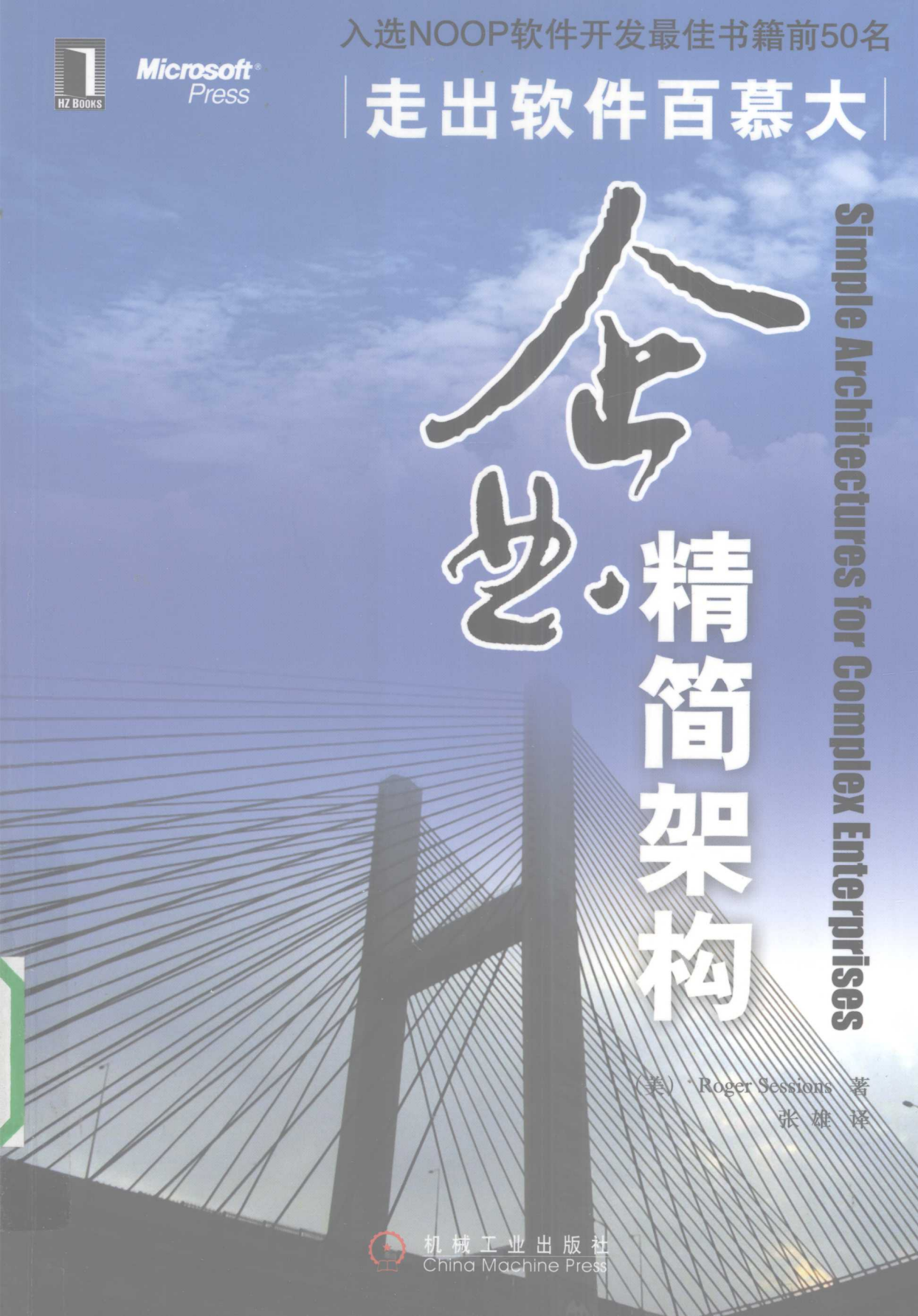
HZ BOOKS
Microsoft Press
入选NOOP软件开发最佳书籍前50名
走出软件百慕大
企业精简架构
Simple Architectures for Complex Enterprises
(美) Roger Sessions 著
张雄 译
机械工业出版社
China Machine Press

