

面向方面的 软件开发

Aspect-Oriented Software Development

Robert E. Filman
Tzilla Elrad
Siobhán Clarke
Mehmet Akşit

主编

莫倩 王恺 刘冬梅 袁臻 译



机械工业出版社
China Machine Press

面向方面的 软件开发

Aspect-Oriented
Software Development

Robert E. Filman
Tzilla Elrad
Siobhán Clarke
Mehmet Akşit

主编

莫倩 王恺 刘冬梅 袁臻 译



机械工业出版社
China Machine Press

面向方面的软件开发 (AOSD) 是一个激动人心的新规范, 它被用以寻找软件系统中新的模块化特性。AOSD 和 OOP (面向对象编程) 不是相互竞争的技术, 实际上它们相辅相成得十分融洽。本书介绍了对实现方面的语言描述, 在编程语言领域中对方面的地位给予了关注; 描述了面向方面软件开发过程的技术; 以及详细讲解了使用方面技术的初始应用经验。

本书适合软件工程技术人員阅读。

Simplified Chinese edition copyright ©2005 by Pearson Education Asia Limited and China Machine Press.

Original English language title: Aspect-Oriented Software Development (ISBN: 0-321-21976-7) by Robert E. Filman, Tzila Elrad, Siobhán Clarke, Mehmet Aksit, Copyright ©2005.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Addison-Wesley.

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

版权所有, 侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号: 图字: 01-2004-6540

图书在版编目 (CIP) 数据

面向方面的软件开发 / 菲尔曼 (Filman, R. E.) 等主编; 莫倩等译. —北京: 机械工业出版社, 2006. 1

书名原文: Aspect-Oriented Software Development

ISBN 7-111-17533-6

I. 面… II. ①菲… ②莫… III. 面向对象语言-软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字 (2005) 第 117126 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: 赵利 李云静

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2006 年 1 月第 1 版第 1 次印刷

787mm×1020mm 1/16 · 30 印张

印数: 0 001-4 000 册

定价: 55.00 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 (010) 68326294

译者序

.....

面向方面的软件开发,是当今最为前沿和热门的一个软件开发技术研究和应用领域,也是一个将很快在我们的经济、生活和工作中产生重大影响的新兴技术领域。

面向对象的编程与传统的结构化编程相比,理论上会使应用程序更易编写、更易维护、更易演化。但这是理论上的优点,真实世界的对象十分复杂,人们需要通过不同的角度来看待和处理它们,也就是面向不同的方面(Aspect)。当面对应用程序中大量传统的编程对象时,如果要实现从不同的角度来看待和处理它们,就需要编写新的代码,这些新的代码会破坏编程对象的模块性,产生所谓的代码混杂。例如 Apache Web 服务器有 43 个功能模块,其中有 37 个模块都包含处理用户登录的代码。

面向方面的编程(Asspect-Oriented Programming, AOP)能够帮助我们解决这类问题。面向方面技术最基本的特性是多量化和不知觉性(第2章)。在上面的例子中,采用面向方面编程技术,只需要定义一个用户登录方面模块,就能同时实现 37 个模块对用户登录的处理,这就是多量化;而另一方面,这 37 个模块完全不需要了解任何有关用户登录方面模块的任何信息,也就是程序员并不需要在 37 个模块中编写和插入任何用户登录的代码,这就是不知觉性。这两个特性所带来的好处是显而易见的:多量化使得原本分散编写的代码集中到一个方面模块中,代码更清晰,编程量更小,程序更易编写、更易维护。不知觉性使得用户登录模块与其他功能模块彼此独立,当修改用户登录功能时,不会影响其他 37 个模块,这样的系统更易维护、更易演化。

简而言之, AOSD 将使软件开发者的工作更轻松!

本书分为三个部分共 31 章,每章独立成篇,但在逻辑上又彼此前后呼应、一脉相承。所有的作者均为 AOSD 领域的领袖人物或知名学者。第一部分“语言和基础”讨论了面向方面语言的基础、历史、本质、模型和实现。这一部分使我们能够深入理解面向方面语言的本质,知其所以然。第二部分“软件工程”立足于软件工程的角度,从软件开发生命周期的多个阶段讨论了面向方面软件的开发。编程只是软件开发生命周期中的一小部分,面向方面的编程如果没有面向方面的分析、设计和建模等相关过程的配合,就可能无法为我们带来它理论上的优点和益处。第三部分“应用”讨论了多个 AOSD 的不同应用领域,其中包括已经在商业应用领域中经受过实践检验的安全性应用,以及必将在未来影响我们的经济和生活的应用感知领域。

本书是目前 AOSD 领域最为全面、最为权威的论著,无论读者想了解或从事 AOSD 领域任何一方面的工作,它都是一本必读之书!

本书前 16 章,由莫倩博士翻译,第 17 章到第 21 章由王恺翻译,第 22 章到第 27 章由刘冬梅翻译,第 28 章到第 31 章由袁臻翻译,全书由莫倩博士组织翻译并负责统稿。

由于译者水平有限,加上 AOSD 又是一个新兴的领域,文中难免存在不妥之处,敬请读者批评指正。

译者

2005 年 9 月 10 日于北京

关于主编



Robert E. Filman 美国国家航空航天局 (NASA) Ames 研究中心高级计算机科学研究学会的高级科学家, 他致力于研究分布式应用框架, 在软件工程领域、分布式计算、网络安全、程序语言、人工智能、算法及人机交互方面发表了大量论著。他于 1979 年在斯坦福大学获得计算机科学的博士学位。

