



软件工程技术丛书

设计系列

面向模式的软件体系结构

卷2：用于并发和网络化对象的模式

Pattern-Oriented Software Architecture

Volume 2 Patterns for Concurrent and Networked Objects

Douglas Schmidt
Michael Stal 著
Hans Rohnert
Frank Buschmann

张志祥 任雄伟 肖斌 等译 贲可荣 审校



机械工业出版社
China Machine Press

软件工程技术丛书

设计系列

面向模式的软件体系结构

卷2：用于并发和网络化对象的模式

Pattern-Oriented Software Architecture
Volume 2 Patterns for Concurrent and Networked Objects

Douglas Schmidt
Michael Stal 著
Hans Rohnert
Frank Buschmann

张志祥 任雄伟 肖斌 等译 贵可荣 审校



机械工业出版社
China Machine Press

本书讲述用于创建并发和网络化系统的模式,共涉及17种模式与惯用法。这些模式构成了用于解决有关分布式、并发和网络化问题的模式语言的基础。本书的模式是创建并发和网络化系统的核心元素,包括服务访问、事件处理、并发控制、连接管理和初始化、事务、安全性等。

本书强调实际的解决方案,读者可以使用本书的方法解决特定的软件开发问题,获得对构造分布式及并发应用和中间件最佳实践的基本理解。书中含有多语言(C、C++和Java)编写的大量例子和已知使用,适于专业软件开发人员及计算机专业高校师生使用。

Douglas Schmidt, Michael Stal, Hans Rohnert & Frank Buschmann: Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects (ISBN: 0-471-60695-2).

Authorized translation from the English language edition published by John Wiley & Sons, Inc.
Copyright © 2000 by John Wiley & Sons, Ltd.
All rights reserved.

本书中文简体字版由约翰-威利父子公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号:图字:01-2002-0817

图书在版编目(CIP)数据

面向模式的软件体系结构 卷2:用于并发和网络化对象的模式 / 施密特(Schmidt, D.)等著;张志祥等译. -北京:机械工业出版社, 2003.2

(软件工程技术丛书 设计系列)

书名原文: Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects.

ISBN 7-111-11686-0

I. 面… II. ①施… ②张… III. 软件工程-系统结构 IV. TP311.5

中国版本图书馆CIP数据核字(2003)第010175号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:张金梅

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2003年8月第1版第1次印刷

787mm×1092mm 1/16·26.75印张

印数:0 001-5 000册

定价:59.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线电话(010)68326294

服务访问和配置模式

包装器外观 (Wrapper Facade) (第28页) 设计模式把现有的非面向对象的API所提供的函数和数据, 封装在更加简洁的、健壮的、可移植的、可维护的和内聚的面向对象的类接口中。

组件配置器 (Component Configurator) (第46页) 设计模式允许应用程序在不必修改、重新编译、静态地重新链接应用程序的情况下, 在运行时链接和解链它的组件实现。组件配置器进一步支持在不必关闭和重启运行的进程的情况下, 把组件重新配置到不同的应用程序进程。

拦截器 (Interceptor) (第66页) 体系结构模式允许透明地把服务加到框架中, 并且在某些事件发生时, 能自动地触发服务。

扩展接口 (Extension Interface) (第85页) 设计模式允许组件导出多个接口, 当开发人员扩展或修改组件的功能时, 此模式能防止接口的膨胀和客户机代码的破坏。

事件处理模式

反应器 (Reactor) (第108页) 体系结构模式, 使事件驱动的应用可以多路分解并分配从一个或者多个客户机发送给应用的服务请求。

主动器 (Proactor) (第130页) 体系结构模式使事件驱动的应用程序能有效地多路分解和分配由于异步操作的完成而激活的服务请求, 这样在不存在不利条件时能获得并发所带来的好处。

异步完成标记 (Asynchronous Completion Token, ACT) (第158页) 设计模式可以使应用程序能有效地多路分解和处理对调用服务的异步操作的响应。

接受器-连接器 (Acceptor-Connector) (第173页) 设计模式将网络化系统中合作的对等服务器的连接和初始化工作, 与该服务在连接和初始化之后所执行的处理分开。

同步模式

定界加锁 (Scoped Locking) (第199页) 是C++惯用法, 能确保当控制进入到某一范围时, 自动获得锁, 而当控制离开该范围时, 自动释放锁, 不管从该范围返回的路径是什么。

策略化加锁 (strategized locking) (第205页) 设计模式把同步机制参数化, 这一机制保护组件的临界区免受并发访问。

线程安全接口 (Thread-Safe Interface) (第212页) 模式将加锁开销减至最小, 并能保证组件间的方法调用不会因为想再次获得一个已经被组件拥有的锁而导致“自死锁”。

双检查加锁优化 (Double-Checked Locking Optimization) (第217页) 设计模式, 如果代码的临界区必须在程序执行期内只以线程安全的方式获得一次锁时, 该模式能够减少争用和同步开销。