走出软件百慕大

企业精简架构

Simple Architectures for Complex Enterprises

Roger Sessions 著
张雄 译

机械工业出版社
China Machine Press

本书主要论述如何精简企业架构。全书分为两部分：第一部分，通过一些直观的问题展示复杂性带来的问题，接着从数学的角度进行分析；第二部分，讨论解决复杂性问题的过程。这个过程就是SIP，即简单迭代分割。

读完本书之后，你会成为反复杂联盟中优秀的一员，面对各种难题，你都可以使用你定义的企业架构来简化它。本书内容详实，图文并茂，可作为企业架构师的参考用书。

Roger Sessions：Simple Architectures for Complex Enterprises(ISBN：978-0-7356-2578-5)

版权登记号：图字：01-2009-1148

图书在版编目(CIP)数据

企业精简架构/(美)塞瑟斯(Sessions, R.)著；张雄译. —北京：机械工业出版社，2009.9
(走出软件百慕大)
书名原文：Simple Architectures for Complex Enterprises

ISBN 978-7-111-26626-6

Ⅰ. 企…　Ⅱ. ①塞…　②张…　Ⅲ. 企业管理－研究　Ⅳ. F270

中国版本图书馆CIP数据核字(2009)第040245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：陈佳媛
北京瑞德印刷有限公司印刷
2009年9月第1版第1次印刷
170mm×242mm · 11.25印张
标准书号：ISBN 978-7-111-26626-6
定价：39.00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

前　言

那是最美好的时代，那是最糟糕的时代；那是智慧的年头，那是愚昧的年头；那是信仰的时期，那是怀疑的时期；那是光明的季节，那是黑暗的季节；那是希望的春天，那是失望的冬天；我们拥有一切，我们一无所有……

这是查尔斯·狄更斯(Charles Dickens)1775 年写的《双城记》(A Tale of Two Cities)的开头部分，书中的两个城市是伦敦和巴黎。如果狄更斯活到现在，我猜他一定会写一本企业架构领域的书，一本涉及整合商业需求和 IT 解决方案的书。

这是最美好的时代。企业架构的目标就是通过 IT 投资获取最大的商业价值。对大多数企业而言，无论是大规模的还是小规模的，盈利性质的还是非盈利性质的，公有的还是私有的，它们都越来越强烈地希望通过 IT 投资使商业回报最大化，并帮助 IT 更高效地配合商业的需求、促进业务增长。毫无疑问，他们对企业架构的兴趣变得空前高涨。

这是最糟糕的时代。企业架构的目的是确保 IT 系统带来商业价值，但实际往往事与愿违。执行官们现在正在对企业架构失去信心，他们觉得企业架构和一般的 IT 没什么区别。这种信任危机正逐渐蔓延到各种规模、各种领域、各种类型的公司。2007 年 10 月，Gartner 就曾预言：到 2010 年，有 40% 的现存的企业架构将会消亡。在那本影响广泛的书《Enterprise Architecture as Strategy》(哈佛商学院出版社，2006)中，作者 Ross、Weill 和 Robertson 说：真正能够有效利用企业架构的企业还不到 5%。从我对企业架构的研究和该书作者提到的实现中，我发现了一个共同点：企业架构在高付出后却没有高回报。因此，对企业架构回报能力的质疑达到空前也就不足为奇了。

企业架构以一种高层次的企业视野，聚焦于组织的 IT 架构和业务架构之间。IT 架构描述 IT 系统，业务架构描述业务过程。IT 系统如果不能满足商业需求，那将是大大的浪费；业务过程如果没有良好的 IT 支持，效率很难提高。企业架构描述这两者之间如何互为补充，以确保组织中的 IT 系统能够高效地支持业务过程。

显然，这是个好主意。然而，企业架构却正趋于失败。

哪里出问题了呢？根据我的经验，目前实现企业架构的途径有三个基本问题尚待解决。首先，这些途径执行起来代价不菲；其次，耗费的时间太多；最后，没有方法验证结果。我们耗费大量资金、大量时间来创建架构，而为了测试这些架构的效果，我们还必须构造数量众多、费用昂贵的实现。尽管如此，我们不仅没有一种方法来评价某个给定架构的优劣与好坏，甚至大多数企业架构方法论都没有一个标准来衡量什么是“优”，什么是“劣”。

