

SysML精粹

SysML Distilled *A Brief Guide to the
Systems Modeling Language*

(美) Lenny Delligatti 著 侯伯薇 朱艳兰 译



机械工业出版社
China Machine Press

软件工程技术丛书

SysML精粹

SysML Distilled *A Brief Guide to the
Systems Modeling Language*

(美) Lenny Delligatti 著 侯伯薇 朱艳兰 译



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

SysML 精粹 / (美) 德里吉提 (Delligatti, L.) 著; 侯伯薇, 朱艳兰译. —北京: 机械工业出版社, 2014.11

(软件工程技术丛书)

书名原文: SysML Distilled: A Brief Guide to the Systems Modeling Language

ISBN 978-7-111-48458-5

I. S… II. ①德… ②侯… ③朱… III. VRML 语言 IV. TP312

中国版本图书馆 CIP 数据核字 (2014) 第 258286 号

本书版权登记号: 图字: 01-2014-2862

Authorized translation from the English language edition, entitled *SysML Distilled: A Brief Guide to the Systems Modeling Language*, 9780321927866 by Lenny Delligatti, published by Pearson Education, Inc., Copyright © 2014.

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 © 2015.

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(不包括中国台湾地区和香港、澳门特别行政区)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。

SysML 精粹



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: 谢晓芳

责任校对: 董纪丽

印刷: 北京市荣盛彩色印刷有限公司

版次: 2015 年 1 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 13.25

书号: ISBN 978-7-111-48458-5

定价: 59.00 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: hzjsj@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

本书赞誉

“Lenny 的这本书没有让我失望，它秉承了 Addison-Wesley 技术出版物的优秀品质。Lenny 所完成的工作非常精湛，他抓住了 OMG SysML 的精髓，那是一种实际可用、基于标准的建模语言，可以帮助系统工程师解决日益增长的系统复杂性的问题。这本书具有很重要的观点，首先介绍了基本的 MBSE 概念，以区别用例和场景之间的细微差别，然后阐明了命名空间以及 SysML 包，甚至还提到了不为多数人所知的 SysML 语义，如令牌流等。”

——Jeff Estefan, NASA 喷射推进实验室首席工程师

“像 SysML 这样的建模语言的强大之处在于，它不仅可以促进系统工程师之间的沟通，还可以促进跨规则以及跨开发生命周期的沟通。很多语言都具有改善沟通的潜力，但由于缺少有效的指导，它们都很难达到那个目的。在本书中，Lenny 把恰到好处的技术和一般通用方法结合，从而让 SysML 实现那种沟通。我在过去 30 多年间曾经参与过很多领域的系统和软件工程，并且还在很多组织和大学中讲授过计算机语言、UML 和 SysML，我发现 Lenny 的书籍是无价之宝。他清晰地展现了那些概念，并提供了有用且注重实效的实例，让你快速入门并能够成为有效的建模者。”

——Thomas W. Fagnoli, 洛克希德马丁公司工程技术领导成员

“这本书对 SysML 提供了非常棒的介绍。Lenny 的说明非常简洁且易于理解；例子非常透彻且有趣。”

——Susanne Sherba, 丹佛大学计算机科学系高级讲师

“Lenny 的这本书既可以作为 SysML 的参考书来查阅，也可以作为一本有趣的书来完整阅读，从而帮助我们学习这种语言。它是一本好书，秉承了著名的《UML 精粹》的风格。”

——Tim Weilkiens, oose 公司 CEO

“这本书比单纯的 PPT 具有更多的信息量，却没有 OMG 规格说明那么枯燥。它为从业的系统工程师提供了学习纯粹的 OMG SysML 所需要的动机、概念和标识法，

从而更流畅地使用图形化语言来编写规格说明，并分析他们复杂的实际系统。”

——Lonnie VanZandt, No Magic 公司首席架构师

“Lenny 的这本书是最适用的一本书；它浓缩了讲授和在业界环境中使用 SysML 的精华经验。作者为这种强大且复杂的建模语言提供了非常清晰且可读的视图，通过易于模仿的实际例子说明了它的用法。尽管最初它的目的是介绍 SysML，但我毫无疑问可以把它作为资深实践者的随身参考。”

