



软件架构建模和仿真

Palladio方法

拉尔夫·H. 雷乌斯纳 (Ralf H. Reussner) 斯特芬·贝克尔 (Steffen Becker)

詹斯·哈佩 (Jens Happe) 罗伯特·海因里希 (Robert Heinrich)

[德]

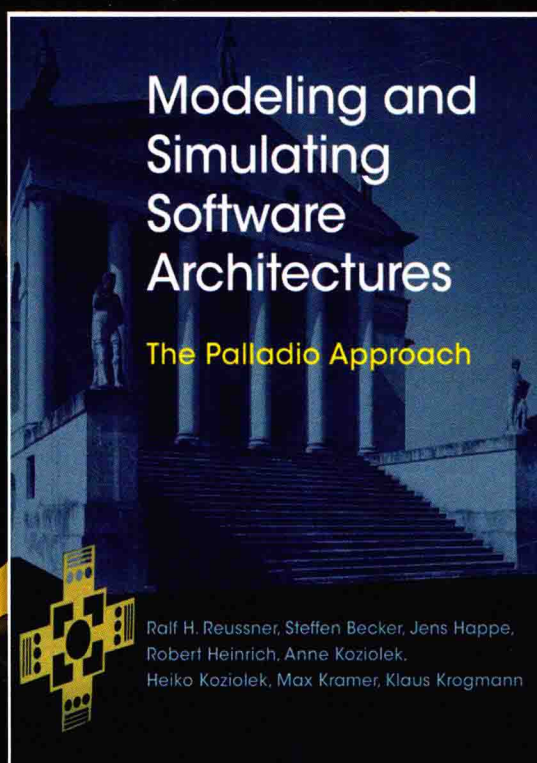
安妮·科齐奥勒克 (Anne Koziolk) 海科·科齐奥勒克 (Heiko Koziolk)

马克斯·克莱默 (Max Kramer) 克劳斯·克罗曼 (Klaus Krogmann)

李必信 王璐璐 孔祥龙 周颖 廖力 等译

编著

Modeling and Simulating Software Architectures
The Palladio Approach



计 算 机 科 学 丛

软件架构建模和仿真 Palladio方法

拉尔夫·H. 雷乌斯纳 (Ralf H. Reussner) 斯特芬·贝克尔 (Steffen Becker)

詹斯·哈佩 (Jens Happe) 罗伯特·海因里希 (Robert Heinrich)

[德] 安妮·科齐奥勒克 (Anne Koziolk) 海科·科齐奥勒克 (Heiko Koziolk) 编著

马克斯·克莱默 (Max Kramer) 克劳斯·克罗曼 (Klaus Krogmann)

李必信 王璐璐 孔祥龙 周颖 廖力 等译

Modeling and Simulating Software Architectures

The Palladio Approach

Modeling and
Simulating
Software
Architectures

The Palladio Approach



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

软件架构建模和仿真: Palladio 方法 / (德) 拉尔夫·H. 雷乌斯纳 (Ralf H. Reussner) 等编著; 李必信等译. —北京: 机械工业出版社, 2018.9
(计算机科学丛书)

书名原文: Modeling and Simulating Software Architectures: The Palladio Approach

ISBN 978-7-111-60722-9

I. 软… II. ①拉… ②李… III. 软件设计 IV. TP311.1

中国版本图书馆 CIP 数据核字 (2018) 第 192034 号

本书版权登记号: 图字 01-2017-0497

Ralf H. Reussner, Steffen Becker, Jens Happe, Robert Heinrich, Anne Kozirolek, Heiko Kozirolek, Max Kramer and Klaus Krogmann: Modeling and Simulating Software Architectures: The Palladio Approach (ISBN 978-0-262-03476-0).

Original English language edition copyright © 2016 by Massachusetts Institute of Technology.

Simplified Chinese Translation Copyright © 2018 by China Machine Press.

Simplified Chinese translation rights arranged with MIT Press through Bardon-Chinese Media Agency.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or any information storage and retrieval system, without permission, in writing, from the publisher.

All rights reserved.

