

/THEORY/IN/PRACTICE

架构之美

Beautiful Architecture

顶级业界专家揭密软件设计之美

O'REILLY®

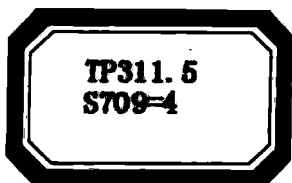


机械工业出版社
China Machine Press



Diomidis Spinellis & Georgios Gousios 编

王海鹏 蔡黄辉 徐锋 等译



-2

架构之美

Diomidis Spinellis & Georgios Gousios 编

王海鹏 蔡黄辉 徐锋 等译

TP311.5
S709-4

O'REILLY®

Beijing · Cambridge · Farnham · Köln · Sebastopol · Taipei · Tokyo

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社

图书在版编目 (CIP) 数据

架构之美 / (美) 斯宾耐立思 (Spinellis, D.), (美) 郭西奥斯 (Gousios, G.) 编; 王海鹏等译. —北京: 机械工业出版社, 2010.1

书名原文: Beautiful Architecture: Leading Thinkers Reveal the Hidden Beauty in Software Design

ISBN 978-7-111-28356-0

I. 架… II. ①斯… ②郭… ③王… III. 软件设计 IV. TP311.5

中国版本图书馆CIP数据核字 (2009) 第172092号

北京市版权局著作权合同登记

图字: 01-2009-1579

Copyright © 2009 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2010.
Authorized translation of the English edition, 2009 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2009。

简体中文版由机械工业出版社出版2010。英文原版的翻译得到O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

本书法律顾问 北京市展达律师事务所

书 名 / 架构之美

书 号 / ISBN 978-7-111-28356-0

责任编辑 / 周茂辉

封面设计 / Karen Montgomery, 张健

出版发行 / 机械工业出版社

地 址 / 北京市西城区百万庄大街22号 (邮编 100037)

印 刷 / 北京京师印务有限公司

开 本 / 178毫米×233毫米 16开本 24.5印张

版 次 / 2010年1月第1版第1次印刷

定 价 / 69.00元 (册)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzsj@hzbook.com

O'Reilly Media, Inc.介绍

为了满足读者对网络 and 软件技术知识的迫切需求,世界著名计算机图书出版机构O'Reilly Media, Inc. 授权机械工业出版社, 翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc. 是世界上在UNIX、X、Internet和其他开放系统图书领域具有领导地位的出版公司,同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》(被纽约公共图书馆评为20世纪最重要的50本书之一)到GNN(最早的Internet门户和商业网站),再到WebSite(第一个桌面PC的Web服务器软件),O'Reilly Media, Inc. 一直处于Internet发展的最前沿。

许多书店的反馈表明,O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比,O'Reilly Media, Inc. 具有深厚的计算机专业背景,这使得O'Reilly Media, Inc. 形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc. 所有的编辑人员以前都是程序员,或者是顶尖级的技术专家。O'Reilly Media, Inc. 还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家,而现在编写著作,O'Reilly Media, Inc. 依靠他们及时地推出图书。因为O'Reilly Media, Inc. 紧密地与计算机业界联系着,所以O'Reilly Media, Inc. 知道市场上真正需要什么图书。

如何看到一滴水的美丽

支付宝（中国）公司业务架构师
《大道至简》作者
周爱民 (aimingoo)

【一】

架构是一个过程，而非一个结果。

【二】

在大多数人的谈论中，架构是一个目标产物，而作为架构师的责任就是去生产它。所以无论如何，架构是可以“做”出来的，而且也应该有一些“做”的方法、技术、技巧。

有人问过我：架构的最主要产出是什么？我的答案是：图。这里面有两层含义：一层含义是如同建筑师描绘的蓝图一样，用于引导实施者；另一层含义是架构师头脑中清晰的目标系统。如果架构师头脑中没有系统清晰的图像，他是没有办法把它画出来的。