Tzilla Elrad 伊利诺斯理工学院计算机科学系的研究教授, 是并发软件系统研究组的负责人。她在 Hebrew 大学获得数学和物理学的学士学位, 分别在纽约的 Syracuse 大学和以色列的 Technion 大学获得计算机科学的硕士和博士学位。她的研究方向主要包括并发程序语言设计、自适应软件系统开发和形式化推理。

Siobhán Clarke 都柏林 Trinity 学院计算机科学系的讲师。她的研究方向主要是面向方面的软件开发, 以及针对移动、感知和上下文感知系统的上下文感知的计算和编程模型。她在都柏林的城市大学 (Dublin City University) 获得了学士和博士学位。在从事研究职业之前, 她曾在 IBM 任软件工程师 11 年。

Mehmet Aksit 在 Eindhoven 工学院获得硕士学位, 在 Twente 大学获得博士学位。现任 Twente 大学计算机科学系的教授, 他还在远程信息处理和信息技术中心任职。他及他的工作组开发了组合过滤器面向方面的编程技术、基于合成的体系结构/软件设计、用于产生稳定的软件系统的合成结构技术、在软件设计中引入模糊逻辑, 并创建了一些设计形式, 如用来在软件开发的阶段平衡质量因子的 Design Algebra。

关于撰稿人

Franz Achermann 是 Swisscom 公司的软件工程师。他感兴趣的研究方向主要有面向对象的语言和工具、软件工程，以及形式化方法。他由于组合语言 Piccola 方面的研究在计算机科学获得了博士学位。

Omar Aldawud 是朗讯公司的软件架构师。他拥有伊利诺斯理工学院计算机科学的博士学位，现任并发软件系统研究组的讲师和高级成员。Omar 感兴趣的研究方向包括 OO、AO、.net，以及使用 UML 的先进建模。

Gustavo Alonso 是瑞士联邦理工学院计算机科学系的教授。他在 Madrid 工程学院获得电信工程学士学位（1989），在 Santa Barbara 的加利福尼亚大学获得计算机科学硕士学位（1992）和博士学位（1994）。

João Araújo 是葡萄牙新里斯本大学的助教。他在巴西 UFPE 获得计算机科学学士学位和硕士学位，在英国兰卡斯特大学获得软件工程博士学位。他的研究方向包括需求工程、面向对象、面向方面的软件开发。

Michael Austermann 开发了 JMangler 框架，作为在 Bonn 大学的学位论文的一部分。拿到学位之后，他在德国科隆的 SCOOP Software GmbH 公司任顾问。期间他继续从事 JMangler 工作，发起并领导了 CC4J（一项使用 JMangler 开发代码覆盖应用）的开发工作。

Atef Bader 是朗讯的技术成员，伊利诺斯理工学院的助教，也是该院并发软件系统研究组的成员。他的研究兴趣包括面向对象技术、面向方面的软件开发、自动软件测试、并发软件系统、软件结构，他也拥有伊利诺斯理工学院的计算机科学博士学位。

Elisa L. A. Baniassad 于 2003 年获得大不列颠哥伦比亚大学计算机科学的博士学位，现在在都柏林 Trinity 大学攻读 NSERC 的博士后。研究方向为开发者在寻找和处理代码中的有关方面的经验研究。基于这些观察，她着眼于对系统需求中方面的阐述进行软件设计分析，并建立到详细设计和实现的可跟踪链接。

Ted Bapty 是 Vanderbilt 大学的电子工程研究助教，美国软件集成系统学会的高级研究科学家。他的研究兴趣主要是从不同粒度层上建立大型的、实时、分布式、嵌入系统的技术。

Stu Barrett 是德克萨斯州 Austin 的顾问。在 Texas Instruments, Raytheon, Microsoft 和 CALEB Technologies 从事过研究工作，包括嵌入式操作系统的设计、网络、人工智能、分布式系统/应用，以及中间件。1972 年获得密歇根理工大学的电子工程学士学位。不当顾问时，以点火高能火箭为乐趣。

Lodewijk Bergmans 是荷兰 Twente 大学的助教，在那里他获得了博士学位，主要研究兴趣是理解如何从分离的模块组合稳定、可扩展的软件。他通过开发基于组合过滤器的 AOP 模型来实现这一目标。

Gordon S. Blair 是兰卡斯特大学的分布式系统教授。他是国际中间件年会的负责人，1998

年担任该学会的主席。他的研究兴趣包括下一代中间件和反射。

Lynne Blair 是兰卡斯特大学计算机系的高级讲师。她具有形式化规范和验证分布式多媒体系统的背景, 目前正在从事面向方面的软件开发, 以及发生在这样的复杂系统中的交互问题(特征)的研究工作。

Noury Bouraqadi 于 2001 年建立了在法国 Douai 的 Ecole des Mines 的计算机科学实验室。从 1995 年开始, 他主要从事简化复杂软件的开发。为达到这一目标, 他主要从事反射和面向方面的编程, 以及分布式系统中的软件组件的开发。

Marcelo R. Campo 是 UNICEN 大学(阿根廷 Tandil)计算机科学系的副教授, ISISTAN 研究中心的负责人。1997 年获得 Federal de Rio Grande do Sul 联合大学(巴西 Porto Alegre)计算机科学博士学位。他的研究方向主要是智能软件工程辅助、软件结构和框架、agent 技术, 以及软件可视化。他是阿根廷国家科学技术研究理事会(CONICET)的研究成员。

Richard Cardone 是纽约 IBM Waston 研究中心的成员, 于 2002 年在 Austin 的 Texas 大学获得计算机科学博士学位。主要研究兴趣是开发有用的软件, 寻找使开发软件变得更有趣且高效的方法, 除了计算机之外, 他还爱好滑雪、自行车和摩托车。