本书中文简体字版由 MIT Press 通过 Bardon-Chinese Media Agency 授权机械工业出版社在中华人民共和国境内 (不包括香港、澳门特别行政区及台湾地区) 独家出版发行。未经出版者书面许可, 不得以任何方式抄袭、复制或节录本书中的任何部分。

本书介绍了一种新的、定量的架构仿真方法来讨论软件设计和架构的质量和属性, 让软件工程师在早期设计阶段就能够建模和评估软件服务的质量。书中提出了第一个针对软件架构的仿真器——Palladio, 向学生和专业人员展示了为了分析软件服务的性能如何建模可重用、参数化的组件以及可配置、可部署的系统。

本书适合作为高等院校软件工程相关专业的教科书, 以及软件架构师和其他工程技术人员的参考书籍。

出版发行: 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: 唐晓琳

责任校对: 李秋荣

印刷: 三河市宏图印务有限公司

版次: 2018 年 9 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 16.75

书号: ISBN 978-7-111-60722-9

定价: 79.00 元

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

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

文艺复兴以来,源远流长的科学精神和逐步形成的学术规范,使西方国家在自然科学的各个领域取得了垄断性的优势;也正是这样的优势,使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中,美国的产业界与教育界越来越紧密地结合,计算机学科中的许多泰山北斗同时身处科研和教学的最前线,由此而产生的经典科学著作,不仅擘划了研究的范畴,还揭示了学术的源变,既遵循学术规范,又自有学者个性,其价值并不会因年月的流逝而减退。

近年,在全球信息化大潮的推动下,我国的计算机产业发展迅猛,对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇,也是挑战;而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下,美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此,引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用,也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始,我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力,我们与Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage等世界著名出版公司建立了良好的合作关系,从他们现有的数百种教材中甄选出Andrew S. Tanenbaum, Bjarne Stroustrup, Brian W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson等大师名家的一批经典作品,以“计算机科学丛书”为总称出版,供读者学习、研究及珍藏。大理石纹理的封面,也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助,国内的专家不仅提供了中肯的选题指导,还不辞劳苦地担任了翻译和审校的工作;而原书的作者也相当关注其作品在中国的传播,有的还专门为其书的中译本作序。迄今,“计算机科学丛书”已经出版了近两百个品种,这些书籍在读者中树立了良好的口碑,并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化,教育界对国外计算机教材的需求和应用都将步入一个新的阶段,我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

华章网站: www.hzbook.com

电子邮件: hzjsj@hzbook.com

联系电话: (010) 88379604

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037



华章科技图书出版中心

译者序

Modeling and Simulating Software Architectures: The Palladio Approach

众所周知，在物理世界中，房屋的架构、桥梁的架构、汽车的架构等都是决定它们质量好坏和生命周期长短的重要因素之一；同样，在软件世界中，软件架构也是决定软件质量好坏和软件生命周期长短的重要因素之一。好的软件架构会使得软件自身的性能、可靠性、安全性、可维护性等质量指标很好地满足用户的需求，不好的架构会导致所有这些指标或部分指标达不到用户要求。可见，“软件架构的好与不好”这个问题非常重要！那么，如何判断或者评价软件架构的好与不好呢？在软件工程领域，这是个很难解决的问题，至少到目前为止，无论学术界还是工业界都缺少非常有效的解决方法。早期的方法主要包含两类：一类是直接对架构文档或者模型进行度量和评价，代表性的方法有 ATAM、SAAM 等定性评估方法；另一类是通过直接运行软件来获取监控信息（性能、可靠性等方面的数据），再反过来评价软件架构，代表性的方法有性能监控和测试、可靠性测试和评估等。但是，这些方法由于没有很好的机制把软件架构和相关的质量指标关联起来，都具有一定的片面性和局限性。

本书是一本关于如何评价软件架构的书，全面且系统地介绍了在 Palladio 方法（Palladio 方法能够很好地把软件架构和相关的质量指标关联起来）中如何进行软件架构建模，如何进行软件架构仿真，以及如何进行软件架构质量预测等。本书共有 16 章，在介绍 Palladio 方法基本原理和实施步骤的同时，还结合三个大型的应用案例详细演示了如何在实际项目中运用 Palladio 方法。这是一本难得的既有理论阐述，又有工具开发介绍，还有实践案例详细解析的好书。

本书面向的读者对象如下：（1）作为高等学校软件工程相关专业的教科书，面向高年级本科生和研究生；（2）作为软件架构研究人员的参考书，本书讨论的软件架构仿真和实验研究方法，为相关研究人员提供了很好的思想启迪和行动指南；（3）作为软件架构师及其他工程技术人员工具书，本书提供的软件架构仿真原理为软件全生命周期早期阶段的质量保证提供了很好的、可行的方案。