【三】

画家画的无非是物我。画物的画家，最终画的还是我见。所以，画家的笔最终描绘的是他自己心里的映像。

【四】

艺术是不可能被“生产”出来的，生产出来的，叫“艺术品”。

【五】

架构这个过程，是架构师洞见系统内在结构、规律、原则和逻辑的过程。真正的架构师是可以将自己放在系统中去的（例如作为系统中的任何一个角色），只有清晰地理解系统，才能简洁地描述它。而当架构师拿出了他所描述的“作品”的时候，架构这一过程就已经结束了。

【六】

一滴水滴落的过程中，有多少个形态的变化？

推荐序二

架构的架构

北京无限讯奇信息技术有限公司产品技术高级总监
黄冬

从编辑手里拿到厚厚的《架构之美》译稿时，恰巧是我刚刚讲完一场消息系统架构的讲座之后。而正是在昨天，一位想要创业的朋友跟我说要寻找一位懂得“架构”的高人与他一起创业。要知道与代码不同的是，“虚幻”的架构常常让人认为其有很多玄妙之处，只因它大多难以落在纸上。特别是与很多大师谈及架构时，经常落入他们的一些“陷阱”，并往往为自己达不到大师的精明与技巧而叹息。殊不知，被我们所津津乐道的这些架构，是他们在日常工作里经历了大量的错误、重复的尝试、无数的代码、长久的考验所积淀下来的只言片语。

本书将数十人的经历与只言片语，经过深思熟虑后抽象出规律，使之可以不断复用。而另一方面，又将架构的过程娓娓道来，尝试让读者思考架构的过程与思路。在这里，更多的过程与思考被展现出来，更多的原因与为什么让我们了解。

这本书里展现了很多绚丽的故事，犹如士兵阅读将军的传记一样，阅读本书将会让你更鼓起勇气追寻大师们的脚步，但永远要记住，每一滴汗水才真正是你成长路上的每一个记号，要在自己的工作里更深地去理解每一处不同，架构出属于自己的系统。

感谢译者和出版者为我们带来这样一本传奇的架构故事书。

推荐序三

美丽架构的含义

腾讯R&D研发总监 (Tencent Director Of R&D Development)
资深技术专家 (Senior Technology Expert)

王速瑜

古人形容美女之美：“……增之一分则太长，减之一分则太短……”，深刻地揭示了“恰到好处”的美丽含义。当我拿到《架构之美》书稿时，我发现美丽的含义如此相似。

美丽至简。美丽的架构应尽可能简单，但不要过于简单。书中通过多种例子表达了这个最基本的道理。我见过很多大型的软件架构，从大型的电信网络管理系统，到大规模应用的互联网架构，以及企业级的ERP软件，系统总是遵循从无到有，从简单到复杂，再到简单这样的过程。最终，支撑这些大型系统稳定可靠运行的就是这个最基本的道理。

美丽的架构应尽可能精益，并且是演进式发展的。当你架构一个亿万人同时在线的大规模网站系统的时候，你无法从一开始就提供最完善的解决方案，它应该是随着用户的增长而可扩展的。精益的思想让你避免了过度设计，也使架构不断演进，趋于完美。书中从企业级应用架构、用户级应用架构等多个角度提供了相应的解决方案，对于架构师无不是一顿美味的大餐。

深夜看完这本书稿后，我发现，架构之美并不简单，它没有定法。但是，它将为架构师们提供一把进入“美丽架构艺术馆”大门的钥匙。拿起它，您将会开启这扇大门！

推荐序四

美丽架构之道

《构建高性能Web站点》作者
Web架构实践者
郭欣

我无法给架构下一个简单的定义，因为任何定义都会束缚你对架构的无限想象。不可否认，架构师早已出现在人类几千年前的各项生产活动中，比如建筑、音乐。而在计算机软件及Web领域，架构的设计直接影响着系统的生产，比如开发过程和效率、代码和组件复用性等，同时也影响着系统的可用性、可伸缩性、性能、容量可预测性等。

本书更加关注架构之美。美丽的架构同样无法定义，可它却一定是自然的、简单的、可复用的、人文的，甚至是外行人也可以细细品味其思想的。当我看到超市的多个收银台排满长队时，便想到服务器并发处理性能和容量；当我看到十字路口的车辆等待转弯时，便想到它通过缓存思想来提高交通吞吐率。

