

系统架构

复杂系统的产品设计与开发

[美] 爱德华·克劳利 (Edward Crawley) 布鲁斯·卡梅隆 (Bruce Cameron)
丹尼尔·塞尔瓦 (Daniel Selva) 著
爱飞翔 译

SYSTEM ARCHITECTURE

Strategy and Product Development for Complex Systems



机械工业出版社
China Machine Press



架构师书库

SYSTEM ARCHITECTURE

Strategy and Product Development
for Complex Systems

系统架构

复杂系统的产品设计与开发

[美] 爱德华·克劳利 (Edward Crawley) 布鲁斯·卡梅隆 (Bruce Cameron) 著
丹尼尔·塞尔瓦 (Daniel Selva)
爱飞翔 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

系统架构: 复杂系统的产品设计与开发 / (美) 爱德华·克劳利 (Edward Crawley) 等著; 爱飞翔译. —北京: 机械工业出版社, 2016.11

(架构师书库)

书名原文: System Architecture: Strategy and Product Development for Complex Systems

ISBN 978-7-111-55143-0

I. 系… II. ①爱… ②爱… III. 计算机系统 IV. TP30

中国版本图书馆 CIP 数据核字 (2016) 第 252211 号

本书版权登记号: 图字: 01-2015-5200

Authorized translation from the English language edition, entitled System Architecture: Strategy and Product Development for Complex Systems, 9780133975345 by Edward Crawley, Bruce Cameron, Daniel Selva, published by Pearson Education, Inc., Copyright © 2016 by Pearson Higher Education, Inc., Hoboken, NJ07030.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by Pearson Education Asia Ltd., and China Machine Press Copyright © 2017.

本书中文简体字版由 Pearson Education (培生教育出版集团) 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。

本书封底贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

系统架构: 复杂系统的产品设计与开发

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: 关 敏

责任校对: 殷 虹

印 刷: 中国电影出版社印刷厂

版 次: 2017 年 1 月第 1 版第 1 次印刷

开 本: 186mm × 240mm 1/16

印 张: 31

书 号: ISBN 978-7-111-55143-0

定 价: 119.00 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) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@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

.. 系统架构原则 ..

