

计 算 机 科 学 丛 书



基于模型的测试

一个软件工艺师的方法

[美] 保罗·C. 乔根森 (Paul C. Jorgensen) 著

王轶辰 王轶昆 曹志钦 译

The Craft of Model-Based Testing

THE CRAFT OF
MODEL-BASED
TESTING



Paul C. Jorgensen



机械工业出版社
China Machine Press

计 算 机 科 学 从 书

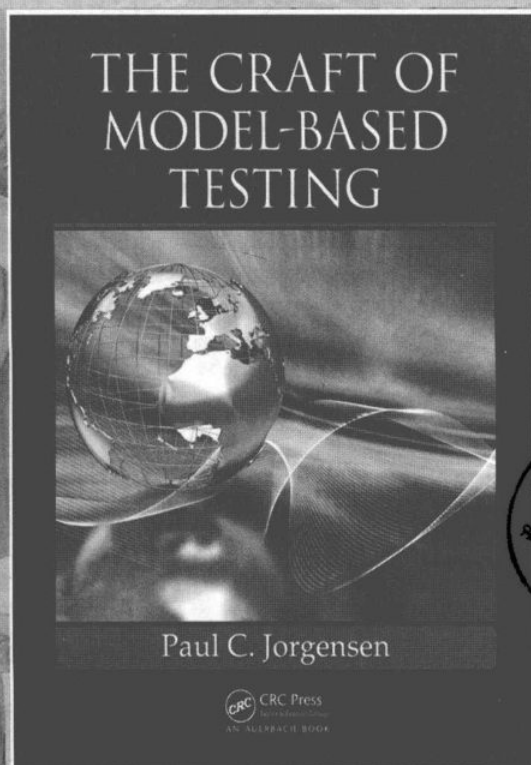
基于模型的测试

一个软件工艺师的方法

[美] 保罗·C. 乔根森 (Paul C. Jorgensen) 著

王轶辰 王轶昆 曹志钦 译

The Craft of Model-Based Testing



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

基于模型的测试: 一个软件工艺师的方法 / (美) 保罗·C. 乔根森 (Paul C. Jorgensen) 著; 王轶辰, 王轶昆, 曹志钦译. —北京: 机械工业出版社, 2019.7
(计算机科学丛书)

书名原文: The Craft of Model-Based Testing

ISBN 978-7-111-62898-9

I. 基… II. ①保… ②王… ③王… ④曹… III. 软件-测试 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2019) 第 110945 号

本书版权登记号: 图字 01-2018-4590

The Craft of Model-Based Testing by Paul C. Jorgensen (ISBN: 978-1-4987-1228-6).
Copyright © 2017 by Taylor & Francis Group, LLC.

Authorized translation from the English language edition published by CRC Press, part of Taylor & Francis Group LLC. All rights reserved.

China Machine Press is authorized to publish and distribute exclusively the Chinese (Simplified Characters) language edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Copies of this book sold without a Taylor & Francis sticker on the cover are unauthorized and illegal.

本书原版由 Taylor & Francis 出版集团旗下 CRC 出版公司出版, 并授权翻译出版。版权所有, 侵权必究。

本书中文简体字翻译版授权由机械工业出版社独家出版并仅限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) 销售。未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何内容。

本书封面贴有 Taylor & Francis 公司防伪标签, 无标签者不得销售。

本书是知名的“Craftsman”系列软件测试书籍中的新作, 主要讨论基于模型的测试 (MBT) 技术。第一部分讲解理论知识, 介绍了 9 种不同的测试模型。第二部分关注实践方法, 涵盖 6 个商用的 MBT 产品和 6 个开源工具。书中设计了两个贯穿各章的例子, 一个简单的保费计算系统, 一个事件驱动的车库门控系统, 以帮助读者深入理解建模过程和工具的应用技巧。

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: 蒋 越

责任校对: 殷 虹

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2019 年 7 月第 1 版第 1 次印刷

开 本: 185mm × 260mm 1/16

印 张: 17.5

书 号: ISBN 978-7-111-62898-9

定 价: 79.00 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 88379833

投稿热线: (010) 88379604

购书热线: (010) 68326294

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

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与 Pearson, McGraw-Hill, Elsevier, MIT, John Wiley & Sons, Cengage 等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出 Andrew S. Tanenbaum, Bjarne Stroustrup, Brain W. Kernighan, Dennis Ritchie, Jim Gray, Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman, Abraham Silberschatz, William Stallings, Donald E. Knuth, John L. Hennessy, Larry L. Peterson 等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方式如下：