那么如何设计出美丽的架构呢？从代码逻辑到物理网络，从单机到分布式，无数的技术可供架构师选择，如分层、组件化、服务化、标准化、缓存、分离、队列、复制、冗余、代理等，不过它们仍然只是“术”的范畴，而何时何处如何恰到好处地使用它们才是“道”的范畴，比如顿悟变化的道理，在博弈中寻找平衡，以系统化的角度来分析问题，寻找相对与绝对的奥秘、开放的心态……

然而，这个领域实在是太年轻了，我们需要更多的例子和经验，本书将让你大开眼界！

译者序

架构与美

王海鹏

人们在生活和工作中发现美并创造美，软件开发和架构设计也不例外。

架构之美体现了关注点的分离与结合。在软件设计中，设计师需要考虑多方面的关注点。漂亮的架构设计让这些关注点尽可能分离，然后以最简单的机制结合在一起，从而得到高内聚、低耦合的系统。例如在Darkstar项目中，架构师们考虑的重点就是如何将多人在线游戏的游戏逻辑与系统的可伸缩性分离开来，让游戏的开发者只要遵守少量的规则，就能够像编写单机游戏一样编写大规模多人在线游戏。又如REST架构风格，体现了对资源命名、请求处理和资源物理表现形式的关注点分离。资源的名称与请求资源时服务器的处理方式无关，请求者无需知道服务器端采取的技术，并且请求者本来就不关心服务器端的处理技术。资源的物理表示形式可以通过内容协商来决定，使系统可以支持多种物理表示形式，并可以方便地扩展。

架构之美注重表达的简洁性。“Don't Repeat Yourself”，好的架构致力于消除各种类型的信息重复。从结构化程序设计中的子程序和函数，到面向对象程序设计中的继承，无不体现了对表达简洁性的特殊偏爱。在敏捷方法学中，消除重复则是重构的主要目的之一。爱因斯坦说：“让它尽可能简单，但不要过于简单。”我们需要考虑所有必须考虑的关注点，然后用简洁漂亮的架构体现我们的关注。同时，简洁的架构之美也降低了软件的总体成本，从这个意义上说，“简洁性”又可以称为“经济性”。

架构之美需要解决实际问题，它既是艺术，也是生活。软件像建筑一样，它的美不能脱离它的实用价值。Bjarne Stroustrup说，人类文明运行于软件之上。每一个软件都有自己的架构，这些架构有的很美，有的不太美。从艺术的角度来说，美是创造矛盾并解决矛盾。架构的多关注点和表达简洁性就是一种矛盾，美丽的架构提供了这一矛盾的解决方法，让我们的内心产生一种愉快的感觉。

架构之美需要经过专业的学习才能更好地欣赏和创造。和所有的艺术之美一样，不是说不经专业学习就不能欣赏，但是经过了专业的学习，就能更好地欣赏这种美的种种精妙之处。如果想要创造出这种美，那就必然要经过长期的专业学习。

架构之美经过时间打磨。像Facebook面向数据的Web服务、FQL和FML架构，是在对应不同实际需求的过程中逐渐发展起来。在应用程序架构形成的过程中，设计者不断面对新的关注点需求，不断对已有的架构进行修改，并发展出新的架构组件。这就是所谓的“演进式架构”。只有变化是永恒不变的。在架构设计初期，设计者会将一些关注点有意推迟到将来考虑，例如持久和并发。对于这些暂不考虑的关注点，设计者对它们的实现方式尽可能不做任何假定，从而保留更多的可能性，让不同关注点之间的耦合尽可能小。

架构之美没有定法。虽然有一些法则可供我们参考，却没有非如此不可的。《金刚经》云：“一切贤圣，皆以无为法而有差别。”

参加本书翻译工作的人员还有蔡黄辉、徐锋、王海燕、李国安、周建鸣、范俊、张海洲、谢伟奇、林冀、钱立强、甘莉萍。

在这本书的翻译过程中，我受益良多，因此郑重地向大家推荐它。

作译者简介

作者简介

Till Adam在年轻时学习了哲学、比较文学、美国研究和音乐学，职业是音乐人。由于没有发财和出名，他转而攻读科学硕士，学习了数学、计算机科学和商业。多年从事自由软件的经历（特别是对KDE的贡献）教会了他编程，也为他带来了在Klarälvdalens Datakonsult AB工作的机会，目前他在该公司负责协调KDE的开发和其他与自由软件相关的活动。他和他的妻子、女儿住在德国柏林。