你怎么知道一个典型的企业架构是优还是劣呢？很简单，尝试着实现它。对于存在的业务过程，构造支持它的IT系统。如果你成功地交付了这些IT系统，并且它们满足商业需求，那么你的企业架构一定相当棒。如果不满足，祝你下次好运吧！

在其他科学领域里，这种途径根本就是行不通的。没有人会在发射登月火箭之前不通过行星的数学模型计算运行轨道；不通过重力压力模型计算所需的燃料。没有人认为建一座大桥之前不需要进行压力、负载、流体流的架构测试。

那为什么我们不在实现一个昂贵的大企业架构的时候首先通过数学模型来测试它是否有效呢？原因很简单，我们根本就不知道如何来测试。换句话说，站在数学的角度上，我们缺乏对“优”的理解，也缺乏一种测试“优”的模型，我们甚至缺乏“优”的基本定义。

没有这些模型(和定义)，就没有方法来验证企业架构，没有方法预算成本，没有方法确保它的交付能否带来商业价值，甚至没办法知道它是否能够交付。这就是企业架构领域有这么多麻烦的原因所在，也是我们经常听到IT失败的原因：项目经费超支、交付延期、无法满足需求等，不一而足。

最近，在一个影响力很大的IT论坛上，我曾和两个高级架构师就企业架构问题讨论过。我问他们，他们的IT项目中有多大比例能够准时地交付，项目符合预算，好钢用在刀刃上。他俩面面相觑，其中一人说，“按时交付、符合预算、好钢用在刀刃上？我觉得我的项目中没有一个项目能够达到这样的标准。”他把脸转向另外一位，“你的呢?”另外一人遗憾地摇了摇头。我也和很多企业的专业人士聊过相同话题，聊天对象包括架构师、首席信息官(CIO)和首席技术官(CTO)。

《IEEE Spectrum》最近有篇文章中写出了如此令人沮丧的断言：

> 看看最近五年新建的软件项目(包括政府项目和商业项目在内)的总投资，我估计项目失败引起的损失至少有250亿美元，甚至可能达到750亿美元。当然，这750亿不是反映在那些超支的项目上(大多数项目都会超支，也不反映在那些延期发布的项目上)，相当多的项目都延期发布了。这750亿也不包括那些遗弃后又可能重新启动的项目成本，也不包括那些充满Bug

而不得不返工的项目成本。[⊖]

鉴于这种情况，我们该如何应对呢？放弃企业架构？Gartner 不是预言 40% 的企业架构会消亡吗？我们该放弃吗？不，我们不能放弃这个领域。企业架构的目标实在是太重要了，我们不应放弃。相反，我们应该解决这个问题：怎样构建优秀的企业架构。

谈到“优”，我指的是五个方面。第一，我们必须定义好一个“优”的企业架构的含义；第二，利用这个定义，建立“优”的数学模型；第三，扩展这种数学模型，使之成为一个至少看起来优秀的企业架构的正式模型；第四，创建一个基于这个模型的“优”的企业架构的开发过程；第五，在实现模型之前，通过模型来验证架构的结果。

以上这几点都是以“优”的良好定义为基础。这儿，我的定义是：一个“优”的企业架构就是一个“简单”的企业架构。在两个能够有效整合商业需求和 IT 性能的架构中，如果哪个简单，哪个就优秀。两个架构中，较复杂的那个就是拙劣的。

在解决已存在的复杂问题时，我们不应创造新的复杂问题，否则只会越弄越麻烦，毕竟商业方面的问题已经够复杂了。商业活动历尽千辛万苦去适应新的技术，与日益严格、不断变化的需求作斗争。所有这些都是复杂的问题，并且只会越来越多。至于 IT 方面呢，复杂变成了家常便饭。对公司或组织机构而言，软件变得更加分布式，种类日趋繁多，交互日益频繁。所有的这些都是复杂的问题，并且越来越多。

鉴于业务和软件系统都日益复杂，它们之间的关系也变得更难理顺。业务部门和 IT 部门处理各自的工作也变得更加专业。他们开发了自己的语言，甚至创造了自己的文化。他们都没有时间关注他们之外的世界。因此，业务部门和 IT 部门之间的隔阂也越来越大。