Ruzanna Chitchyan 是兰卡斯特大学计算机系的博士生。其主要研究兴趣包括多维分解关注点, 面向方面的综合设计方法, 开发动态组合机制。她在 ECOOP 2003 参与组织了面向方面软件分析的第一个工作组。

Pedro J. Clemente 1998 年毕业于 Extremadura 大学计算机科学系, 目前就任讲师。他是 Quercus 软件工程组的成员, 致力于研究基于组件的软件工程和面向方面的软件开发。

Yvonne Coady 毕业于大不列颠哥伦比亚大学, 刚刚获得博士学位, 现在 Victoria 大学任助教。她的研究兴趣包括面向方面的软件开发、系统工程及系统架构的持久性。

Geoff Cohen 是未来软件技术的研究者和顾问。他是 OOPSLA 的“Onward! Seeking New Paradigms and New Thinking”程序委员会的创始成员之一, 在 2004 年度任主席。他是主要研究成员, 建立了有关计算机科学和生物学的交叉学科的美国国家科学委员会。他拥有 Duke 大学的计算机科学博士学位, 在普林斯顿大学公共国际事务的 Woodrow Wilson 学校获得学士学位。

Adrian Colyer 是英国 IBM 的高级技术专员。在 IBM, 他领导着一支团队, 不但研究面向方面(AO)的核心技术, 而且帮助开发组成员运用这一技术。此外, 他还领导了开源项目 AspectJ 和在 Eclipse.org 上的 AspectJ 开发工具项目。在投身于 AO 之前, 他主要以不同的角色研究企业中间件, 包括产品开发、策略和不断出现的技术。

Pascal Costanza 是 Bonn 大学计算机科学学会 III 的助理研究员。他的研究兴趣包括程序语言的设计, 特别是为未曾预料到的软件演化和面向方面的编程进行语言构建。他还参与了编程语言, 如 Gilgul 和 Lava、JMangler 框架、通用 Lisp 的面向方面扩展的设计和实现。

Bart De Win 是 Katholieke Universiteit Leuven 的计算机科学系的分布式系统和计算机网络(Distri-Net)研究组的助理研究员。其主要研究兴趣在于安全, 特别是安全软件工程, 匿名控制, 及嵌入系统环境下的安全。

J. Andrés Díaz Pace 是阿根廷 Tandil UNICEN 大学计算机系的助教, 于 2001 年取得硕士学位, 目前正在攻读博士学位。他的研究方向包括面向对象的框架、基于体系结构的设计、人工智能(AI)技术, 以及关注点的分离。

Remi Douence 从 1998 年开始在 Ecole des Mines de Nantes 任助教。1996 年, 他完成了关于正式学习功能语言实现的论文答辩。1997 年, 他在卡内基梅隆大学研究软件体系结构语言。他的研究兴趣是通用软件工程, 特别是 AOP 的语言支持。

Laurence Duchien 目前是 INRIA Lille 和法国 Lille 大学的教授, 自 2001 年就在 Jacquard 组从事研究

工作。她曾是法国巴黎 CNAM 的计算机科学系任副教授。1988 年获得巴黎大学 6 LIP6 实验室的博士学位, 1999 年取得格勒诺布尔市 Joseph Fourier 大学的 *diriger des* 研究资格。她的研究方向包括设计方法学、分布式面向对象应用的验证系统、反射和面向方面的编程。

Mike Feeley 是大不列颠哥伦比亚大学的副教授。他目前的研究项目主要在点对点的文件系统、移动专用网, 以及可扩展的系统结构领域。

G rard Florin 现在是法国巴黎 CNAM 计算机科学教授。他自 1990~1998 年, 任 CEDRIC (Centre d'Etudes et de Recherche) 的负责人。他毕业于 ENSET Cachan (“Ecole Normale Sup rieure de l'Enseignement Technique”)。1975 年获得巴黎大学 6 (University Paris 6) 的博士学位, 攻读博士学位期间主要研究关于形式化定义 Petri 网, 以及一套实现理论结果的工具。目前的研究兴趣包括分布式计算、容错性、反射及面向方面的编程。

Pascal Fradet 是 INRIA 的研究员。他的研究兴趣主要在编程语言方面: 功能编程、领域专用的语言、编译、翻译、分析等, 自 1997 年开始研究面向方面的编程, 研究兴趣点在于 AOP 的基础和语义问题。他认为方面是强加在程序上的正式属性。

Daniel P. Friedman 是印度大学计算机科学系的教授。他获得 Houston 大学的学士学位, Austin 的 Texas 大学的硕士和博士学位。参与合著了一系列重点在于编程功能性方法的小册子和巨著。

Aniruddha S. Gokhale 是 Vanderbilt 大学的电子工程及计算机科学系的助教, 软件集成系统学院的研究科学家。主要负责一些前沿的研究项目, 如中间件求解的建模。

Jeff Gray 是伯明翰 Alabama 大学计算机与信息科学系的助教, 也是面向对象分布式计算实验室的负责人之一。他的研究方向是允许在不同的抽象层次构建方面编织器的独立于语言的框架。

William G. Griswold 是圣地亚哥加利福尼亚大学的计算机科学与工程系的教授。他 1991 年获得华盛顿大学的计算机科学博士学位。2003 年, 他出任“面向方面软件开发”第二届国际会议的主席。研究兴趣包括普遍计算、面向方面的软件开发、软件演化和设计、软件工具和程序分析。

Thomas Gross 是瑞典 ETH 苏黎世的计算机科学教授。他拥有斯坦福大学电子工程的博士学位, 是卡内基梅隆大学计算机学院的荣誉教授。他的兴趣在于工具、技术和软件构建的抽象上, 致力于研究软件系统的设计和实现的许多方面。