华章网站：www.hzbook.com

电子邮件：hzsj@hzbook.com

联系电话：(010) 88379604

联系地址：北京市西城区百万庄南街1号

邮政编码：100037



华章科技图书出版中心

在艺术领域，每个亟待成名的人都希望自己能掌握“golden touch”（点金术），因为一旦有了这种本领，成功是迟早的事。但是在工程界，尤其是软件行业，我们连 silver bullet（银弹）都没有，只有“一万小时定律”。

本书的作者 Paul C. Jorgensen 是软件工程界的名人，相信很多软件测试的从业者都曾经读过他的著作《Software Testing: A Craftsman's Approach》。我们把编程人员称为码农，Jorgensen 将软件测试工程师称为匠人，两者并无本质区别。作为译者，我们是在中国从事软件测试研究和工程实践超过 20 年的匠人，其中一位是曾经的码农，在软件行业已经浸淫超过三万小时。

在翻译此书之前，我们一直在思考一个问题：基于模型的测试（MBT）这种理论性极强并且基本上还处于象牙塔里的技术，是如何撑起 20 章的？如何把带有泳道概念的 Petri 网，运用到软件测试项目当中呢？带着这样的疑问，我们开启了“边学习边翻译”的历程。

翻译这本书的时间远远超过了预期。并不是我们不能很好地将英文转换成中文，我们也不是拖延症患者，而是因为翻译这本书是一个非常赏心悦目的过程，让我们不由得放慢了脚步。

本书的前 10 章详细介绍了基于模型的测试的基本理论和方法，深入浅出，层层递进。阅读和翻译的过程，就好像跟随一位经验丰富的老师上了一学期带有实践的 MBT 课程。书中配备着表格和图的案例分析，能让 MBT 小白在短时间内了解 MBT 的前生今世。而本书的后 10 章则详细介绍了具有代表性的、支持 MBT 的商业工具和开源工具，读者可以按照前 10 章给出的各项 MBT 技术，在后 10 章寻找对应的工具。因此，不管是新手，还是已经在软件测试行业摸爬滚打多年的老将，都能从本书中有所收获。作为译者，我们力求使用“行话”而非绝对正确的“原话”来表达书中的技术内容。对于有些术语，我们加上了英文标注，一方面便于读者理解，另一方面读者也能够使用原文做进一步的查阅。

MBT 有很多毋庸置疑的优势，但它不是软件测试的银弹。学会了 MBT，也绝不是掌握了点金术。这本书带给我们的最大启发恰恰是“匠人”这两个字。本书作者以做木匠活为例，说明了在完成一件作品的过程中工具不是关键，了解物料本身的特质，因材施教，才是匠人的本领。

希望本书能够让读者在成为软件测试匠人的路上知己知彼，少走弯路。

首先是免责声明：我所使用的“工匠”和“工匠精神”这两个词是完全中性的，无意冒犯任何人。我相信，基于模型的测试（MBT）技术能够成为也应该成为一门手艺，而非艺术。工匠精神包含 3 个关键部分：对物料的深入理解，选择合适工具的能力，以及使用这些工具的经验。工具与手艺之间的关系是很有趣的，一个工匠就算使用很破旧的工具也能做出让人满意的产品，但一个新手就算使用精妙的工具也制作不出好产品。对于 MBT 这门手艺来说，这一点尤其如此。

除了软件测试之外，我个人最喜欢的手艺是木工活。作为一门手艺来说，木工需要了解物料，也就是木头。不同的木头有不同的特质，了解这些特质的木工才能做出正确的选择。枫木非常坚硬，需要非常锋利的工具；松木则很软，而且很容易塑型。我最喜欢的木头是樱桃木，它虽然不像枫木或者橡木那样坚硬，但是它有非常漂亮的花纹，而且好用。工具部分就更明显了。拿手锯来说吧，一个工匠可以有横切锯和粗木锯，有镶边手锯、轴锯箱、钢丝锯，也许还有些特制的日本锯用来进行更精细的切割。每一样工具都有某一种特殊的用途，没有一把锯能够符合所有的要求。但仅有工具是远远不够的。未来的工匠必须知道如何使用这些工具来达到自己的目的。此时经验就发挥作用了。在我看来，也许史上练就手艺的最好方法是学徒制，包括学徒期和熟练工时期，最终是大师级的工匠。整个学徒过程的核心是，在成为一个公认的、值得信赖的工匠之前，一个人必须经过长期的、受督导的学习历程。