并发模式

主动对象 (Active Object) (第226页) 设计模式将方法执行和方法调用分离, 加强并发和简化对驻留在自身控制线程中对象的同步访问。

监视器对象 (Monitor Object) (第245页) 设计模式使并发方法的执行同步化, 以确保任一时刻仅有一个方法在对象内运行。它也允许对象方法相互协调, 调度方法的执行顺序。

半同步/半异步 (Half-Sync/Half-Async) (第261页) 体系结构模式将并发系统中的异步和同步服务处理分离, 简化了编程, 同时又没有降低性能。该模式介绍了两个通信层, 一个用于异步服务处理, 另一个用于同步服务处理。

领导者/追随者 (Leader/Followers) (第273页) 体系结构模式提供一个高效的并发模型。在该模型中, 为了检测、多路分解、分配和处理事件源上引发的服务请求, 多线程轮流共享一个事件源集合。

线程特定的存储器 (Thread-Specific Storage) (第289页) 设计模式允许多线程使用一个“逻辑上全局的”访问点获得一个局限于某一线程的对象, 而不会导致对象访问中的加锁开销。

译者序

模式系统包括很多通用的模式，而本书集中深入地讨论用于并发和网络化的通用的、与应用领域无关的模式。这些模式是对《面向模式的软件体系结构 卷1：模式系统》一书中介绍的通用模式在两个软件开发日益重要的领域上的补充。

除了关注用于并发和网络化的与领域无关的通用模式之外，本书对已有的关于并发网络程序设计和面向对象设计的资料做了如下扩充：(1) 关于并发的网络程序设计的资料通常关注操作系统API的语法和语义。相反，本书介绍了如何在设计和实现高质量的并发和网络化的系统时有效地使用这些API。(2) 本书讨论了并发和网络应用程序的开发过程。(3) 本书中的模式并不仅仅构成一个模式分类或系统，它们相互补充，形成了用于并发和网络化软件的模式语言。本书描述了如何使用这些模式语言来构建成熟的并发和网络化软件系统和应用程序、Web服务和分布式对象计算中间件，以及底层的操作系统网络协议和机制。

本书中介绍的很多模式还可用于并发和网络化的语境之外。

本书对软件开发的初学者和专家都有帮助。它可以帮助初学者不需要积累多年的经验，就可在适当规模的项目上像专家一样工作，它可以帮助专家在大型复杂的软件设计上利用已定义属性，也可以促使他们学习其他专家的经验。

本书力图做到既是教材又是参考手册，帮助软件开发人员以一种新的方式思考软件体系结构，并提供一些技术来解决特定的不断再现的设计问题。把本书用做软件工程课程的指南，可以给学生提供大型软件设计的完整的新观点。把本书用做参考手册，可以方便查找全面的技术并随时可以使用。本书包含了涉及模式实际应用的许多指导原则和强制限制。

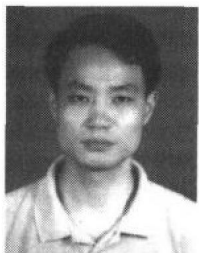
本书第1章、第2章由任雄伟翻译，第5~8章由肖斌翻译，其余章节由张志祥翻译。全书由贲可荣审校。孙宁、王淑雪、朱继梅等参与了本书的部分工作，在此表示感谢。

由于各种原因，译稿难免存在错误和疏漏，欢迎读者批评指正。

本书可作为计算机专业高年级本科生、计算机专业研究生的软件工程教材或参考书，亦可作为软件开发人员的参考手册。

译者
2003年3月

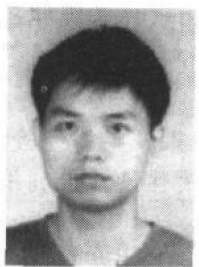
译者简介



张志祥 男, 1967年10月生, 湖北省天门市人。现任海军工程大学副教授。

1988年和1991年在海军工程大学计算机系分别获工学学士、工学硕士学位。1991年3月至今在海军工程大学计算机系任教。2000年9月起在华中科技大学计算机学院攻读博士学位。主要研究方向: 语义Web、程序设计方法。

先后参与多项军内科研项目的研制, 多项成果获军队科技进步奖。在《计算机研究与发展》等刊物发表论文多篇。译著包括《净室软件工程——技术与过程》(电子工业出版社, 2001年6月), 《Oracle 9i宝典》(电子工业出版社, 2002)等。



任雄伟 男, 1970年1月生, 湖北省武汉市人。现任海军工程大学讲师。

1992年在解放军通信工程学院无线通信工程专业获工学学士学位, 1997年在海军电子工程学院通信与电子系统专业获工学硕士学位。1998年3月至今在海军工程大学计算机系任教。2000年9月起在华中科技大学电信系攻读博士学位。主要研究方向: 多媒体通信、计算机网络。