——Bran Selic, Malina 软件公司总裁

“SysML 是一种让人望而生畏的建模语言，但在这本书里，Lenny 让它变得非常易于理解，而书中的建议会帮助实践者避免各种各样的陷阱，帮助他们领会并应用 SysML 的核心元素以及精神。如果你计划应用 SysML，那么这本书就是你所需要的！”

——Celso Gonzalez, IBM Rational 团队高级开发人员

“本书对于开始钻研基于模型的系统工程的工程师来说是一本非常棒的书。空间系统示例以简单而有效的方式记录了状况，并说明了概念。”

——Matthew C. Hause, OMG UPDM 小组首席顾问工程师

“从 20 世纪 90 年代开始我就深入参与到 OMG 中，但我的专业需求并没有让我过多接触 SysML 领域。所以我觉得自己应该是 Lenny 的书很好的 beta 测试员。让我高兴的是，通过阅读这本书我学到了很多知识，并且知道你也会一样。”

——Doug Tolbert, Unisys 的杰出工程师，OMG 委员会成员，架构委员会总监

“本书对基于模型的系统工程的语言组件提供了清晰且易于理解的描述，同时对于在哪里找到工具和方法组件相关的信息提供了建议。本书处处都能够证明，作者对 SysML 以及它在系统开发过程中的应用都有非常深入的理解。我肯定会在自己讲授的 MBSE 课程中使用它作为教科书。”

——J. D. Baker, OCUP, OCSMP, OMG 架构委员会成员

“本书是一本必备书，很多 SysML 建模者都应该在书架上放一本。Lenny 拥有丰富的经验和大量认证，能够帮助你解决日常的建模问题。这本书并不是一个教程，也不是关于 SysML 所有内容的百科全书。如果你使用 SysML 建模，那么它会成为你的

日常伙伴，那意味着你会经常用到它。我相信你的书很快就会有折角和笔记。”

——Robert Cloutier 博士，斯蒂文斯理工学院

“SysML 现在在很多领域都会得到应用，包括深度空间机器人飞船和深入地下的农业设备。这本书简洁地说明了 SysML，展现的方式既可以让新手快速上手，又可以让老手感觉非常方便。”

——Russell Peak，乔治亚科技空间系统设计实验室，MBSE 分部总监

“本书在系统建模领域是一本写得非常好、内容丰富并且简洁的书籍。简单易懂的说明能够带领新手为相当复杂的系统建立模型，而对 SysML 建模语言最常用部分丰富、有深度的阐述，使其可以适用于大多数系统的描述。它还可以作为手边的参考书。感谢 Lenny 先生给了系统建模领域这样好的一件礼物。”

——Bobbin Teegarden，OntoAge 公司 CTO、
首席架构师，No Magic 公司董事会成员

译者序

之前在对日软件项目开发的时候，项目组里面有各种各样的角色——PG、SE、PL、PM等。其中PG是编写代码的程序员（Programmer），而SE是系统工程师（System Engineer）。在那个时候，对SE这个职位的理解还不是很清晰，似乎他们更多的是在从事沟通和测试的工作。

当看到这本书的时候，我才发现系统工程师并不是那么简单。真正意义上的系统工程师要比当初那个职位的意义丰富很多。最重要的就是，系统工程师要创建并维护系统的模型。而软件项目中的系统局限在软件系统的范围之内，其实在很多领域中都会有系统，因此在更多的领域中都会有系统工程师这个重要的角色存在。

在敏捷还没有像现在这么流行的时候，我们的软件开发过程还是遵循瀑布式的流程，需求分析，概要设计，详细设计，编码，测试，一个一个步骤按部就班地进行，而且每个阶段都会有完备的文档作为产出物，下一个阶段会以上一个阶段的产出物作为输入。然而，相信经历过这类流程的朋友都会有过很惨痛甚至至于不堪回首的体会，那就是在出现需求变更的时候，我们不得不把所有的文档都修改一遍，当然，也包括我们作为最终产出物的代码。所以，那个时候，需求变更真的是所有项目成员的噩梦。