涌现原则（2.2.2 节）

当各实体拼合成一个系统时，实体之间的交互会把功能、行为、性能和其他内在属性涌现出来。

整体原则（2.4.3 节）

每个系统都作为某一个或某些个大系统的一小部分而运作，同时，每个系统中也都包含着更小的一些系统。

聚焦原则（2.4.4 节）

在任何一个点上都能发现很多影响系统的问题，而其数量已经超出了人们的理解能力。因此，我们必须找出其中最关键、最重要的那些问题，并集中精力思考它们。

二元原则（4.6.1 节）

所有由人类构建而成的系统，其本身都同时存在于物理领域和信息领域中。

受益原则（5.3.2 节）

好的架构必须使人受益，要想把架构做好，就要专注于功能的涌现，使得系统能够把它的主要功能通过跨越系统边界的接口对外展示出来。

价值与架构原则（6.1 节）

价值是有着一定成本的利益。架构是由形式所承载的功能。由于利益要通过功能而体现，同时形式又与成本相关，因此，这两个论述之间形成一种特别紧密的联系。

与特定解决方案无关的功能原则（7.1.2 节）

糟糕的系统规范书总是把人引向预先定好的某一套具体解决方案、功能或形式中，这可能会令系统架构师的视野变窄，从而不去探索更多的潜在选项。

架构师角色原则（9.2.1 节）

架构师的角色是解决歧义、专注创新，并简化复杂度。

歧义原则（9.2.2 节）

系统架构的早期阶段充满了歧义。架构师必须解决这种歧义，以便给架构团队定出目标，并持续更新该目标。

现代实践压力原则（9.3.2节）

现代产品开发过程是由同时工作着的多个分布式团队来进行的，而且还有供应商的参与，因此，它更加需要有优秀的架构。

架构决策原则（10.1节）

我们要把架构决策与其他决策分开，并且要提前花一些时间来谨慎地决定这些问题，因为以后如果想变更会付出很高的代价。

遗留元素复用原则（10.9节）

要透彻地理解遗留系统及其涌现属性，并在新的架构中把必要的遗留元素包括进来。

产品进化原则（10.9.3节）

系统必须进化，否则就会失去竞争力。在进行架构时，应该把系统中较为稳固的部分定义为接口，以便给元素的进化提供便利。

开端原则（11.2.1节）

在产品定义的早期阶段列出的（企业内部和企业外部的）利益相关者会对架构产生极其重大的影响。

平衡原则（11.3.4节）

有很多因素会影响并作用于系统的构想、设计、实现及操作。架构师必须在这些因素中寻求一个平衡点，使大多数重要的利益相关者得到满足。

系统问题陈述原则（11.4.2节）

对问题所做的陈述会确定系统的高层目标，并划定系统的边界。就问题陈述的正确性进行反复的辩论和完善，直到你认为满意为止。

歧义与目标原则（11.5节）

架构师必须解决这些歧义，以便提出几条有代表性的目标并持续地更新它们。这些目标要完备且一致，要兼具挑战性和可达性，同时又要能够为人类所解决。

创新原则（12.2.4节）

在架构中进行创新，就是要追求一种能够解决矛盾的好架构。

表面复杂度原则（13.2.2节）

我们要对系统进行分解、抽象及分层，将其表面复杂度控制在人类所能理解的范围之内。

必备复杂度原则（13.2.3节）

系统的必备复杂度取决于它的功能。把系统必须实现的功能仔细描述出来，然后选择一个复杂度最低的概念。

第二定律原则（13.2.3节）

系统的实际复杂度总是会超过必备复杂度。架构师要令实际复杂度尽量接近必备复杂度。

分解原则（13.3.1 节）

分解是由架构师主动做出的选择。分解会影响性能的衡量标准，会影响组织的运作方式及供应商的价值捕获潜力。

“2下1上”原则（13.3.1 节）

要想判断出对 Level 1 所做的分解是否合适，必须再向下分解一层，以确定 Level 2 中的各种关系。

优雅原则（13.3.2 节）

对于身处其中的架构师来说，如果系统的必备复杂度较低，而且其分解方式能够同时与多个分解平面相匹配，那么该系统就是优雅的。

架构健壮程度原则（15.5 节）

好的架构要能够应对各种各样的变化。能够应对变化的那种架构，要么是比较健壮的架构，要么是适应能力比较强的架构。前者能够处理环境中的变化，而后者则能够适应环境中的变化。

架构决策的耦合与整理原则（15.6.4 节）

可以按照指标对决策的敏感度以及决策之间的连接度来排定架构决策之间的先后顺序。

.. 译者序 ..

系统和系统架构，是两个让人觉得既熟悉又模糊的词语。说熟悉，是因为很多领域都频繁地提到它们；说模糊，则是因为不同的领域和不同的人，对这两个词的定义及范围，都有着不同的理解。而本书正是要从各领域对系统架构的理解和构建中，总结出一套共识，使大家能够更为清晰地看到复杂的产品在研发策略方面所体现出的共性。

本书第一部分讲解了什么是系统，其中主要强调整体思维的作用，并强调系统的整体功能必须大于各部件的功能之和。多出来的这些功能，就称为涌现物。

为了使系统涌现出良好的功能，我们需要对它进行分析。于是，第二部分就从形式和功能这两个方面讲解了如何分析系统，并且把系统架构这个概念描述为对形式和功能之间的映射关系所做的分配，同时，还强调了各部件之间的交互以及本系统与周边环境之间的互动。这些问题对于系统的价值来说至关重要。

