HZ BOOKS
华章科技
PEARSON

软件架构领域开创性著作，资深软件架构师数十年工作经验结晶，具有里程碑意义

深刻阐述如何用架构视点和架构视图的方法定义软件架构，如何用架构视角的方法确保软件质量，以及如何用架构视点和架构视角的方法与利益相关者合作

华章程序员书库

Software Systems Architecture

Working with Stakeholders Using Viewpoints and Perspectives

Second Edition

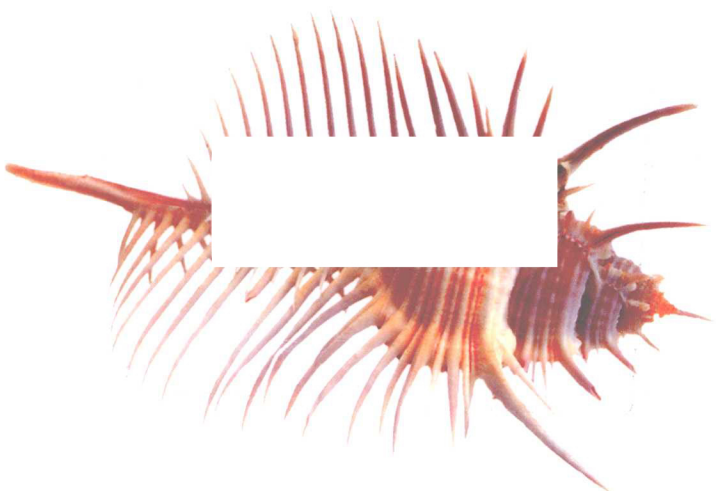
软件系统架构

使用视点和视角与利益相关者合作

（原书第2版）

（英）Nick Rozanski　Eoin Woods 著

侯伯薇　译



机械工业出版社
China Machine Press

华章程序员书库

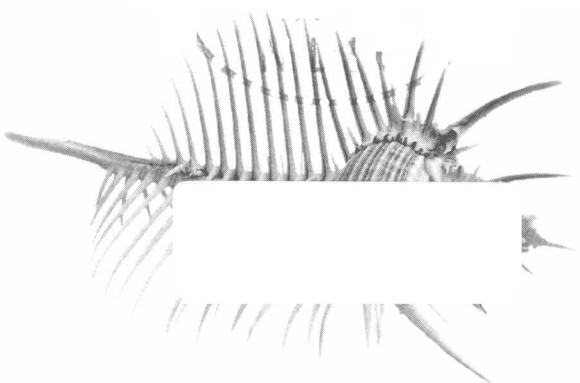
Software Systems Architecture
Working with Stakeholders Using Viewpoints and Perspectives
Second Edition

软件系统架构

使用视点和视角与利益相关者合作

（原书第2版）

（英）Nick Rozanski　Eoin Woods 著
侯伯薇　译



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

软件系统架构：使用视点和视角与利益相关者合作（原书第2版）/（英）鲁然斯基（Rozanski，N.），（英）伍兹（Woods，E.）著；侯伯薇译. —北京：机械工业出版社，2013.5
（华章程序员书库）
书名原文：Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives，Second Edition

ISBN 978-7-111-42186-3

I. 软… II. ① 鲁… ② 伍… ③ 侯… III. 软件开发 IV. TP311.52

中国版本图书馆CIP数据核字（2013）第077228号

本书版权登记号：图字：01-2012-1276

本书是软件系统架构领域的开创性著作，是两位拥有数十年软件行业工作经验的架构师工作经验的结晶，围绕利益相关者、视点和视角三大主题，创新性地提出了如何用架构视点和架构视图的方法来定义软件架构，如何用架构视角的方法来确保软件质量，以及如何用架构视点和架构视角的方法与利益相关者合作，具有里程碑意义。本书还展示了一种实用的、经过验证的框架，你可以应用它来处理架构定义过程，并应对创建软件架构工作所带来的挑战。

本书分为五个部分，共30章。第一部分（第1~5章）阐释利益相关者、架构描述、视点、视图和视角等基本概念，并描述软件架构师的角色；第二部分（第6~14章）描述作为架构师所要从事的重要活动，如协商项目的范围、识别并管理利益相关者、使用场景和模式、创建模型以及为架构创建文档并对其加以验证等；第三部分（第15~23章）集合了在创建架构描述时最重要的七种视点：情境、功能、信息、并发、开发、部署和运维视点；第四部分（第24~29章）集合了对于信息系统最重要的视角，包括安全性、性能和可伸缩性、可用性和适应性、演进、位置、开发资源、国际化等；第五部分（第30章）把这些概念融合在一起，并阐释了如何把这些理论应用到实践中。

