O'REILLY®

TURING 图灵程序设计丛书

演进式架构

Building Evolutionary Architectures

敏捷之父Martin Fowler作序推荐

ThoughtWorks CTO等技术大牛详细讲解

先进架构思想

[美] 尼尔·福特　丽贝卡·帕森斯 著

[澳] 帕特里克·柯

周训杰 译

中国工信出版集团　人民邮电出版社 POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

演进式架构

Building Evolutionary Architectures

[美] 尼尔·福特 [美] 丽贝卡·帕森斯 [澳] 帕特里克·柯 著
周训杰 译

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北 京

图书在版编目（CIP）数据

演进式架构 / （美）尼尔·福特（Neal Ford），（美）丽贝卡·帕森斯（Rebecca Parsons），（澳）帕特里克·柯（Patrick Kua）著；周训杰译. -- 北京：人民邮电出版社，2019.8
（图灵程序设计丛书）
ISBN 978-7-115-51617-6

Ⅰ. ①演… Ⅱ. ①尼… ②丽… ③帕… ④周… Ⅲ. ①程序设计 Ⅳ. ①TP311.1

中国版本图书馆CIP数据核字(2019)第137818号

内 容 提 要

在软件开发流程中，为了尽可能快地响应各种变化，理应把结构渐进改变作为设计的首要原则。本书详尽阐述了演进式架构的必要性、构建方法以及需要注意的问题。各章结合案例分别讨论了软件架构、适应度函数、开展增量变更、架构耦合、演进式数据、构建可演进的架构、演进式架构的陷阱和反模式，以及实践演进式架构。

本书适合所有架构师及开发人员阅读。

◆ 著　[美] 尼尔·福特 [美] 丽贝卡·帕森斯 [澳] 帕特里克·柯
译　周训杰
责任编辑　朱　巍
责任印制　周昇亮

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
天津翔远印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：9.75
字数：230千字　2019年8月第1版
印数：1-3 500册　2019年8月天津第1次印刷
著作权合同登记号　图字：01-2018-8084号

定价：59.00元

读者服务热线：(010)51095183转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

译者介绍

周训杰

ThoughtWorks高级咨询师，拥有十余年企业级应用和互联网应用的开发和架构经验，在ThoughtWorks带领多个团队完成不同规模的海外项目交付，目前负责大型企业级项目的服务化平台搭建及技术团队管理。

微信连接

回复“架构”查看相关书单

微博连接

关注@图灵教育 每日分享IT好书

QQ连接

图灵读者官方群I：218139230
图灵读者官方群II：164939616

图灵社区
iTuring.cn

在线出版，电子书，《码农》杂志，图灵访谈

此为试读，需要完整PDF请访问：www.ertongbook.com

版权声明

© 2017 by Neal Ford, Rebecca Parsons, and Patrick Kua.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2019. Authorized translation of the English edition, 2019 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，2017。

简体中文版由人民邮电出版社出版，2019。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。

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O'Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 *Make* 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务还是面授课程，每一项 O'Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar 博客有口皆碑。”

——*Wired*

“O'Reilly 凭借一系列非凡想法（真希望当初我也想到了）建立了数百万美元的业务。”

——*Business 2.0*

“O'Reilly Conference 是聚集关键思想领袖的绝对典范。”

——*CRN*

“一本 O'Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——*Irish Times*

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野，并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——*Linux Journal*

序

长久以来，软件行业都奉行这样一个理念：在开始编写第一行代码前就应该完成架构开发。受到建筑行业的影响，人们认为成功的软件架构在开发过程中不需要修改，而且重新架构往往会导致高成本的报废和返工。

随着敏捷软件开发方法的兴起，这样的架构愿景受到了很大的挑战。预先规划的架构要求在编码前就确定需求，因而催生了分阶段（即瀑布式）的开发方法——在完成需求分析后才开始设计架构，然后再进入构建（编码）阶段。然而，敏捷软件开发方法并不认同“需求是确定的”这样的观点。它发现现代商业中需求不断变化是必然的，并进一步提供了项目规划技术来拥抱受控的变化。