在明确了系统和系统架构的含义之后，第三部分开始讲解如何创建良好的系统架构。产品上游和下游的影响因素，以及利益相关者所提出的各种需求，都是架构师必须要考虑的问题，因此，第11章开头给出了一套思维框架，使我们可以把庞杂的需求转化为凝练的目标，并确定这些目标之间的先后次序。接下来，讨论了怎样把这些目标汇聚成与特定解决方案无关的概念。之所以要先把特定的解决方案抛开，是为了使架构师能够把抽象的概念细化为多个片段，并运用创造力来探索它们的各种实现及组合方式，然后把那些能够满足客户需求的方式逐渐演化为具体的架构。由于很多系统本质上是非常复杂的，因此，在把概念演化为架构的过程中，架构师需要对系统进行分解，以看清这些组件的结构以及它们之间的交互情况。

同一个组件可能有多种实现方式，而组件之间的布局及交互情况，也是多种多样的，这就需要对其中较为关键的几个决策点做出选择。于是，第四部分从决策角度探讨系统架构。对每一个这样的架构决策来说，各选项之间的搭配方式可能极为繁多，因此，为了进行定量分析，我们需要根据一些衡量指标来构建权衡空间，并把各种候选架构都展示在该空间中的对应坐标上，以便使用优化算法找出优势较大的架构。

以上四个部分就是全书的主线。作者把系统架构的分析和综合过程中所提到的相关理念，总结成了20多条架构原则，使我们可以在系统架构的各环节中把握住重点，这也

是本书的一个特色。它的另外一个特色在于：作者对照建筑学中的模式语言，以及软件开发中的设计模式，把各种架构决策问题也归纳成 6 种模式，使架构师能够在不同的工作场景中选用合适的模式及算法来对这些问题进行程序化的处理。

本书提供了丰富的架构展示方法和架构分析公式，也给出了适用面较为广泛的原则和模式，在面对具体的项目时，我们应该根据自己的经验和判断力，灵活地运用这些工具。

翻译过程中，我得到了机械工业出版社诸位编辑和工作人员的帮助，在此深表感谢。

由于译者水平有限，错误与疏漏之处，请大家发邮件至 eastarstormlee@gmail.com，或访问 github.com/jeffreybaoshenlee/zh-translation-errata-sysarch/issues 留言，给我以批评和指正。

.. 推 荐 序 ..

Norman R. Augustine[⊖]

在医疗健康领域中，有一种趋势特别有前途，这就是生物医学研究与工程实践的结合。我有一位朋友是工程师，他最近告诉我，美国一家知名大学曾经开了这样一个会，工程系与心脏病学科系的教研人员，在会上探讨了这两种学科的结合方式。会议的重点是构建一颗可供人类使用的机械心脏，心脏病学科系的主管刚开始描述人类心脏的各项特征，就有工程师打断他，并问道：“机械心脏必须放在胸腔中吗？有没有可能放在其他更容易够到的地方？例如大腿中？”开会的人以前从来没考虑过这种可能。主管接着描述心脏的特征，但是过了一会儿，又有另一个工程师打断他，并提问：“能不能不要只放一个心脏，而是把三颗或四颗心脏组合成分布式系统？”这个问题，也是大家从未考虑过的。

本书由系统架构领域内三位备受崇敬的领军人物撰写，他们的观点很有见地。书中讨论的就是如何提出并回答上面那样的问题。我在工作中曾经碰到工程、商务、政府等方面的各种系统架构问题，当运用系统架构领域中的一些经验来解决这些问题时，我发现结果会好很多。

然而，单单运用这些经验是不够的。刚开始工作时，我记得自己总是问同事各种问题。当时我们正在“合作”完成一个导弹项目，我问他们为什么要采用某种特定的方式来设计产品的某个部件。有人给出的原因是“这种设计方式重量最轻”，有人说他设计的那一部分雷达横截面（radar cross-section）最小，还有人说自己设计的那个部件成本低、体积小，等等。

