Table of Contents

Preface (Second Edition)........................................................................................................... xi
Preface (First Edition)............................................................................................................. xiii

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 Enterprise .............................................................................. 18
  1.3.5 Service-Oriented System Engineering ................................................................ 20
  1.4 Service-Oriented Software Development and Applications ...................................... 21
  1.4.1 Traditional Software Development Processes .................................................... 21
  1.4.2 Service-Oriented Software Development ............................................................ 21
  1.4.3 Applications of Service-Oriented Computing ..................................................... 25
  1.4.4 Web Application Composition .......................................................................... 26
  1.5 Discussions ................................................................................................................ 27
  1.6 Exercises and Projects ............................................................................................... 31

Chapter 2  Distributed Computing with Multithreading ......................................................... 39
  2.1 Introduction to C# and .Net ...................................................................................... 39
  2.1.1 Getting Started with C# and .Net ..................................................................... 39
  2.1.2 Comparison between C++ and C# ..................................................................... 42
  2.1.3 Namespaces and the using Directive .................................................................. 44
  2.1.4 The Queue Example in C# ................................................................................ 45
  2.1.5 Class and Object in C# ...................................................................................... 47
  2.1.6 Parameters: Passing by Reference with ref & out ............................................. 50
  2.1.7 Base Class and Base Calling Class Constructor ............................................... 51
  2.1.8 Constructor, Destructor, and Garbage Collection ............................................. 52
  2.1.9 Pointers in C# .................................................................................................... 52
  2.1.10 C# Unified Type System ................................................................................. 53
  2.2 Memory Management and Garbage Collection ........................................................ 54
  2.2.1 Static Variables and Static Methods ................................................................... 55
  2.2.2 Runtime Stack for Local Variables .................................................................... 56
  2.2.3 Heap for Dynamic Memory Allocation ............................................................. 58
Chapter 3  Essentials in Service-Oriented Software Development ............................. 121

3.1  Overview of Service-Oriented Software Development Environments ................... 121

3.2  Service Provider: Creating and Hosting Services ........................................... 123
3.2.1  Using ASP.Net to Create Web Services .................................................. 124
3.2.2  Program Your Services in C# ................................................................... 125
3.2.3  Testing Your Web Services ......................................................................... 126
3.2.4  Hosting Your Web Services as a Service Provider ..................................... 127

3.3  Service Brokers: Publishing and Discovering Services ....................................... 129
3.3.1  An Ideal Service Broker with All Desired Features .................................... 129
3.3.2  UDDI Service Registry .............................................................................. 132
3.3.3  ebXML Service Registry and Repository .................................................... 139
3.3.4  Ad Hoc Registry Lists ................................................................................ 141

3.4  SOAP ............................................................................................................... 141
3.4.1  SOAP Format ............................................................................................. 142
3.4.2  SOAP Over HTTP ....................................................................................... 143

3.5  WSDL: Web Service Description Language ....................................................... 144
3.5.1  Elements of WSDL Documents .................................................................. 145
3.5.2  WSDL Document Example ....................................................................... 146

3.6  Service Requesters: Building Applications Using Services ................................. 147
3.6.1  Connecting Endpoint and Proxy .................................................................. 148
3.6.2  Creating a Web Application Project in ASP.Net ......................................... 148
3.6.3  Creating GUI and Composing an Application Based on Remote Web Services .................................................. 151