究其原因，就是因为当时对系统的所有记录都是基于文档的，所以每次变更都会导致所有文档的变更，我们不得不深入到每个部分去做修改，而即便如此，也难免会有遗漏和错误的情况发生。

本书介绍了和基于文档的系统工程完全不同的方式——基于模型的系统工程，这对系统工程师来说是一大福音，特别是对于需求变更非常频繁的软件行业的从业者来说更是如此。

书中首先就说明了基于模型的系统工程方法，并与基于文档的方法做了对比，这样就从根本上解决了出现变更时所需要解决的问题。当需求发生变更的时候，我们不再需要把所有文档都翻出来修改一遍，而是只需要修改一下模型，当需要文档的时候，都是从模型直接生成，这样就不需要到处修改了。

除此之外，书中还详细介绍了各种图，让读者可以在读过之后就了解如何创建SysML表述的模型，所有内容都不依赖于特殊的建模工具，所以读者不必担心对自

已选择的工具不适用。此外，图书的网站上还提供了适用于各种工具的示例，结合阅读这本书，读者可以更好地了解 SysML。

SysML 中的各种图和 UML 很类似，而本书的编写风格和经典的《UML 精粹》又非常类似，所以，如果你读过关于 UML 的那本著作，那么阅读这本书就会有驾轻就熟的感觉，由此也看出作者是多么体贴和用心。

总之，对系统工程师们来说，不管是属于软件系统领域还是属于其他领域，本书都非常有价值，我们既可以把它当成教程从头至尾仔细阅读，也可以将其作为手册放在手边随时拿过来作为参考。

愿系统工程师朋友们享受阅读本书以及使用 SysML 的快乐！

侯伯薇

序 一

系统工程这个学科很难讲授。在职业生涯早期，就有人向我强调，我们无法在教室中学习系统工程，而只能通过实践学习。尽管这种说法并没有得到明确验证，但在系统工程实践中，确实有些概念很微妙，而且晦涩难解。

在模型中表达这些概念需要一种合适而健壮的语言，于是我们这群人于 2002 年早期开始专门做这方面的工作，SysML 由此而来。在设计这种语言的时候，我们试图让它足够简洁和直接，特地把这种语言的目的设定为用于系统工程实践。我确信我们创建的语言既灵活又实用。我很高兴它已经成为沟通系统相关想法的领域标准。

然而，就像系统工程实践一样，我们发现 SysML 也很难有效地讲授。系统工程涉及的范围很广泛，尽管 SysML 是一种相对简洁的语言，但同学们还是经常会因为其复杂性而手足无措。学习 SysML 和基于模型的系统工程的资源现在还相对有限，但已经在逐步丰富。某些大学以及教育机构，已经在定期讲授正式的 MBSE 和 SysML 课程，而且现在已经有了至少一本综合性的教材。

但想要偶尔抽时间学习 SysML 基础知识的工程师和经理可能不会去上课。高级系统工程师也会忙于期限紧张的项目，而没有时间去上课。对于上述两种情况，这本书都价值非凡。

这本书和 Martin Fowler 的畅销书《UML 精粹》结构类似，它以清晰、简洁的术语介绍了 SysML 图的基础知识。它以轻量级的非正式方式编写，但传递了每种概念的主旨以及图形化的表示方法。关于这本书我最喜欢的一点就是，它会让我手不释卷，而不会陷入到“元说法”和“UML 主义”中。书中有很多有趣且切实可行的建议。

这并不是一本应用 SysML 或者部署 MBSE 的教材和指导书籍，也不会详细解释每个系统工程概念的方法论原理。尽管它没有在各个章节中使用一系列一致的例子，但也不会仅仅带着读者浏览任何特定的 MBSE 过程。它不是一本工作手册，没有包含问题，也没有让读者解决的示例练习。作为 SysML 的使用者或者高级的 MBSE 实践者，你可能最终需要其他资源，但这本书是个非常不错的开端。