Jim Blandy在1990年至1993年间为自由软件基金会维护GNU Emacs，和Richard Stallman一起发布了Emacs的第19个版本。他是Subversion版本控制系统的最初设计者之一。他也是CVS版本控制系统、GNU调试器（GDB）、Guile扩展语言库和一个编辑基因序列的Emacs程序的贡献者。他现在为Mozilla公司工作，工作内容是SpiderMonkey，即Mozilla的Javascript编程语言的实现。Jim和他的妻子、两个女儿住在俄勒冈州的波特兰。

Mirko Boehm从1997开始就是KDE的开发者，在1996年至2006年间是KDE e.V.委员会的成员。他毕业于德国汉堡Helmut Schmidt大学的商业专业。在他的闲暇时间里，他阅读纸版书籍、与家人在一起，试图远离计算机。他目前在德国柏林为Klarälvdalens Datakonsult AB工作，负责跨平台软件和嵌入式软件开发。

Christopher Dennis自2005年JCP项目开始时就是项目的主要开发者。Chris在牛津大学读博士时开始使用Java。此前，他使用过各种编程语言，从十六进制小键盘上编写的Z80机器码到PHP和JavaScript。他对特殊情况、编码技巧和偶尔有点丑陋的临时编码很有兴趣，喜欢用各种语言编写紧凑的、优雅的代码。

Dave Fetterman是Facebook的工程经理，他在那里创建了Facebook平台项目。在2006年加入Facebook之前，他是一名软件工程师，参加Microsoft开发者部门的项目，包括.NET的通用语言运行环境（CLR）。他喜欢为其他开发者创建软件，也喜欢对愿意听的人发表长篇大论。他拥有应用数学的学士学位，并在2003年获得了哈佛大学的计算机科学硕士学位。

Keir Fraser是XenSource的创始人之一，XenSource现在是Citrix Systems公司的一部分。他也是Xen系统管理程序的首席架构师。Keir在2002实现了Xen的第一个版本，作为他

在剑桥计算机实验室攻读博士学位时的一项娱乐。在该项目成为大规模的社区合作的过程中，他继续作为主要的开发者。他因在无锁并发控制方面的工作于2004年获得了博士学位，并在同年成为一名教师。

Pete Goodliffe是一名程序员、专栏作家、演说家和作家，从来不在同一软件领域做过多的停留。Pete的热门书籍《Code Craft》(No Starch Press)是对整个编程追求的实际而有趣的调查——大约600页，真是了不起！他对制革很有热情，而且不穿鞋。

Georgios Gousios是一名职业研究者，接受的教育和软件工程有关，热衷于软件开发。目前，他正在希腊的雅典经济与商业大学完成他的博士论文。他的研究兴趣包括软件工程、软件质量、虚拟机和操作系统，他拥有英国曼彻斯特大学的科学硕士学位。Gousios为多个开源软件项目贡献过代码，并参与了各种学术项目和商业项目的研究与开发。他是SQO-OSS项目的项目经理、设计权威和主要开发成员，为评估软件质量探索一些创新的方法。在他的学术生涯中，Gousios在会议和杂志上发表了10篇技术论文。Gousios是ACM、IEEE、Usenix Association和Technical Chamber of Greece的成员。

Dave Grove是IBM的T.J. Watson研究中心动态优化组的一名研究员。他的主要研究兴趣包括分析和优化面向对象语言、虚拟机设计和实现、JIT编译、在线反馈导向的优化和垃圾收集。他在1998年参加了Jalapeño项目，是这个优化编译器和适应式优化系统首个实现的主要贡献者。自Jalapeño在2001年作为Jikes RVM开放源码以来，他一直是Jikes RVM核心团队和指导委员会的活跃成员。