上述内容与 MBT 有何关系呢？MBT 中，类似工匠的角色有哪些呢？物料，也就是被测软件或者被测系统。它们简单的区别在于，软件可能是转换类型的或者交互类型的。类型不同将影响如何选择合适的 MBT 工具。

MBT 工具包括用于描述软件的模型，在书中第一部分会涵盖这些内容。能够生成并且可能运行从模型派生出来的测试用例的商用或开源产品，会在第二部分介绍。第一部分首先简单介绍一些基本知识，第 2 ~ 10 章则分别讲解了 9 个模型，它们有不同的复杂度和表现力。有些非常有名，比如流程图和决策表。我们特别关注有限状态机，因为大多数的商用工具或者开源工具对其支持力度最大。第二部分展示了 6 个商用的 MBT 产品，最后一章简单描述了 6 个开源的 MBT 工具。

写作本书最大的挑战是如何传授经验。有两个贯穿全书的例子，保费计算问题是一个转换型应用的例子，车库门控系统是一个交互型应用（事件驱动）的例子。在第 2 ~ 10 章，我们用教学的方式对这两个问题予以建模。之后把这两个例子交给了 6 个商用工具，看看这些产品是如何支持这两个贯穿全书的例子的。所有 MBT 社区都承认，MBT 的成功在很大程度上取决于被测系统的建模好坏。因此第 2 ~ 10 章非常重要。

我父亲是一位工具和骰子制造商，他的父亲和祖父是丹麦的橱柜制造商。我外公是一位画家，我妻子是一位出色的厨师。我的家庭成员都将各自的才能作为手艺，并以各自的工作为荣。我相信，这种自豪感能够将普通的工作升华为手艺。我希望读者能够使用本书所展现的内容成为一个 MBT 的手艺人。

致 谢 |

The Craft of Model-Based Testing

非常感谢我的同事和研究生对本书所做的贡献。在 MBT 供应商中，感谢 sepp.med GmbH 的 Anne Kramer、Smartesting 的 Bruno Legard、Elvior LLC 的首席执行官 Andrus Lehtmetts、TestOptimal LLC 的首席执行官 Yaxiong Lin、德国不来梅大学的 Jan Peleska 教授，以及 RTtester 和 Conformiq 的首席技术官 Stephan Schulz。

还要感谢我在大峡谷州立大学（位于密歇根州的阿伦达尔）的研究生团队：Mohamed Azuz、Khalid Alhamdan、Khalid Almoqhim、Sekhar Cherukuri、James Cornett、Lisa Dohn、Ron Foreman、Roland Heusser、Ryan Huebner、Abinaya Muralidharan、Frederic Paladin、Jacob Pataniczak、Kyle Prins、Evgeny Ryzhkov、Saheel Sehgal、Mike Steimel、Komal Sorathiya 和 Chris Taylor。

在大峡谷州立大学，计算与信息系统学院院长 Paul Leidig 博士、帕德诺斯工程与计算学院院长 Paul Plotkowski 博士和大学教务长 Gayle Davis 博士都批准了我的休假，让我有时间完成这本书。

最后，对我的妻子 Carol 表示深深的谢意，感谢她过去几个月对我的耐心。

Paul C. Jorgensen 博士从事电话交换系统软件开发工作 20 年，这是他的第一份职业。他于 1986 年开始了大学教学工作，先在位于亚利桑那州坦佩市的亚利桑那州立大学教授研究生的软件工程课程，后于 1988 年在位于密歇根州阿伦达尔的大峡谷州立大学担任正教授。在从事学术工作之余，他还短暂从事过“软件范型”的咨询业务。他曾服务于数据系统语言会议 (CODASYL)、计算机协会 (ACM)、电气和电子工程师协会 (IEEE) 标准委员会。2012 年，他所在大学以“杰出学科贡献奖”表彰了他这一生的成就。

除了他的软件测试书籍《Software Testing: A Craftsman's Approach》(第 4 版) 之外，他还是《Modeling Software Behavior: A Craftsman's Approach》一书的作者，以及《Mathematics for Data Processing》(McGraw-Hill, 1970) 一书和《Structured Methods—Merging Models, Techniques, and CASE》(McGraw-Hill, 1993) 一书的合著者。最近，Jorgensen 博士参与了国际软件测试评定委员会 (ISTQB) 的工作，他与人合作编写了相关工作的高级教学大纲，并担任了 ISTQB 术语工作组的副主席。他是 ISTQB “基于模型的测试”教学大纲的审阅者。