3.7  Java-Based Web Service Development ............................................................... 156
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.7.1</td>
<td>Web Application Building Using AJAX</td>
<td>156</td>
</tr>
<tr>
<td></td>
<td>programming</td>
<td></td>
</tr>
<tr>
<td>3.7.2</td>
<td>Java-Based Web Service Development and</td>
<td>158</td>
</tr>
<tr>
<td></td>
<td>Hosting</td>
<td></td>
</tr>
<tr>
<td>3.8</td>
<td>Discussions</td>
<td>160</td>
</tr>
<tr>
<td>3.9</td>
<td>Exercises and Projects</td>
<td>161</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 4</td>
<td>XML Data Representation and Processing</td>
<td>167</td>
</tr>
<tr>
<td>4.1</td>
<td>XML Fundamentals</td>
<td>168</td>
</tr>
<tr>
<td>4.2</td>
<td>XML Data Processing</td>
<td>173</td>
</tr>
<tr>
<td>4.3</td>
<td>XPath</td>
<td>180</td>
</tr>
<tr>
<td>4.4</td>
<td>XML Type Definition Languages</td>
<td>182</td>
</tr>
<tr>
<td>4.5</td>
<td>Extensible Stylesheet Language</td>
<td>191</td>
</tr>
<tr>
<td>4.6</td>
<td>Other Web Data Formats</td>
<td>197</td>
</tr>
<tr>
<td>4.7</td>
<td>Discussions</td>
<td>205</td>
</tr>
<tr>
<td>4.8</td>
<td>Exercises and Projects</td>
<td>207</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chapter 5</td>
<td>Web Application and Data Management</td>
<td>213</td>
</tr>
<tr>
<td>5.1</td>
<td>Structure of Web Applications</td>
<td>213</td>
</tr>
<tr>
<td>5.2</td>
<td>Models of Web Applications</td>
<td>226</td>
</tr>
<tr>
<td>5.3</td>
<td>State Management</td>
<td>230</td>
</tr>
<tr>
<td>5.4</td>
<td>Exercises and Projects</td>
<td>236</td>
</tr>
<tr>
<td>5.5</td>
<td>Application State and Service Execution</td>
<td>237</td>
</tr>
</tbody>
</table>
8.2.1 BPEL Activities and Constructs .................................................. 320
8.2.2 BPEL Process ........................................................................... 321
8.2.3 WSDL Interface Definition of BPEL Process ................................. 323
8.2.4 BPEL Process ........................................................................... 325
8.2.5 An Example Invoking Real Web Services .................................... 328
8.3 Stateless versus Stateful Web Services ........................................... 334
8.3.1 BizTalk’s Singleton Object Approach ........................................ 335
8.3.2 BPEL’s Correlation Approach .................................................. 335
8.4 Frameworks Supporting BPEL Composition ..................................... 338
8.4.1 Oracle SOA Suite ..................................................................... 338
8.4.2 ActiveBPEL ............................................................................. 339
8.4.3 BizTalk .................................................................................... 341
8.5 Mashup for Web Application Composition ....................................... 342
8.6 Other Composition Languages ....................................................... 345
8.6.1 OWL-S .................................................................................... 345
8.6.2 SCA/SDO ................................................................................ 346
8.6.3 Workflow Foundation and Silverlight ......................................... 347
8.6.4 WSFL: Web Services Flow Language ......................................... 349
8.7 Discussions ................................................................................ 350
8.8 Exercises and Projects .................................................................. 351

Chapter 9 Service-Oriented and Event-Driven Robotics Applications .......... 357
9.1 Service-Oriented Robotics Computing ............................................. 357
9.2 Event-Driven Robotics Applications ............................................... 359
9.3 Robot as a Service in Cloud Computing ........................................... 360
9.4 Robotic Studio and Visual Programming Language ............................ 363
9.5 Simulating an ALU Using VPL ..................................................... 367
9.5.1 Logic Design of an Arithmetic and Logic Unit ............................... 368
9.5.2 Creating VPL Activities ........................................................... 368
9.5.3 VPL Diagram and Testing ......................................................... 370
9.6 Finite State Machine and VPL Diagram .......................................... 373
9.7 Developing Service-Oriented Robotics Applications .......................... 377
9.7.1 Service Repository in Robotics Developer Studio ........................... 377
9.7.2 Sensor Service .......................................................................... 378
9.7.3 Maze Navigation Algorithms .................................................... 379
9.7.4 Implementing a Maze Navigation Algorithm in VPL ........................ 380
9.8 Mapping VPL to Other Platforms .................................................. 387
9.9 Discussions ................................................................................ 389
9.10 Exercises and Projects ................................................................. 391