John Grundy 是 Auckland 大学的软件工程教授, 发表了 100 多篇有关论文, 包括软件工具和方法、软件体系结构和基于组件的系统、面向方面的软件工程技术、软件过程技术和用户界面技术。

Jan Hannemann 目前是加拿大大不列颠哥伦比亚大学的计算机科学系的在读博士。他曾获得佐治亚理工学院的计算机科学硕士学位, 德国 Osnabr ck 大学的应用系统科学的证书。

William Harrison 自 1970 年以来一直任 IBM Thomas J. Watson 研究中心的研究员。他在优化编译器技术、软件开发技术、合作软件开发, 以及组合软件等方面做出了巨大贡献。他也是面向主题编程的共同发起人及 Concern Manipulation Environment 的倡导者之一。

Juan Hern ndez 是 (西班牙) Extremadura 大学计算机科学系的副教授, 在那里领导着 Quercus 软件工程组。拥有 Extremadura 大学的数学学士学位, Madrid 理工大学的计算机科学博士学位。他的研究兴趣在于基于组件的软件开发、面向方面以及分布式系统。

Jos  Luis Herrero 目前是 Extremadura 大学计算机科学系副教授, 是 Quercus 软件工程组的成员。他拥有 Granada 大学的计算机科学的学士学位, 2003 年获得 Extremadura 大学的计算机科学博士学位。他的研究兴趣是使用 UML 的面向方面和电子商务。

John Hosking 是 Auckland 大学计算机科学系的教授和负责人。他的研究兴趣及发表文章主要集中在软件工具、软件工程、一致性管理、有关医疗信息的应用及构建信息技术等方面。

Norm Hutchinson 是大不列颠哥伦比亚大学的副教授。他的研究兴趣主要包括并发和分布式操作系统、

软件实时系统，以及分布系统中的移动性。

Wouter Joosen 是 Katholieke Universiteit Leuven 的计算机科学系的分布式系统和计算机网络研究组的教授。他的主要研究兴趣在于分布式系统的软件结构领域、面向方面和面向组件的软件开发、中间件、软件安全和嵌入式系统。

Gregor Kiczales 是大不列颠哥伦比亚大学的教授，是 Palo Alto 研究中心的首席科学家、AOP 和 AspectJ 的创始者。他与 Danny Bobrow 和 Jim des Rivieres 合作撰写了《The Art of the Metaobject Protocol》一书。

Günter Kiesel 是 Osnabrück 大学的应用计算机科学的候补教授，是 Bonn 大学的讲师。他的研究兴趣主要在于基于对象的继承、角色、面向方面的编程，动态软件演化、程序转换、重构、对转换之间干涉的分析等方面。1989 年他在 Dortmund 大学获得了计算机科学的荣誉证书，2000 年在 Bonn 大学获得计算机科学荣誉博士证书。

Thomas Ledoux 自 1998 年开始就是 EMN (Ecole des Mines de Nantes) 的助教。他的博士工作主要是 OpenCorba，一个在 Smalltalk 元对象协议上实现的反射代理。他目前在 Obasco 工作组 (EMN/INRIA) 工作，致力于研究上下文感知的中间件计算和基于组件的自适应应用。

Diana Lee 是科学应用国际组织的数学科学家，在 NASA Ames 研究中心的神经工程实验室工作。Lee 女士还是 Santa Clara 大学工程学院的荣誉成员。在到 NASA 工作以前，曾就任于微电子和计算机研究组织、Lockheed-Martin、IBM。她获得了 Santa Clara 大学的数学硕士学位，Rice 大学的数学学士学位。

Fabrice Legond-Aubry 从 2001 年起开始攻读法国巴黎国会艺术学院的博士学位。目前在巴黎大学 6 LIP6 实验室工作，主要从事 JAC (Java 方面组件) 核心框架的开发。他还通过方面和约束来研究组件的适应性和重用性。他于 2000 年毕业于 Ecole 法国电子信息工程学院。

Wesley Leonç 拥有圣地亚哥加利福尼亚大学的硕士学位。目前在 NVIDIA 公司工作。

Karl Lieberherr Lieberherr 博士是美国东北大学计算机信息科学学院的教授，自 1985 年开始从事 Demeter 项目研究。他是面向方面软件开发 (AOSD) 会议的执行委员会成员，在 2004 年英国兰卡斯特召开的“世界软件工程会议 2004”担任主题发言人和程序委员会主席。

Calvin Lin 是 Austin Texas 大学计算机科学系的副教授。他于 1985 年在普林斯顿大学获得了 B. S. E 学位，1992 年在华盛顿大学获得博士学位。他的研究方向是语言和编译器。业余时间喜好飞盘游戏。

Ted Linden 在前微电子和计算机技术协会 (MCC) 管理对象基础设施项目 (OIP)。他的其余兴趣和发表文章包括人工智能、计划和调度、软件安全性和可靠性、机器人技术、灵活的软件开发方法等。1968 年在 Yeshiva 大学获得数学博士学位，目前在加利福尼亚 Palo Alto 担当顾问。

Cristina Videira Lopes 是加利福尼亚大学 Irvine 分校计算机学院的助教，从事程序语言和普遍计算方面的研究。她是 PARC 的 AOP 项目的发起人，拥有东北大学计算机科学系的博士学位。

David Lorenz 是东北大学计算机学院的助教。在 Technion 获得了博士学位，研究方向包括软件组件概念，特别是自适应组件和基于组件的设计 (尤其是 JavaBeans 技术)。

Laurent Martelli 是 AOPSYs 公司 (该公司致力于提供基于面向方面编程规范的，完备且易用的编程工具) 的合作发起人。1997 年毕业于法国 Evry 信息学院 (Institut d'Informatique d'Entreprise)，是 JAC (Java 方面组件) 框架的开发者之一，目前正使用 JAC 开发 AOPSYs 客户的应用。

