

基于 UML 状态图的软件测试充分性准则研究

杨志伟¹, 吴 兵^{1,2}

(1. 中国航天系统科学与工程研究院软件工程研究部, 北京 100048;
2. 中国航天科技集团公司软件评测中心, 北京 100048)

摘 要:文中详细讨论了一组基于 UML 状态图的软件测试充分性准则。在软件测试时,还没有测试充分性准则明确定义循环应该被执行多少次,而循环只被执行一次是很不充分的。针对这种情况,在现有测试准则的基础上提出了两种扩展的基于 UML 状态图测试的充分性准则:ZOT-R 循环覆盖准则和全 ZOT-R 路径覆盖准则,丰富并完善了现有的测试准则体系。并且通过结合一个具体实例给出了这些覆盖准则在具体测试实践中的应用方法,探讨了这些覆盖准则之间的包含关系。

关键词:UML 状态图;软件测试;充分性准则

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2013)08-0043-04

doi:10.3969/j.issn.1673-629X.2013.08.011

Research on Adequacy Criteria for Software Testing Based on UML Statecharts

YANG Zhi-wei¹, WU Bing^{1,2}

(1. Department of Software Engineering Research of China Academy of Aerospace Systems Science &
Engineering, Beijing 100048, China;

2. Software Test & Evaluation Center of China Aerospace Science and Technology Corporation, Beijing 100048, China)

Abstract: Discussed a set of adequacy criteria for testing based on UML statecharts. In software testing, there is no adequacy defines cycle should carry out how many times, and cycle perform only once is not sufficient. Based on the existing testing criteria, put forward two expanded adequacy criteria based on UML statecharts: ZOT-R cycle coverage criteria and full ZOT-R path coverage criteria, enriching and completing the existing testing criteria system. Also produced the methods of applying the test criteria combined with a specific example, and discussed the relationship between these coverage criteria.

Key words: UML statecharts; software testing; adequacy criteria

0 引 言

为了发现软件中的错误,需要在软件的输入域中选择合适的用来运行被测软件系统,对其进行测试。测试用例是以发现错误为目的而精心设计的一组测试数据。但是,软件的输入域是非常庞大的,甚至是无穷的,工程上不可能采取穷举测试的方法。工程中的做法是选择输入域中的一部分数据对软件进行测试。那么,怎样来选择这部分数据呢?选择这部分数据的指导方针就是测试充分性准则。

软件测试的重点是找到判别测试用例集是否足够

充分的准则。软件测试无法确认程序中没有错误,不能保证程序的质量,但是通过测试发现程序中的错误,可以提高软件的质量,这在 J. B. Goodenough 和 S. L. Gerhart 的关于测试理论^[1]的文献中已经特别提出。文中综合以上论断,定义测试充分性准则(又称测试覆盖准则或测试准则, Test Adequacy Criteria):指导测试工程师开发设计有效的,且能够增强软件系统可靠性和功能正确性的测试用例的规则集称作测试充分性准则。

测试充分性准则在软件测试过程中起到非常重要

收稿日期:2012-10-18

修回日期:2013-01-22

网络出版时间:2013-04-08

基金项目:国家自然科学基金资助项目(91118007)

作者简介:杨志伟(1987-),男,浙江嘉兴人,硕士研究生,研究方向为 UML 建模和软件测试技术;吴 兵,高级工程师,硕士研究生导师,研究方向为软件工程、软件测试技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20130408.1559.016.html>

的作用,它可以衡量不同的测试用例集测试的质量和充分性;判定测试的有效性,决定什么时候可以停止测试;指导生成测试数据^[2]。

1 UML 状态图

在基于 UML 模型的测试中,状态图有着十分重要的应用。一个状态图是一个状态机,它刻画了对象状态变迁的控制流^[3]。在软件中,并不是每个类都有重要的,事件次序相关的行为,一般只为那些表现出这种行为的类画出状态图^[4]。