参加本书翻译工作的人员主要来自东南大学软件工程研究所、东南大学计算机科学与工程学院的教师和部分高年级研究生，除封面署名外，还包括王桐、刘辉辉、宋启威、韩伟娜、李慧丹、谢仁松、杨安奇、杜鹏程、尹强、宋震天、葛丹薇、熊壬浩、汪小飞、苏晓威、段鹏飞、王家慧、汤立辉等。同时感谢东南大学“软件开发方法与技术”课程的 30 多位 2017 级研究生，他们在课堂上详细报告了本书内容。在翻译过程中，还得到了来自华为公司吴文胜先生的指导和帮助。在此，对他们的辛勤劳动表示衷心的感谢。也特别感谢机械工业出版社编辑姚蕾老师的无私帮助。

限于水平，我们对本书内容的理解和中文语言的表达难免存在不当之处，在此敬请读者批评指正。但无论如何，本书是一本非常优秀的软件架构读物，本人也十分荣幸地向读者推荐，认真阅读本书一定使你受益匪浅。

李必信

2018 年于南京九龙湖

利用很多细粒度的元素构建软件是一项艰巨任务，因为在软件开发中完成这项任务的基础是“一般的编程技术”，即基于运行环境和编程语言中可用的基本元素进行程序合成。正是这种一般编程技术的高自由度促成了当今软件世界的所有生态。同时，各种库、框架和事先开发的软件包的出现大大便利了编程工作。尽管一般的编程技术看似无所不能，但是由它所构建软件的功能正确性却难以预测，同时，像安全性这类定性的质量需求以及像性能这类定量的质量需求也难以满足。

利用粗粒度的、遵从良好规范的单元来构建软件在很久以前就被认为是一种更有效率的方式，这些粗粒度单元自身已经带有强大的功能和属性，可以提供更多的信息帮助预测软件的属性。粗粒度单元的种类很多，其中包括在已构建系统中的足够独立、有固有结构的单元。这些单元称为软件组件。

如果计划用给定的和专门构建的组件来构建软件，会让开发人员在实际构建系统之前对其功能和其他属性有更多的想象，但是这不代表对系统各种属性的预测已经完成。要真正地对系统的属性进行准确的预测，还需要系统的整体架构和各个组件的属性信息，以及能够使用这些信息做出预测的方法和工具。

这就引出了本书的主题方法——Palladio。Palladio 是一个系统仿真器，可以在给定软件架构的情况下运行，能够预测系统的各种独立属性，例如性能和可靠性、可维护性和成本。此外，Palladio 仿真器本身是以组件的方式构建的，因此具有良好的可扩展性和可配置性。

Palladio 项目是我们过去十多年里从事科学和工程研究的重头戏，书中也吸收和总结了研究过程中一些深刻的经验教训。本书作者致力于从一个更广泛的视角说明 Palladio 到底是什么——一个工具、一个方法，甚至是一种实践。Palladio 使得实践者不必通过构建大量原型来进行系统度量，而是建议使用一种清洁的工程化方法对待建系统进行模型驱动分析，从而预测系统的属性。这使得性能瓶颈或难以忍受的成本等问题都能够在早期被发现，进而可以通过调整架构、选择不同的组件或在理解需求的情况下开发特定的组件等方法解决这些问题。

Clements Szyperski

美国华盛顿州雷德蒙德

2016 年 2 月

又一本谈软件架构的书？

关于软件架构方面的书已经很多，其中不乏好的书籍。这并不奇怪，因为几乎在二十多年前软件架构就已经在研究和实践上成为一个雷打不动的关注领域。

那么本书有什么独特之处呢？大多数书都是在“文档化软件架构”，本书是在“仿真软件架构”。文档化软件架构并不是错误的，事实上，一些软件工程活动能从软件架构文档中获益，例如成本规划、组件定义、制定工作计划以及规划重用等。然而，文档化软件架构有了一个最新的用途——软件架构的仿真。软件架构对于软件工程师来说是非常重要的，因为可以在软件架构设计的基础上预测制品的属性，而不用去实现系统，这也是工程学科的核心特征之一。从这个角度看，“软件工程”几乎还不能算是一门工程学科，因为我们通常在软件设计时还不了解设计决策对系统质量属性（如性能或可靠性）的影响。一般的做法是，我们要想了解最终软件的质量，就要尝试通过昂贵且有风险的“试错”周期来测试质量。这样做的最大问题是，等到测试时我们实际上已经为系统的实现付出了很多精力和工作量，而且后期发现问题再去修改架构以提高系统性能的过程会有很大的额外开销。本书描述了如何对软件架构进行建模和仿真，以便在实现和文档化系统之前就自动获得软件系统的定量质量预测。