这本书是一本有干货、可以自主学习、轻量级的 SysML 参考指南。世界对其充满期待。

Rick Steiner

《A Practical Guide to SysML》合著者

序 二

采用新技术要花时间

我很幸运能够在 20 世纪 70、80 年代技术及实业领域最好的大学之一读书。正如史蒂夫·乔布斯所说，未来是发明出来的，而不是发现出来的。在那所大学中，黑客们通宵达旦，帮助世界创建了雷达、闪光摄影术以及因特网。那些技术有助于改变世界；更重要的是，人们创建了很多公司和其他组织来使用那些技术，经济也随之繁荣起来。计算爆炸从 20 世纪 60 年代开始，到 1980 年在麻省理工结束。

我在学术领域最早的贡献是在麻省理工的 11 年间做出的，随着我在学术上的兴趣转移和改变，贡献也有了变化，它从人工智能领域开始（最早关注于手写识别技术），在研究生阶段转移到计算系统架构，最终是二者的融合。我在所关注领域做出的贡献，当我还在 MIT 的时候就已经被五家创业公司采用（尽管可能那是在 MIT 期间做出的一个大贡献），但不仅限于那些。人工智能先驱——像 Symbolics 和金山计算机——对我关于应用技术的理解都非常重要；我自己的创业公司，A.I. Architects（其中有我所见过的最好的系统工程师），也非常依赖于一种冲突，那就是对人工智能的需求和早期个人电脑有限的计算能力之间的冲突。

我在这段时间产生的最重要的认识可能是，使技术从实验室进入实际生产的时间花销，要比任何学术领域想象的还要多。20 世纪 80 年代的专家系统，现在是诊断系统以及世界范围内其他系统的主要补充（尽管一般来说要使用“基于规则的系统”的标记），很明显基于 20 世纪 60 年代 PLANNER 和 CONNIVER 之类的系统。20 年看起来像是弹指一挥间；让一种技术经过工程需求的检验，足够稳定，并在工业范围成为一种方法，然后再面向市场生产和整合，都需要花费大量时间。

OMG 敬畏的对象

不管怎样，当对象管理组织（Object Management Group, OMG）在 1989 年创建的时候，人们承诺对象技术和分布式对象将会改变计算领域的面貌。随着因特网慢慢

变成 3 维网，人们开始看清，一致、标准化的中间件对于让集成——不仅仅是世界范围内的文字页面，还包括应用程序的互操作性——更可能成为现实。“混合”全世界计算能力和数据源的能力（正如我们在 20 年后所说的），以及使用标准化 API 和在线协议，要比面向对象的方法更简单。

尽管 OMG 在最初做得不错，没有大肆宣传，避免了 20 世纪 80 年代由于过分宣传 AI 市场而导致的“人工智能的冬天”，但它对于采用技术需要花费的时间似乎认识不够。在“混合”真正成为现实之前可能需要 15 到 20 年，而面向对象语言（最初是 C++，它本身就是从 Simular 发展了 20 年的产物；现在是 Java、C# 和 Ruby-on-Rails）会渗透到计算领域中。像所有技术一样，OMG 敬畏的对象会成为世界各地软件开发中所使用的日常工具，但那会花费很多年。

建模造就内行

同时，OMG 发现了另一个机会。在 1996 年有人提出，面向对象分析和设计市场（那时就是这么叫的）已经到了穷途末路，因为没有基于继承技术，基于群集的方法（甚至更糟的标记法）在市场上已经泛滥成灾。即使是喜爱那些方法的技术专家也因为有太多选择而感到手足无措（一旦做出选择，很难保证可移植性和互操作性）。20 世纪 90 年代中期分析和设计市场的巩固，创建了以厂商为中心的市场力量，它们创建了标准，并被慢慢增长的用户社区所接受。在 1997 年它们创建了统一建模语言（Unified Modeling Language, UML）标准，尽管只有共享的标识法，而没有共享的方法，但足以在接下来的 15 年内让市场进入到超过百分之百的 CAGR。由于应用程序的生命周期不仅仅是分析和设计，还包括开发、测试、实现和维护，之前针对“分析和设计”的活动称为建模。