Chapter 10 Interfacing Service-Oriented Software with Databases ............. 395
10.1 Databases in Service-Oriented Software .......................................... 395
10.2 Relational Databases in Service-Oriented Software ........................... 396
10.2.1 Interface between Database and Software ................................... 396
10.2.2 SQL Database in ADO.Net ....................................................... 398
10.2.3 DataAdapter and DataSet in ADO.Net ....................................... 404
Chapter 11 Ontology and Semantic Web ................................................................. 425
11.1 Semantic Web and Ontology ........................................................................... 425
11.2 Ontology Languages RDF ............................................................................ 426
11.3 RDF Schema .................................................................................................. 428
11.4 Reasoning and Verification in Ontology ......................................................... 434
11.5 OWL: Web Ontology Language ..................................................................... 436
  11.5.1 From RDF to OWL ..................................................................................... 436
  11.5.2 The OWL Class and Property ................................................................. 436
  11.5.3 Boolean Combinations of Classes ......................................................... 437
  11.5.4 Property Restrictions .............................................................................. 437
  11.5.5 Synopsis of OWL Lite, DL, and Full ......................................................... 439
11.6 Ontology Development Environments ......................................................... 440
11.7 Discussions .................................................................................................... 441
11.8 Exercises and Projects .................................................................................... 443

Chapter 12 Service-Oriented Application Architecture ........................................ 447
12.1 Introduction ..................................................................................................... 447
12.2 Application Architectures .............................................................................. 449
  12.2.1 Dynamic Architecture via Dynamic Composition ..................................... 451
  12.2.2 Dynamic Re-Composition ................................................................. 452
  12.2.3 Lifecycle Management Embedded in Operation Infrastructure .............. 452
12.3 Examples of Service-Oriented Application Architectures ............................. 454
  12.3.1 IBM WebSphere Architecture .............................................................. 454
  12.3.2 Enterprise Service Bus ............................................................................ 456
  12.3.3 SAP NetWeaver ...................................................................................... 457
  12.3.4 User-Centric Service Oriented Architecture ....................................... 458
12.4 Discussions .................................................................................................... 459
12.5 Exercises and Projects .................................................................................... 461

Chapter 13 A Mini Walkthrough of Service-Oriented Software Development ......... 465
13.1 Introduction ..................................................................................................... 465
13.2 Sample Domain Model ................................................................................... 469
  13.2.1 Ontology Systems .................................................................................... 469
13.2.2 Published Services .......................................................... 473
13.2.3 Published Workflows .................................................... 476
13.2.4 Shipping Domain Collaboration Templates .......................... 477
13.3 Specific Requirements for a Project ...................................... 479
13.4 A Worked Example .......................................................... 480
13.5 Discussions .................................................................... 488
13.6 Exercises and Projects ...................................................... 491