在意大利生活和工作的三年，使他成为一个坚定的“Italophile”（喜爱意大利的人）。他和妻子 Carol 以及女儿 Kirsten 和 Katia 曾多次访问那里的朋友。自 2000 年以来，Paul 和 Carol 每年夏天都会在南达科他州 Pine Ridge 保护区的 Porcupine 学校做志愿者。他的大学电子邮件地址是 jorgensp@gvsu.edu，他在 2017 年夏天成为荣誉退休教授，也可以通过 pauljorgensen42@gmail.com 联系他。

出版者的话
译者序
前言
致谢
作者简介

第一部分 基于模型测试的模型理论

第 1 章 基于模型测试概述	2
1.1 基本术语	2
1.2 事件	3
1.3 测试用例	4
1.4 测试用例的执行框架	4
1.5 MBT 中的模型	4
1.6 ISTQB 中的 MBT 扩展	5
1.7 MBT 的形式	5
1.8 案例集	5
1.8.1 单元级问题：保费计算	5
1.8.2 系统级问题：车库门控系统	6
1.8.3 其他案例	7
1.9 MBT 的技术现状	8
参考文献	9
第 2 章 流程图	10
2.1 定义与表示法	10
2.2 技术详解	10
2.3 案例分析	12
2.3.1 日期计算函数	12
2.3.2 风寒指数表	12
2.3.3 保费计算流程图	13
2.3.4 车库门控系统流程图	13
2.4 基于流程图派生的测试用例	15
2.4.1 保费计算问题的测试用例	15
2.4.2 车库门控系统的测试用例	17

2.5 优势与局限	18
2.6 经验教训	20
参考文献	20

第 3 章 决策表

3.1 定义与表示法	21
3.2 技术详解	22
3.2.1 决策表的精简	22
3.2.2 有互斥条件的决策表	22
3.2.3 冗余和不一致的决策表	24
3.2.4 决策表引擎	24
3.3 案例分析	25
3.3.1 日期计算函数	25
3.3.2 汽车刮水器控制器	26
3.3.3 铁路道口门控制器	27
3.4 基于决策表派生的测试用例	28
3.4.1 保费计算问题的决策表	28
3.4.2 车库门控系统的决策表	30
3.4.3 车库门控系统的测试用例	31
3.5 优势与局限	32
3.6 经验教训	32
参考文献	33

第 4 章 有限状态机

4.1 定义与表示法	34
4.1.1 有限状态机的矩阵表达	35
4.1.2 有限状态机的文本表达	36
4.1.3 有限状态机的惯例与约束	36
4.2 技术详解	38
4.2.1 有限状态机的解释	38
4.2.2 有限状态机的实践	40
4.2.3 有限状态机引擎	40
4.3 案例分析	41
4.3.1 汽车刮水器控制器	41
4.3.2 铁路道口门控制器	43