在这个新的敏捷世界里，很多人质疑架构的作用。当然，预先规划的架构无法适应动态的现代业务，但是，还是会有另外一种架构，它以敏捷的方式拥抱变化。如此看来，架构是不断努力的结果，是一个与开发工作紧密结合的过程，这样它才能同时响应不断变化的需求和开发人员的反馈。我们称之为“演进式架构”，正是要强调当无法预测变化时，该架构仍然可以朝着正确的方向发展。

在 ThoughtWorks，我们无时无刻不秉持演进式架构的理念。Rebecca 在 21 世纪初领导了多个重要项目，作为 CTO，她提升了 ThoughtWorks 的技术领导力。Neal 一直密切关注着我们的工作，将我们学到的经验汇总并传播开来。Pat 在开展项目工作的同时也加强了我们的技术领导力。我们始终认为架构是极其重要的，不容忽视。我们虽然也犯过错误，但我们从错误中学习，更好地理解了如何构建出能正确应对各种变化的系统。

构建演进式架构的核心是采取小步变更，然后通过反馈环让团队的每个成员不断地从系统的发展过程中学习。持续交付的兴起使得演进式架构变得切实可行。三位作者通过适应度函数监控架构的状态。他们探索了架构演进的不同方式，并且重视那些通常被别人所忽视的长存数据。在讨论中也理所当然地会经常提及康威定律。

关于构建演进式软件架构，虽然我们还需要了解很多东西，但本书基于当前的理解给出了基本线路图。随着越来越多的人意识到软件系统在 21 世纪人类社会中的核心地位，每个软件领导者都应该知道如何在发展中以最好的姿态应对变化。

Martin Fowler

前言

排版约定

本书使用如下排版约定。

- **黑体字**
  表示新术语或重点强调的内容。
- 等宽字体（`constant width`）
  表示程序以及段落内引用的程序元素，如变量、函数名、数据库、数据类型、环境变量、语句和关键字。
- 加粗等宽字体（**`constant width bold`**）
  表示应该由用户输入的命令或其他文本。
- 等宽斜体（*`constant width italic`*）
  表示应该被替换为由用户提供的值或由上下文确定的值的文本。

该图标表示提示或建议。

O'Reilly Safari

Safari（原来叫 Safari Books Online）是一个会员制的培训和参考咨询平台，面向企业、政府、教育从业者和个人。

会员可以访问来自 250 多家出版商的上千种图书、培训视频、学习路径、互动式教程和精选播放列表，这些出版商包括 O'Reilly Media、Harvard Business Review、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Adobe、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 等。

要了解更多信息，可以访问 http://oreilly.com/safari。

联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到书的相关信息，包括勘误表[1]、示例代码以及其他信息。本书的网站地址是：http://oreil.ly/2eY9gT6。

对于本书的评论和技术性问题，请发送电子邮件到 bookquestions@oreilly.com。

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：
http://www.oreilly.com。

我们在 Facebook 的地址如下：http://facebook.com/oreilly。

请关注我们的 Twitter 动态：http://twitter.com/oreillymedia。

我们的 YouTube 视频地址如下：http://www.youtube.com/oreillymedia。

附加信息

可以通过 http://evolutionaryarchitecture.com 访问本书的配套站点。

注 1：本书中文版的勘误请到 http://www.ituring.com.cn/book/2440 查看和提交。——编者注

此为试读，需要完整PDF请访问：www.ertongbook.com

致谢

Neal 要感谢过去几年里在各种大会倾听过他演讲的所有人，他们帮助打磨和修订了本书。他也要感谢本书的技术审阅人员，他们为本书提供了非常宝贵的建议和意见，特别是 Venkat Subramanium、Eoin Woods、Simon Brown 和 Martin Fowler。Neal 还要感谢他的猫 Winston、Parker 和 Isabella，它们的干扰总能为他带来顿悟。他还想感谢他的朋友 John Drescher、ThoughtWorks 的所有同事、Norman Zapien 敏锐的耳朵、每年的 Pasty Geeks 度假团和附近的鸡尾酒俱乐部，感谢他们的支持和友谊。最后，Neal 要感谢他坚忍的妻子，她用笑容包容了他的经常出差和为事业所做的其他牺牲。