这些理由都不错，可是其中缺了一样东西。那就是系统架构师（system architect）。

系统架构师的缺位很常见，但表现方式一般比较微妙。几年前，我曾经参与了近音

⊖ Norman R. Augustine 在商业界曾经是洛克希德·马丁公司（Lockheed Martin Corporation）的董事长兼首席执行官（CEO），在美国政府中曾经担任陆军副部长（Under Secretary of the Army）。在学术界，他是普林斯顿大学（Princeton University）工程学院的成员，是麻省理工学院（MIT）、普林斯顿大学和约翰·霍普金斯大学（Johns Hopkins）的董事，也是马里兰大学系统（University System of Maryland，含有12所大学）的校务委员。

速运输机 (Near-Sonic Transport aircraft) 的早期研发工作。一份市场调查表明, 乘客想要更快地到达目的地。近音速运输的理念, 就是想使速度尽量接近音速 (也就是接近 1 马赫, 1 马赫 $\approx 1225\text{km/h}$), 但又不超过它, 以避免超过音速之后所引发的各种问题。然而空气动力学者 (我早前的研究领域就是空气动力学) 发现: 这样做使得飞机在阻力曲线上会进入一个耗油量陡增的区域。

从系统架构的观点来看, 我们要解决的问题并不是怎样飞得更快, 而是如何缩短乘客从家中到机场、办理登机手续、过安检、上飞机、飞行、落地后取行李并驶往最终目标所需的总时间。把这个问题放在刚才那个情境中考虑, 我们就会发现, 更基本的问题其实应该是: 节省这 5 分钟或 10 分钟的飞行时间, 究竟能给乘客带来多大的好处? 答案是: 带不来多大好处。因此, 这项近音速运输机计划就提前终止了。如果我们想使乘客更快地到达目的地, 那么显然还有其他更好的方案可供探索。这个项目之所以失败, 是因为大家没有意识到自己正在处理的是系统架构问题, 而不单单是空气动力学或飞机设计方面的问题。

这些年来, 我一直在不断地完善自己对“系统”这个词所下的定义。系统是“可以交互的两个或多个元素”, 而本书作者又明智地补充了一点, 那就是系统的功效必须大于各自元素的功效之和。尽管在概念层面很简单, 但是现实世界中的系统相当复杂。实际上, 对于由若干元素所构成的 (而且这些元素之间都以最简单的方式来交互) 系统来说, 描述其可能具备的状态数量所用的那个公式非常吓人, 有“怪兽公式”之称。此外, 很多系统里面还有人的因素在内, 如果系统里面还包括人, 那么系统架构所面临的困难就会更大, 因为人的因素会带来不确定性。我们在现实中遇到的就是这种系统, 本书作者要分析和解决的架构问题, 也针对的是这种系统。

我曾经分析过这样一个给美国南极站进行补给的系统。与其他系统一样, 我需要非常谨慎地设定具体的评估目标。是要缩减在可以预期的状况下所产生的消耗, 还是要减少在无法预期的最差状况 (例如糟糕的天气) 下所产生的消耗? 又或者是要尽量避免在补给品根本无法送达时所发生的那些“令人遗憾的”状况? 可以考虑的目标还有很多。

在这个系统中, 有很多个必须互相配合的元素, 例如运输船、破冰船、各种飞机、用于卸货的冰码头、存储设施、交通工具、通信等。而且在做各种决策时, 还要考虑架构中一直可能会发生的一种危险, 那就是单点故障。

我还在商务领域中遇到了一个比刚才更复杂的问题, 那就是能不能把 7 家公司的所有部门或主要部门, 合并为洛克希德·马丁公司, 如果能, 应该如何去做。这个系统中的每个“元素”都有其优缺点, 每个都涉及很多人, 都有自己的目标、能力和局限性。这项决策的关键在于, 合并之后的新公司, 其效能是否远远高于原有那些部分各自的效能总和。如果做不到这一点, 那就没有理由耗费资金去进行合并与收购了。

对于这一类复杂问题来说, 并没有简单的数学公式能够给出“正确”答案。然而, 系统思维 (systems thinking) 的训练是一项极为有用的工具, 可以帮助我们去评估系统