UML 状态图可以形式化地表示为一个五元组 $K < S, \Sigma, T, s_0, F >$, 其中, S 是非空状态集合, 状态可以是简单状态或复合状态; Σ 是非空输入集合; T 是迁移集合; s_0 是初始状态; F 是非空可接受终态集合。 T 是迁移 t 的集合, 迁移 t 是一个五元组: $(\text{SrcState}, \text{DstState}, \text{event}, \text{guard-condition}, \text{action})$, 其中, SrcState 表示迁移 t 的源状态, DstState 表示迁移 t 的目标状态, event 表示激发迁移的事件, guard-condition 表示监视条件, action 表示动作。

图 1 是一个简化的电子商务网站网上购物模块的状态图的例子。

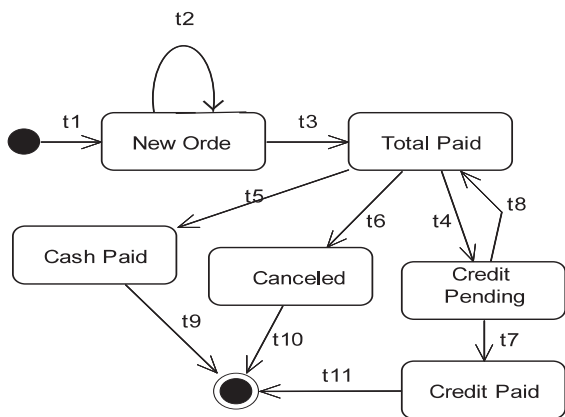


图 1 UML 状态图

2 基于 UML 状态图的测试准则

A. J. Offutt 结合较为成熟的结构测试准则,提出了四种基于状态图测试的测试准则^[5]。文中参考相关文献^[5,6], 定义了两种扩展的基于 UML 状态图测试的充分性准则: ZOT-R 循环覆盖准则, 全 ZOT-R 路径覆盖准则。对于文中定义的测试准则, 文中都结合图 1 的实例, 示范如何使用每一条覆盖准则, 产生测试路径, 设计测试用例集。这些测试用例集满足对应的覆盖准则。

2.1 状态覆盖准则

定义 1 状态覆盖准则: 执行测试用例集 TCS, 被测对象的 UML 状态图中的每一个状态应该至少被访问

一次。

为了满足状态覆盖准则, 需要设计的测试用例数量常常是最少的, 在基于 UML 状态图做测试的覆盖准则中, 它是最容易达到的, 但是, 它挖潜缺陷的能力有限。

生成满足状态覆盖准则的测试用例的算法是, 先对 UML 状态图做广度优先遍历, 生成一棵广度优先搜索树。然后对生成的广度优先搜索树中的每个叶节点, 找到其从根节点开始的路径。对于图 1, 得到 3 条测试路径: $t_1, t_3, t_4, t_7, t_{11}$; t_1, t_3, t_5, t_9 ; t_1, t_3, t_6, t_{10} 。

2.2 迁移覆盖准则

定义 2 迁移覆盖准则: 执行测试用例集 TCS, 被测对象的 UML 状态图中的每一个迁移应该至少被激发一次。

迁移覆盖准则兼顾了测试成本和测试充分性, 所以应用面较广, 很多面向对象软件的类层次的测试仅要求满足迁移覆盖准则。

为了满足迁移覆盖准则, 可以按如下算法来生成测试用例集。先生成一个测试用例集满足状态覆盖准则。然后检查没有被覆盖的迁移, 对每一个没有被覆盖的迁移做如下动作: 列出这个迁移的源状态和目标状态; 从已有的状态覆盖序列中找出一条以这个迁移的源状态为目标状态的迁移前缀序列, 找出一条以这个迁移的目标状态为源状态的迁移后缀序列; 按顺序把迁移前缀序列, 当前未覆盖的迁移, 迁移后缀序列组合形成一条测试序列。