读者对象

本书是关于软件架构的质量建模和仿真的，主要针对软件工程领域的研究生以及他们的导师或者已经在软件工程领域有学术研究背景的从业者。在大学中，学生和老师可以把这本书作为质量工程、性能工程等研究生课程的教材，或是作为软件架构方面的更多普通课程特定部分的辅助材料。

阅读本书，有助于计算机科学方向的学生或从业者成为定量软件工程师，掌控软件架构的性能和可靠性，具体包括：

- 了解如何定量地构建更好的软件架构。
- 仿真和预测软件架构的影响。
- 避免并修复性能和可靠性问题。
- 了解软件架构变化对质量的影响。
- 规避在软件构建和设计过程中的典型风险。

内容概要

本书主要介绍对软件架构的仿真，以明确架构对于软件质量的影响。书中详细介绍了如何对软件系统的各个质量属性（例如，性能和可靠性）在最终软件敲定之前进行建模、分析和改善。本书展示了软件架构师如何利用架构仿真更好地工作，并提供可靠且经过验证的质量建模和预测技术。它不是一本阐述软件架构文档的书，而是描述了世界上第一个软件架构

仿真器：Palladio。通过仿真，你就可以知道怎样避免高昂的花费和在“试错”周期中徘徊，更好地用工程化方法进行软件设计。通过本书，你会一步一步地学习和体会到如何对可重用、参数化的组件以及可配置、可部署的系统进行建模，以便在早期设计阶段能够分析系统的服务质量。

本书首先详细介绍了用于软件架构的特定领域建模语言的一些主要概念，并提供了相应的开发和分析过程。然后描述了如何使用质量信息来校准架构模型，从中可以自动地导出详细的仿真模型用于质量预测。然后，结合三个工程案例，沿着那些建立好的、具有革新意义的过程（例如，双向工程和设计空间探索）解释获得的预测结果是如何使用的以及有什么样的益处。

通过阅读本书你可以学到什么？简单来说，你将习得如何系统地回答下列问题：

- 我的系统可扩展吗？
- 实现某些性能目标需要什么硬件资源？
- 如何设计软件才能实现其高效性和高可靠性？
- 我需要与我的服务提供商达成什么样的服务级协议？

在本书中可以找到上述问题的答案，因为你可以在早期设计阶段通过建模和分析了解可重用的、参数化的组件的服务质量和可配置的、可部署的系统的服务质量。

组织结构

可运行案例

本书在第2章有一个可运行案例，该案例建模和分析了一个多媒体存储系统，多媒体文件可以在该系统上进行上传和下载。在部分章节中会引用该系统来辅助解释某些任务和方法。另外，在第五部分中全面介绍和讨论了三个工业案例。

相关问题、练习、拓展阅读和小贴士

在第二、三、四部分中每章章末都有专门的几节来帮助评估你理解的准确性，帮助你从外部资源获取更多信息，并且检查你是否掌握了所有关键点。首先，在每章最后都设置了一系列问题来帮助你检查自己是否对书中的内容有正确的理解。然后，你可以使用练习获得进一步的实践观察。这些问题和练习都根据所需的认知工作量（cognitive effort）进行分类：是对知识的复制（*），还是重组（**）、迁移（***）或创造（****）。问题的答案和练习的解决方案在本书的相关网站（后面会说到）上可以找到，而且配有教程和截屏。接着是“拓展阅读”节，会给出一些额外的参考信息，它简要列出了一些推荐文献，这些文献中会更为详细地介绍基本概念，或者深化书中的内容。最后是“小贴士”节，第二、三、四部分的各个章节都包含此节，其简要总结了该章内容的关键点。

定义、题外话和索引

对于有争议且需要加以说明的重要术语，书中会有简短的定义。对于不需要更深入理解的材料，就不在正文中给出更详细的信息，而是把这类额外的信息放在“题外话”框中。另外为了让你能快速导航到书中的某个关键字，重要的术语将被编入索引。

相关网站

由于本书中提到的许多主题已经在大学课程和几个工业合作伙伴的研讨会上进行了教

学，因此我们在 MIT 出版社的官方网站上以教程、截屏录像和演讲材料的形式提供丰富的在线资料：<https://mitpress.mit.edu/books/modeling-and-simulating-software-architectures>。