在大多数组织里，业务部门和 IT 部门之间的裂痕正在扩大。大多数读者对这些不足为奇。尽管众所周知，但知道裂痕原因的人却寥寥无几。IT 部门责怪业务部门，而业务部门又归咎于 IT 部门，两边互相埋怨。彼此逐渐不信任，相互指责成了家常便饭。业务部门的人对 IT 部门提出过分的需求，这样使 IT 部门的人工作压力越来越大，工作难以完成。反过来，IT 部门的人阻碍业务进展，使销售额在竞争日益激烈的环境里难以提升。

其实，问题既不出在业务部门，也不出在 IT 系统。这根本就是在 IT 部门和业务部门普遍存在的一个基本问题。实际问题就是复杂，而复杂其实就是每个人的问题。

所以，问题就复杂了。但是复杂的问题，它本身并不需要一个复杂的解决方案。反之，本书的前提就是用简单的方案来解决复杂的问题。复杂的解决方案太麻烦了。

⊖ “Why Software Fails”，《IEEE Spectrum》2005 年 9 月，作者 Robert N. Charette。

对付复杂问题的杀手锏就是简化。如果能够化繁为简，那么你已经成功了一大半。当然，化繁为简并非易事。本书会教你如何做。

要简化问题，首要事情是把焦点放到简单上，并把它当作核心价值。我们都在讨论软件系统的敏捷、安全、性能、可靠性等重要性，好像它们是所有需求中最重要的一样。实际上，我们应该把简单放到与它们同等重要的位置上。我们在理解怎样让架构系统变得安全、高效、可靠上肯定不遗余力，同样我们应该多花精力让系统变得更简单。实际上，我并不是说简化问题同其他的特性有同样的优先级，而是说它的优先级应该更高。在众多特性中，它才是核心。

就安全而言。如果一个简单的系统不安全，想要让它变得安全并不难。而那些看起来安全的复杂系统往往并不安全，但想让这些不安全的复杂系统变得简单或者安全却几乎不可能。

现在，我们再看看敏捷。那些定义良好、互操作达到了最小化的简单系统，可以把它们以新的方式整合在一起，尽管这些整合方式是在系统创建之初没有考虑过的。复杂的系统就不能以这种敏捷的方式整合，它们太复杂了。当然，想让它们变得简单也不是一件容易的事情。

尽管对一个系统需求来说，简单的重要性非常高，但在架构计划、开发、评审过程中却往往被我们忽视了。最近，我在不少国家和组织做过很多演讲，演讲的对象(至少上百人)是企业架构师、CIO 和 CTO。几乎每次演讲中我都问过同样的问题：在他们参加过的项目中，是否有人曾经把简单作为系统的一个特性考虑进去。回答是没有。

对简单的寻求还没结束。甚至那些在设计之初就把简单考虑进去的系统(确实，很少系统会这么做)在以后仍然会陷入永无止境的挑战。当系统的性能需要提高的时候，这儿做个补丁；当需要改进互操作性的时候，那儿做点改进。差不多一两年后，曾经简单的系统就会变得臃肿，甚至一团糟。所以，本书不仅仅是教你如何创建一个简单的系统，而且还教你如何使系统保持简单。

首先声明，本书并非适合所有的人。如果你们的系统基本上都能准时交付，符合预算，满足业务需求，那么本书对你来说完全没用。要么，你做的系统比我们讨论的简单得多，要么你已经找到了如何管理复杂问题的方法。无论属于哪种情况，我恭喜你，你属于比较幸运的少数。

最近，我认识了一个人，Kevin Drinkwater，他是 Mainfreight 公司的 CIO。Mainfreight 是新西兰最大的跨国公司，它每年给新西兰政府提供的税收就高达 5 亿多美金。Kevin 因为他在 IT 方面的创新和解决方案的成本效率和敏捷而著称。他曾经一夜之间放弃了公司购买的价值 1300 万的 JD Edwards ERP 实现，而用一个 2.5 万美金的国产系统取而代之，就因为这事，在新西兰曾经轰动一时。他是 ComputerWorld 的

年度 CIO 候选人，也是一位知名的演讲家。同时，Kevin 在他的商业部门还是一位值得信赖的顾问，很少有 CIO 能够享有这种地位。

在由 Fronde 发起、ComputerWorld 新西兰分公司主办的圆桌会议中，Kevin 和我交换了企业架构中关于简单性的意见。我们彼此描画了对简单企业架构的看法，结果显示，我们双方都对简化持赞成态度。Kevin 就不需要我来传达简化的思想。他和他的 IT 组织每天都在简单地吃饭、喝水、呼吸。简单就是他们每天所做的事情的核心需求。这也就是为什么 Kevin 每发布一个系统都能按时、符合预算的原因，而且系统也完全符合业务需求。