先后参与“编队电子战训练模拟器”和“舰载电子战作战效能评估”等项目的研制。在《电视技术》、《计算机应用》等刊物发表论文多篇。



贵可荣 男, 1963年8月生, 江苏省海安县人。现任海军工程大学教授。

1983年在苏州大学数学系获理学学士学位, 1986年在南京大学数学系获理学硕士学位。1986年8月至1990年3月在海军工程大学计算机系任教。1994年6月在国防科技大学获工学博士学位。博士期间, 由国防科技大学计算机学院陈火旺院士指导, 主修计算机软件。1994年12月任海军工程大学计算机系副教授。1995年起担任计算机应用技术专业硕士生导师、海军工程大学学位评定委员会委员。经教育部批准, 2000年3月至2001年3月任武

汉大学软件工程国家重点实验室访问学者; 2000年被教育部确定为海军首批十名骨干教师之一。2000年12月晋升为教授。

主要译著有《净室软件工程——技术与过程》(电子工业出版社, 2001年6月), 《能力成熟度模型(CMM): 软件过程改进指南》(电子工业出版社, 2001年7月), 《基于项目的软件工程——面向对象研究方法》(机械工业出版社, 2002年6月), 《面向模式的软件体系结构卷1: 模式系统》(机械工业出版社, 2003年1月), 审校《实用软件测试指南》(电子工业出版社, 2003年1月), 参加《计算机科学技术百科全书》(清华大学出版社, 1998)的编写。先后承担国家自然科学基金项目、国家863项目和军队科研等五个项目。在“中国科学”、“软件学报”、“计算机科学”等刊物和会议发表论文50余篇。成果获军队科技进步奖。主要研究方向: 软件可靠性、软件质量保证技术、形式化方法等。兼任中国计算机学会计算机理论专业委员会委员、中国造船工程学会电子技术学术委员会委员, 《海军工程大学学报》、《舰船电子工程》等刊物编委。

译者序

模式系统包括很多通用的模式，而本书集中深入地讨论用于并发和网络化的通用的、与应用领域无关的模式。这些模式是对《面向模式的软件体系结构 卷1：模式系统》一书中介绍的通用模式在两个软件开发日益重要的领域上的补充。

除了关注用于并发和网络化的与领域无关的通用模式之外，本书对已有的关于并发网络程序设计和面向对象设计的资料做了如下扩充：(1) 关于并发的网络程序设计的资料通常关注操作系统API的语法和语义。相反，本书介绍了如何在设计和实现高质量的并发和网络化的系统时有效地使用这些API。(2) 本书讨论了并发和网络应用程序的开发过程。(3) 本书中的模式并不仅仅构成一个模式分类或系统，它们相互补充，形成了用于并发和网络化软件的模式语言。本书描述了如何使用这些模式语言来构建成熟的并发和网络化软件系统和应用程序、Web服务和分布式对象计算中间件，以及底层的操作系统网络协议和机制。

本书中介绍的很多模式还可用于并发和网络化的语境之外。

本书对软件开发的初学者和专家都有帮助。它可以帮助初学者不需要积累多年的经验，就可在适当规模的项目上像专家一样工作，它可以帮助专家在大型复杂的软件设计上利用已定义属性，也可以促使他们学习其他专家的经验。

本书力图做到既是教材又是参考手册，帮助软件开发人员以一种新的方式思考软件体系结构，并提供一些技术来解决特定的不断再现的设计问题。把本书用做软件工程课程的指南，可以给学生提供大型软件设计的完整的新观点。把本书用做参考手册，可以方便查找全面的技术并随时可以使用。本书包含了涉及模式实际应用的许多指导原则和强制限制。

本书第1章、第2章由任雄伟翻译，第5~8章由肖斌翻译，其余章节由张志祥翻译。全书由贾可荣审校。孙宁、王淑雪、朱继梅等参与了本书的部分工作，在此表示感谢。

由于各种原因，译稿难免存在错误和疏漏，欢迎读者批评指正。

本书可作为计算机专业高年级本科生、计算机专业研究生的软件工程教材或参考书，亦可作为软件开发人员的参考手册。

译者
2003年3月

前言

中间件是一组服务、协议和支撑工具，提供了构建现代分布式系统和应用程序的“管道”，而这些系统和应用是支持Web服务、分布式对象、协作应用程序、电子商务系统以及其他重要平台的基础设施。以前，很少听到中间件这个词，从事中间件开发的人更少。但是在过去的十年间，这个名词、有关它的研究和实践以及它的影响无处不在。但是，到目前为止，还没有一本书介绍如何构建网络化和并发的面向对象（OO）中间件，所以中间件设计还仍然像魔法一样神秘。本书阐明中间件的构建，希望能像专家一样，对于一般的设计问题、强制条件、成功的解决方案以及结论，为你提供合理的和带有经验指导的说明。