的观感、系统中潜藏的机遇以及系统对参数的敏感程度等。在刚才那个企业合并的案例中，大多数人都认为合并是“对的”，顺便说一下，在相似的案例中，有 80% 的情况也是如此。

我与本书的一位作者及其他同事，曾经向美国总统提出一项载人航天计划，这项计划是为将来的几十年而制定的。在这个案例中，最困难的地方是怎样合理地定义任务 (mission)，虽说确定适当的硬件配置也是个不小的工作，但与前者比起来，难度还是要低一些。所幸这种问题都可以通过系统思维来解决。

正如作者在书中所说的那样，系统架构的建立过程，既是科学，又是艺术。尽管这听起来很美好，但现实相当残酷。与达尔文式的物种进化现象一样，用过去的错误架构所搭建出来的系统是无法生存的；反之，用良好的架构所搭建的系统不仅可以生存，而且会越来越越好。

所谓复杂系统的架构，也就是这么一回事。

我们写这本书，是为了阐述一种强大的思想。越来越多的人已经开始有了这种思想，这就是“系统的架构”（architecture of a system）。从电网的架构到移动支付系统的架构，很多领域都出现了系统架构的思维。架构就是系统的DNA，也是形成竞争优势的基础所在。拥有系统架构师这一头衔的专业人士，现在已经超过10万人，此外还有更多的人以其他身份参与架构工作。

对于强大的思想，其边界一般都比较模糊。我们发现许多同事、客户和同学都能够意识到系统架构问题，但他们对这个词的用法有所区别。这个词一般用来区分两个已有的系统，例如“这两种山地自行车的架构不同”。

系统的架构到底是由什么组成的？这个话题通常会引发巨大的争论。在某些领域中，架构用来指代一项能够在抽象层面上区分两类系统的决策，例如“封包交换的架构”（packet-switched architecture）与“电路交换的架构”（circuit-switched architecture）。而在另外一些领域中，这个词则用来在忽略某些小细节的情况下描述整体的实现，例如“我们的软件是用来充当服务架构的”。

我们的目标是阐述架构思维的强大之处，并且使其边界变得更加明晰。架构思维的强大，源自它能够使我们在项目的早期阶段权衡各种架构、展望后续的发展情况，并发现各种约束以及对提升项目价值较为重要的机遇。如果架构把全部细节都包括进去，那么我们就无法在各种粗略的想法之间轻易跳跃，但如果架构中缺少了重要的价值驱动力，那它又显得没有意义。

我们写本书时所持有的理念与Eberhardt Rechtin相同，都认为架构师应该是专才，而非通才。我们想在书中描述系统架构的分析与构建过程，也想展示系统架构的“科学性”。与产品设计规范相比，本书的措辞在某种程度上较为宽松一些，因为我们要处理的系统更加复杂。产品研发人员所重视的是设计问题，而我们要强调的则是系统中的各个部件如何才能凝聚成一个连贯的整体，我们重视的是这个奇妙的涌现过程。

我们把过去的经验融入了本书中。我们有幸参与了许多复杂系统的早期研发工作，这些系统遍布通信、运输、移动广告、财经、机器人、医疗设备等各个领域，其复杂程度也各有不同，从农具到国际空间站，我们都接触过。

此外，书中的案例研究还涉及混合动力车 (hybrid car) 及商用飞机 (commercial aircraft) 等其他领域的系统架构师所总结出的一些经验。我们认为，本书必须要能够应对当前系统架构师所面临的各项挑战，因为只有这样，才能推进系统架构的发展。

本书的核心受众有两类人，一类是专业的架构师，另一类是工程学的学生。系统架构这一理念，是相关行业的从业者从实践和尝试中得来的，这些从业者运用自身的智慧，试着总结出一些经验，以应对研发新系统时所面临的挑战。本书的一部分目标读者是进行架构决策的资深专业人士。这些专业人士包括高级技术人员，也包括软件、电子、工业用品、航空、汽车及消费用品等科技产业中的管理人员。

本书的另一部分目标读者是工程学的学生。本书是根据过去 15 年间我们在麻省理工学院讲授研究生课程时的教学经验而写成的，其中有很多学生后来成了私人企业和政府部门中的佼佼者，对此我们深感荣幸。架构思维不仅可以帮助我们理解系统当前的运作方式，而且对于科技组织的管理来说，这也应该是一项必备的能力。

.. 致 谢 ..