John Klein是软件工程研究所（SEI）的高级技术人员，他的研究方向是“众系统之系统”的架构方法，并帮助个人、团队和组织机构改进他们的软件架构能力。在加入SEI之前，John是Avaya公司的首席架构师。在Avaya，他负责开发多模式的代理、通信分析的架构，以及为各种客户交互产品创建并改进架构。在此之前，John是Quintus的一名软件架构师，在那里他设计了第一款获得商业成功的多渠道集成联系中心产品，并导致了Quintus兼并了另外两家公司，实现了产品组合的技术集成。在加入Quintus之前，John曾为多家视频会议和视频网络业的公司服务。他的职业生涯开始于Raytheon，在那里他为雷达信号处理、多光谱图像处理、并行处理架构和算法提供硬件和软件解决方案。John拥有Stevens技术学院的学士学位和Northeastern大学的硕士学位。他是ACM和IEEE计算机学会的成员。

Greg Lehey的漫长职业生涯在德国和澳大利亚度过，他曾为德国空间研究所工作，也曾为Univac、Tandem、Siemens-Nixdorf和IBM等计算机制造商工作，也曾作为一些没名气的软件公司的大客户，还曾做过独立的咨询顾问。他的活动范围很广，包括从内核开发到产品管理，从系统编程到系统管理，从处理卫星数据到为油泵编程，从生产

CD-ROM到把自由软件移植到DSP指令集上。他是FreeBSD核心团队的成员，也是澳大利亚UNIX用户协会的主席。他是FreeBSD和NetBSD项目的开发者，也是《Porting UNIX Software》和《The Complete FreeBSD, Fourth Edition》(O'Reilly)的作者。他还以编写商业应用软件而闻名。Greg在2007年退休，将多出来的时间用于品味生活。现在，休闲活动占据了他的大多数时间，但这还不够，他还听古典木纹唱片、烹饪、酿啤酒（他开发了一个计算机控制的发酵系统）、做园艺、骑马和摄影。他也对一些历史题材感兴趣，包括古代的难解的欧洲语言。他的主页是：<http://www.lemis.com/grop/>。

Panagiotis Louridas在20世纪80年代通过一台Sinclair ZX Spectrum开始涉足计算机。从那时起，他就开始用机器语言进行编程，而且非常喜欢编程。他在雅典大学信息系获得了计算机科学学士学位，在曼彻斯特大学获得了计算机硕士和博士学位。这些年来，他一直为私人机构开发软件，现在，他在希腊研究和教育网络（GRNET）工作。他也是雅典经济与商业大学（AUEB）软件工程和安全（SENSE）研究组的成员。他发表的文章范围很广，从人类学到加密，从仪表展示到软件工程。他特别喜欢寻找计算机世界和其他领域的联系。

Stephen J. Mellor在为软件开发创建有效的工程方法方面，是国际公认的先行者。在1985年，他出版了广为阅读的Ward-Mellor三卷本《Structured Development for Real-Time Systems》(Prentice Hall)；在1998年，他的书首次定义了面向对象分析。Stephen还在2002年出版了《Executable UML: A Foundation for Model-Driven Architecture》(Addison-Wesley Professional)。他最近的一本书《MDA Distilled: Principles of Model-Driven Architecture》(Addison-Wesley Professional)在2004年出版。他在对象管理组织（OMG）中活动积极，是为UML添加可执行动作的协会的主席，他最近完成了可执行UML的标准。他是敏捷宣言的签名者之一。他是OMG架构委员会的两任成员，IEEE软件顾问委员会的主席，最近，他成为Mentor Graphics的嵌入式软件部门的首席科学家。

Bertrand Meyer是ETH Zurich的软件工程教授，也是Eiffel软件的首席架构师，他领导并设计了EiffelStudio环境和大量的库。他是一些畅销书的作者，其中包括获得Jolt大奖的《Object-Oriented Software Construction》(Prentice Hall)。他也因为在对象技术和Eiffel方面的工作获得了ACM软件系统大奖和Dahl-Nygaard大奖，并获得了St. Petersburg州立技术大学的荣誉博士学位。他的研究对象涉及面向对象技术、编程语言、软件验证（包括测试、并发和规范方法）。他也是一名活跃的顾问和讲师。