和大多数概念一样，确定中间件的范畴是很困难的。通常，中间件由构建系统和应用程序所需要的软件组成，但它并不是操作系统内核的固有部分。不过，中间件不太容易一眼就识别出来，它可能出现在库和框架、操作系统及其附件、Java虚拟机以及其他运行时系统中，也可能出现在大粒度软件组件中和像Web服务这样的部分终端产品中。

本书不是一本笼统介绍中间件，或者描述哪类应用程序和分布式系统体系结构可用中间件构造的教科书。相反，本书提出了一种模式语言，这种语言记录了构建大多数中间件具有的面向对象通信支持所用到的设计步骤。本书介绍的很多模式对于不是直接基于中间件的各层的系统和应用程序也是有用的。

本书强调实际的解决方案，而不是形式化理论。这里介绍的很多模式背后的基本思想对有经验的系统开发人员来说是熟知的——例如分配、多路分解、回调和配置，有时候是更一般的面向对象模式的变体——例如代理、适配器和外观。本书的主要贡献是基于这些思想的彻底的工程化解决方案。中间件开发人员必须解决范围很广的强制条件，包括吞吐量、响应能力、依赖性、互操作性、可移植性、可扩展性以及遗留软件的适应等等。这些强制条件的多样性和重要性说明了中间件模式的复杂性，这一点和在较小规模的面向对象应用程序和并发程序设计中看到的不一样。

这些强制条件的多样性，以及多年的工程实践，通常产生了很多设计考虑和工程化的权衡方案，它们将思想和思想在中间件框架中的表示分开。本书所使用的模式描述格式将解决方案表示成一组具体的设计步骤，从而简化这一分开的过程。很多步骤依次调用了其他的模式。将这些综合在一起，就形成一种模式语言，开发人员在设计服务和应用程序时可以从模式转到模式。

正如作者所提到的，本书所讨论的一些思想和技术是对W. Richard Stevens那些关于网络编程的开拓性著作（如[Ste98]）所阐述思想的补充。二者主要差别在于，本书更加关注更高层次的设计问题。例如，在讨论UNIX `select()`调用的输入和输出时，本书解释如何基于`select()`和其他操作系统调用，构建可组合的和灵活的框架，如反应器。

本书的一个隐含的主题是，如何应用由当前流行的平台所提供的处理I/O、线程、同步

和事件多路分解的功能作为基础，构建更高层框架和组件。强调在UNIX和微软操作系统中使用C/C++不会减弱这一主题。而在如下情况下，Java程序员会发现一些小小的不连贯：一是Java已经直接实现了一些本书中讨论的模式（如定界加锁），或者Java已经提供了按照某些模式的特殊实现而构造的框架（如对可配置组件的JavaBeans框架支持），以及一些Java不能访问底层系统机制（如同步事件多路分解）。

但是，熟悉Java、Smalltalk以及其他面向对象程序设计语言的读者将会受惠于模式所表达的中心思想，能更好地理解如何以及为什么有一些模式已经在语言功能和库中直接得到了支持，并可能基于其他模式建立有用的组件。例如，Java一直不提供对异步I/O有用的系统结构的访问，直到有了java.nio。不过在引用了本书中对主动器模式的一段描述后，我整理了一个Java版本，它用简单的spin-loop线程在多个通道上检测I/O的可用性，来模拟多路分解步骤。这样做效率不高，但对于它的使用语境已经足够了。

近几年来，本书中介绍的一些模式，如反应器模式，已经从设计发明的描述进化为设计模式的描述。每个正在实现可移植的面向对象中间件的人，都已经写了或者使用了至少一个包装器外观。本书对以往其他几种模式的讨论还涉及其设计的一些新作用。起初不能肯定是否应该将这些描述看成是模式，模式应该是被时间证明的、独立（重新）发现的解决方案。不过，随着时间的推移，本书作者和面向对象中间件业界越来越坚信本书中介绍的模式确实抓住了关键强制条件和设计问题的本质，并已经看到所介绍的解决方案在很多不同的使用语境中被反复用到。

我建议你分享这一现象。通过阅读——特别是使用——本书中的内容，你将会明白为什么像反应器和主动器这样的模式名称在面向对象中间件开发人员之间很普及，就像装饰器和观察者在面向对象GUI开发者中很普遍一样。

Doug Lea

纽约州立大学Oswego分校

内容简介

模式已经引起了软件开发业界的重视。自从第一本开创性的著作《设计模式——可复用面向对象软件的基础》[GoF95]问世以后,软件开发人员对模式产生了很高的热情。随后的一些工作,如“程序设计的模式语言(Pattern Languages of Program Design, PLoPD)”丛书[PLoPD1][PLoPD2][PLoPD3][PLoPD4]以及《面向模式的软件体系结构,卷1:模式系统》[POSA1]^①进一步激起了人们对模式的高度兴趣,这些兴趣最早是由更早期的对软件惯用法[Cope92]、构造建筑的模式[Ale79][AIS77]以及文化人类学模式[Bat79]的工作引起的。