即便是使用像 UML 这样的早期标准，在几年内建模的重要性也显现出来了。早期的科学分析显示，使用建模方法（相对于低级的编程语言开发）提升了 35% 的生产力；可能更重要的是，在软件开发生命周期中维护和支持占据了 80% ~ 90% 的工作量，而一些关键的分析显示，那 35% 的生产力提升恰好在维护和整合方面。对这些标准的认可——在写这本书的时候，根据 Gartner & Forester 的市场分析，包括了 71% 以上的软件开发团队——使得在 OMG 出现了与建模相关标准的爆发性增长。

在 OMG UML 标准（及其相关的强大父标准，元对象工具（Meta Object Facility，

MOF)) 出现后的 15 年, 出现了很多标准化的领域特定建模语言。各种功能的语言, 包括定义在芯片上的系统、面向服务架构 (SoaML)、用于业务建模和分析 (BPMN)、记录企业架构 (UPDM)、定义基于规则的系统 (SBVR), 甚至是记录系统开发背后的动机 (BMM), 所有这些都加入了 OMG。更重要的是, OMG 的大多数工作都转移到“垂直市场”, 解决医疗信息技术、财务服务、生命科学、自动化以及其他消费设备依赖性的分析等问题, 所有这些都基于系统的视图, 而系统又基于高层次的模型。

为系统的散播提供服务

OMG 中最重要的一个成员就是系统建模语言 SysML。SysML 被定义为一种特殊的 UML, 作为能够完成多项任务的语言, 它可以整合很多大型系统工程的视图: 不仅仅是软件和硬件, 还包括需求、数学参数、工具管理、维护设计, 甚至包括对人力和其他资源, 以及设计中的系统行为的管理。我在 2001 年提出的愿景叫做模型驱动架构 (model driven architecture), 可能会通过这样的方法实现。这种方法可以整合各种工程, 而不仅仅是“软件架构”, 还包括复杂系统的总体架构, 如载人飞行器和化工厂等。就像 IDEF 系列规范在 20 世纪 80 年代早期承诺的一样, SysML 可以集成在众多需要它的领域中, 以构建设计良好、符合目标且可维护的大型系统。

从最早提出模型驱动架构到现在已经有 12 年, 从统一建模语言的发布到现在也快要 20 年了, 当前一书在手, 就可以整合各位专家关于如何思考并使用 SysML 来交付真正的系统方面的观点。现在我们有了这本书; 它把重要的内容以及最重要的概念都提取出来, 展现给大家。

复杂系统开发本来就是一项团队活动。单独一个人无法管理甚至无法搜集大型系统的需求; 规模本身就这样的系统非常复杂了。由于设计真正关注的是简洁性, 还有一个或多个维度, 我们需要标识法和过程, 不仅要沟通简化的愿景, 还需要让设计师、开发者和工程师深入到系统设计中, 并以不规则的方式探寻底层的设计部分、期望和需求以及整合方法。你要知道, 像 SysML 这样的标识法——大型且复杂, 当然包含各种的不同工具——可以支持大型系统开发。想要有效地使用那些工具, 需要努力地学习。我的岳父熟知如何使用螺丝刀来完成房子周围的手工任务 (包括敲钉子); 我更喜欢使用一些合适的工具, 并了解如何以集成的方式来使用那些工具。此外, SysML 建模语言不仅仅用于实现大型复杂系统, 还用来和系统的用户、维护人

员，以及可能会对系统进行调试和集成扩展、修正、变更的人来沟通设计。

这本书介绍了那些工具集；更好的是，它说明了如何一起使用多种工具，以获取需求、构件设计、分析设计并和其他团队中的其他人（你们可能在将来组成一个团队）沟通这个过程。那是工程师所做的工作，而 SysML 是完成那项工作的最好方式。

Richard Mark Soley 博士

OMG 总裁及首席执行官

前 言

为什么要写本书呢？很简单，因为忙碌的你需要了解 SysML。你已经有一些系统建模工作要做。你不需要知道语言的所有细节。你只是想要有本书能够让你专注于 SysML 在日常工作中最常用、最有用的部分。本书就是你想要的那本书。

你可以选择把本书作为必备参考书，当遇到难题并且快到最后期限的时候随手翻阅。或者你可以选择每次深入阅读一章，为即将到来的工作向你的工具箱添加新的建模技能。或者你可以选择逐页阅读，从而为 OMG 认证系统建模专家（OCSMP, OMG Certified Systems Modeling Professional）的前两级认证——OCSMP 模型用户和 OCSMP 模型构建者：基础——做准备。本书的设计可以满足你所有这些方面的要求。