Palladio 网站

你可以从下面的网站了解关于 Palladio 作为一种方法、工具和架构语言的主要信息：www.palladio-simulator.com。

在这里，你可以下载“PCM Workbench”工具，还可以下载教程、截屏录像以及各种出版物和案例研究文档。

致谢

作者想对许多帮助写这本书的人表示感谢。首先，感谢 Marie、Kathleen、Katherine 和 MIT 出版社的整个团队在写作和编辑过程中的指导。另外，感谢我们的家人、合作伙伴和孩子们，因为在过去三年中，他们经常看到我们在写书，而不是与他们在一起。除此之外，我们也要感谢所有创造 Palladio 的人。在支持我们工作的资助机构中，我们想着重感谢德国研究基金会（German Research Foundation, DFG），当整个项目的成功与否不是很清楚时，他们便投资初步的 Palladio 项目，并派来了年轻的调查人员艾美·恩特（Emmy Noether）。所有写这本书的人都以各种形式参与了 Palladio 项目；然而，除了本书的作者之外，还有很多参与者。许多前雇员以及担任研究助理或者撰写硕士或者学士论文的学生都帮助构造了 Palladio，使之成为世界上第一个能运行的软件架构仿真器，并希望能够使软件设计成为一门工程学科。非常感谢大家！

关于作者

本书的作者是一群学术界和工业界的科学家和工程师，他们致力于开发、扩展和成功应用软件架构的质量仿真和预测技术。

- Ralf H. Reussner——博士、教授，卡尔斯鲁厄理工学院（KIT）软件设计与质量研究院的软件工程全职教授，也是信息技术研究中心（FZI）执行董事。
- Steffen Becker——博士、教授，开姆尼茨工业大学（TUC）的软件工程全职教授。
- Jens Happe——博士，SAP 公司的高级研究员，也是初创公司 avenyou 的联合创始人。
- Robert Heinrich——博士，卡尔斯鲁厄理工学院软件设计与质量研究院持续质量工程研究组主管。
- Anne Kozirolek——博士，卡尔斯鲁厄理工学院软件工程初级教授，架构驱动需求工程研究组主管。
- Heiko Kozirolek——博士，ABB 公司研究部首席科学家和全球化研究领域协调员。
- Max E. Kramer——卡尔斯鲁厄理工学院软件设计与质量研究院的科学家。
- Klaus Krogmann——博士，Citrix、GetGo 质量工程平台经理。

其他贡献者

Fabian Brosig——乌兹堡大学

Erik Burger——卡尔斯鲁厄理工学院

Axel Busch——卡尔斯鲁厄理工学院

Zoya Durdik——ABB 公司研究部
 Lucia Happe——卡尔斯鲁厄理工学院
 Christoph Heger——Novatec 咨询公司
 Jorg Heness——FZI 信息技术研究中心
 Nikolaus Huber——乌兹堡大学
 Oliver Hummel——iQser 股份有限公司
 Benjamin Klatt——inovex 公司
 Martin Küster——Senacor 技术公司
 Michael Langhammer——卡尔斯鲁厄理工学院
 Sebastian Lehrig——开姆尼茨工业大学 (TUC)
 Philipp Merkle——卡尔斯鲁厄理工学院
 Florian Meyerer——iQser 股份有限公司
 Qais Noorshams——IBM 德国研究开发公司
 Kiana Rostami——卡尔斯鲁厄理工学院
 Simon Spinner——乌兹堡大学
 Christian Stier——FZI 信息技术研究中心
 Misha Strittmatter——卡尔斯鲁厄理工学院
 Alexander Wert——Novatec 咨询公司

目 录

Modeling and Simulating Software Architectures: The Palladio Approach

出版者的话

译者序

推荐序

前言

第一部分 软件架构的工程方法

第 1 章 引言 2

- 1.1 软件设计的工程方法 4
- 1.2 软件架构的角色 5
- 1.3 软件组件的角色 5
- 1.4 Palladio 方法 6
 - 1.4.1 Palladio 方法可以预测哪些质量属性和度量值 6
 - 1.4.2 Palladio 方法由哪些部分组成 7
 - 1.4.3 和遗产系统的关系 8
- 1.5 本书结构和阅读路径 8