2.1 节中的测试用例集没有激活迁移 t_2 和迁移 t_8 。对于图 1 所示的 UML 状态图, 在 2.1 节满足状态覆盖的测试用例集的基础上, 增加下面 2 条测试路径就能满足迁移覆盖准则: $t_1, t_2, t_3, t_4, t_7, t_{11}$; $t_1, t_3, t_4, t_8, t_4, t_7, t_{11}$ 。

2.3 全谓词公式覆盖准则

定义 3 全谓词公式覆盖准则: 执行测试用例集 TCS, 被测对象的 UML 状态图中的每一个迁移的谓词公式 F 的每一个主要谓词 P 取真和取假的情况下应该各测试一次。

为了满足全谓词公式覆盖准则, 可以按如下算法来生成测试用例集。先生成一个测试用例集满足迁移覆盖准则。因为一个迁移不含监视条件, 则它的谓词公式就是永真式, 只需要激发这个迁移一次, 就代表对这个谓词公式为真的情况测试了一次, 满足全谓词公式覆盖准则。然后检查每一个迁移, 对含有监视条件的迁移做如下操作: 以图 1 中的迁移 t_8 为例, 监视条件是“password wrong or money not enough”, 确定谓词公式 $F = P \vee M$, 分别以 P 和 M 为主要谓词, 计算得到全谓词公式覆盖测试的 3 组真值组合; 找到迁移 t_8 的源

状态,找出以迁移 t_8 的源状态为目标状态的迁移前缀序列,分别和迁移 t_8 的 3 组真值情况组合,形成 3 条测试序列。

2.4 迁移对覆盖准则

定义 4 迁移对覆盖准则:执行测试用例集 TCS,被测对象的 UML 状态图中的每一对邻近迁移 $\langle (S_i:S_j), (S_j:S_k) \rangle$ 都应该被测试一次。

考虑图 2,应用迁移对覆盖准则测试状态 Total Paid,共有 6 个测试需求,在测试用例集中应该包含六个迁移对: $T = \{ (t_3, t_4), (t_3, t_5), (t_3, t_6), (t_8, t_4), (t_8, t_5), (t_8, t_6) \}$ 。

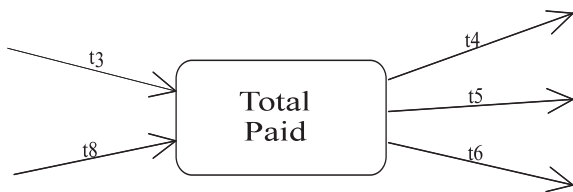


图 2 状态 Total Paid 的迁移对

2.5 ZOT 循环覆盖准则

定义 5 ZOT 循环覆盖准则:执行测试用例集 TCS,被测对象的 UML 状态图中的每一个循环应该分别执行 0 次,1 次,2 次的测试。

在测试含有循环的问题时,目前还没有一个测试准则明确规定循环执行的次数。一个循环执行两三次以后就把握了这个循环迭代的基本规律,所以测试循环就让循环执行 0 次、1 次、2 次 (Zero-One-Two)。根据不完全归纳法的思想,若循环通过 ZOT 测试,则它的可靠性会显著提高^[6,7]。

为了满足 ZOT 循环覆盖准则,按如下算法来生成测试用例集。先找出 UML 状态图中的循环结构;然后让每一个循环分别执行 0 次,1 次,2 次,生成 3 条测试序列。

在 UML 状态图中找出循环的方法是先以状态图的初始状态为根节点产生迁移树;根据迁移树可以得到从根节点开始到叶节点终止的由迁移组成的路径;检查每一条路径,如果在路径中出现了两个相同的状态,这两个相同的状态之间的迁移就构成了一个循环^[8,9]。