我们要感谢使本书得以面世的诸位人士。首先，感谢 Bill Simmons、Vic Tang、Steve Imrich、Carlos Gorbea 和 Peter Davison，他们在本书的相关章节中提供了自己的专业意见，并对本书的初稿做了点评。Norman R. Augustine 为本书撰写了推荐序，并帮助我们成形系统架构方面的想法，为此我们深表感激。

感谢评审者 Chris Magee、Warren Seering、Eun Suk Suh、Carlos Morales、Michael Yukish 及 Ernst Fricke 给我们提供了明确的意见，并帮我们指出了未能传达出关键思想的那些段落。还要感谢很多匿名评审者，他们给出的反馈意见使我们能够对本书加以改进。感谢 OPM (Object Process Methodology, 对象过程方法) 的研发者 Dov Dori，他是我们的优秀合作伙伴。

感谢 Pat Hale 为我们在 MIT 的教学活动提供支持并对本书初稿给出反馈。感谢 MIT 系统设计与管理 2011 班 (MIT System Design and Management Class of 2011) 的 63 位同学详细阅读本书的每一章，并给出大量建议。尤其要感谢 Erik Garcia、Marwan Hussein、Allen Donnelly、Greg Wilmer、Matt Strother、David Petrucci、Suzanne Livingstone、Michael Livingstone 及 Kevin Somerville。感谢 MIT 图书馆的 Ellen Finnie Duranceau 帮我们明智地选择了出版社。

本书的编写得益于历届的研究生，他们所贡献的内容以各种形式出现在本书中。除了上面提到的那些人之外，还要感谢 Morgan Dwyer、Marc Sanchez、Jonathan Battat、Ben Koo、Andreas Hein 及 Ryan Boas。

感谢 Pearson 团队的 Holly Stark、Rose Kernan、Erin Ault、Scott Disanno 及 Bram van Kempen 为本书的出版所付出的辛勤劳动。

最后，感谢 Crawley 的妻子 Ana、Cameron 的妻子 Tess，以及 Selva 的妻子 Karen，感谢你们在周末和假期对我们著书工作的理解，使我们不至于把它拖成一本“永远都写不完的书”。

——Edward Crawley Bruce Cameron Daniel Selva

马萨诸塞州剑桥市

.. 作者介绍 ..

Edward F. Crawley

Edward Crawley 是俄罗斯莫斯科斯科尔科沃科学与技术学院的校长，也是 MIT 的航空航天学及工程系统学教授。他从 MIT 取得航空与航天专业的学士学位及硕士学位，并获得航空航天结构专业的博士学位。

Crawley 于 1996~2003 年担任 MIT 航空航天学系的主管。他与其他人共同主导了一项国际协作，以推动工程学教育的改革。Crawley 是《Rethinking Engineering Education: The CDIO Approach》一书的第一作者。Crawley 于 2003~2006 年担任剑桥-MIT 研究所 (Cambridge-MIT Institute) 的执行董事，这是由 MIT 与剑桥大学合办的机构，受到英国政府及业界的资助。该机构的目标是了解大学如何有效地发挥创新与经济增长引擎的作用，以及如何推广这种效用。

Crawley 博士创立了多家公司，其中包括产品研发与生产公司 ACX、生物分子探测器公司 BioScale、互联网广告投放公司 Dataxu，以及针对企业的能源投资组合分析公司 Ekotrope。2003~2012 年，他任职于轨道科学公司 (Orbital Sciences Corporation) 的董事会。