第 2 章 Palladio 方法简介 10

- 2.1 Media Store 10
 - 2.1.1 Media Store 需求 10
 - 2.1.2 Media Store 用户交互 11
 - 2.1.3 初始设计 12
 - 2.1.4 初步设计决策 13
- 2.2 Palladio 的角色 14
- 2.3 仿真结果 15
- 2.4 容量规划 16
- 2.5 Media Store: 可选的设计 18
 - 2.5.1 将 ReEncoder 组件分配到另一个服务器 19
 - 2.5.2 增加缓存组件 20
 - 2.5.3 替换 TagWatermarking 和 ReEncoder 组件 21
- 2.6 结论 22

第二部分 架构建模

第 3 章 架构级视点 24

- 3.1 概述 24
- 3.2 模型、视点、视图类型和视图 25
 - 3.2.1 模型和建模语言 25
 - 3.2.2 视图、视图类型和视点 27
 - 3.2.3 架构视点 28
- 3.3 结构视点 29
 - 3.3.1 组件存储库 29
 - 3.3.2 装配 32
 - 3.3.3 组件类型层次结构 33
- 3.4 行为视点 34
 - 3.4.1 组件内部行为 34
 - 3.4.2 组件间行为 35
 - 3.4.3 场景行为 36
- 3.5 部署视点 37
- 3.6 决策视点 38
 - 3.6.1 决策类型分类 39
 - 3.6.2 建模决策 39
 - 3.6.3 决策视图类型 40
 - 3.6.4 扩展示例 44
- 3.7 问题和练习 45
 - 3.7.1 问题 45
 - 3.7.2 练习 45
- 3.8 拓展阅读 46
- 3.9 小贴士 47

第 4 章 架构重用 48

- 4.1 术语 48
- 4.2 组件和接口重用 49
- 4.3 架构风格重用 52
- 4.4 架构模式重用 53
- 4.5 参考架构重用 54

4.6 产品线架构	54
4.6.1 产品合成	55
4.6.2 产品配置	55
4.6.3 上下文解释	56
4.7 问题和练习	56
4.7.1 问题	56
4.7.2 练习	56
4.8 拓展阅读	56
4.9 小贴士	57

第三部分 架构分析

第 5 章 质量建模	60
5.1 质量属性	60
5.1.1 性能	60
5.1.2 可靠性	61
5.1.3 成本	61
5.2 目标驱动方法	61
5.2.1 实用主义	62
5.2.2 表示	62
5.2.3 简化	63
5.3 组件质量	64
5.4 使用画像及其传播	67
5.5 执行环境	69
5.6 领域特定质量建模	71
5.7 整体 Palladio 模型装配	72
5.8 问题和练习	72
5.8.1 问题	72
5.8.2 练习	73
5.9 拓展阅读	74
5.10 小贴士	74

第 6 章 数据获取	75
6.1 概述	75
6.2 真实用户和应用程序性能监控	78
6.2.1 使用和适用性	78
6.2.2 数据解释	79
6.2.3 常用工具	80
6.2.4 示例	81
6.2.5 讨论	81

6.3 资源需求估算	81
6.3.1 使用和适用性	81
6.3.2 数据解释	82
6.3.3 常用工具	83
6.3.4 示例	84
6.3.5 讨论	84
6.4 估算失效概率	84
6.4.1 使用和适用性	85
6.4.2 数据解释	86
6.4.3 常用工具	86
6.4.4 示例	86
6.4.5 讨论	86
6.5 常见的缺陷	87
6.5.1 错误的资源需求	87
6.5.2 不准确的计时器	87
6.5.3 缺失资源	87
6.5.4 没有验证	87
6.5.5 太多细节	88
6.6 问题和练习	88
6.6.1 问题	88
6.6.2 练习	88
6.7 拓展阅读	89
6.8 小贴士	90

第 7 章 回答设计问题	91
7.1 设计问题	91
7.1.1 常见的质量设计问题	91
7.1.2 其他性能设计问题	92
7.1.3 其他可靠性设计问题	92
7.2 理解结果	93
7.2.1 理解性能结果	93
7.2.2 理解可靠性结果	101
7.3 提高质量的策略	103
7.3.1 性能策略	103
7.3.2 可靠性策略	104
7.4 自动搜索更好的架构	105
7.4.1 评估标准	105
7.4.2 自由度	105
7.4.3 PerOpteryx: 自动化探索	106
7.5 问题和练习	108

