session is for seniors and the CSE598 session is for graduate students. The course started in Fall 2006, and first edition of the book was developed for this course in 2008. Now the contents in Part I of this text are used for this course.

A second course CSE446/598 (Software Integration and Engineering) was piloted in 2010 and 2011, and then the course becomes a regular course in 2012. Part II of this text is used for this course.

Both CSE445 and CSE446 are required courses of the Software Engineering Concentration in the Computer Science program at Arizona State University.

The following table illustrates the lectures of CSE445/598 and CSE446/598. Each lecture is 75-min long and counts as 1.5 lecture hours. Each course is completed in about 44 lecture hours.

CSE445/598 Lecture by Lecture Contents	CSE446/598 Lecture by Lecture Contents
L01 - 1. Intro to Computing	Unit 1-1 Introduction
L02 - 1. Intro to Distributed Architecture	Unit 1-2 Web Data and Standards
L03 - 1. Intro to SOC Concepts	Unit 1-3 Self Hosting Service
L04 - 1. Intro to SOC Development	Unit 1-4 Advanced WCF Service
L05 - 2. Multithreading concepts	Unit 1-5 RESTful Concepts
L06 - 2. Multithreading in Java	Unit 1-6 RESTful Services
L07 - 2. Multithreading in C Sharp	Unit 2-1 Enterprise Architecture and Process
L08 - 2. Event-driven programming	Unit 2-2 Workflow 1 Concepts
L09 - 2. Threading-Multicore Performance	Unit 2-3 Workflow 2 Case Studies
L10 - 3. SOC Service Development	Unit 2-4 BPEL 1 Process
L11 - 3. SOC Hosting and Brokerage	Unit 2-5 BPEL 2 Case Study
L12 - 3. SOC Application Builder	Unit 2-6 BPEL 3 State and Frameworks
L13 - 3. SOC Development in Java	Unit 2-7 Message-Based Integration
L14 - 4. XML basics	Unit 2-8 Mashup
L15 - 4. XML processing	Mid-Term Review
L16 - 4. XML Schema	Unit 3-1 Device Integration
L17 - 4. XML Transformation	Unit 3-2 Robotics Studio and VPL
L18 - 5. Web Application Structure	Unit 3-3 VPL Maze
L19 - 5. Web Application Controls	Unit 4-1 ADO
L20 - 5. Config-Global-DLL-Cookies	Unit 4-2 LINQ to Object
L21 - 5. Session and File System	Unit 4-3 LINQ to SQL
L22 - 5. Web Caching	Unit 4-4 LINQ to XML
L23 - 6. Security-Reliability Concepts	Unit 4-5 XML Database
L24 - 6. Forms Security	Unit 5-1 Big Data Concepts and Domains
L25 - 6. Error Control and SSL	Unit 5-2 Big Data Processing
L26 - 6. Data Encryption-Hash-Reliable Msg	Unit 5-3 Big Data App and RDF-RDFS
L27 - 7. Dynamic Graphics - Animation	Unit 5-4 OWL and Ontology Frameworks
L28 - 7. Mobile Apps Development	Unit 6-1 Cloud Computing
L29 - 7. MVC Summary and Outlook	Unit 6-2 Cloud Computing Case Studies

We recommend teaching the two courses in a sequence. However, the two courses can be taught independently without making one to be the prerequisite of the other. In this case, the basic concepts and principles from Part I, including those from a part of Chapter 1 and the first section of Chapter 4, should be reviewed or be assigned as reading materials for preparing the course using Part II.

We like to thank many of our sponsors, colleagues, and students in this project, particularly in preparing this edition of the book, including Prof. Xiaoying Bai of Tsinghua University, Prof. Farokh Bastani of University of Texas at Dallas, Mr. Calvin Cheng of Banner Health, Dr. J. Y. Chung of IBM, Prof. Zhongshi He of Chongqing University, Prof. Mei Hong of Shanghai Jiaotong University, Dr. K. W. Hwang of IBM, Prof. Zhi Jin of Peking University, Prof. Yingxu Lai of Beijing University of Technology, Prof. Y. H. Lee of Arizona State University, Prof. Yisheng Li of Fudan University, Prof. K. J. Lin of University of California at Irvine, Mr. John Oliver of Intel, Dr. Raymond Paul of DoD OSD NII, Profs. Qian Wang and Yongkang Xing of Chongqing University, Profs. Ruzhi Xu and Zhizheng Zhou of Qilu University of Technology, Prof. S. S. Yau of Arizona State University, and Prof. I-Ling Yen of University of Texas at Dallas. They contributed to our understanding of the materials. We also acknowledge the generous support from Intel, the U.S. Department of Education, the U.S. Department of Defense, and the National Science Foundation. Without their support, this book would not be possible. We also thank the teaching assistants and research assistants at Arizona State University. They validated many of the examples and assignments used in the book. We also thank the teaching assistants and research assistants at Arizona State University, including Calvin Cheng, Gennaro De Luca, Jay Elston, Qian Huang, Adam Lew, Wu Li, Gavin Liu, Mengxue Liu, Sheng Liu, Sami Mian, Xin Sun, Jingjing Xu, Xinyu Zhou, and Peide Zhong. They implemented and validated many of the examples and assignments used in the book. Finally, we would like to thank our families for their support and understanding of taking on such a project while carrying out a full research and teaching load at the university.

Note for Instructors

All the assignments and projects have been classroom-tested at Arizona State University. Furthermore, all the code presented in this book has been developed and tested. Contact the authors if you are interested in obtaining more materials in this book. Instructor-only resources, such as complete presentation slides, assignments, and tests, can be obtained by directly contacting the authors at yinong@asu.edu.

Yinong Chen

Wei-Tek Tsai

June 2015

Part I

Distributed Service-Oriented Software Development and Web Data Management

Chapter 1 Introduction to Distributed Service-Oriented Computing

Chapter 2 Distributed Computing with Multithreading

Chapter 3 Essentials in Service-Oriented Software Development

Chapter 4 XML Data Representation and Processing

Chapter 5 Web Application and State Management

Chapter 6 Dependability of Service-Oriented Software