4.4 基于有限状态机派生的测试用例	44	6.2.2 多原因输出事件	78
4.4.1 保费计算问题	45	6.2.3 事件静默	78
4.4.2 车库门控系统	46	6.2.4 事件驱动 Petri 网的引擎	78
4.5 经验教训	47	6.2.5 事件驱动 Petri 网的优势 与局限	79
4.6 优势与局限	48	6.3 案例分析	79
参考文献	49	6.3.1 铁路道口门控制器	79
第 5 章 Petri 网	50	6.3.2 汽车刮水器控制器	80
5.1 定义与表示法	50	6.4 基于事件驱动 Petri 网派生的测试 用例	82
5.1.1 可变迁与可点火	51	6.4.1 保费计算问题	83
5.1.2 惯例	51	6.4.2 车库门控系统	83
5.1.3 非图形化的表达方式	52	6.5 经验教训	89
5.2 技术详解	53	6.6 优势与局限	91
5.2.1 顺序、选择和循环	54	参考文献	92
5.2.2 可用、不可用和激活	55	第 7 章 状态图	93
5.2.3 触发	55	7.1 定义与表示法	93
5.2.4 挂起、恢复和暂停	55	7.2 技术详解	95
5.2.5 冲突和优先级	56	7.2.1 基于广播机制的交互	96
5.2.6 互斥	56	7.2.2 状态图引擎	96
5.2.7 同步	56	7.2.3 基于状态图派生的测试 用例	97
5.2.8 标记和可用序列	57	7.3 案例分析	98
5.2.9 Petri 网和有限状态机	58	7.3.1 铁路道口门控制器	98
5.2.10 Petri 网引擎	58	7.3.2 汽车刮水器控制器	98
5.3 案例分析	59	7.4 后续问题	99
5.3.1 生产者-消费者问题	59	7.4.1 保费计算问题	99
5.3.2 汽车刮水器控制器	61	7.4.2 车库门控系统	99
5.4 基于 Petri 网派生的测试用例	62	7.5 经验教训	104
5.4.1 保费计算问题	63	7.6 优势与局限	105
5.4.2 车库门控系统	65	参考文献	105
5.5 经验教训	69	第 8 章 泳道型事件驱动的 Petri 网	106
5.6 优势与局限	70	8.1 定义与表示法	106
参考文献	71	8.1.1 可变迁与可点火	106
第 6 章 事件驱动的 Petri 网	72	8.1.2 泳道型事件驱动的 Petri 网中 的事件	107
6.1 定义与表示法	73	8.2 技术详解	107
6.1.1 可变迁与可点火	74	8.2.1 使用泳道模型	107
6.1.2 惯例	76		
6.1.3 非图形化的表达方式	76		
6.2 技术详解	77		
6.2.1 上下文敏感输入事件	78		

8.2.2 “模型检验”	109	第二部分 基于模型测试的实践
8.2.3 基于泳道型事件驱动的 Petri 网派生的测试用例	110	第 11 章 国际软件测试评定 委员会
8.3 后续问题	112	11.1 ISTQB 组织
8.3.1 保费计算问题	112	11.2 认证等级
8.3.2 车库门控系统	112	11.3 ISTQB 的 MBT 大纲
8.4 泳道型事件驱动的 Petri 网派生的 测试用例	120	11.3.1 基于模型测试的简介
8.5 经验教训	120	11.3.2 基于模型测试的建模
参考文献	121	11.3.3 测试用例设计的选择标准
第 9 章 面向对象的模型	122	11.3.4 MBT 测试的实施与执行
9.1 定义与表示法	123	11.3.5 评估和部署 MBT 的方法
9.1.1 用例图	124	参考文献
9.1.2 活动图	125	第 12 章 在组织内实施 MBT
9.1.3 状态图	126	12.1 开始
9.1.4 顺序图	126	12.1.1 识别改变的必要性
9.2 案例分析	127	12.1.2 技术捍卫者
9.3 后续问题	127	12.2 起步
9.3.1 保费计算问题	127	12.2.1 候选的 MBT 产品
9.3.2 车库门控系统	128	12.2.2 成功标准
9.4 基于 UML 模型派生的测试 用例	138	12.2.3 试点项目
9.4.1 基于活动图的测试用例	138	12.3 培训与教育
9.4.2 基于用例的测试用例	138	12.4 经验教训
9.4.3 基于用例图的测试用例	139	12.4.1 物料
9.4.4 基于顺序图的测试用例	139	12.4.2 工具
9.4.5 基于状态图的测试用例	139	12.4.3 使用工具的能力
9.5 优势与局限	139	参考文献
参考文献	141	第 13 章 MBT 测试工具供应商 的信息
第 10 章 业务流程建模和标识	142	13.1 模板
10.1 定义与表示法	142	13.2 单元级问题: 保费计算问题
10.2 技术详解	143	13.2.1 问题描述
10.3 案例分析	143	13.2.2 问题模型
10.4 基于业务流程建模和标识定义 派生的测试用例	143	13.2.3 保费计算问题的程序代码 (VB 语言)
10.4.1 保费计算问题	143	13.3 系统级问题: 车库门控系统
10.4.2 车库门控系统	144	13.3.1 问题描述
10.5 优势与局限	144	13.3.2 问题模型