William J. Mitchell是MIT架构和媒体艺术与科学系的Alexander Dreyfoos教授，他领导着MIT媒体实验室和MIT设计实验室的Smart Cities团队。他以前曾担任MIT架构和计划学院的院长。他最近的新书是《World's Greatest Architect》和《Imagining MIT》（都由

MIT出版社出版)。

Derek Murray是剑桥大学计算机实验室的博士生。他在2006年加入Xen项目，主要工作是通过重新设计控制栈来改进Xen的安全性。他现在的研究主要是改进大规模分布式系统的容错性，但他还是偶尔会涉及系统核心。Derek在2006年从爱丁堡大学获得了高性能计算专业的硕士学位，2005年获得了Glasgow大学的计算机学士学位。

Rhys Newman在十多年前于牛津大学完成博士学位时，就开始使用Java，那时Java也只有几年历史。在他早期的研究中，他利用纯Java环境展示了高性能实时场景处理的实现方法，即使当时还是使用早期JIT化的JVM。从那时起，他同时在学界和业界工作，一次次证明Java平台实际有多灵活、多高效、多快。在超过20年的软件工程生涯中，他获得了多个业界杰出技术奖项，最近他回到了牛津，承担了网格计算领域的突破性研究工作。JPC是最新研究工作的一部分。

Michael Nygard致力于在全国帮助开发者提高水平和减少痛苦。他和他遇到的每一个人分享他对改进的热情和活力，有时甚至没有得到对方的同意。Michael花了20年中的大部分时间学习对专业程序员有意义的事，他关心艺术、质量和技艺。他总是愿意在那些全职的、真心投入工作的开发者（那些“觉醒的”开发者）身上花时间。在另一方面，他不能容忍缺乏兴趣或浪费潜力。Michael在近20年来一直是专业的程序员和架构师。在这段时间里，他为美国政府、军方、银行、金融业、农业和零售业交付了运营系统。通常，Michael都要面对他自己开发的系统。这种实际运营的经历改变了他对软件架构和开发的看法。他参与了一个Tier 1零售网站的初期开发，并且常常作为其他在线业务的“流动解决问题专家”。这些经验让他对在相当不友好的环境下构建高性能、高可靠性的软件有了独特的看法。最近，Michael编写了《Release It! Design and Deploy Production-Ready Software》(Pragmatic Programmers)，该书获得了2008年的Jolt生产力大奖。他的其他文字可以在<http://www.michaelnygard.com/blog>上读到。

Ian Rogers是曼彻斯特大学高级处理器技术研究组的研究员。他的博士研究工作是关于Dynamite二进制翻译器的，该技术实现了商用，现在是许多二进制翻译器产品的一部分，包括Apple的Rosetta。他最近的学术研究工作一直是编程语言设计、运行时环境和虚拟机环境，特别是如何自动创建它们并有效地使用并行技术。他是Jikes研究虚拟机的主要贡献者，是开发团队的核心成员。

Brian Sletten是自由的、受过艺术教育的软件工程师，关注forward-learning技术。他曾担任过系统架构师、开发者、现场指导者和培训师。他在世界各地的会议上发表演讲，

编辑注： 此书由机械工业出版社引进出版，书名为《代码质量》（注释版），书号为：978-7-111-22671-0。

并为一些在线出版物编写关于面向Web技术的文章。他的经验涉及国防、金融和商业领域。他曾设计并建造了网络矩阵式交换控制系统、在线游戏、3D仿真/可视化环境、因特网分布式计算平台、P2P和基于Web的语义系统。他拥有William and Mary大学的计算机科学学士学位，目前居住在弗吉尼亚的Fairfax。他是Bosatsu咨询公司的总裁，该公司为Web架构、面向资源的计算、语义Web、高级用户界面、可伸缩系统、安全和其他20世纪末21世纪初的技术提供专业的咨询服务。