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：秦 健
三河市杨庄长鸣印刷装订厂印刷
2013年5月第1版第1次印刷
186mm×240mm · 27.5印张
标准书号：ISBN 978-7-111-42186-3
定 价：99.00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
购书热线：（010）68326294 88379649 68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

译 者 序

作为一名程序员，对未来的职业规划会有很多种。有人想要走管理路线，基本目标是成为项目主管或者项目经理；有人对数据库感兴趣，期望成为DBA；有人对测试感兴趣，期望成为测试方面的专家或者质量保证方面的高手；还有很多人希望做架构师，能够从总体上把握一个系统或者一个组织中的所有系统。

和其他角色相比，在一个系统的设计和开发过程中，架构师的地位显得尤为重要，颇有“运筹帷幄之中，决胜于千里之外”的架势。但是，很多时候，在很多组织中，架构师只不过是一个职位的概念，很多人即便在拥有这个头衔之后，还是会有很多的疑问亟待解答，比如：

- 架构师需要哪些能力？
- 架构师需要从事哪些工作？
- 想要设计出良好的架构，需要考虑哪些问题？
- 如何与系统的各种利益相关者相互协调？他们的关注点一般又会有哪些？
- 如何编写出能够让相关人员都很好了解的架构文档？
- 如何评估设计出的架构？
- 在解决某些特定系统问题时，应该采用何种合适的方法？

……

很少有一本书能够系统地回答这么多问题，能够为迷茫的架构师做出有效的指导。我自己也是一样，作为一名程序员，对架构方面还是如履薄冰，了解了一些皮毛，绝不敢以架构师来自居。

直到偶然与这样的一本书相遇，我才发现，其实想要成为架构师，或者在成为架构师之后，想要更好地完成相应的工作，这本书都能够提供很好的指导和帮助。其中提出的各种视图、视点和视角，可以帮助我们更好地了解架构的诸多方面，从而在设计架构的时候，能够根据最重要的利益相关者的关注点，更好地进行平衡。

也正因为如此，我才争取到这样的机会，把这么好的一本书翻译成中文，呈献给国内的程序员朋友们，希望他们能够通过阅读这本书，更好地了解架构设计和架构师这种职业，也更好地为他们解决架构方面的问题。

不得不说，翻译这样一本大部头的确是很辛苦的事情，这也是我所翻译的最厚的一本书，所以首先要感谢我的家人对我的理解和支持，让我抽出晚上和周末的时间来从事该书的翻译工作。还要感谢关敏老师，容忍我一再延期交稿。还要感谢张勇等人的参与，正是大家的帮助，才让我最终完成了这本书的翻译。

愿这本书能够帮助更多有志成为架构师或者已经是架构师的朋友们！

前　言

和十年前我们开始撰写本书第 1 版时相比，当前 IT 业界的状况已经发生了天翻地覆的变化。计算机和互联网已经成为很多人日常工作和生活的重要组成部分，这使得整个世界更紧密地连接在一起。同时，这也让用户和其他利益相关者对系统寄予更大的期望，他们希望系统功能完备、易于使用、健壮稳定、方便扩展且安全可靠。我们认为，在实现这些目标的过程中，架构师扮演了非常重要的角色，而且广大专业的软件开发者以及高级业务和技术经理也深以为然。

我们非常高兴地看到，本书第 1 版得到了软件从业者、有抱负的软件架构师以及学术界的认可。读者认为它有用、可理解且内容丰富。然而，架构是一种不断变化的规则，因此本书第 2 版会反映出从第 1 版的出版到现在我们在实践中所学到和已经做出的改善。它还包括了读者发来的大量非常有用的评论和改进建议，对此我们深表感谢。

然而，本书的基础内容保持不变。本书的首要关注点依然在于：架构是为利益相关者提供的服务，并且是确保信息系统能够满足他们需求的一种方式。本书还是会强调视图（view）的极端重要性，它会以利益相关者可以理解的方式来显示架构的复杂性。我们还坚持认为，架构必须定义系统能如何提供利益相关者所要求达到的质量属性——如可扩展性、弹性和安全性——还要定义静态和动态结构，以及能够提供有效方式达到这些目的的透视图。