Mira Mezini 是德国 Darmstadt 工程学院 (Darmstadt University of Technology) 的计算机科学教授。她发表了多篇关于面向方面编程领域的论文，而且负责了几个项目，不仅研究面向方面的技术，而且将技术在公司之间传播。她 10 多年来一直致力于研究改进面向对象系统的粒度问题。

Ana Moreira 是新里斯本大学 (New University of Lisbon) 的助教。她的研究兴趣包括对象技术、需

求工程以及面向方面。她多年来一直是 ECOOP 和 UML 程序员组织的成员，组织了多次工作室和会议。1994 年在 Stirling 大学获形式方法的博士学位。

Juan Manuel Murillo 是 Extremadura 大学计算机科学系的副教授，Quercus 软件工程组织的成员。在 Barcelona 理工大学获计算机科学的学士学位，2001 年在 Extremadura 大学获计算机科学博士学位。他的研究兴趣包括面向方面，Web 服务编排（orchestration），以及软件体系结构。

Gail C. Murphy 任大不列颠哥伦比亚大学的副教授。她拥有华盛顿大学的硕士学位。她的研究兴趣包括软件演化、软件设计、源代码分析、面向方面的软件开发及经验评估。

Sandeep Neema 是 Vanderbilt 大学 ISIS 的研究科学家。他的研究兴趣包括设计空间探索和基于约束的并行/分布式合成、实时、嵌入式系统，以及嵌入式实时环境下软硬件的动态重配置。

Oscar Nierstrasz 是 Berne 大学计算机科学学会的计算机科学教授，在那儿他领导着软件合成工作组（Software Composition Group）。参与编写了《Object-Oriented Reengineering Patterns》（Morgan Kaufmann, 2003）一书。他的研究兴趣在于面向组件软件技术的许多方面，特别是高层规范语言的设计和实现，以及支持重用性和演化的工具。他拥有滑铁卢大学（University of Waterloo）的数学学士（1979）学位，多伦多大学的硕士（1981）和博士学位（1984）。

Martin E. Nordberg III 是 Blueprint 技术公司的高级软件架构师，拥有近 20 年的软件开发经验，既包括技术复杂的项目，也包括组织复杂的项目。他的兴趣包括程序语言细节、面向对象和面向方面的分析和设计、设计模式，以及模块驱动的软件架构。他拥有美国 Rensselaer 理工研究所管理学院（Rensselaer Polytechnic Institute）的学士和工商管理硕士学位，以及麻萨诸塞大学的博士学位。

Joon Suan Ong 刚刚毕业于大不列颠哥伦比亚大学，获得了博士学位。他的研究兴趣包括体系结构、用户级网络和分布式系统。

Harold Ossher 自 1986 年以来一直在 IBM Thomas J. Watson 研究中心从事研究工作。他是面向主题编程、多维分解关注点和 Hyper/J，以及 Concern Manipulation Environment 的发起者。目前领导着研究组，进行该领域的研究和技术传播。

Klaus Ostermann Darmstadt 理工大学的副研究员。在 Bonn 大学取得硕士学位，在 Darmstadt 理工大学取得博士学位。研究方向为面向方面的语言设计与实现，以及面向方面的类型系统、基础和应用。

Renaud Pawlak 于 2002 年在 Conservatoire National des Arts et Métiers 获得计算机科学博士学位。目前在 INRIA Lille 攻读博士后，参与了 Jacquard 组的工作。他的研究方向包括软件工程，特别是面向方面的编程。目前正在研究分布式环境下的 AOP，推进 AOP 在业界的应用。

Frank Piessens 在 Katholieke Universiteit Leuven 的计算机科学系的分布式系统和计算机网络（Distri-Net）研究组任教授。他的主要研究兴趣是软件安全，特别是基于语言的安全性、软件工程中的安全性、中间件安全性和软件的弱点。

Andrei Popovici 1998 年获得 Darmstadt 理工大学的计算机科学硕士学位，之后就职于 ETH 苏黎士计算机科学系，作为助理研究员和助教。于 2003 年 6 月获得博士学位，学位论文是动态自适应和面向方面的编程。

Awais Rashid 是兰卡斯特大学计算机系的讲师。他的主要研究方向是面向方面的需求工程、混合面向方面编程、面向方面的数据库，以及对象数据管理。

Martin P. Robillard 最近毕业于大不列颠哥伦比亚大学的计算机系，获得博士学位，目前在蒙特利尔的 McGill 大学担任助教。他的研究方向包括程序理解、演化及模块化。

Isabelle Rouvellou 是 IBM T. J. Watson 研究中心成员，高级企业中间件（Advanced Enterprise Middleware）部门的负责人，从事有关促进下一代分布式应用中间件的设计和实现工作。由于她对 IBM Websphere 家族的产品所做出的贡献而获得多项 IBM 的奖励。她还参与合作撰写了有关分布对象技术、

消息、软件工程方面的论文，并获得专利。她拥有哥伦比亚大学的博士学位。

Fernando Sánchez 在西班牙 Extrmadura 大学计算机科学系任副教授，是 Quercus 软件工程组的成员。1992 年在 Sevilla 大学获得计算机科学学士学位，1999 年在 Extrmadura 大学计算机科学系取得博士学位。他的研究兴趣包括面向方面和 Web 工程。参与了互联网公司的多个项目和协议。

Douglas C. Schmidt 是 Vanderbilt 大学的电子工程和计算机科学系的教授。研究方向集中在模式、优化技术、分布式实时系统面向对象框架的经验分析，以及运行于高速网上的和嵌入系统互联的嵌入式中间件。