如果你和 Kevin 是同一类的人，你压根就没必要看这本书。然而，这类人少之又少。很有可能，你们组织开发的系统超时、超过预算，并且可能你的技术部门和业务部门不和睦。那么，你非常需要这本书。如果你是 IT 执行官、IT 经理、软件架构师，或者项目的业务分析师，如果你发现你的项目的复杂性成指数增加，那么，你将会发现本书相当有价值。

为何我要说本书相当有价值呢？因为它可能会改变你对企业架构的理解，它可能会改变你在某些方面的观念：为什么我们需要它们？怎样才能做得更好？怎样高效地实现它们，它们怎样才能提供更多的业务价值？

所有的这些，都归于简单。简单是核心价值，简单是推进器，简单是业务价值。正如一个大航空公司的首席架构师曾经对我说过的，“我同很多人、很多组织讨论过企业架构的问题。他们讲的都千篇一律，没有一人切中要害。你是第一个与众不同的人，也是第一个一语中的的人。”其实，并非我一语中的，而是简单。

怎样让问题变得简单呢？很简单，避免复杂性。认知它、理解它、消灭它。等你读完这本书的时候，你就会知道怎么做了。你会从数学的角度理解复杂，从数学的角度理解以复杂性为基础的模型，从数学的角度理解消灭复杂性的过程，从数学的角度理解那些确保复杂性不再光临企业架构的方法。到那时候，你的生活、你的架构，都会变得更简单。

本书表面上是在讲企业架构，实质是在讲述更基本的东西：简单。本书中提到的控制复杂的方法不仅可以成功地应用于商业架构，还可以应用于 IT 架构。不过，如果它应用于某个层面（这个层面包括既应用于商业架构，也应用于 IT 架构），它的效果将会充分发挥。这个层面就是企业架构的层面。

本书组织结构

本书开始从一个比较直观的复杂性问题着手，接着转到一个正式的理解，最后转到一个以过程为中心的讨论。这个特殊的过程叫做 SIP，也就是简单迭代分割。

SIP 是唯一一个以问题复杂性为中心的企业架构方法论。

第一部分，“复杂性问题”给出了企业架构上关于复杂性问题的基本理解。

第 1 章，“当今企业架构”介绍了一般的企业架构，包括当今主流的方法论和它们在对付复杂性问题的优缺点。

第 2 章，“初识复杂性”以一种非数学的方式介绍了划分、遍历、简化的主要内容，以及在复杂性控制中，这三者之间的关系。在读这章的时候，你会发现你在企业架构方面可以学到很多，尤其是执行官午餐、紧急救护和国际象棋这几节。

第 3 章，“复杂性的数学原理”正式地从数学角度介绍复杂性。别担心，本章并不需要很深的数学基础。文中涉及的掷骰子、分割、布尔问题这些内容都很通俗易懂。而且，所有这些都会从基础讲起，它们都是基础的复杂性模型。这些模型帮助我们更好地理解：当维护企业划分时，复杂性是如何改变的。

第二部分：“简化的需求”描述了我提倡的解决企业架构复杂性的一种方法论。

第 4 章：“企业划分的 ABC”，介绍了自治的商业能力（Autonomous Business Capability，ABC）。ABC 是划分子集的企业等价。理解了 ABC 的属性和它们之间的相互关系，我们就能创建一个以简单为核心的企业架构。

第 5 章：“SIP 过程”详细地描述简单迭代分割（Simple Iterative Partitions，SIP）。这是我们控制复杂的方法。它着眼于数学复杂性，基于鉴别、操纵、再划分和再认知 ABC。

第 6 章：“复杂性的案例研究”。本章举了一个相当复杂的项目实例，IT 方面的国家项目，英国的国家保健系统（National Health Care System）的一部分。如果你认为你见过复杂的项目，先别忙着说。这个项目已经花费了几十亿美金，把好几个公司带到了财政崩溃的边缘。说不定，最后将会以世界上最大的 IT 失败项目而告终。本章将讨论到底是哪儿出了问题，如何运用 SIP 方法挽救这个项目。