13.3.3 车库门控系统的程序代码 (VB 语言).....	167	第 17 章 Elvior 公司产品.....	210
第 14 章 Smartesting 公司的 Yest 和 CertifyIt 工具.....	170	17.1 简介.....	210
14.1 简介.....	170	17.1.1 Elvior 的 TestCast 工具集.....	210
14.1.1 产品架构.....	170	17.1.2 相关的测试服务.....	211
14.1.2 用户支持.....	171	17.2 保费计算问题的测试结果.....	211
14.2 使用 Yest 测试保费计算问题.....	171	17.2.1 被测系统建模.....	211
14.3 使用 CertifyIt 测试车库门控 系统.....	174	17.2.2 测试覆盖与测试生成.....	212
14.4 供应商的建议.....	180	17.3 车库门控系统的测试结果.....	215
参考文献.....	181	17.3.1 被测系统建模.....	215
第 15 章 TestOptimal 公司产品.....	182	17.3.2 测试覆盖与测试生成.....	215
15.1 简介.....	182	17.4 供应商的建议.....	219
15.1.1 产品架构.....	183	第 18 章 sepp.med 公司产品.....	221
15.1.2 TestOptimal 产品套件.....	183	18.1 简介.....	221
15.1.3 用户支持.....	184	18.1.1 sepp.med 概述.....	221
15.2 保费计算问题的测试结果.....	184	18.1.2 MBTsuite 概述.....	221
15.3 车库门控系统的测试结果.....	185	18.1.3 用户支持.....	222
15.4 供应商的建议.....	189	18.2 保费计算问题的测试结果.....	222
第 16 章 Conformiq 公司产品.....	190	18.2.1 问题输入.....	222
16.1 简介.....	190	18.2.2 生成的测试用例.....	225
16.1.1 产品特性.....	190	18.2.3 其他供应商提供的信息.....	229
16.1.2 Conformiq 360° 自动化 测试套件.....	192	18.3 车库门控系统的测试结果.....	229
16.1.3 用户支持.....	193	18.3.1 问题输入.....	229
16.2 保费计算问题的测试结果.....	193	18.3.2 生成的测试用例.....	230
16.2.1 Conformiq Creator 的输入.....	194	18.3.3 其他供应商提供的信息.....	234
16.2.2 生成的测试用例.....	195	18.4 供应商的建议.....	234
16.2.3 测试覆盖率分析.....	198	参考文献.....	235
16.3 车库门控系统的测试结果.....	203	第 19 章 国际验证系统公司产品.....	236
16.3.1 输入图和 QML 文本 文件.....	203	19.1 简介.....	236
16.3.2 生成的测试用例.....	204	19.1.1 RT-Tester 工具箱.....	236
16.3.3 追踪矩阵.....	206	19.1.2 基于模型的测试组件 RTT-MBT.....	238
16.4 供应商的建议.....	208	19.2 案例分析: 保费计算问题.....	242
参考文献.....	209	19.3 案例分析: 车库门控系统.....	244
		19.4 供应商的建议.....	248
		参考文献.....	250
		第 20 章 开源的 MBT 工具.....	251
		20.1 ModelJUnit 2.5.....	251

20.1.1	ModelJUnit 2.5 概述	251	20.4.1	Auto Focus 3 概述	264
20.1.2	利用 ModelJUnit 2.5 测试 车库门控系统	251	20.4.2	Auto Focus 3 使用方法	265
20.1.3	小结	254	20.4.3	小结	265
20.2	Spec Explorer	254	20.5	Graphwalker	266
20.2.1	Spec Explorer 概述	254	20.5.1	Graphwalker 概述	266
20.2.2	Spec Explorer 使用方法	254	20.5.2	Graphwalker 使用方法	266
20.2.3	小结	260	20.5.3	小结	267
20.3	MISTA	261	20.6	fMBT	267
20.3.1	MISTA 概述	261	20.6.1	fMBT 概述	267
20.3.2	MISTA 使用方法	262	20.6.2	fMBT 使用方法	268
20.3.3	小结	264	20.6.3	小结	268
20.4	Auto Focus 3	264	参考文献	参考文献	268

| 第一部分 |

The Craft of Model-Based Testing

基于模型测试的模型理论

基于模型测试概述

无论是软件还是硬件，甚至是日常生活中，所有的测试都可以视为系统因某个激励产生响应，然后对其进行检查的过程。事实上，早期的需求方法也主要是针对激励 - 响应进行研究的。在基于模型的测试 (MBT) 中，我们认为模型在某种程度上就是激励 - 响应的一种表达方式。其中有一些说明性的词汇有助于我们对这个问题的理解。