由状态图产生迁移树的算法是:定义状态为树的节点,迁移是树的边,初始状态为树的根节点;检查树中每一个非终端叶节点对应的 UML 状态图中的每一个离开的迁移和对应的目标状态,对应每一个迁移在树中画一条新的边和新的节点;如果新节点是一个终止状态或者新节点表示的状态已经存在于迁移树中,就把这个新节点标记为一个终端叶节点,它不用再画新的边;重复上述过程直到所有的叶节点都标记为终端叶节点。

下面利用 ZOT 循环覆盖准则来对图 1 所示的 UML 状态图生成测试用例。首先画出迁移树,如图 3 所示。

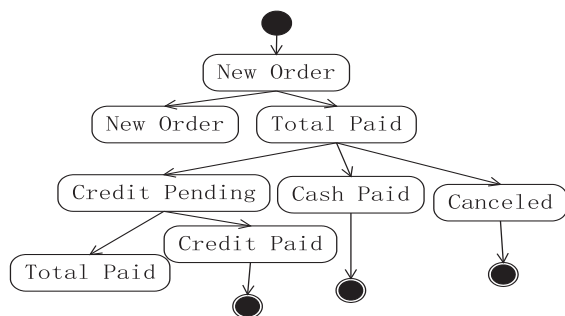


图 3 迁移树

得到两个循环: t_2 和 t_4, t_8 。考察循环“ t_4, t_8 ”。当循环执行 0、1、2 次,得到 3 条测试路径: $t_1, t_3, t_4, t_8; t_1, t_3, t_4, t_8, t_4, t_8$ 。同理,循环“ t_2 ”也有 3 条测试路径。

2.6 全 ZOT 路径覆盖准则

定义 6 全 ZOT 路径覆盖准则:执行测试用例集 TCS,被测对象的 UML 状态图中的每一个循环分别执行 0 次,1 次,2 次的测试,非循环的独立路径至少被访问一次。

覆盖 UML 状态图中所有的路径,这样的测试是不现实的。针对这个困难,提出了全 ZOT 路径覆盖准则,最大限度地压缩测试中需要覆盖的路径数量^[6]。

状态图中的每一条全 ZOT 路径都可以按如下方式分类:不含有循环结构和含有循环结构。全 ZOT 路径都要保证开始状态是初始状态,结束状态是终止状态。

下面利用全 ZOT 路径覆盖准则来对图 2 所示的 UML 状态图生成测试用例。

在 2.5 节已经给出 6 条测试路径,它们满足 ZOT 循环覆盖,使这 6 条测试路径以终止状态结束,成为全 ZOT 路径。另外迁移树中还有 3 条不含循环的全 ZOT 路径。因此,全 ZOT 路径共有 9 条。

2.7 ZOT-R 循环覆盖准则

在介绍 ZOT-R 循环覆盖准则之前,先来认识随机测试的概念。随机测试 (Random Testing),是指测试中所有的输入数据都是随机生成的,其目的是模拟用户的真实操作,并发现一些边缘性的错误。

有人也把随机测试叫做猴子测试,有这样一种想法:如果让一百万只猴子在一百万只键盘上敲一百万年,他们最终就有可能写出莎士比亚的巨著^[10]。当软件发布后,就会有成千上万的人像猴子一样对软件乱敲乱点。在软件发布之前,尽量模拟用户随机的操作,就有可能发现一些隐蔽的错误。

定义 7 ZOT-R 循环覆盖准则:执行测试用例集

TCS,被测对象的 UML 状态图中的每一个循环应该分别执行固定次数的 0 次,1 次,2 次的测试,还应该执行若干次(至少一次)随机次数的测试。

ZOT-R 循环覆盖测试含有两个参数形成的一个二元组, $\langle N, V \rangle$ 。N 代表的是“猴子”的数量,也就是模拟用户的规模,这个值应该由测试人员根据当前的测试条件、成本来控制,V 是一个随机值向量,向量的维数由 N 来动态确定。