本书读者对象

SysML 是一种图形建模语言，可以使用它来可视化并沟通各种规模的社会技术（sociotechnical）系统的设计——由硬件、软件、数据、人和过程组成的系统。系统工程师会负责对社会技术系统进行规范、分析、设计、验证和检验。因此，系统工程师以及系统工程的学生是本书的目标读者。

但这样说十分笼统。很多作者和老师都说过：一切都是系统。让我来补充一下：一切工程师都是系统工程师。不管你的领域和工作职责是什么，你都可能会执行我提到的某些或者全部系统工程任务。这本书的前提是，相对于通过非标准化的沟通模式，加上一系列文档和图，通过集成的 SysML 模型的标准化媒介，可以更有效地执行这些活动。你是一名系统工程师，你想要更有效地完成工作。因此你也是这本书的目标读者。

在阅读本书之前需要了解什么呢？你应该至少对系统叙述、分析、设计、验证和检验有概念性的了解。提前知道其中包括这些活动，会帮助你内化 SysML 能够帮助你更好完成这些活动的方式。国际系统工程委员会（International Council On Systems Engineering, INCOSE）的《系统工程手册》(Systems Engineering Handbook) 是本权威的参考书。

在阅读本书之前，你并不需要在建模语言方面有任何经验。你可能已经知道，SysML 基于统一建模语言 (UML)。事实上，你可能已经读过 Martin Fowler 的《UML 精粹》。我把本书设计为系统工程师的必备书，他们需要在软件系统 (创建 UML 的原因) 之外更广泛的领域进行系统建模。因此，你不需要了解 UML 对作为阅读本书的前提。本书的结构和内容使得阅读它本身就足以学习 SysML 了。

本书结构

本书包含 12 章以及 2 个附录。第 1 章介绍了基于模型的系统工程 (Model Based Systems Engineering, MBSE) 的概念，并提供了学习 SysML 的情境和业务案例。第 2 章讨论了为什么创建 SysML，并介绍了可以创建的 9 种 SysML 图。第 2 章还提到了能够应用在所有 9 种图上的通用概念。

第 3 ~ 11 章详细说明了每种 SysML 图的细节，介绍了可以在其中显示的元素和关系。尽管能够在这些图上显示的元素和关系种类偶尔有重复，但我还是每次在一章中专注于一个图，以有效地把相关的观点分组，帮助你在需要的时候更容易定位特定主题。第 3 ~ 11 章分别介绍以下 SysML 图。

- 模块定义图
- 内部模块图
- 用例图
- 活动图
- 序列图
- 状态机图
- 参数图
- 包图
- 需求图

最后一章也就是第 12 章，介绍了分配的概念——它是可以横跨所有 9 种 SysML 图关联元素的一种关系。

书中的示例图显示了 DellSat-77 卫星系统的各个方面，这完全是我为了撰写本书假设的一个系统 (在此我确保没有泄露任何航空公司的任何专利信息)。我选择一个卫星系统来演示可以如何使用 SysML 来建立复杂、现实的社会技术系统的模型——而不采用在建模工作坊中常见的典型模型 (ATM 和巡航控制系统)。本书通篇

选择使用唯一的系统作为运行的示例，也是为了展示 9 种类型的 SysML 图如何对底层的系统模型展现相互补充且一致的视图。

你可以进入我的网站 www.lennydelligatti.com，在 Articles and Publications 页面下载 DellSat-77 卫星系统的 SysML 模型。我分别以 XML 格式和各种建模工具的本地格式提供数据文件。这些资源让自学者、老师和学生能够选择自己的建模工具，实际操作本书中出现的系统模型。

附录 A 是书中图形表示法的简单总结，标明了对其进行详细讨论的章节。附录 B 介绍了 SysML v1.3 引入的元素种类，那也是在撰写本书时 SysML 的最新版本。