Chapter 14 Cloud Computing and Software as a Service .................. 493
14.1 Introduction ................................................................... 493
14.2 SaaS Maturity Model ....................................................... 496
14.3 Database Design for Multi-tenancy SaaS ............................... 499
14.3.1 Resource Isolation Patterns ............................................ 500
14.3.2 Security .................................................................... 504
14.3.3 Scalability .................................................................. 506
14.4 Google App Engine .......................................................... 506
14.4.1 Services .................................................................... 507
14.4.2 Datastore ................................................................... 508
14.4.3 Hints for Developing Scalable Applications ...................... 508
14.4.4 Development and Tooling ............................................. 509
14.4.5 Other Constraints ........................................................ 509
14.5 Google File System ........................................................... 509
14.5.1 GFS System Structure and Operations ............................ 510
14.5.2 Lessons Learned from Developing GFS .......................... 511
14.5.3 Other Similar Projects .................................................. 512
14.6 BigTable ........................................................................ 512
14.6.1 Major Components ...................................................... 513
14.6.2 BigTable Overview ...................................................... 513
14.7 MapReduce .................................................................... 519
14.7.1 MapReduce Programming Model ................................. 519
14.7.2 Example .................................................................... 520
14.7.3 Applications .............................................................. 521
14.7.4 Execution Overview .................................................... 521
14.7.5 Task Granularity and Fault-Tolerant Computing ............... 522
14.8 Hadoop ......................................................................... 522
14.8.1 Hadoop Ecosystem ...................................................... 522
14.8.2 Hadoop HDFS & MapReduce ...................................... 523
14.8.3 Example .................................................................... 524
14.9 Microsoft Azure ............................................................. 525
14.9.2 Azure Architecture ....................................................... 525
14.9.3 Azure Elements ........................................................ 527
14.9.4 Azure Cloud Applications .......................................... 528
14.10 Salesforce.com ............................................................... 528
14.11 Prioritization and Scheduling ............................................ 534
14.12 Cloud Computing Algorithms .......................................... 540
14.13 Applications of Data Differencer ...................................... 550
Discussions .............................................................................................................. 551
Exercises and Projects .......................................................................................... 553

Part III  Appendix: Tutorials on Component-Based and Service-Oriented Software Development ............................................................................................................. 555

Appendix A Component-Based Movie and Game Programming .......................... 557
A.1 Developing a Game in an Engineering Process ............................................. 557
A.2 Basic Programming Concepts in Alice ......................................................... 558
A.3 Graphic Programming ................................................................................... 560
A.4 Online Tutorials and Examples .................................................................. 561

Appendix B Web Application Development ......................................................... 563
B.1 Design of Graphical User Interface ............................................................. 563
B.2 Discovering Web Services Available Online ............................................. 568
B.3 Access Web Services in Your Program: Cinema Service .............................. 570
B.4 Access Web services in Your Program: Weather Forecasting Service ........ 574
B.5 Access Web Services in Your Program: USZip Service ............................... 577

Appendix C Service-Oriented Robotics Applications ........................................... 579
C.1 Getting Started with Microsoft Robotics Studio and VPL Programming ........ 579
C.2 Programming Conditions in VPL ................................................................. 581
C.3 Programming Loop in VPL ........................................................................ 582
C.4 Programming a Robot in a Simulation Environment .................................... 582
C.5 Deploying the Program to a Real Robot ...................................................... 588
C.6 Programming the Arm of the Robot ............................................................. 589
C.7 Autonomous Robot in an Obstacle Course .................................................. 591

Appendix Exercises and Projects ........................................................................ 597

References .............................................................................................................. 601
Index ......................................................................................................................... 609
Preface (Second Edition)

Service-Oriented Software Development (SOSD) based on Service-Oriented Architecture (SOA) and Service-Oriented Computing (SOC) represents modern software engineering practices and technologies. The first edition of the book covered the concepts, principles, methodology, and the latest technologies in SOSD. As SOSD is still a rapidly developing young field, many new concepts as well as technologies have emerged since the publication of the first edition of the book two years ago. In the new edition of the book, we have embraced a large part of the new knowledge, including concepts, principles, and technologies developed in the past two years. Five new chapters are added and all the other chapters have been significantly revised and extended. The new chapters are:

Chapter Five on Web application and data management, which discusses stateful Web application development using different state management techniques, including view state, session state, application state, file management, and Web caching.

Chapter Seven on service-oriented and resource-oriented computing, which introduce Web service development, service hosting, and RESTful service development in Windows Communication Foundation.

Chapter Nine on service-oriented computing in robotics applications, which studies Visual Programming Language (VPL) and uses it develop robotics services and applications. Distributed robotics applications and Robot-as-a-Service (RaaS) are also presented.

Chapter Ten on service-oriented database management, which presents interface between service-oriented software and relational database, XML database, and LINQ (Integrated Language Query) and using LINQ to access object, relational database, and XML database.

