THIRD 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

ISBN 978-0-7575-9239-3

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Table of Contents

Preface (This Edition)

Web software development and cloud computing based on Service-Oriented Architecture (SOA) and Service-Oriented Computing (SOC) represent the modern software engineering theory, practices and technologies. The book takes a comprehensive and coherent approach to address these issues. The goal is to learn the concepts, principles and methods in advanced software architecture, software engineering, and software development. The approach is learning by developing. We assume students have basic understanding of software architecture, and this book takes an architecture-driven approach to help students creating working solutions for their architecture design, including programming and code deployment. The text consists of fourteen chapters and two appendices, which are organized into three parts.

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 Data Management
Chapter 6	Dependability of Service-Oriented Software

Chapter 7	Advanced Services and Architecture-Driven Application Development
Chapter 8	Enterprise Software Development and Integration
Chapter 9	Service-Oriented and Event-Driven Robotics Applications
Chapter 10	Interfacing Service-Oriented Software with Databases
Chapter 11	Ontology and Semantic Web
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

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

Each part is relatively independent and 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 SOC software development. It introduces fundamental concepts and principles, in addition to technologies and tools, which have not

been taught in traditional software engineering courses. This part covers 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 software composition and integration using services and components. The approach is based on higher-level of data management and application building techniques. Part II covers advanced service and application development and integration in Windows Communication Foundation, Workflow Foundation for application integration, Business Process Execution Language (BPEL) for enterprise software integration, and Visual Programming Language (VPL) for event-driven software development and robotics applications, interfaces between service-oriented software and databases, ontology languages and applications, service-oriented application architecture, and cloud computing. The materials in Part II can be used for a senior or graduate course on advanced software engineering and software integration. While most contents in Part II are 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 Appendix A and Appendix 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 can also be used for a freshman level course to introduce computing concepts through robotics programming and Web application programming. Appendix C lists the deployed examples and URLs of services, applications, and other resources used in this text.

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.

Part II of the book is used for a newly developed course CSE494/598 (Software Integration Engineering) for seniors and graduate students.

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 One and the first section of Chapter Four, should be reviewed or be assigned as reading materials for preparing the required concepts to start the course using Part II.

The first edition of the book covered the concepts, principles, methodology, and the latest technologies in service-oriented software development. As this field is still a rapidly developing young field, many new concepts as well as technologies have emerged since the publication of the first edition in 2008. In the latest edition of the book, we have embraced a large of part of the new knowledge, including concepts, principles, and technologies developed in the past years. Five new chapters are added, and all the other chapters have been significantly revised and extended. The new chapters include the following:

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, Web caching, and dynamic graphics generation.

Chapter Seven on service-oriented and resource-oriented computing, which introduces Web service development, service hosting, RESTful service development in Windows Communication Foundation, and workflow development and application integration in Workflow 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 introduces the most recent trend in SOC. 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. Software-as-a-Service is the focus of the chapter.

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, 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, Prof. S. S. Yau of Arizona State University, and Prof. Lian Yu of Peking University. 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, including Jay Elston, Wu Li, Guanqiu Qi, Edward Raleigh, Qihong 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. Instructor-only resources, such as presentation slides, assignments, and tests, 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

June 2011 (Third Edition)

Preface (First Edition)

Software development has evolved for several generations from imperative, procedural, object-oriented, to distributed object-oriented paradigms. As the emergence of distributed computing and service-oriented computing, 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.

However, concepts, principles, and methods are not only explained in text and diagram, but also demonstrated in working code.

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.

This book covers SOA, SOC, and SOD topics in breadth and depth.

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 (first edition) 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 the following:

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

Chapter 8 Service-Oriented Application Architecture

Chapter 9 A Mini Walkthrough of Service-Oriented Software Development

Appendix Tutorials on Component-Based and Service-Oriented Software Development

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, includingZhibin 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 (First Edition)