Diomidis Spinellis是希腊雅典经济与商业大学管理科学与技术系的副教授。他的研究领域包括软件工程、计算机安全和编程语言。他也编写了两本“开放源码方面”的书，由Addison-Wesley出版：《Code Reading》（获得了2004年的软件开发生产力大奖）和《Code Quality》（获得了2007年软件开发生产力大奖，编辑注）。他也写了几十篇科学论文。他是IEEE Software编辑委员会的成员，负责定期的“Tools of the Trade”栏目。Diomidis是FreeBSD的提交者，也是UMLGraph和其他开源软件包、库和工具的开发者的。他拥有软件工程的硕士学位和计算机科学博士学位，都是在Imperial College London获得的。Diomidis是ACM的高级成员，也是IEEE和Usenix Association的成员。

Jim Waldo是Sun微系统实验室的杰出工程师，负责研究下一代大规模分布式系统。他目前是Project Darkstar的技术负责人，该系统是针对大规模多人在线游戏和虚拟世界而设计的多线程、分布式基础设施。在此之前，他曾是Jini的首席架构师，Jini是基于Java的分布式编程系统。Jim编写了《The Evolution of C++: Language Design in the Marketplace of Ideas》（MIT出版社），也是《The Jini Specification》（Addison-Wesley）的合著者之一。他曾是美国国家学术委员会的共同主席，编辑并出版了《Engaging Privacy and Information Technology in a Digital Age》一书。Jim也是哈佛大学的辅助教师，在计算机科学系教授分布式计算和策略与技术相关的内容。Jim拥有马萨诸塞大学（Amherst）的哲学博士学位。

David Weiss拥有Union College的计算机科学学士学位，并拥有马里兰大学的计算机科学硕士和博士学位。他目前是Avaya实验室的软件技术研究部的领导，他关注软件开发效率改进的普遍问题和Avaya软件开发过程改进的特殊问题。在第二个问题上，他领导了Avaya软件技术研究中心。以前，他曾是朗讯技术贝尔实验室软件生产研究部的主任，该部门负责研究如何改进软件开发的效率。在加入贝尔实验室之前，他是软件生产力协会（SPC）复用和度量部门的主任，该协会由14个大型的美国航空公司组成。在加入SPC之前，Weiss博士在技术评估办公室度过了一年的时间，在那里他与同事共同完成了Strategic Defense Initiative的技术评估。在1985—1986学年，他是Wang Institute的访问学者，在许多年里，他一直是华盛顿特区Naval研究实验室（NRL）计算机科学和系统部门的研究员。他也是一名程序员和数学家。Dave的主要研究方向是软件工程领域，特

别是软件开发过程和方法学、软件设计和软件测量。他最为人知的是发明了软件测量的“目标-问题-测量指标”方法，软件系统模块化结构的工作，以及软件生产线工程的工作。他是Synthesis过程及其后继FAST过程的共同发明人。他与别人共同编著了两本书：《Software Product-Line Engineering》和《Software Fundamentals: Collected Papers of David L. Parnas》（都由Addison-Wesley出版）。

译者简介

王海鹏 1994年毕业于华东师范大学。拥有理学士（物理）和文学士（英国语言文学）学位。独立的咨询顾问、培训讲师、译者和软件开发者。已翻译十余本软件开发书籍，主题涵盖敏捷方法学、需求工程、UML建模和测试。拥有15年软件开发经验，目前主要的研究领域是软件架构和方法学，致力于提高软件开发的品质和效率。

蔡黄辉 江苏启东人。1999年毕业于上海交通大学，毕业后一直从事软件开发工作，主要使用Java做Web方面的底层开发。现居住在上海。联系方式：caihuanghui@hotmail.com。

徐锋 中国系统分析员顾问团（CSAI）软件工程首席顾问，中国软件技术大会杰出贡献专家，资深咨询顾问。主要研究领域为需求工程、系统分析与设计、软件估算，致力于推动软件工程方法论的落地应用。曾在《程序员》等媒体发表了《实战OO》、《项目管理三步曲》、《大话Design》等多个专栏文章，著有《软件需求最佳实践》、《UML面向对象建模基础》等多本书籍，翻译了《UML 2.0实战》、《AOSD中文版》、《Cloud to Code中文版》等多本相关技术书籍。



机械工业出版社
CHINA MACHINE PRESS

华章分社

为每一个团队提供最有价值的阅读服务

每一本书都值得您和您的团队一起阅读

分享阅读 分享成功