本书的主要受众是架构师或者期望成为架构师的人，但是，我们希望其他可能会和架构师一起工作的 IT 专业人士、某天可能会成为架构师的学生，都认为这本书值得一读。

第 2 版中做出的最重要改变如下。

- 本书引入了一种叫做情境的新视角（Context viewpoint），它会描述系统及其所处环境（人、系统以及与之交互的外部实体）之间的关系、依赖和交互。这部分内容还会将第 1 版第 8 章中对范围和情境相对简要的讨论进行扩展、格式化和标准化。
- 第二部分进一步讨论了架构角色的不同方面。
- 修订了大多视角和透视图的定义，特别是功能（Functional）和并发（Concurrency）视图以及性能（Performance）和可扩展性（Scalability）透视图。
- 修订和扩展了大多数章节的参考文献和延伸阅读部分。
- 更新了第 1 版中的部分内容，使其可以与新的国际架构标准 ISO 42010（它继承自 IEEE 标准 1471）中的概念和术语保持一致。
- 更新了 UML 建模的建议和示例，从而适应 UML 在第 2 版中所引入的变更。

我们希望你能够发现，本书第 2 版对第 1 版做出了很有用的提升和扩展，我们也邀请你访问本书的网站：www.viewpoints-and-perspectives.info，在那里你可以获得更多软件架构的资源，或者可以与我们联系，提供对本书的反馈。

致谢

除了在第 1 版中感谢的人，我们还要感谢审阅了本书第 2 版的各位朋友：Paul Clements、Tim Cull、Rich Hilliard、Philippe Kruchten 和 Tommi Mikkonen，还有勤勉负责的文字编辑 Barbara Wood。特别要感谢 Paul 彻底、具有洞察力和挑战性的意见与改进建议，这些对我们帮助很大。

第1版前言

本书的两位作者都是实践经验丰富的软件架构师，在信息系统开发项目中担当这个角色多年，在有些项目中是共同担任架构师，在有些项目中是分别担任的。在那期间内，我们看到软件架构师的数量得到了显著增加，而且该角色的重要性也得到同事、管理层和客户的认可。现在，所有大型的软件开发项目都需要架构师的参与，甚至在先进的开发团队中，会存在架构师小组。

尽管人们已经达成共识，认为软件架构师的角色非常重要，但是关于这一角色所包含的工作并没有一致的意见。谁是我们的客户？我们应该对谁负责？他们期望我们交付什么？一旦架构设计完成，我们又该做些什么？以及可能最重要的是，需求、架构和设计之间的边界是什么？

因为当前软件项目（特别是项目的架构师）中有很多亟待解决的问题，所以缺少对架构师角色的清晰定义，这导致了更多问题。

- 用户和其他利益相关者在功能、容量、推向市场的时间以及灵活性方面提出了更多苛求。
- 漫长的系统开发时间导致范围持续改变，进而导致系统架构和设计经常变更。
- 现在的系统要比以往功能更加丰富，结构更加复杂，经常会混用直接可用的和自定义构建的组件。
- 很少有系统能够独立存在，大多数都需要和很多其他系统交互操作并交换信息。
- 让系统的功能结构（即设计）正确只是问题的一部分。系统如何表现（例如，它的质量属性）与其对有效性同样重要。
- 技术持续改变的步伐让架构师的技术专业知识无法保持最新。

当第一次担任软件架构师的时候，我们试图寻找某种软件架构手册，希望它能够指导我们完成创建架构设计的过程。毕竟，其他架构方面的准则，在理论上和确立的最佳实践上都要领先软件架构几个世纪。

例如，在公元 1 世纪，罗马人 Marcus Vitruvius Pollio 就写出了第一本架构手册《建筑十书》（Ten Books on Architecture），这本书描述了建筑架构师的职责和所需要的技能，为标准的架构结构提供了丰富的资料。1670 年，日记作家 Samuel Pepys 的朋友 Anthony Deane，曾是英国 Harwich 镇的镇长，后来又成为一位议员，出版了一本开创性的书《船舶工程学》（A Doctrine of Naval Architecture），其中详细描述了当时大型船舶设计方面的先进方法。Deane 的观点和原则在很多年间帮助人们将船舶工程的实践系统化。1901 年，英国化学工业的顾问工程师 George E. Davis 发表了《化学工程学手册》（A Handbook of Chemical Engineering），从而创建了一种新领域的工程学。这是第一本定义很多实践性原则的书籍，这些原则支持了工业化化学过程，并在之后的很多年引领了该领域的发展。

这些最佳实践存在的重要结果就是统一了方法。如果你想要给几位架构师和工程师一项设计楼房、旅游船或者化工厂的任务，他们创建出来的设计可能会彼此不同。然而，他们使用的过程、