Chapter Fourteen on cloud computing, which introduce the most recent trend in SOC, and it covers multi-tenancy architecture, Web databases and file systems, scheduling, fault-tolerant techniques, and real-time computing. It also presents cloud computing platforms and development environments from Google, Microsoft, and Salesforce.com, with the Software-as-a-Service as the main focus.

We organize the chapters into three parts, each of which can be used in a different course. 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, data management, and programming language-based software development. It introduces fundamental concepts and principles, in addition to technologies and tools, that have not been taught in traditional software engineering courses, such as multithreading, event-driven programming, service-oriented development, Web-based programming, Web data management, and reliability and security mechanism development.

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 detail of the first six chapters. This part emphasizes composition of higher-level data management and application building using services and large components. It covers Web service and RESTful service development in Windows Communication Foundation, Business Process Execution Language (BPEL) for enterprise software development, Visual
Programming Language (VPL) for event-driven software development and robotics applications, interfacing service-oriented software to databases, ontology languages and applications, service-oriented application architecture, case studies, and cloud computing. The materials can be used for a senior or graduate course. While most content in Part II is based on matured knowledge and technologies, many research questions are also discussed to help graduate students to identify their research directions and topics.

Part III contains tutorial-based materials that provide step-wise instructions to build working applications from scratch. The materials in this part are critical to students with an insufficient programming background who are learning Part I and Part II. These tutorials and exercises can simply help these students to connect the concepts and turn them into working programs. This part can be used for a freshman level course to introduce computing concepts through game programming, robotics programming, and Web programming.

At Arizona State University, we use the book as the text for two 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. This course mainly teaches the content from Part I. Biweekly programming assignments and projects are given at the end of each chapter. We also use the book for a newly developed course CSE494/598 (Service-Oriented Computing and Information Management), which teaches from the content of Part II. These two courses can be taught independently without making one to be the prerequisite of the other. However, basic concepts and principles from Part I, including those from Chapter One, Chapter Three, and the first section of Chapter Four, should be reviewed or be assigned as reading materials for preparing the required concepts to start Part II.

We like to thank many of our sponsors, supporters and colleagues in this project, particularly in preparing the second edition of the book, including Prof. Xiaoying Bai of Tsinghua University, Dr. Shuyuan Chen of SAP, Dr. J. Y. Chung of IBM, Prof. Zhihui Du of Tsinghua University, Mr. Marcos Garcia-Acosta of Intel, Prof. Mei Hong of Peking University, Dr. K. W. Hwang of IBM, Prof. Zhi Jin of Peking University, Prof. Y. H. Lee of Arizona State University, Prof. Yisheng Li of Fudan University, Mr. John Oliver of Intel, Dr. Raymond Paul of DoD OSD NII, and Prof. S. S. Yau of Arizona State University. They contributed to our understanding of the materials. We also acknowledge the generous support from Intel, U.S. Department of Education, U.S. Department of Defense, and National Science Foundation. Without their support, this book will not be possible. We also thank the teaching assistants and research assistants at Arizona State University involving Jay Elston, Wu Li, Guanqiu Qi, Qihong Edward Raleigh, Shao, Xin Sun, Le Xu, and Peide Zhong. They 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. This book also has a corresponding Web site at http://asusrl.eas.asu.edu/share/services/book/ where you can download resources related to this book. Instructor-only resources can be obtained by directly contacting the authors at {yinong, wtsai}@asu.edu.

Yinong Chen, Arizona State University
Wei-Tek Tsai, Arizona State University and Tsinghua University
Software development has evolved for several generations from imperative, procedural, object-oriented, to distributed object-oriented paradigms. As the emergence of service-oriented computing, distributed software development is shifting from distributed object-oriented development, represented by CORBA (Common Object Request Broker Architecture) developed by OMG (Object Management Group) and Distributed Component Object Model (DCOM) developed by Microsoft, to distributed service-oriented development. Service-oriented computing and service-oriented software development have been adopted and supported by all major computer companies, including BEA, Google, HP, IBM, Intel, Microsoft, Oracle, SAP, and Sun Microsystems, and their technologies have been standardized by OASIS, W3C, and ISO.