生成满足 ZOT-R 循环覆盖准则的测试用例的算法是,生成满足 ZOT 循环覆盖准则的测试用例集;然后对每一个循环做如下动作:确定模拟用户的规模 N,生成一个 N 维的随机值向量 V,分别取随机值向量 V 中的随机值 x ,使循环执行 x 次的测试。

考察循环“ t_4, t_8 ”,生成满足 ZOT 循环覆盖准则的测试用例集,根据 2.5 节的结果,得到 3 条测试路径。接下来,假设模拟 3 个用户的随机操作,且取随机向量(5,3,8)。可以得到如下 3 条测试路径: $t_1, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_8; t_1, t_3, t_4, t_3, t_4, t_3, t_4, t_8; t_1, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_3, t_4, t_8$ 。所以总共对循环“ t_4, t_8 ”产生 6 条测试路径。

2.8 全 ZOT-R 路径覆盖准则

定义 8 全 ZOT-R 路径覆盖准则:执行测试用例集 TCS,被测对象的 UML 状态图中的每一个循环分别执行 0 次,1 次,2 次和一组随机值次数的测试,非循环的独立路径至少被访问一次。

A. J. Offutt 等人提出一条覆盖准则是“完全序列准则”。这条测试准则实践非常困难,它对测试工程师要求很高,需要他们具备丰富的经验和领域知识,然后能够从数量庞大的或者无穷多个迁移序列中选择出部分“有意义”的迁移序列^[11,12]。而文中提出的全 ZOT-R 路径覆盖准则能够产生一些非常有意义的测试路径,因为是模拟用户的实际操作而产生的路径,可以想象,如果模拟得越逼真,那么软件在用户使用过程中越可靠。

生成满足全 ZOT-R 路径覆盖准则的测试用例的算法是,生成满足 ZOT-R 循环覆盖准则的测试用例集,对其中的每条测试路径进一步完善,使之以终止状态结束,成为全 ZOT 路径;把迁移树中不含循环的全 ZOT 路径,加入到测试用例集中。采用 2.7 节的例子,循环“ t_4, t_8 ”产生 6 条测试路径,用同样方法循环“ t_2 ”也产生 6 条测试路径,另外迁移树中还有 3 条不含循环的全 ZOT 路径。所以图 1 例子中运用全 ZOT-R 路径覆盖准则一共产生了 15 条测试用例。

3 基于 UML 状态图测试准则之间的关系

如果满足测试准则 A 的任何测试用例集都满足

测试准则 B,则定义测试准则 A 包含测试准则 B^[6]。测试准则之间的包含关系具有自反性和传递性。

ZOT 循环覆盖准则不包含任何其它的覆盖准则,但和其他覆盖准则之间有交集。全 ZOT 路径覆盖准则包含 ZOT 循环覆盖准则。ZOT-R 循环覆盖准则只包含 ZOT 循环覆盖准则。全 ZOT-R 路径覆盖准则包含 ZOT-R 循环覆盖准则和全 ZOT 路径覆盖准则,它不包含全谓词公式覆盖准则与迁移对覆盖准则。