7.5.1 问题	108
7.5.2 练习	108
7.6 拓展阅读	108
7.7 小贴士	109

第 8 章 在后台 110

8.1 质量分析工具	110
8.2 Palladio 模型的性能模拟	112
8.2.1 工作负载生成器	112
8.2.2 模拟用户	112
8.2.3 模拟系统请求	113
8.2.4 模拟资源	114
8.2.5 模拟调度策略	114
8.2.6 收集度量数据	116
8.3 性能分析工具	119
8.3.1 SimuCom: 生成式模拟	119
8.3.2 EventSim: 解释式模拟	120
8.3.3 SimuLizar: 模拟中的 自适应	120
8.3.4 ProtoCom: 性能原型开发	121
8.3.5 进一步的分析工具	121
8.3.6 选择性能分析工具	122
8.4 可靠性分析	123
8.4.1 求解参数依赖性	123
8.4.2 确定可能的物理系统状态 及其概率	123
8.4.3 为单个 PSS 生成 DTMC	124
8.4.4 评估 DTMC 和聚合结果	124
8.4.5 总结	124
8.5 成本分析	125
8.6 问题和练习	125
8.6.1 问题	125
8.6.2 练习	126
8.7 拓展阅读	126
8.8 小贴士	126

第四部分 嵌入软件工程过程

第 9 章 软件工程过程 128

9.1 何时使用(或不使用)基于模型	
--------------------	--

驱动的软件质量预测	128
-----------	-----

9.1.1 基于模型驱动质量预测的 成本	128
9.1.2 基于模型驱动质量预测的 好处	129
9.1.3 Palladio 应用指标	129
9.1.4 嵌入软件系统活动	130
9.2 基于组件的质量感知的开发 过程	133
9.2.1 开发人员角色	134
9.2.2 组件开发	135
9.2.3 利用组件进行开发	135
9.2.4 规范工作流	137
9.2.5 质量分析工作流	140
9.2.6 对质量进行建模时的交互	141
9.3 开发过程中的应用	142
9.3.1 顺序过程模型	143
9.3.2 迭代过程模型	144
9.4 问题和练习	146
9.4.1 问题	146
9.4.2 练习	146
9.5 拓展阅读	147
9.6 小贴士	147

第 10 章 与需求工程的关系 148

10.1 需求工程基础	148
10.1.1 需求来源	148
10.1.2 需求类型	149
10.1.3 精化质量需求	151
10.1.4 软件质量术语	151
10.2 需求活动和架构级活动的关系	153
10.2.1 与架构级设计的关系	154
10.2.2 与架构级分析的关系	156
10.3 在生命周期后期阶段的需求 检查	157
10.4 问题和练习	158
10.4.1 问题	158
10.4.2 练习	158
10.5 拓展阅读	158
10.6 小贴士	159