Rebecca 要感谢所有同事、大会出席者及演讲者和作者，感谢他们多年来为演进式架构领域贡献的想法、工具、方法以及提出的澄清问题。她也要感谢本书的技术审阅人员，感谢他们细致的阅读和评注。此外，Rebecca 还要感谢本书的合著者，感谢创作过程中所有启发性的对话和讨论。她要特别感谢 Neal，几年前与他有过一场关于应急式和演进式架构区别的大讨论或者说是辩论，之后这方面的思想才逐渐地成型。

Patrick 要感谢 ThoughtWorks 的所有同事和客户，感谢他们推动了需求，并为阐明演进式架构的构建想法提供了测试平台。他还要同 Neal 和 Rebecca 一起向本书的技术审阅人员表示感谢，他们的反馈极大地提升了本书的质量。最后，他要感谢本书的合著者们。过去几年中，他们位于不同的时区，相距甚远，很少有机会能面对面地一起工作，特别感谢能有机会在这个项目上和他们密切合作。

电子书

扫描如下二维码，即可购买本书电子版。

目录

第1章

软件架构

一直以来，由于软件架构涉及范围广且内涵不断变化，开发人员不断尝试给它一个简洁的定义。Ralph Johnson 就将其定义为“重要的东西（无论那是什么）”。架构师的工作就是理解和权衡那些“重要的东西”（无论它们是什么）。

为了给出解决方案，架构师工作的第一步是理解业务需求，也即领域需求。这些需求是使用软件来解决问题的动机，但终究只是架构师在构建架构时需要考虑的因素之一。架构师还必须考虑其他很多因素，其中一些比较明确（比如清楚地写在性能服务水平协议里），还有一些则隐含在商业活动中不言自明（比如公司正着手并购重组，软件架构显然也要有变动）。所以对于软件架构师来说，架构水平体现了他们在权衡业务需求和其他重要因素后找到最佳方案的能力。软件架构涵盖了所有这些架构因素，如图 1-1 所示。

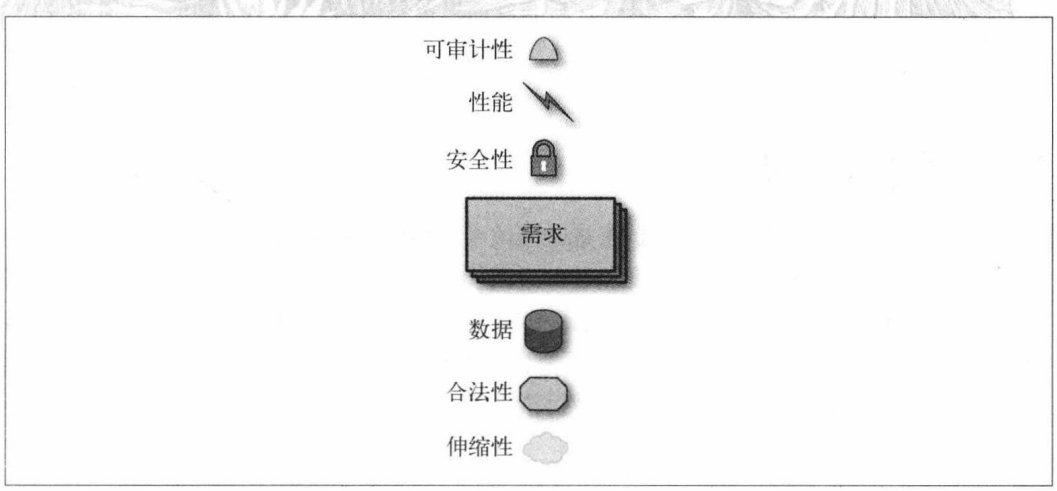

图 1-1：架构的完整概念，包括需求及“各种特征”

此为试读，需要完整PDF请访问：www.ertongbook.com