Before we start to introduce what this book is about, let us first clarify three fundamental concepts: service-oriented architecture, service-oriented computing, and service-oriented software development.

**Service-Oriented Architecture (SOA)** is a distributed software architecture, which considers a software system consisting of a collection of loosely coupled services that communicate with each other through standard interfaces and protocols. These services are platform independent. Services can be published in public or private directories or repositories for software developers to compose their applications. As a software architecture, SOA is a conceptual model that concerns the organization and interfacing among the software components (services). It does not concern the development of operational software.

**Service-Oriented Computing (SOC)** refers to the computing paradigm that is based on the SOA conceptual model. However, SOC goes a step further to include not only the concepts and principles, but also the methods, algorithms, coding, and evaluation, which are a large part of the software development process.

**Service-Oriented Development (SOD)** concerns the entire software development cycle based on SOA concepts and SOC paradigm, including requirement, specification, architecture design, composition, service discovery, service implementation, testing, evaluation, deployment, and maintenance. SOD also involves using the current technologies and tools to effectively produce operational software.

We use “Distributed Service-Oriented Software Development” as the title of the book to compare with the widely used “Distributed Object-Oriented Software Development” approach, and to emphasize the fact that service-oriented software development is distributed naturally. Not only is the software under development distributed in different computers in different locations, but also the development process is distributed, in the sense that the application builders, service brokers, and service providers are developers working independently in different locations, but following the same interfaces and standards. Furthermore, we have a chapter (Chapter Two) to discuss distributed computing in general and how SOA, SOC, and SOD fit into the framework of general distributed computing.
Recently, many SOA, SOC, and SOD books have been published in response to the growing requirements in these areas. These books fall into one of the three categories:

1. high-level concepts and principles in SOA;
2. one of the aspects of the SOC, such as BPEL, Ontology, or XML;
3. SOD using a specific platform, such as Visual Studio .Net, Oracle SOA Suite, Java EE, or WebSphere. Most of these books are written by developers, and are largely focused on the language, platforms, and tools.

Different from the existing books, this book takes a balanced approach to teach all three topics of SOA, SOC, and SOD in one course, and covers a large portion of each topic in depth. The main concern of the book is to teach the SOA/SOC concepts, principles, and methods.

We believe that students can better understand concepts, principles, and methods if they see a piece of working code that implements them.

We also introduce the cutting-edge technologies and tools that can be applied to develop operational software with reasonable size and functionality, such as an operational online bookstore, trading site, or a robotics program manipulating a real robot to traverse a maze with artificial intelligence. Such software can never be developed in a course assignment without the latest development tools and without using the services and components made available by professional service providers. Many exercises and at least one large project are given at the end of each chapter of the book for students to practice SOA and SOC concepts and to develop operational software.

The book is based on the materials taught by the authors in CSE445/598 (Distributed Software Development) course in Computer Science and Engineering at Arizona State University every semester since Fall 2006. The CSE445 session is for seniors and the CSE598 session is for graduate students. The CSE598 also has an online session that is taught to students in the executive master's program in engineering. Many of these students are on the side of software project management. A part of the advanced materials of the text was also taught in CSE 565 (Software Verification, Validation, and Testing). The objectives and outcomes of a course based on the text can include:

1. To develop an understanding of the software engineering of programs using concurrency and synchronization, with the following outcomes:
   * Students can identify the application, advantages, and disadvantages of concurrency, threads, and synchronization.
   * Students can apply design principles for concurrency and synchronization.
   * Students can design and write programs demonstrating the use of concurrency, threads, and synchronization.
2. To develop an understanding of the development of distributed software, with the following outcomes:

* Students can recognize alternative distributed computing paradigms and technologies;

* Students can identify the phases and deliverables of the software lifecycle in the development of distributed software;

* Students can create the required deliverables in the development of distributed software in each phase of a software lifecycle;

* Students understand the security and reliability attributes of distributed applications.