本书是“面向模式软件体系结构(Pattern-Oriented Software Architecture, POSA)”系列丛书的第2卷。和第1卷《模式系统》^②[POSA1]一样,本书介绍了模式和最佳实践,这些模式和实践代表了用于构建产业化软件系统的具体的、经过证明了的有用技术。这些模式和最佳实践能够并且已经应用于广泛的领域,包括电信和数据通信、金融服务、医学工程、航天、制造过程控制以及科学计算。同时,它们也成为流行的分布式对象计算中间件的基础,这些中间件如CORBA[OMG98c]、COM+[Box97]、Java RMI[WRW96]和Jini[Sun99a]。

而且,本书中的所有模式和卷1都建立在同样坚实的概念基础之上。例如,它们使用同样的模式分类大纲、同样的模式描述格式、用多种编程语言(如C++、Java和C)表示实例和已有的使用。因此,本书遵循和《模式系统》一样的体系,有着同样的表现形式。

《模式系统》包括很多通用模式,与此相反,本书的内容比较集中,只讨论并发和网络化。本书中的所有模式集中在这两个领域,这样可以更深入地讨论与并发和网络化有关的问题,涉及其他不相关领域的模式时,这种讨论就不会太深入。因此,本书中的模式是对《模式系统》一书中介绍的通用模式在日益重要的软件开发领域上的补充。

我们也关注用于并发和网络化应用程序和中间件的通用的、与应用领域无关的模式。我们的目标是努力使书中的模式对日常的项目有所帮助,因此,在本书中并不讨论只针对于特殊应用领域的模式,如在[DeBr95][Mes96][ACGH+96]中的模式,只考虑属于电信领域的连网问题。

在集中讨论针对并发和网络化的通用的、与应用领域无关的模式之前,本书也对现存的有关并发网络程序设计和面向对象设计的资料做了补充:

- 关于并发的网络程序设计的文献通常关注操作系统API(如Sockets[Ste98]、POSIX Pthreads[Lew95]、Win32线程[Ric97])的语法和语义,流行的操作系统往往提供了这些API,作为对内核层通信框架(如System V STREAMS[Ris98][Rago93])的访问媒介。

① 我们引用《模式系统》一书时标引[POSA1]而不是标引作者。本书也一样,标引为[POSA2]。我们采用这种记法避免了在读者心目中留下只是第一作者撰写这本POSA著作的印象。

② 本书中文版已由机械工业出版社引进出版。——编者注

而本书着重介绍如何在设计和实现高质量的并发和网络化系统时有效地使用这些API。

- 讨论高层软件设计和质量因素的文献[Boo94][Mey97][DLF93]一般并不关注并发和网络应用程序的开发过程。本书可以填补这一空白。

本书和《模式系统》一书的另一个不同之处在于，本书中的模式并不仅仅构成一个模式分类或模式系统。相反，这些模式相互补充，共同为并发和网络化软件的模式语言提供基础。将这些模式和其他文献中的模式结合起来，我们描述了如何使用这些模式语言来构建高级的并发和网络化软件系统和应用程序、Web服务和分布式对象计算中间件，以及底层的操作系统网络协议和机制。

我们还将对单个模式的描述与它们形成模式语言的方式的讨论分开。首先相对完整地介绍模式本身，以便可以在最有用的语境中应用这些模式。随后的一章则描述模式如何交互，以及如何用其他模式进行补充。

不过，本书中的很多模式可应用于并发和网络化语境之外，指出这一点很重要。为了说明其应用领域的广泛性，我们列出了一些其他领域（如基于组件或交互式软件系统）中的应用实例。另外，我们还给出了一些实例，说明这些模式应用于日常生活中的情形。

大家可能对一些模式很熟悉，因为在PLoP丛书[PLoPD1][PLoPD2][PLoPD3][PLoPD4]以及《C++报告》(C++ Report)杂志已经发表了它们的初期版本。不过本书对这些早期版本做了相当大的改进：

- 首次将它们组织在一个文档中，这样有助于强调它们所表示的模式语言。
- 根据多方意见对这些模式进行了重写并做了重大修改。这些意见有些来自会议和专题学术讨论会上，有些通过电子邮件发来，有些通过大量的内部评审提出，还有我们的领导人的评审意见。
- 这些模式已经被转换成POSA模式格式，具有一致的书写风格。

本书的读者

和前一卷《模式系统》一样，本书的目的是供专业软件开发人员，尤其是那些正在构建并发和网络化系统的人员使用。本书可以帮助这些软件从业人员以一种新的方式考察软件体系结构，并帮助他们进行大规模的、复杂的中间件和应用程序的设计和编程。