第 11 章 与实现的关系 160

11.1 正向工程和逆向工程概述 160

11.2 正向工程 162

11.2.1 SimuCom 仿真映射 163

11.2.2 Java SE 和 EE 原型映射 165

11.2.3 Java SE 和 EE 代码存根
映射技术 167

11.2.4 SCA 映射技术 168

11.2.5 映射总结 168

11.3 逆向工程 168

11.3.1 组件结构检测 169

11.3.2 组件行为检测 171

11.3.3 控制流和数据流的参数化 172

11.3.4 Archimatrix 176

11.4 问题和练习 178

11.4.1 问题 178

11.4.2 练习 178

11.5 拓展阅读 178

11.6 小贴士 179

第五部分 案例研究**第 12 章 1&1 电子邮件系统的工作
负载感知监控** 182

12.1 引言 182

12.2 目标和问题 183

12.3 系统描述 183

12.4 建模 184

12.5 数据采集 185

12.6 分析 185

12.6.1 工作负载感知性能监控过程
的应用 186

12.6.2 STORE 子系统 186

12.6.3 性能建模研究 186

12.7 评估 194

12.8 经验教训 195

**第 13 章 IBM 存储虚拟化中的设计
权衡问题** 197

13.1 目标和问题 197

13.2 系统架构 197

13.2.1 虚拟化层的设计 198

13.2.2 同步设计 198

13.2.3 异步设计 199

13.2.4 同步决策的作用 199

13.3 结构与行为建模 200

13.3.1 组件交互和控制流 200

13.3.2 建模限制 202

13.4 数据采集 202

13.5 分析与评估 203

13.5.1 模型验证 203

13.5.2 仿真结果讨论 205

13.6 经验教训 206

**第 14 章 ABB ASP.NET 服务器的
设计空间探索** 208

14.1 待研究系统 208

14.2 目标和问题 209

14.3 建模 210

14.3.1 方法和工具选择 210

14.3.2 模型构建 211

14.4 数据采集 212

14.4.1 工具选择 212

14.4.2 工作负载建模 213

14.4.3 测量执行 214

14.4.4 模型校准 214

14.5 分析 215

14.5.1 手动设计空间探索 215

14.5.2 正式 RDS 成本模型 216

14.5.3 自由度和目标 216

14.5.4 自动探测结果 217

14.6 经验教训 221

14.6.1 性能建模增加理解 221

14.6.2 预测转移利益相关者讨论 221

14.6.3 经济效益必须存在 222

14.6.4 性能建模工具的能力有限 222

14.6.5 值得投资的好工具 222

第 15 章 未来趋势 223

15.1 拓展局限性的未来工作 223

15.2 未来的主题	224
15.2.1 云计算	224
15.2.2 绿色计算	224
15.2.3 DevOps	225
第 16 章 结论	226

后记 Palladio 简史	228
----------------------	-----

参考文献	230
------------	-----

索引	245
----------	-----

软件架构的工程方法

第一部分通过应用场景描述软件架构的仿真问题，并解释基于软件架构的质量预测在软件工程中的作用。引入 Palladio 方法，并通过一个软件系统的例子，概述 Palladio 方法中使用的软件架构建模语言的基本元素和概念。最后，概括本书的章节结构，并针对不同读者讨论可能的阅读路径。

引言

Ralf H. Reussner, Jörg Henss, Max Kramer

本章通过应用场景讨论对软件架构仿真的必要性，并解释基于软件架构的质量预测在软件设计工程中的作用。本章引入 Palladio 方法，以及 Palladio 方法的主要度量值、建模语言、承载方法的流程、相关的角色和职责等。本章结尾讨论本书结构以及不同读者阅读本书的可能路径。

目前，大家可以找到许多关于软件架构的好书。这没有什么好惊讶的，因为大约 20 年前软件架构就是一个在研究和实际应用中已经建立起来的领域了。

正如前言所述，本书特殊的地方就是它对文档化软件架构和仿真软件架构的关注。

为什么我们应该仿真软件架构呢？对于工程师来说，在实际创造某个制品之前使用模型对该制品进行仿真是常见的实践方法。汽车、电子电路、桥梁等的设计模型都可以用来进行仿真，以便了解设计决策对各种质量属性（例如安全性、能量耗损或稳定性等）的影响。在一个制品设计模型的基础上预测它的属性，而不必真正地实现它，这种能力是工程学科的主要特征之一。从已经建立的工程学科的观点来看，“软件工程”勉强算是一个工程学科。正如大家经常看到的，软件工程师缺少对设计决策如何影响质量属性（例如性能或可靠性）的了解，他们通常试图以成本很高的循环试错（trial-and-error）方式来“测试”质量。

如果我们不仿真会发生什么？不了解设计决策的影响可能带来的后期质量保障成本很高，也非常冒险：走到软件测试这一步，意味着我们已经把很多精力投入到软件的实现中了。例如，如果测试暴露了性能存在问题，很有可能需要修改软件架构，这会导致大量额外的成本。实际上，在很多企业软件系统中，系统的低性能主要是因为不合理的软件架构设计，而不仅是因为代码写得不好。

Glass 在他的书中指出：历史上失败的软件项目或富有挑战的软件项目清单很长（Glass 1998）。大量的近期项目也存在明显的延期或预算严重超支的情况。导致这些不良结果的原因可以在一定程度上追溯到软件架构及其质量属性问题，例如性能和可靠性。因此，问题的根源在于系统高层级的组织结构。关于性能的典型例子是由 Schmietendorf 和 Scholz（2001）以及 Koziolok（2008）提供的：丹佛国际机场和希思罗机场的自动行李处理系统以及 SAP 的商业设计（Business By Design, BBD）项目。

行李处理系统：行李处理系统的问题导致丹佛国际机场比原计划晚了 16 个月投入使用，超出预算差不多 20 亿美元，而且没有一个是全自动的行李处理系统。这是因为该系统的最初计划是服务于机场的一个航站楼，但最终要求服务于所有的航站楼（Montealegre and Keil 2000）。系统不能处理这种增加的需求，意味着它的可扩展性不是很好。类似的问题发生在 2008 年希思罗机场新建的 5 号航站楼上，行李处理系统所产生的消息数量对于可扩展性不好的系统来说实在是太多了^[24]，因此，在运营的最初几周中，就弄丢了 23 000 个行李箱，