第 7 章：“警戒边界：软件城堡”。ABC 的软件组件，讨论在维护自治系统的边界时面临的挑战。我将会讨论一种称之为“软件城堡（Software Fortresses）”的模式，它允许你把 SIP 的简单数学模式应用到软件系统。

第 8 章，“复杂性展望”，回顾本书讲到的关键点，怎样能够把你带到对复杂性的更深层次的理解，使用它使简单融入到公司文化中。

本书包含一个附录，“本书一览”，它对本书中用到的主要数学规则、SIP 方法论、软件城堡模型作了简单的描述。读完本书之后，你就成了反复杂联盟中优秀的一员，面对各种难题，你都可以使用你定义的企业架构来简化它。

致 谢

在本书的编写过程中，我得到了很多人的帮助，包括评审人员、摄影师、生产人员等。

首先需要感谢的是 Beverly Bammel。他是 ObjectWatch 公司的主席，同时也是我工作上的亲密战友，我俩一起探讨复杂性管理的问题，一起为本书的出版而努力。

本书早期的草稿得到了众多评审人员的反馈，在此对他们表示衷心的感谢。其中有几位主要的评审我不得不提，他们分别是：

- Paulo Rocha，就职于新西兰 Fronde Systems Group，LTD 公司，企业架构经理。
- Darko Bohinc，就职于新西兰 Fronde Systems Group LTD 公司，策略服务首席顾问。
- Dan Schwartz，就职于 EDS 公司。
- Matt Peloquin，就职于 Construx 公司，首席技术官(Chief Technology Officer，CTO)。

和微软出版社(Microsoft Press)合作是一件非常愉快的事情。在此，我特别感谢 Ben Ryan(策划)、Lynn Finnel 和 Devon Musgrave(编辑)，还有 Roger Leblanc(版面文字编辑)。

在本书中用到了一些图片，原图片的摄影师或拥有者都乐于让我在本书中使用它们。这些图片分别是：

- 第 1 章中梭罗(Thoreaus)的小木屋(旧屋不复存在，此屋是仿建的)，这张照片是我的侄女 Laura Raney 拍的。她上次去瓦尔登湖(Walden's Pond)旅游时给我带回了这张照片。
- 第 2 章中魔方的照片在使用前得到了 Seven Towns LTD 公司的允许，该公司的网站为 http：//www. rubiks. com。
- 第 2 章中驾驶舱的照片来源于米格飞机大冒险(MIG Jet Adventures)网站，在使用前我经过了 Richard 和 Susie McDonald 的允许，他们的网址是 http：//www. migjet. com。
- 第 4 章中兰花的照片来自 Orchids of Wickford 网站，我从 N. Kingston，RI 那儿得到了授权，网址为 http：//www. wickfordorchids. com。

- 第 4 章中凤仙花的照片得到了 Red Planet, Inc 公司 Bruce Marlin 的授权。

除此之外，还有几位也为本书作出了贡献：

- 就职于 Neoris 公司的 Rodrigo Estrada，我们经常讨论 IT 系统的复杂性问题。
- Mainfreight 公司的首席信息官（Chief Information Officer，CIO）Kevin Drinkwater。本书中使用的不少材料来自我们平常讨论的话题。当然，在使用前我也征得了他的同意。
- 微软公司的 John Devadoss 和 Simon Guest，他们为我的白皮书的早期工作提供了很多支持。
- 位于英国威尔特郡的 Al Summer，我和他下过国际象棋，在征得他的同意后，我把我们曾经对弈过的棋局作为例子来讲解划分方面的内容。
- 伯明翰（Brenham）星巴克的咖啡师，感谢他们为我提供热情周到的服务。回想起来，好想再来双份浓缩咖啡、一份糖、其他配料，哦，对了，别忘了预热茶杯。