在软件测试领域，有 3 种被广泛接受的测试级别：单元测试、集成测试和系统测试。每个级别的测试都有明确的测试目标和测试方法。单元测试主要针对类或者过程进行测试，集成测试主要针对单元之间的交互进行测试，而系统测试则更加关注被测系统 (SUT) 的对外端口 (也称为端口边界)。每个级别的测试都有一个测试覆盖矩阵 (更详细的讨论请见 [Jorgensen 2013])，同时每个级别的测试用例都包含相似的要素：用例名称和标识 (ID)、前置条件、输入序列 (也可能是交互输入) 及预期输出、记录的实际输出、后置条件，以及通过准则。

1.1 基本术语

定义：被测系统通常缩写为 SUT，即将要被测试的系统。

SUT 可能是：一个由软件控制的硬件系统，一个仅有硬件的系统或者仅有软件的系统，甚至可能是由若干个 SUT 组成的大系统。SUT 也可以是一个单独的软件单元或者是由多个单元组成的集合。

定义：被测系统的端口边界，是被测系统中所有能够施加输入激励以及接收输出响应的端口集合。

无论是硬件、软件还是固件抑或是 3 种的组合，每一类系统都有端口边界。识别出 SUT “真正的” 端口边界对于基于模型的测试 (MBT) 过程而言是至关重要的。为什么说是 “真正的”？因为在测试过程中我们很容易将用户引起的物理事件与由此产生的电子信号 (激励) 相混淆。在基于 Web 的应用中，用户接口很可能就是系统级的输入和输出所在位置。在汽车刮水器控制器中，端口边界通常包括控制杆、决定刮水器速度的拨盘以及驱动刮水器叶片的电机。本书中还用到了车库门控系统 (后面会详述)，该系统的端口边界包括发送控制信号的设备、安全设备、末端传感器，以及一个驱动电机。一个单元的端口边界则是激活该单元的某种机制 (可能是面向对象软件中的消息，也可能是传统软件中的某个过程调用)。

定义：端口输入事件是针对给定 SUT 端口边界的一个激励。同样，端口输出事件是发生在 SUT 端口边界的一个输出响应。

在看待系统的端口输入事件上，开发人员和测试人员在观念上有明显的不同。测试人员的想法是针对 SUT 如何产生或引发一个输入激励，而开发人员考虑更多的是输入激励会导致软件产生什么样的行为和动作。这种区别也同样明显地反映在输出事件上，测试人员需要知道如何观察或者探知到系统的输出响应，而开发人员则关心如何生成或者引发输出响应。这些观念上的不同，有一部分是由软件开发团队构建的设计和开发模型所造成的。设计和开

发模型是从开发者的视角进行构建的，而测试用例的产生却是从测试人员的视角构建的。这个不同之处可能会在基于模型的测试中产生一定的影响。

1.2 事件

下面的术语基本上都是同义词，但略有不同：端口输入事件、激励、输入。同样，下面这些词也基本上算是同义词：端口输出事件、响应、输出。事件是一级一级发生的，当然，或许使用“按顺序发生”会更好些。我们来看车库门控系统（见 1.8.2 节详述），重点是包含光束传感器的安全设备。当车库门正在向下关闭的时候，如果任何事物阻断了光束（在靠近地板部分），那么电机会立刻停止并反向打开车库门。这个事件序列就是从某一个物理事件开始的，这个事件有可能是向下关门过程中有一个小动物恰巧穿过了光束。当光束传感器探测到中断时，它会给控制器发送一个信号。这就是端口输入事件，而且是一个真实的电信号。内部的控制器软件就会将此视为一个逻辑事件。

端口输入事件可能发生在不同的逻辑场景中。“一只猫穿过了光束”是一个物理事件，而这个物理事件有可能发生在好几种场景中，例如车库门已经打开的时候，正在打开车库门的时候，或者正在关闭车库门的时候。我们所关心的逻辑事件却只是发生在“正在关门”的时候。通常，事件发生环境在某些有限状态机（FSM）中表示为状态。在评估不同类型的模型对 MBT 的支持能力的时候，有一个很重要的指标就是该类型的模型能否表达和识别出对环境敏感的输入事件。同样，这也要求我们关注端口输入设备本身。假设一个测试人员想要测试光束传感器，尤其是其失效模式。通常的设备失效包括“在位置 1 失效”（SA-1）和“在位置 0 失效”（SA-0）。对于 SA-1 失效，光束传感器在发送信号位置失效，即保持一直发送信号的状态，无论物理输入事件有没有发生。请注意，在这种失效情况下是没法关上车库门的（请见下表中的 EECU-SA-1 用例）。而 SA-0 失效更加隐蔽一些，它表示光束传感器始终保持不发送信号的状态，这将导致即使物理中断发生之后，门也不会反向打开。我相信，如果律师得知在安全设备中会产生 SA-0 失效之后，那么他一定会非常郁闷。在第 8 章我们会对此进行建模。