本书也适合于大学高年级学生和研究生阅读，他们应具备足够的网络和操作系统基础知识，渴望掌握有效地设计和实现这样的系统所必备的核心原则、模式和技术。

结构和内容

本书可作为教科书，从头读到尾；也可以作为参考书，用于研究特定的模式在细节上的差别。

本书第1章“并发和网络化对象”，概述并发和网络化的面向对象应用程序和中间件的设计人员所面临的困难。使用了真实的例子——一个并发的Web服务器来说明该领域的一些关键问题，如服务访问和配置、事件处理、同步和并发。

第2章至第5章是本书的主要部分，这几章包含各种模式（用U2乐队的歌词来说就是

“实货，real things” [U2]），其中编纂了为开发高品质并发和网络化系统所确立的原则和技术。我们希望这些模式能成为你开发并发和网络化应用程序以及对你发现的模式建立文档时的有用的角色模型。

第6章“将模式组织在一起”讨论第2章至第5章中的模式之间相互联系的方式。我们还说明这些模式如何与文献中其他模式联系，构成用于并发网络化系统和中间件的模式语言。正如前面所提到的，一些模式还可应用于并发和网络化系统语境之外的场合。对于这些模式，我们总结了它们的应用范围。

第7章“模式的过去、现在和未来”回顾我们1996年在POSA卷1中做出的“模式向何处去”的预言，讨论过去4年间模式的实际发展方向，并分析模式和模式业界的现状。基于这些回顾，我们对自己关于模式应用和模式语言未来的研究方向的想法做了修正。

本书结束部分是对模式总的看法，常用术语的词汇表，本领域研究工作的参考文献，模式索引以及总的主题索引。

可以在<http://www.posa.uci.edu/>上找到与本书有关的补充材料。该URL中还包含有到ACE和TAO源代码的链接，其中包含本书所有模式的C++和某些Java实例。

毫无疑问，我们忽略了并发和网络化系统的一些方面，某些内容要随实践中应用和扩展模式语言时才会出现。如果你有任何评论、建设性的批评意见，或者是改进本书的风格和内容的建议，请用电子邮件通过patterns@mchp.siemens.de发给我们。也欢迎对我们在模式上所做的工作进行公开讨论，请用我们的邮件地址siemens-patterns@cs.uiuc.edu将反馈、评论和建议发给我们。在模式主页上可以找到订阅的方法，其URL是<http://hillside.net/patterns/>。该链接中还提供了有关于模式的多方面重要信息源，如已出版和将要出版的书籍，关于模式的会议和论文，等等。

致谢

很高兴地在此感谢那些帮助我们出版这本书的同仁，他们或者与我们分享了知识，或者对早期的草案进行了审查并提供了有用的反馈意见。

首先的评审是由我们尊敬的同事们做出的，他们是Regine Meunier、Christa Schwanninger、Martin Botzler、Lutz Dominick、Prashant Jain、Michael Kircher、Karl Präse和Dietmar Schütz。他们花费了许多宝贵的时间，来评审本书手稿，对本书进行润色并定稿。还要感谢分布式对象计算（Distributed Object Computing, DOC）组的成员——Tim Harrison、Prashant Jain、Carlos O’Ryan和Irfan Pyarali，我们和他们一起合作完成了本书的六个模式的最初版本。这些研究人员和本书的四位作者一起，组成了分别位于慕尼黑的西门子公司、华盛顿大学圣路易斯分校以及加利福尼亚大学欧文分校的POSA团队。

还要感谢Peter Sommerlad、Chris Cleeland、Kevlin Henney和Paul McKenney。Peter是我们的领队，他深入地审查了我们的所有材料，特别关心正确性、完整性、一致性和质量。我们的同行评审员Chris、Kevlin和Paul向我们提供了详细的反馈意见。他们对于改进本书做出了重大贡献。

也要感谢伊利诺依大学厄巴纳-尚佩恩分校的软件体系结构小组的成员，其中包括

Frederico Balaguer、John Brant、Brian Foote、Alejandra Garrido、Peter Hatch、Ralph Johnson、Dragos Manolescu、Brian Marick、Hiroaki Nakamura、Reza Razavi、Don Roberts、Les Tyrrell、Joseph W. Yoder、Wanghong Yuan、Weerasak Witthawaskul和Bosko Zivaljevic，他们主持了关于许多POSA2模式的作者专题讨论会。他们的评论对于我们提高本书的正确性和可理解性很有帮助。

世界各地的许多同仁对本书的早期版本提供了一些反馈意见，他们是Giorgio Angiolini、Brad Appleton、Paul Asman、David Barkken、John Basrai、Joe Bergin、Rainer Blome、Don Box、Martina Buschmann、Tom Cargill、Kobi Cohen-Arazi Chuck，以及Lorrie Cranor、Robert Crell、James O. Coplien、Ward Cunningham、Mike Curtis、Gisela Ebner、Ed Fernandez、Erich Gamma、Sonja Gary、Luciano Gerber、Bob Hanmer、Neil Harrison、Michi Henning、David Holmes、Tom Jordan、Christopher Kohlhoff、Fabio Kon、Bob Laferriere、Greg Lavender、Doug Lea、John MacMillan、Mittal Monani、Duane Murphy、Jaco van der Merwe、Michael Ogg、Bill Pugh、Patrick Rabau、Dirk Riehle、Linda Rising、Eric Samuelsson、Stefan Scherer、Wolfgang Schroeder、Richard Toren、Siva Vaddepuri、John Vlissides、Roger Whitney、Johnny Willemsen和Uwe Zdun。各种模式的“致谢”一节简述了他们的建议对完善本书的帮助。