SysML v1.2 是当前 OCSMP 认证考试所采用的版本。SysML v1.2 和 v1.3 之间最大的区别在于端口——一种可以出现在模块定义图 (Block Definition Diagram, BDD) 和内部模块图 (Internal Block Diagram, IBD) 中的元素。第 3 章介绍了 BDD，第 4 章介绍了 IBD。在这些章节中，出于以下三种原因我专注于 SysML v1.2 的定义。

- ❑ 在 SysML v1.3 发布之前开始的建模项目中（这些项目很多现在仍然在进行中），它们是系统模型中主要的端口类型。
- ❑ 某些建模工具还没有跟上 SysML 的改变，还没有实现 SysML v1.3 中对端口的定义。
- ❑ OCSMP 认证考试在 SysML v1.3 发布之后还没有修订，仍然采用 SysML v1.2 中对端口的定义。

但是，不需要担心。附录 B 完整介绍了 SysML v1.3 对端口的定义。如果你的建模团队想要创建新的系统模型，那么我建议使用新的端口来替换旧的（假设你的 SysML 建模工具支持）。

章节的顺序大至基于图的一般使用频率。它并不反映每种图的相对价值，当然，它也无法反映。毕竟价值是很主观的东西，你的团队会基于所采用的建模方法以及为客户产出的交付物来决定。

章节的顺序也没有反映——也不应该建议——任何特定的建模方法。毕竟这不是一本方法学方面的书籍，而是一本语言方面的书。第 1 章讨论了建模方法和建模语言之间的区别。我列举了几种众所周知的建模方法，并给出了一些全面讨论它们的参考书。

本书的目标是，以简洁、专注目标的方式向你展示 SysML 最常用、最有用的特

性——不管采用什么建模方法都有用的特性。关键的一点是，SysML 只是一种语言，它与方法无关。我把本书也设计为与方法无关。我希望你能够知道，SysML 是用于沟通的具有附加值的媒介，不管你的团队为了完成工作、满足利益相关者的需求采用的是什么过程、流程和工具。

我希望你觉得这本书在学习 SysML 的过程中有用。它是一种丰富、利于表达的语言——具有足够的广度和深度，可以让你可视化并沟通系统设计的所有方面。要了解的知识还有许多，但你并不需要知道所有内容就可以创建能够清晰沟通的有效系统模型。阅读本书并获得你所需要的内容。你会发现如何快速把那些知识应用到工作中，并向你的客户交付价值。

致谢

很多聪明和敬业的人都为本书的出版付出了很多。我首先要特别感谢 Jim Thompson，他是我的朋友、同事、精神导师以及每周寿司学习班的搭档。在我撰写初稿的时候，他花了几个月来帮我审校，并且提供了重要、精辟的反馈。本书很大程度受益于他敏锐的技术头脑以及优秀的沟通技巧。

我要特别感谢 Addison-Wesley 出版社的 Chris Guzikowski。他带领我完成这个项目，让我这个菜鸟作者度过了最初的学习阶段。我特别要感谢他提出的宝贵建议：要不断精简，从而让我最终能够完成这本书。

Chris Zahn 是 Addison-Wesley 出版社的策划编辑，Betsy Hardinger 是一位优秀的文字编辑，他们为我提供了所需的特别支持，让这本书最终出版成册。他们教我把手好的想法转换成好书以及好的稿件的艺术。这本书的因为他们的贡献而质量优越。

Elizabeth Ryan 是 Addison-Wesley 出版社的项目编辑，她和产品团队协作，设计了这本书的版式，并把各个部分集中起来以便印刷。他们把复杂的过程变得简单，并创建了最终的产品。非常感谢他们辛苦的工作。

我还要深深感激由工程师和系统建模者组成的外部团队，他们是这本书的技术审校者：Celso Gonzalez、Robert Cloutier、Susanne Sherba、John Pantone、Michael Engle 和 Michael Chonoles。他们的专业能力和洞察力让我把粗糙的初稿，修改成更加专注的最终版本——它可以更好地为系统工程师社区服务。非常感谢他们。

我还要特别感谢下面这些人：Jeff Estefan、Susanne Sherba、Lonnie VanZandt、Bran Selic、J. D. Baker、Tim Weikiens、Tom Fagnoli、Robert Cloutier、Matthew