以上各位，真的很感谢你们。

法律声明

目 录

前 言
致 谢

第一部分 复杂性问题

第 1 章 当今企业架构 …… 2
为何困扰 …… 3
问题：不可靠的企业信息 …… 3
问题：不及时的企业信息 …… 3
问题：新的潜在复杂项目 …… 3
问题：并购新的公司 …… 4
问题：企业想分离部门 …… 4
问题：需要鉴别外包机会 …… 4
问题：规章制度要求 …… 4
问题：需要自动调节与外部伙伴的关系 …… 5
问题：需要自动调节与客户的关系 …… 5
问题：IT 部门和业务部门糟糕的关系 …… 5
问题：IT 系统糟糕的互操作性 …… 5
问题：难以管理的 IT 系统 …… 6
企业架构的价值 …… 6
一些定义 …… 6
什么是企业架构 …… 7
企业架构中的复杂性 …… 9
企业架构的 Zachman 框架 …… 14
开放组架构框架 …… 20
Federal 企业架构 …… 24
小结 …… 30
第 2 章 初识复杂性 …… 32
分而治之 …… 32
执行官的午餐 …… 32
合唱团排练 …… 33
紧急响应 …… 34
服装店 …… 35
国际象棋游戏 …… 35
孩子们在星巴克 …… 37
魔方 …… 37
分区的五条法则 …… 39
法则一：分区必须是正确的分区 …… 39
法则二：分区的定义必须恰当 …… 40
法则三：所划分的子分区必须合理 …… 41
法则四：分区中子分区大小应基本相等 …… 41
法则五：子分区间的相互影响必须最小化且明确定义 …… 42
简化 …… 43
迭代 …… 43
小结 …… 49
第 3 章 数学复杂性 …… 50
复杂性 …… 51
复杂性规则 …… 54
同态 …… 57
骰子系统中的复杂性控制 …… 57

增加桶 …… 59
分割 …… 61
等价关系 …… 63
等价类 …… 67
反等价关系 …… 68
等价关系和企业架构 …… 69
实践中的相互作用 …… 72
减少面 …… 73
减少桶 …… 75
其他复杂性 …… 76
理论和实践中的复杂性 …… 77
小结 …… 79

第二部分 简化问题

第 4 章 企业划分中的 ABC …… 82
复习数学知识 …… 82
分割企业 …… 83
企业等价类的 ABC …… 84
ABC 类型的关系 …… 85
实现和配置 …… 87
ABC 类型 …… 89
类型层次 …… 91
组合关系 …… 92
伙伴关系 …… 93
关系和分割简化 …… 95
再回到零售操作 …… 96
小结 …… 100
第 5 章 SIP 过程 …… 101
概述 …… 101
状态 0：企业架构评估 …… 102
问题：不可靠的企业信息 …… 102
问题：不及时的企业信息 …… 103
问题：新的潜在复杂项目 …… 104
问题：并购新的公司 …… 104
问题：企业想分离部门 …… 104
问题：需要鉴别外包机会 …… 105
问题：规章制度要求 …… 105
问题：需要自动调节与外部伙伴的关系 …… 106
问题：需要自动调节与客户的关系 …… 106
问题：IT 部门和商业部门糟糕的关系 …… 106
问题：IT 系统糟糕的互动性 …… 107
问题：难以管理的 IT 系统 …… 107
禁忌 …… 107
状态 1：SIP 筹备 …… 108
保证准备就绪 …… 109
培训 …… 109
管理模型 …… 109
SIP 混合 …… 110
企业相关的工具 …… 111
状态 2：分割 …… 111
状态 3：分割简化 …… 114
状态 4：ABC 优先级 …… 117
状态 5：ABC 迭代 …… 120
小结 …… 120
第 6 章 复杂性的案例研究 …… 121
概述 NPfIT …… 121
NPfIT 的当前状态 …… 124
SIP 途径 …… 127
小结 …… 136
第 7 章 警戒边界：软件城堡 …… 138
技术划分 …… 138
规则 1：自治 …… 143
规则 2：清晰的边界 …… 143
规则 3：功能分割 …… 143
规则 4：依赖定义 …… 144
规则 5：异步 …… 144
规则 6：数据分割 …… 145

规则 7：无交叉事务 …… 145
规则 8：单点安全 …… 146
规则 9：内部信任 …… 146
规则 10：保持简单 …… 147
小结 …… 147
第 8 章　复杂性展望 …… 149
复杂：真正的敌人 …… 150
值得简化 …… 151
简化的哲学 …… 154
本书内容回顾 …… 155
告别 …… 156
附录　本书一览 …… 158
数学内容 …… 158
分区的数学定义 …… 158
分区的五个法则 …… 158
骰子系统中的状态计算 …… 159
同态 …… 159
等价关系 …… 159
反等价关系 …… 160
分区 …… 160
等价关系的分割计算 …… 160
企业架构内容 …… 161
企业架构的合适定义 …… 161
理想架构的定义 …… 161
Boyd 迭代原则 …… 161
企业复杂性法则 …… 161
协作和自治 …… 161
SIP 内容 …… 162
SIP 定义 …… 162
SIP 过程 …… 162
ABC …… 163
软件城堡模型 …… 163
ABC 的三种通信方式 …… 164
SIP 信条 …… 164