还非常感谢华盛顿大学圣路易斯分校、加利福尼亚大学欧文分校、对象计算公司以及Riverace公司的DOC小组的新老成员，他们将本书中介绍的所有模式具体化、细化和优化，在ACE和TAO中间件项目中实现为组件和框架。这个富有灵感的小组中有Everett Anderson、Alex Arulanthu、Shawn Atkins、Darrell Brunsch、Luther Baker、Matt Braun、Chris Cleeland、Angelo Corsaro、Sergio Flores-Gaitan、Chris Gill、Pradeep Gore、Andy Gokhale、Priyanka Gontla、Myrna Harbison、Tim Harrison、Shawn Hannan、John Heitmann、Joe Hoffert、James Hu、Steve Huston、Prashant Jain、Vishal Kachroo、Ray Klefstad、Yamuna Krishnamurthy、Michael Kircher、Fred Kuhns、David Levine、Ebrahim Moshiri、Michael Moran、Sumedh Mungee、Bala Natarjan、Ossama Othman、Jeff Parsons、Kirthika Parameswaran、Krish Pathayapura、Irfan Pyarali、Carlos O’Ryan、Malcolm Spence、Marina Spivak、Naga Surendran、Selcuk Uelker、Nanbor Wang、Seth Widoff以及Torben Worm。还要感谢全球几千ACE和TAO的用户，他们在过去10年间的应用完善了本书中介绍的模式和框架组件。没有他们的支持、不断的反馈和鼓励，我们不会完成这本书。

还要对Johannes Nierwetberg、Lothar Borrmann和Monika Gonauser表示感谢，感谢他们对德国慕尼黑西门子的总公司技术部软件工程实验室在管理上的支持。还要感谢慕尼黑西门子通信设备事业部的Calinel Pasteanu，他对于撰写本书的繁重的工作与“Internet时代”交付产品的压力之间的冲突给予了充分的理解。

还要感谢在模式和ACE及TAO中间件框架等研究上的同事和赞助商，特别是Ron Akers(摩托罗拉)、Al Aho(朗讯)、Steve Bachinsky(SAIC)、Detlef Becker(西门子)、Jim Blaine(华盛顿大学)、John Buttitto(摩托罗拉)、Becky Callison(波音)、Wei Chiang(诺基亚)、Russ Claus(美国宇航局)、Joe Cross(洛克希德)、Bryan Doerr(波音)、Karlheinz Dorn(西门

子)、Sylvester Fernandez(洛克希德)、Andreas Geisler(西门子)、Helen Gill(DARPA)、Trey Grubbs(Raytheon)、Jody Hagins(ATD)、Andy Harvey(Cisco)、Thomas Heimke(西门子)、Kalai Kalaichelvan(北电)、Arvind Kaushal(摩托罗拉)、Steve Kay(Tellabs)、Chandra Kintala(朗讯)、Gary Koob(DARPA)、Sean Landis(摩托罗拉)、Rick Lett(Sprint)、Joe Loyall(BBN)、Mike Masters(NSWC)、Ed Mays(美国海军)、John Mellby(Raytheon)、Dave Meyer(Virtual Technology)、Eileen Miller(朗讯)、Stan Moyer(Telcordia)、Russ Noseworthy(Object Sciences)、Guru Parulkar(Cisco)、Dan Paulish(西门子)、James Plamondon(微软)、Dieter Quehl(西门子)、Lucie Robillard(美国空军)、Allyn Romanow(Cisco)、Rick Schantz(BBN)、Steve Shaffer(柯达)、Dave Sharp(波音)、Naval Sodha(爱立信)、Brian Stacey(北电)、Paul Stephenson(爱立信)、Umar Syid(休斯)、Dave Thomas(OTI)、Lothar Werzinger(Krones)、Shalini Yajnik(朗讯)和Tom Ziomek(摩托罗拉)。

特别感谢我们手稿的文字编辑, Wordmongers公司的Steve Rickaby, 他为我们内容增色不少。另外, 要感谢编辑Gaynor Redvers-Mutton以及其他在John Wiley & Sons出版社工作的人们, 是他们的帮助使本书得以出版。本书是Gaynor和Steve精心培育的我们的第二本书。他们的支持是巨大的, 我们期望在将来的POSA丛书中再次与他们合作。