Crawley 教授是 AIAA (American Institute of Aeronautics and Astronautics, 美国航空航天学会) 及英国皇家航空学会 (Royal Aeronautical Society) 的会员，也是瑞典皇家工程科学院 (Royal Swedish Academy of Engineering Science)、英国皇家工程院 (Royal Academy of Engineering)、中国工程院 (Chinese Academy of Engineering) 及美国国家工程院 (National Academy of Engineering) 的成员。

Bruce G. Cameron

Bruce Cameron 是咨询公司 Technology Strategy Partners (TSP) 的创始人，也是 MIT System Architecture Lab 的董事。Cameron 博士从多伦多大学 (University of Toronto) 取得学士学位，从 MIT 取得硕士学位。

身为 TSP 的合伙人，Cameron 博士为系统架构、产品研发、技术策略及投资评估提供咨询服务。他曾在 60 多家高科技、太空、运输及消费品行业的财富 500 强企业任职，

其中包括英国石油公司 (British Petroleum, BP)、戴尔 (Dell)、诺基亚 (Nokia)、卡特比勒 (Caterpillar)、安进 (AMGEN)、威瑞森 (Verizon) 及美国国家航空航天局 (National Aeronautics and Space Administration, NASA)。

Cameron 博士在 MIT 的斯隆管理学院 (Sloan School of Management) 及工程学院 (School of Engineering) 讲授系统架构与技术策略课程。Cameron 博士曾经开办 MIT Commonality Study, 这是由 30 多家公司所组成的研究项目, 持续了 8 年。

Cameron 博士原来曾经在高科技企业和银行任职, 并构建了用来管理复杂研发计划的高级分析工具。在早期职业生涯中, 他曾经是 MDA Space Systems 的系统工程师, 并参与过一些航空设备的构建工作, 这些设备目前还在轨道中运行。他是多伦多大学董事会的前成员。

Daniel Selva

Daniel Selva 是康奈尔大学 (Cornell) 机械与航天工程系 (Mechanical and Aerospace Engineering) 的副教授。他从加泰罗尼亚大学 (Polytechnic University of Catalonia, UPC)、法国国立高等航空航天大学 (Supaero) 及 MIT 获得电气工程与航空工程学位。

Selva 教授的研究重点是在设计活动的初期运用系统架构、知识工程 (knowledge engineering) 与机器学习工具。他的研究成果运用于 NASA 的地球科学十年调查 (Earth Science Decadal Survey)、Iridium GeoScan Program 及 NASA 的跟踪与数据中继卫星系统 (Tracking and Data Relay Satellite System, TDRSS) 等项目, 在这些项目中, 他利用架构分析技术来为系统架构师和管理者提供支持。他也是 Best Paper 及 Hottest Article 奖项的获得者。

Selva 在 2004~2008 年就职于法属圭亚那 (French Guiana) 库鲁 (Kourou) 的阿利安太空公司 (Arianespace), 是阿丽亚娜 5 型火箭发射团队 (Ariane 5 Launch team) 的成员, 专门从事设备的数据处理与制导、导航及控制工作。他以前曾经在 Cambrian Innovation 公司研发供轨道卫星使用的新型生物机电系统, 并在惠普公司从事银行网络的监控工作。他是财富管理公司 NuOrion Partners 的顾问团成员。

.. 目 录 ..

系统架构原则

译者序

推荐序

前言

致谢

作者介绍

第一部分 系统思维

第 1 章 系统架构简介 2

1.1 复杂系统的架构 2

1.2 良好架构的优势 2

1.3 学习目标 5

1.4 本书结构 6

1.5 参考资料 7

第 2 章 系统思维 8

2.1 简介 8

2.2 系统与涌现 8

2.2.1 系统 8

2.2.2 涌现 10

2.3 任务一：确定系统及其形式与功能 13

2.3.1 形式与功能 13

2.3.2 工具 - 过程 - 操作数：这是人类的标准思维模式吗 16

2.4 任务二：确定系统中的实体及其形式与功能 16