Lionel Seinturier 是法国 Jacquard 组 INRIA Lille 的副研究员，曾在巴黎大学 6 LIP6 实验室工作过。他的研究兴趣主要包括关于中间件平台的系统服务，以及反射和面向方面的编程。1997 年在巴黎 CNAM 获得博士学位，在那里主要研究分布式应用的设计原理。

Mario Südholt 是 École des Mines de Nantes (EMN) 的助教，拥有柏林理工大学的博士学位，是 INRIA 的 OBASCO 研究组的成员。他的研究方向是 AOSD，基于组件的编程，特别是 AOP 基础和形式化方法的应用。

Stanley M. Sutton Jr. 拥有 Boulder 科罗拉多大学 (University of Colorado) 的博士学位。目前在 IBM 的 T. J. Watson 研究中心担任咨询软件工程师，主要在软件组合部门从事面向方面和模型驱动的软件开发，同时在位于 Amherst 的麻萨诸塞大学的高级软件工程研究实验室工作，主要从事软件工程和 workflow 领域的研究。

Janos Sztipanovits 是 Vanderbilt 大学电子工程和计算机科学系的杰出教授，是软件集成学会的创办者。目前，他是 DARPA 信息技术办公室的执行代表和程序主席，同时是 IEEE 成员。

Peri Tarr 从 1996 年来一直是 IBM Thomas J. Watson 研究中心的研究员。是多维分解关注点与 Hyper/J 的联合发起人，Concern Manipulation Environment 的联合创建者。研究重点是贯穿软件生命周期的 AOSD 和形态基因软件（软件在生命周期保持延展性）。

Federico U. Trilnik 是阿根廷 Tandil 的 UNICEN 大学计算机科学系的研究助理，于 2000 年取得系统工程学士学位，目前正在攻读硕士学位。研究兴趣主要包括面向对象框架、多代理系统，以及面向方面的技术。

Robert J. Walker 是 Calgary 大学的助教，于 2003 年 3 月获得博士学位，研究方向是执行源模块（称为隐含上下文）的本地转换机制，及其用来简化软件演化和重用的应用，其他的兴趣点包括经验确认和软件可视化。

前言



软件开发是不断变化的。互联网的出现、商业计算、了解计算机的消费群体、计算和通信成本的指数级下降,以及长生命周期系统日益增多的动态环境,这些都迫使软件开发者必须使用更好的方法来建立和发展系统。一股热潮席卷了软件开发过程、系统结构、编程、质量保证以及维护等领域。

软件的目标是要对这个世界的部分元素或信息流建立计算模型。即使是最微不足道的软件系统,实现该系统的工程都需要(有可能是反复地)将系统分解成可以独立创建和管理的模块。20世纪90年代关于系统模块化特性的面向对象观点呈现上升趋势(并且可能成为主流)。面向对象方法的焦点在于选择对象作为模块的主要单元,并将这些对象与系统的所有行为联系起来。对象成为问题领域或者计算过程的主要元素。

面向对象方法正面临它的局限性。在创建软件系统时人们所关心的东西(称为关注点)并不能清晰地定位于某个特殊“东西”的行为上。建立不同的系统需要同时管理许多关注点。关注点的案例范围包括从非功能概念,如安全、可靠性、可控性等,扩展到精确的实现技术,如并发控制、缓存、错误恢复。由于传统的编程技术着眼于产生直接的指令序列,因此要求程序员全面掌握编程过程中的所有关注点。程序员必须清楚地将实现这些关注点的命令与实现主要应用功能的代码混合在一起。这会导致产生包含混乱代码、大量错误并难以维护的系统。

新技术的产生,允许包含更丰富的程序规范和关于这些规范的更好的模块化特性。除了这些新技术之外,还出现了使用这些新技术的崭新的软件工程方法论。在这些技术中,最令人激动的是面向方面的软件开发(AOSD)技术。AOSD编程技术(面向方面的编程,或AOP)为独立表示关注点提供了语言机制,并且提供了将这些关注点编织成为一个工作系统的实现技术。面向方面的软件工程(AOSE, Aspect-oriented software engineering)技术正是为了管理使用这种新规范的开发系统过程而出现的。

关于本书

本书内容源于2001年10月《Communications of the ACM》中面向方面技术专题的经验。当时提交的论文达到了我们出版篇幅的7倍;它已经成为ACM数字图书馆的热点专题。同时我们还看到有关面向方面主题的学术研讨会越来越多。很明显,在计算机科学团体中涌现出了对有关方面技术的更大的学习热情。这令我们产生了两个想法——一是召开有关面向方面软件开发的国际会议;二是出一本书,向一些先进的软件开发团体介绍本领域出现的不同思想。由面向方面软件组织与ACM联合发起的会议最近已经召开了第3次年会,在ACM数字图书馆有会议论文集。

我们希望本书能够成为一本有关软件工程的专辑,为软件开发中对此有学习兴趣的高级软件工程师介绍软件开发领域内的这一新方向。为了出版此书,我们向本领域的一些研究人

员约稿，其中有些文章是有关他们当前工作综述的最新论文，有些则是对一些指定的已有论文的再版。本书就是对所有这些提交论文的编选汇总。

如何阅读本书

本书的第 1 章介绍了面向方面软件开发的概概念，建议所有读者阅读。除此之外，尽管实际的系统开发是依托于一个工程活动的所有过程的，但系统开发的基础却是程序语言和模型。尽管方面技术还没有充分成熟到提供令人信服的实际系统数量，但一种软件技术的最终验证还是其所支持构建的系统。因此，在简介之后，本书按三部分组织：第一部分主要是对实现方面的语言描述，在编程语言领域中对方面的地位给予了关注。第二部分描述了面向方面软件开发过程的技术。第三部分详细介绍了使用方面技术的初始应用经验。每一部分都有一个内容介绍章节，介绍了界定有关主题中方面技术的范围，描述了每一章在此领域中的位置。除此介绍之外，各个章节（有助于成为一本关于研究现状的编选著作）基本上是独立的。除了作者在文中明确引用其他文章之外，几乎所有章节都是独立成篇的。