最后, 要对已故的Richard Stevens表示深深的敬意。许多年前, 他的开创性的作品就激发我们去探究网络程序设计的奇妙世界。他的精神渗透在本书中。

作者简介

Douglas C. Schmidt

Douglas Schmidt博士是加利福尼亚大学欧文分校电子与计算机工程系的副教授。他还是国家高级研究计划局（DARPA）信息技术办公室（ITO）的项目经理，在美国的中间件研究中处于领先地位。在此之前，他是密苏里州华盛顿大学圣路易斯分校计算机科学系的分布式对象计算中心的副教授和主任。他的研究方向集中在模式和模式语言、最优化原理以及面向对象技术的经验分析技术，后者用于开发能运行于高速网络和嵌入式互连系统中的高性能和实时分布式对象计算中间件。

Douglas是一位在分布对象计算模式、中间件框架、实时CORBA和开放源码开发方面的国际知名的专家。他在顶级技术杂志、会议上发表了许多文章，出版了多本专著。他曾担任《C++报告》（C++Report）杂志的主编多年，与人合著了几本在模式[PLOPD1]和框架[FJS99a][FJS99b]方面的知名书籍。除了学术研究之外，Douglas领导了ACE和TAO的开发，二者是广泛使用的开放源码的中间件框架，其中包含了很多可重用组件，这些组件是用本书中介绍的模式实现的。

在“闲暇”时间，他喜欢和妻子Sonja跳交谊舞，他还喜欢举重、弹吉他、读世界历史、开雪弗莱车。

Michael Stal

Michael Stal于1991年加入德国慕尼黑的西门子的总公司技术部。早期他在开发编译器软件和计算机图形软件方面积累了丰富的经验。他研究C++运行时类型信息，并曾经服务于C++标准化组X3J16。从1992年开始，Michael的工作集中在使用Sockets、CORBA、COM和Java开发并发和分布式的面向对象系统。Michael是西门子公司与OMG在CORBA方面的主要联系人，也是西门子总公司技术部的分布式对象计算组的负责人。他是POSA第1卷《模式系统》的作者之一。

Michael的主要兴趣是研究用于有效地开发分布式系统的方法和用于描述中间件平台体系结构的模式。他在许多杂志上发表了多篇这方面的文章，并多次在会议上做报告。另外，他还是《Java Spektrum》的总编以及《Objektspektrum》的专栏作家和顾问委员会成员。前者是德国在Java平台方面的主要杂志，而后者是德国在对象技术方面的主要杂志。

在业余时间，Michael参加足球比赛，支持他喜爱的拜仁慕尼黑足球队，去慕尼黑周围的啤酒园观光，看管小猫别在家搞破坏，看电影，读物理、哲学和幽默方面的书。他是Douglas Adams、Scott Adams以及Terry Pratchett的影迷。

Hans Rohnert

Hans Rohnert博士是德国慕尼黑的西门子通信设备事务部的高级软件工程师。他的主要

研究方向是寻找有前途的软件技术并引入到新产品（如下一代移动电话）中。他的专业兴趣是软件体系结构、设计模式、现实世界程序设计等。他在动态图形算法、嵌入式Java虚拟机等方面做过无数次讲演。

目前Hans是一个专家组成员，该专家组的目标是定义用于小型设备的小型化KVM（键盘、视频和鼠标控制器）Java虚拟机以及类库。他从事的项目包括用于嵌入式服务器的服务端模块、C++中的工作流、对ATM交换的基本支持、CORBA客户端的Java GUI前端以及HTTP客户机。他是POSA第1卷《模式系统》的作者之一，PLoPD丛书的第四卷[PLoPD4]的主编之一。还是一名研究生的时候，他研究过组合算法。在职业生涯的早年，他在这个方面出版了一些著作、发表过一些讲演。

Hans是一个雄心勃勃的网球运动员，但输多赢少。他也喜欢爬附近的山，攀岩，去邻国滑雪。不过他最大的“爱好”是家庭，特别是在本书写作的最后冲刺阶段，家中诞生了一个婴儿。

Frank Buschmann

Frank Buschmann是德国慕尼黑西门子公司总部技术部门的高级工程师。他的研究兴趣包括对象技术、软件体系结构、框架和模式。他在这些领域发表了很多文章，这些文章可以在他与人合著的POSA第1卷中找到。Frank于1992~1996年期间是ANSI C++标准化委员会X3J16的成员。Frank发起并组织了在欧洲举办的第一次关于模式的会议——EuroPLoP 1996，他也是PLoPD丛书第三卷[PLoPD3]的主编之一。Frank参与了一些大规模工业软件项目的设计与实现，这些项目包括商务信息、工业自动化以及电信系统。

工作之余，Frank的大部分时间和妻子Martina一起享受生活，在慕尼黑的啤酒园消磨时光、玩自行车特技、滑雪、骑马，爱看支持的多特蒙德足球队比赛，一听歌剧就犯困，临睡前品苏格兰麦芽酒放松自己。