在纸上（或者计算机屏幕上）展现设计的方式以及用来确保设计合理的技术非常类似。

遗憾的是，我们的行业还没有创建出可流传下去的主流业界的最佳实践。当我们查看的时候，发现很少会有介绍性的书籍，可以为信息系统架构师的工作提供详细指导。

我们承认，在特定的技术上已有很多书籍，不管是 J2EE、CORBA 还是 .NET，还有一些更广泛的话题像 Web 服务或者面向对象（尽管由于软件技术变化的速度，很多像这些书籍在几年内就过时了）。也有大量优秀的软件架构方面的书籍，我们会在部分章节中对其加以引用。但是因为这些书很多旨在创建在各种系统中都可以应用的原则，所以使用很通用的方式进行编写，大多数特定文字的目标都是针对实时和嵌入式系统社区的朋友。

我们觉得，如果你是一位新的信息系统软件架构师，很少有书会告诉你如何完成自己的工作，学到你所需要知道的最重要的知识，并成功地做出架构设计。我们不想取代已经存在的软件架构方面的书籍，也不敢与 Vitruvius、Deane 和 Davis 比肩，让我们决定撰写这本书的驱动力就是要解决上述需求。

具体来说，本书将会向你展示：

- 软件架构的相关内容有哪些，你的角色对于成功交付项目为什么极度重要。
- 如何决定谁对你的架构感兴趣（你的利益相关者），理解什么对他们重要（他们的关注点），并设计反映和平衡他们不同需求的架构。
- 如何以可理解的方式与利益相关者沟通你的架构，从而说明你已经满足了他们的关注点（架构描述）。
- 如何关注对架构重要的内容，安全地把其他方面的设计工作交给设计师，而不会遗漏如性能、弹性和位置等问题。
- 作为架构师，哪些重要的活动是你最需要执行的，例如识别并引入利益相关者、使用场景、创建模型以及编写架构文档并加以验证。

贯穿全书，我们首先关注的是大规模信息系统（即用来自动化大型组织的业务操作的计算机系统）的开发。然而，我们也试图以一种特别的方式来展现内容，这种方式不依赖你正在设计的信息系统的类型、开发者将要使用的技术以及项目所遵循的软件开发生命周期。我们标准化了一些事情，例如在大多数图表中使用了统一建模语言（Unified Modeling Language, UML），但是我们只做了这么多，因为 UML 是最广为理解的建模语言。想要理解这本书，你不需要是一位 UML 专家。

我们并不打算让本书成为创建你的信息系统架构的确切指导——这样一本书可能永远都不可能完成，并且需要大量跨广泛技术领域专家的合作。我们也不想编写一本规范方法的书。尽管我们展示了说明如何产出可交付物的一些活动图，但是设计这些活动图是为了与当前使用中的多种软件开发方法相兼容。

我们期望达到的目的是，创建一种可实践的、面向实践者的指导，说明如何为信息系统设计出成功的架构，以及如何保证它们可以成功实现。这是我们在开始从事软件架构师工作时想要看到的那种书，也是我们现在期望参考的书。

你可以访问我们的网站：www.viewpoints-and-perspectives.info，找到更多有用的软件架构资

源，并和我们联系，提供对本书内容的反馈。我们非常期待能够听到你的声音。

致谢

本书的出版得到了很多人的建议、帮助和支持，否则这本书永远都不会问世。

我们非常感谢多位审校者，他们在我们创作本书的不同阶段对文字给出了意见，包括 Gary Birch、Chris Britton、Kelley Butler、Sholom Cohen、Dan Haywood、Sallie Henry、Andy Longshaw、Robert Nord、Dan Paulish、Martyn Thomas 和 Hans van Vliet。

我们还要感谢 Addison-Wesley 的团队成员，正是他们的工作让这本书成为现实，包括 Kim Boedigheimer、John Fuller、Peter Gordon、Chrysta Meadowbrooke、Simon Plumtree 和 Elizabeth Ryan。

还有很多人为我们提供了建议、鼓励和启发，包括 Felix Bachmann、Dave Borthwick、David Emery、Wolfgang Emmerich、Rich Hilliard、Philippe Kruchten、Roland Leibundgut、Mike Mackay、Dave Maher、Mark Maier、Lucia Rapanotti 和 Gaynor Redvers-Mutton。

我们还要感谢我们的家人，他们始终为我们提供持续的爱、鼓励和支持。

目　录

第二部分 软件架构过程

第三部分 视点类型

第四部分 视角