致谢

我们要感谢 Craig Thompson 和 Tom Poppendieck 仔细阅读了本书的手稿，感谢 Ryan Van Roode 在本书出版上给予的帮助，以及那些撰稿者为使本项目得以实现所付出的努力和耐心。

目 录

译者序		
关于主编		
关于撰稿人		
前言		
第 1 章 概览	1	
1.1 本书的组织	1	
1.2 常用术语	2	
1.3 历史背景	3	
第一部分 语言和基础		
传统工程	5	
软件开发是工程	6	
章节说明	7	
方面语言设计的角度	10	
参考文献	11	
第 2 章 面向方面编程的多量化和 不知觉性	12	
2.1 引言	12	
2.2 局部性和单一性语句	12	
2.3 不知觉性	13	
2.4 多量化	13	
2.4.1 静态多量化	14	
2.4.2 动态多量化	14	
2.5 实现问题	15	
2.6 面向方面的语言	15	
2.7 相关工作	16	
2.8 结束语	16	
致谢	17	
参考文献	17	
第 3 章 N 度分离：关注点的多维 分离	21	
3.1 引言	21	
3.2 动机	22	
3.2.1 第一轮	22	
3.2.2 进化 SEE：环境灾难	23	
3.2.3 事后分析	24	
3.2.4 打破统治地位	25	
3.3 关注点的多维分解	25	
3.3.1 传统软件模型	25	
3.3.2 多维分解：超切片	26	
3.3.3 使用超模块组合超切片	28	
3.3.4 使用模型	30	
3.4 实例化	31	
3.4.1 映射到单元和模块	31	
3.4.2 超切片表示	32	
3.4.3 支持组合	32	
3.5 相关工作	32	
3.6 结论和将来的工作	33	
致谢	34	
参考文献	34	
第 4 章 组合过滤器的原理和设计 原理	37	
4.1 引言	37	
4.2 例子：社会安全服务	38	
4.2.1 应用的概述	38	
4.2.2 软件系统	38	
4.3 用组合过滤器进行对象内横切	39	
4.3.1 关注点实例=对象+过滤器	40	
4.3.2 演化步骤 1：保护文档	41	
4.3.3 组合过滤器解决方案	41	
4.3.4 消息处理	43	
4.3.5 对象内部横切	45	
4.4 对象间横切	45	

4.4.1 演化步骤 2: 添加 workflow 管理	46	6.2.1 联结点	76
4.4.2 演化步骤 3: 添加日志	49	6.2.2 切入点	77
4.5 评估	51	6.2.3 通知	78
4.5.1 CF 模型的实现	51	6.2.4 类型间声明	80
4.5.2 内置式组合过滤器	52	6.2.5 方面	81
4.6 结论	53	6.2.6 进一步阅读	82
参考文献	54	6.3 AspectJ 工具	82
第 5 章 从历史视角探讨 AOP (名称的由来)	58	6.3.1 历史	82
5.1 风格问题	58	6.3.2 编译器	83
5.2 20 世纪 90 年代早期的研究趋势	59	6.3.3 AspectJ 的 IDE 支持	84
5.2.1 问题的描述	59	6.3.4 使用 ajdoc 生成文档	86
5.2.2 对问题和专门解决方案的分析	59	6.3.5 Ant 任务	86
5.2.3 识别可以被分离的关注点	60	6.4 将来的方向	86
5.2.4 分离技术	60	6.5 结束语	87
5.3 AOP 在 PARC 的诞生	62	致谢	88
5.3.1 RG	62	参考文献	88
5.3.2 AML	63	第 7 章 耦合面向方面和自适应编程	89
5.3.3 ETCML	63	7.1 概述	89
5.3.4 DJ	63	7.1.1 关注点含蓄的编程	89
5.3.5 Djava	65	7.1.2 结构含蓄的编程	89
5.3.6 AspectJ	65	7.2 含蓄性作为自适应行为的隐喻	90
5.4 建立团体	67	7.2.1 含蓄性怎样关联到 AP	90
5.4.1 研究者	67	7.2.2 含蓄性怎样关联到 AOP	91
5.4.2 实践者	67	7.2.3 Demeter 法则	91
5.5 回顾	68	7.2.4 针对关注点的 Demeter 法则	91
5.6 AOP 的本质	69	7.3 使用 DJ 的反射自适应编程	91
5.7 将来的挑战	69	7.4 使用 DAJ 的方面自适应编程	97
致谢	70	7.4.1 策略图交叉	97
参考文献	70	7.4.2 访问者类	97
第 6 章 AspectJ	74	7.5 相关工作	98
6.1 AspectJ 的目标	74	7.6 结论	98
6.1.1 通用目的	74	参考文献	99
6.1.2 基于编程语言的直接支持	74	第 8 章 使用 CAESAR 解开横切模型	102
6.1.3 易于学习	75	8.1 引言	102
6.1.4 增量式应用	75	8.1.1 对语言支持的需求	104
6.1.5 易于集成	75	8.1.2 本章的贡献和结构	106
6.1.6 适合真实的应用	75	8.2 联结点介入	106
6.2 AspectJ 语言	75	8.2.1 缺乏对多抽象方面的支持	108
		8.2.2 缺乏对复杂映射的支持	109
		8.2.3 缺乏对可重用方面联编的支持	109
		8.2.4 缺乏对方面多态的支持	109