图 4 给出了这些测试准则之间的包含关系。

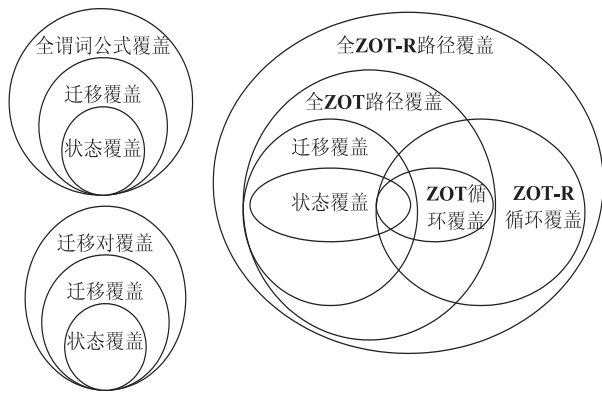


图 4 测试准则之间的关系

4 结束语

在测试时,循环的测试一直是个比较困难的问题,假如只测试循环执行一次,通常达不到充分性要求,而循环的所有次数都测试执行往往很难或者不可能实现。针对这个困难,已有学者提出了 ZOT 循环覆盖准则和全 ZOT 路径覆盖准则。但是这样执行固定次数的循环使得一些边界性的、随机性的缺陷永远没有被发现的可能,基于此文中在已有测试准则的基础上,引入随机测试的思想,提出两种新的测试准则 ZOT-R 循环覆盖准则和全 ZOT-R 路径覆盖准则,有效地模拟了用户的操作,使得测试行为更加接近用户实际操作。通过以上的讨论,基于 UML 状态图测试的充分性准则体系更趋完善,更加丰富。

参考文献:

- [1] Goodenough J B, Gerhart G L. Towards a Theory of Test Data Selection [J]. IEEE Transaction on Software Engineering, 1975, 3(6): 156-173.
- [2] 缪淮扣, 费立志. UML 状态图测试充分性准则的公理化评估[J]. 上海大学学报(自然科学版), 2007, 13(5): 489-496.
- [3] 高莉, 李龙澍. 基于 UML 状态图的测试技术研究[J]. 计算机技术与发展, 2009, 19(5): 68-71.
- [4] 张涌. 面向对象软件测试技术—类级别测试用例生成及回归测试研究[D]. 上海: 复旦大学, 2002.

算法存在的“标签饥饿”问题和确定型防撞算法的延时问题,而标签端仅引入一个特殊寄存器,对标签的成本影响几乎可以忽略。

参考文献:

[1] Finkenzeller K. 射频识别 (RFID) 技术[M]. 陈大才,译. 第 3 版. 北京:电子工业出版社,2005.

[2] 赵云青,徐文军,张晓华,等. 射频识别系统中读写器的设计[J]. 计算机技术与发展,2012,22(7):238-241.

[3] 朱 军,张 元,卢小冬,等. 基于分段搜索的多 RFID 标签抗冲突方法[J]. 计算机应用研究,2011,28(3):1031-1033.

[4] Chen Ying, Zhang Fuhong. Study on Anti-collision Q Algorithm for UHF RFID[C]//Proc. of International Conference on Communications and Mobile Computing (CMC). Hangzhou, China:[s. n.], 2010:168-170.

[5] 谢胜眉,赵军辉. 基于最优化原理的 RFID 系统中的 ALOHA 防撞算法研究[J]. 电路与系统学报,2009,14(4):8-12.

[6] Tao Cheng, Li Jin. Analysis and Simulation of RFID Anti-collision Algorithms[C]//Proc. of the 9th International Conference on Advanced Communication Technology. New York:[s. n.], 2007:697-701.

[7] Hush D R, Wood C. Analysis of Tree Algorithms for RFID Arbitration[C]//Proc. of IEEE International Symposium on Information Theory. Albuquerque, NM:[s. n.], 1998:107-109.

[8] Lai Yuancheng, Lin Chih-Chung. A Blocking RFID Anti-collision Protocol for Quick Tag Identification[C]//Proc. of International Conference on Wireless and Optical Communications Networks (WOCN). Cairo, Egypt:[s. n.], 2009:1-6.

[9] Law C, Lee K, Siu K Y. Efficient Memoryless Protocol for Tag Identification[C]//Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. Boston, Massachusetts, USA:[s. n.], 2000:75-84.

[10] Ryu J, Lee H, Seok Y, et al. A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID Systems[C]//Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc

Networking and Computing. Glasgow, Scotland:[s. n.], 2007:5981-5986.

[11] Choi J H, Lee D W, Lee H J. Bi-Slotted Tree Based Anti-collision Protocols for Fast Tag Identification in RFID Systems[J]. IEEE Communications Letters, 2006, 10(12):861-863.