3. To develop an ability to design and publish services as building blocks of service-oriented applications, with the following outcomes:

* Students understand the role of service publication and service directories;

* Students can identify available services in service registries;

* Students can design services in a programming language and publish services for the public to use.

4. To build skills in using a current technology for developing distributed systems and applications, with the following outcomes:

* Students can develop distributed programs using the current technology and standards;

* Students can use the current framework to develop programs and Web applications using graphical user interfaces, remote services, and workflow.

This book is not for an introductory course in programming. Its main audiences are the seniors and graduate students in computer science and engineering, or software engineers with programming background. The readers are expected to be fluent in one of the object-oriented programming languages such as C++, C#, and Java. Furthermore, students are expected to understand basic software engineering principles.

The book consists of nine chapters and an appendix. Each chapter is a unit that can be taught in six to nine lecture hours, depending on the level of the detail the instructor wants to cover. They are

Chapter 1 Introduction to Distributed Service-Oriented Computing
Chapter 2 Distributed Computing with Multithreading
Chapter 3 Getting Started with Service-Oriented Software Development
Chapter 4 XML and Related Technologies
Chapter 5 Composition Languages for Service-Oriented Software Development
Chapter 6 Dependability of Service-Oriented Software
Chapter 7 Database and Ontology in Distributed Service-Oriented Software
This book is not intended to be a research monograph, but an undergraduate text for teaching senior and graduate students on SOA, SOC, and SOD. However, research students and working professionals may still find this book useful, because of its comprehensive and in-depth discussions of the state-of-the-art content, cutting-edge technologies, and professional development tools. The book is based not only on the teaching experiences of the authors in these areas, but also on the understanding and expertise that the authors have accumulated in their research in these areas.

As SOA, SOC, and SOD are new and dynamic, the technologies and tools are evolving rapidly. Some of the materials may need to be updated soon after the print of the book. It is our intention to cover the latest concepts and technologies, and we must cut in at some point in this process. We have put more emphasis on the SOA and SOC concepts, principles, and methods, which are relatively stable compared to the SOD technologies and tools. We started to teach the material of the book in Fall 2006. A large part of the development examples are initially based on .Net 2005. Now .Net 2008 is released. With little or no revision, we were able to test or convert all the examples into .Net 2008 before the printing of the book. We expect the examples to work for the new editions of the tools in the future.

The tutorials in the appendix of the book are an important addition to the book. They provide full detail of Web application development discussed in Chapter Three and the robotics software development discussed in Chapter Five. On the other hand, the tutorials can be taught independently of the main text to students with no programming experience. In fact, the content of the tutorials has been taught in a service-oriented computing course for high school students.

We would like to thank many of our sponsors, supporters and colleagues in this project including Prof. Xiaoying Bai of Tsinghua University, Prof. Gary Bitter of Arizona State University, Prof. Farokh Bastani of University of Texas at Dallas, Prof. Kuo-Ming Chao of Coventry University, Dr. Shuyuan Chen of SAP, Dr. J. Y. Chung of IBM, Prof. Zhihui Du of Tsinghua University, Dr. K. W. Hwang of IBM, Prof. Kane Kim of University of California at Irvine, Prof. Y. H. Lee of Arizona State University, Prof. Yisheng Li of Fudan University, Prof. K. J. Lin of University of California at Irvine, and Dr. Raymond Paul of DoD OSD NII, Dr. Mary White of Arizona State University, Prof. S. S. Yau, 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 the U.S. Department of Education and the U.S. Department of Defense. Without their support, this book would not be possible. We also thank the teaching assistants and research assistants at Arizona State University, including Zhibin Cao, Calvin Cheng, Sandy Chow, Jay Elston, Qian Huang, Sheng Liu, Zheng Liu, Wu Li, Xin Sun, Jingjing Xu, Xinyu Zhou, and Peide Zhong. They 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.

Yinong Chen
Wei-Tek Tsai

May 2008