8.3 CAESAR 模型	110	10.2 MIXIN 技术	137
8.3.1 方面合作接口	111	10.2.1 Mixin	137
8.3.2 方面实现	111	10.2.2 逐步细化	138
8.3.3 方面联编	112	10.2.3 Mixin 层	138
8.3.4 封装器再利用	112	10.3 FIDGET 的设计	139
8.3.5 最专用的封装器	113	10.3.1 体系结构	139
8.3.6 切入点和通知	114	10.3.2 组件	140
8.3.7 编织片和部署	115	10.3.3 同胞模式	141
8.3.8 虚类和静态类型	117	10.3.4 JL 的隐含的 this 类型参数	142
8.4 评估	118	10.4 FIDGET 的使用	143
8.4.1 问题再现	118	10.5 MIXIN 的编程支持	144
8.4.2 显式与隐含方面的实例 化/部署	119	10.5.1 构造器传播	144
8.5 实现	120	10.5.2 深度一致性	144
8.6 相关工作	120	10.6 将来的工作	144
8.6.1 开放类	120	10.6.1 语义检测	144
8.6.2 自适应性即插即用组件	121	10.6.2 类层次优化	145
8.6.3 基于合作的分解	121	10.6.3 Mixin 作为类型	145
8.6.4 动态部署的实现	122	10.6.4 实现参数化多态	146
8.7 总结和将来的工作	122	10.7 相关工作	147
参考文献	122	10.8 结论	147
第 9 章 基于跟踪的方面	125	致谢	147
9.1 引言	125	参考文献	147
9.2 基于跟踪的方面的特征	126	第 11 章 用首类名字空间分离关注点	150
9.2.1 可观察的执行跟踪	126	11.1 引言	150
9.2.2 方面语言	126	11.2 Piccola	151
9.2.3 编织	127	11.2.1 Piccola 层	151
9.3 表达性方面和等价推导	127	11.2.2 通用封装器	153
9.4 方面交互的检测和消解	128	11.3 例子: Mixin 层组合	154
9.4.1 方面交互	129	11.3.1 Piccola 中的 Mixin 层	155
9.4.2 对冲突消解的支持	129	11.3.2 使用 Mixin 层的软件演化	156
9.5 安全属性的静态编织	130	11.4 用形式结构描述组合风格	157
9.5.1 例子	130	11.5 结论	158
9.5.2 编织阶段	131	致谢	158
9.5.3 即时编织	132	参考文献	159
9.6 结论	133	第 12 章 使用反射支持 AOP	161
参考文献	134	12.1 什么是反射	161
第 10 章 使用 Mixin 技术提高模块化 特性	136	12.1.1 基础级和元级	161
10.1 介绍	136	12.1.2 元对象及其协议	162
		12.1.3 元链和元对象协作	162
		12.1.4 反射编程语言及其 MOP 的 例子	162

12.2 使用反射的 AOP	165	14.4.6 调试转换代码	189
12.2.1 从反射到 AOP	166	14.5 结论	190
12.2.2 使用多个方面的应用例子	166	参考文献	190
12.2.3 分离方面	166	第 15 章 JMangler——一个面向方面编程	
12.2.4 编织方面	168	的强大后端平台	192
12.3 讨论	170	15.1 引言	192
12.3.1 灵活性	170	15.1.1 方面编织的挑战	192
12.3.2 性能	170	15.1.2 加载时间分析和转换 (LAT)	
12.3.3 复杂性	171	的保证	193
12.3.4 工具	171	15.1.3 面向方面的 LAT 和 JMangler	193
12.3.5 重用性	171	15.2 通用拦截	194
12.4 结论	171	15.2.1 Java 的类加载体系结构	194
参考文献	172	15.2.2 依赖类加载器的拦截	195
第 13 章 通过控制通信插入 Ility	175	15.2.3 依赖 JVM 的拦截	195
13.1 通过控制通信实现 Ility	175	15.2.4 依赖于平台的拦截	195
13.2 对象基础框架	176	15.2.5 可移植的通用拦截	195
13.2.1 注入器	177	15.3 开放式体系结构	195
13.2.2 注解	177	15.4 对非本地转换的支持	196
13.2.3 Pragma	178	15.5 BCEL 编织器	197
13.3 已应用 Ility	180	15.5.1 转换和方面	198
13.3.1 可靠性	180	15.5.2 转换器	198
13.3.2 服务质量	180	15.5.3 转换器的使用	198
13.3.3 可管理性	180	15.5.4 非预期的转换器组合	198
13.3.4 安全	181	15.5.5 转换过程	199
13.4 相关工作	181	15.5.6 例子	200
13.5 结论	181	15.5.7 接口转换器的创建	200
参考文献	182	15.5.8 代码转换器的创建	201
第 14 章 使用字节码转换集成新特征	184	15.6 使用场景	201
14.1 方面和转换	184	15.6.1 使用 JMangler 的加载时编织	201
14.2 字节码转换概述	185	15.6.2 使用 JMangler 的运行时编织	202
14.2.1 静态转换	185	15.6.3 先进特征	202
14.2.2 动态转换	185	15.7 相关工作	203
14.3 编程转换	186	15.8 结论和将来的工作	204
14.3.1 选择	186	致谢	204
14.3.2 在 AOSD 中的使用	187	本章附录: 简单的代码覆盖	205
14.4 学到的经验	187	CodeCoverTransformer 的实现	207
14.4.1 接口设计	187	激活	208
14.4.2 安全、可靠和正确	188	参考文献	209
14.4.3 多个名字空间	189	第 16 章 使用 Java 方面组件的面向方面	
14.4.4 反射	189	软件开发	213
14.4.5 处理转换的功能	189		