[12] 张学军,王 娟,王锁萍. 基于标签识别码分组的连续识别防撞算法研究[J]. 电子与信息学报,2011,33(5):1159-1165.

[13] Kim Y H, Kim S S, Ahn K S. A Rapid Tag Identification Method with Two Slots in RFID Systems[C]//Proc. of the 8th IEEE International Symposium on Network Computing and Applications. Massachusetts, USA:[s. n.], 2009:292-295.

[14] Kim Y H, Kim S S, Lee S J. Improved 4-ary query tree algorithm for anti-collision in RFID system[C]//Proc. of International Conference on Advanced Information Networking and Applications. Bradford, United Kingdom:[s. n.], 2009:699-704.

[15] Myung J. Adaptive Binary Splitting for Efficient RFID Tag Anti-collision[J]. IEEE Communications Letters, 2006, 10(3):144-146.

[16] Kim S S, Kim Y H, Ahn K. An Enhanced Slotted Binary Tree Algorithm with Intelligent Separation in RFID Systems[C]//Proc. of IEEE Symposium on Computers and Communications. Sousse, Tunisia:[s. n.], 2009:237-242.

[17] Bonuccelli M A, Lonetti F. Tree Slotted Aloha: a New Protocol for Tag Identification in RFID Networks[C]//Proc. of International Symposium on World of Wireless, Mobile and Multimedia Networks. Washington DC, USA:[s. n.], 2006:603-608.

[18] Wu Haifeng, Zeng Yu, Feng Jihua, et al. Binary tree slotted ALOHA for passive RFID tag anti-collision[J]. IEEE Parallel and Distributed Systems, 2012, 23(3):1-14.

[19] Cui Yinghua, Zhao Yuping. A Modified Q-parameter Anti-collision Scheme for RFID Systems[C]//Proc. of International Conference on Ultra Modern Telecommunications & Workshops. St. Petersburg, Russia:[s. n.], 2009:1-4.

(上接第 46 页)

[5] Offutt A J, Xiong Y, Liu S. Criteria for Generating Specification-based Tests[C]//Proc. of 5th IEEE Intl. Conf. on Engineering of Complex Computer System. Las Vegas, Nevada, USA:[s. n.], 1999:119-129.

[6] 占学德. 基于 UML statecharts 测试方法的研究[D]. 上海:上海大学,2005.

[7] Weyuker E J. Axiomatizing Software Test Data Accuracy[J]. IEEE Transactions on Software Engineering, 1986, 12(12):1128-1138.

[8] 兰景英,王永恒. 基于 UML 状态图的测试场景生成法[J]. 计算机时代,2008(5):12-14.

[9] Offutt A J, Abdurazik A. Generating tests from UML specifications[C]//Proc. of the Second IEEE Intl. Conf. on the Unified Modeling Language. Fort Collins, CO:IEEE Computer Society Press, 1999:416-429.

[10] 赵 斌. 软件测试技术经典教程[M]. 北京:科学出版社, 2011:48-49.

[11] 张毅坤,施凤鸣,姚全珠,等. 基于 UML 状态图的类测试用例自动生成方法[J]. 计算机工程,2003,29(21):91-93.

[12] 江 曼,王天青,潘金贵. 基于 UML 状态图的面向对象软件测试用例生成[J]. 计算机科学,2006,33(6):284-286.

基于UML状态图的软件测试充分性准则研究

作者:	杨志伟, 吴兵, YANG Zhi-wei , WU Bing
作者单位:	杨志伟, YANG Zhi-wei (中国航天系统科学与工程研究院软件工程研究部, 北京, 100048), 吴兵, WU Bing (中国航天系统科学与工程研究院软件工程研究部, 北京 100048; 中国航天科技集团公司软件评测中心, 北京 100048)
刊名:	计算机技术与发展
	<div>ISTIC</div>
英文刊名:	Computer Technology and Development
年, 卷(期):	2013 (8)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjfz201308011.aspx