用例名称	光束传感器在位置 1 失效
用例 ID	EEUC-SA-1
描述	用户尝试使用设备控制信号关闭一个已经打开的门。光束传感器发生 SA-1 失效
前置条件	1. 车库门开
	2. 光束传感器发生 SA-1 失效
事件序列	
输入事件	输出事件
1. 设备控制信号	2. 起动机向下
3. 光束 SA-1 失效	4. 停止并反向运转电机
5. 达到上行轨道终点	6. 停止电机
后置条件	1. 车库门打开
	2. 光束传感器产生 SA-1 失效

“在某个位置失效”这类故障以及其他失效模式是很难被定义的。它们可能不会出现在需求文档中。就算是在需求文档中有所说明，在很多基于模型测试的过程中也很难对其进行

建模。我们会在第 8 章详述这一点。用户有可能会提供类似 EEUC-SA-1 的用例吗？基于以往的经验，这是很有可能的。但在敏捷开发中，这就是个挑战了。

1.3 测试用例

对于一个测试用例而言，它有两种基本的形式——抽象的和真实的，有些 MBT 团队将后者称为具体测试用例。抽象测试用例通常是从一个形式化模型中派生出来的。何为抽象？意指输入通常是以变量方式来表达的。真实的（具体的）测试用例包含输入变量的真实数值，以及预期的输出数值。这两种形式都包括前置和后置条件。

1.4 测试用例的执行框架

图 1-1 是一个通用的自动化测试用例执行框架。它的基础是我的团队在 20 世纪 80 年代早期开发的一个针对电话交换机系统的回归测试项目。图中的计算机包括所有测试用例的处理器，这些处理器控制并观察测试用例的执行过程。测试用例使用简单的语言来描述，

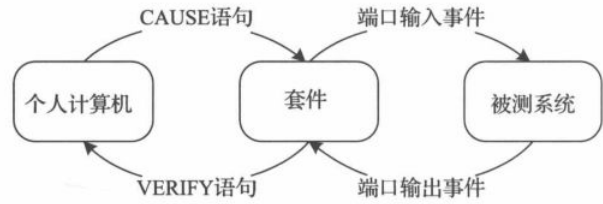


图 1-1 通用的测试执行框架

这个语言可以解释执行。语言中包括 CAUSE 和 VERIFY 语句，它们指代 SUT 的端口边界。CAUSE 语句一般都带有参数，它们指代端口输入事件和发生这些事件的设备（它们还可能带有附加参数）。同样，VERIFY 语句指代预期的端口输出事件。在电话交换机的 SUT 中，一个测试用例可能有以下两种语句：

```

CAUSE InputDigit(9) on Line12
VERIFY DigitEcho(9) on Line12
  
```

在这些语句中，InputDigit 指代 Line12 设备上发生的带有参数的端口输入事件以及该设备上发生的端口输出事件。能够实现这个框架的关键是开发一个能够连接 SUT 与测试用例处理器的“套件”。套件主要完成的工作包括以下内容。在 CAUSE 语句中，将端口输入事件从逻辑形式（抽象形式的用例）转换为物理形式；在 VERIFY 语句中，将端口输出事件从物理形式转换为逻辑形式。

以上内容主要针对系统级测试。在单元测试中，类似 nUnit family 的自动测试工具很常见，其中 CAUSE 和 VERIFY 语句被 ASSERT 语句替代。ASSERT 语句中包括被测单元的输入和输出，由此取代了系统测试中的测试套件。本书中我们忽略了集成测试，因为几乎没有 MBT 工具能够支持集成测试。本章结尾部分给出的 MBT 例子也只是包括了单元测试和系统测试。大家一定要记住，基于模型的测试要想成功，使用的模型必须能够提供激励和响应，不管它是单元级别还是系统级别。

1.5 MBT 中的模型

软件和系统设计模型通常包含两种类型——针对结构的模型和针对行为的模型。在通用建模语言（UML）中（UML 已经成为一种事实上的标准），针对结构的模型集中于类、类的属性、方法和类之间的连接（继承、聚合、相关）。另外，主要有两种针对行为的模型——状态图和活动（或者顺序）图。本书第一部分呈现了 9 种针对行为的模型：流程图、决策表、