PART ONE

第一部分　复杂性问题

本部分包括：

第1章　当今企业架构
第2章　初识复杂性
第3章　数学复杂性

第1章　当今企业架构

本书讲述怎样把企业架构做得更好。于是问题就来了，和什么相比更好？答案是比我们现在做得更好。我将详细讲述那些我认为非常重要、但在已有的方法论中却没有涉及的问题。当然，在所有的这些问题中，最重要的还是复杂性。

在我们讨论如何改进之前，你必须理解这门艺术（不妨把它理解成艺术）的当前状态。哪些方法论需要改进？这些方法论是如何阐述复杂性的？

大多数企业架构师多多少少都有些方法论的经验，但是能够在这个领域有很开阔视野的架构师却为数不多。本章中，我们将讨论为什么这个领域存在以及这个领域的当前状态。我将比较当前使用中的企业架构方法论的优劣点和相互之间的关系。

这些方法论中的每一种都为企业架构师的工具箱的实践性做出了重要贡献。尽管大多数人认为这些方法论是相互排斥的（你可以使用 Zachman、TOGAF 或 FEA），实际上它们是相互补充的，你应该知道在解决你手中的问题时，应该使用哪种方法。

在你知道一种方法的优点时，你还应该知道它的缺点。因为没有哪种方法在创建企业架构时能够提供一个完美的解决方案，甚至把所有的方法都加起来也不能提供一种完美的解决方案。这就是我写这本书的原因：弥补那些别人遗漏的东西。

我指的遗漏的地方就是管理复杂性。这些方法可以帮你理解业务过程，通过这些方法使其更好地服务于那些过程。当然，要想它们高效地工作，首先还得针对企业理顺这些方法的次序。这本书就是告诉你如何驯服你的企业达到理顺次序这个目标，这样它们才能够发挥高效的作用。

因而本章就是告诉你当前企业架构的状态。哪些东西工作，哪些不工作，哪些需要进一步完善。

为何困扰

创建企业架构对一个组织来说至关重要，它需要时间、资源和文化积累。组织为什么会困扰呢？

我将详细讲述在我的经验中，我认为哪些促使企业考虑创建企业架构的典型因素，一个成功的企业架构怎样通过这些因素给这个企业带来价值。如果所有这些企业架构的因素中，有你认为对你所在的组织有意义的，那么你就是实现这个企业架构的最佳人选。如果不是，你应该感到庆幸。

问题：不可靠的企业信息

企业依靠可靠的业务信息来做正确的商业决定。一个企业不能得知它的信息或者获取了不可靠的信息将导致一些问题。最好的情况，事后才做决定；最坏的情况，在不准确信息的基础上做决定。不论哪种情况，都是很严重的问题。

不可靠的信息带来的常见的一种结果就是不同的 IT 系统的数据重复，导致不同的业务过程之间无法有效合作。

一个企业架构可以帮助一个组织理解什么信息是不可靠的，它影响到哪些组织，解决问题需要哪些步骤。

问题：不及时的企业信息

企业不仅依赖可靠的信息（如前所述），而且还依赖企业信息的及时性。企业需要及时的信息来做出敏捷的决定。企业获取的信息不及时将导致做出过时的决定，这有点像下棋的时候你不知道对手上一步走的什么。这些企业将会发现自己很难与信息灵通的企业竞争。

不及时的信息往往导致频繁的人工干预，人工干预也就意味 IT 系统没有很好地与商业需求结合。

企业架构可以帮助组织理解如何更好地应用技术，尽可能地减少人工干预。

问题：新的潜在复杂项目

那些实施复杂 IT 项目的企业，在准备过程中通常涉及管理复杂问题（按道理应该是要涉及的）。如果创建一个新的、高复杂性的 IT 项目却没有完全理解它