

应用 UML2.0 模型的测试用例生成方法

张琛^{1,2}, 段振华^{1,2}

(1. 西安电子科技大学计算理论与技术研究所, 710071, 西安; 2. 西安电子科技大学 ISN 国家重点实验室, 710071, 西安)

摘要: 针对软件开发过程中测试自动化程度低的问题, 在研究基于模型的测试用例生成技术的基础上, 提出了一种基于 UML2.0 序列图与用例描述的测试用例生成方法. 采用事件确定有限自动机来描述系统序列图, 通过命题投影时序逻辑的模型检测技术, 验证了自动机模型的正确性. 使用自动机模型与用例描述来生成测试用例, 该用例满足事件与全路径覆盖准则. 通过对图书管理系统的分析表明, 该方法不仅能够提高软件的测试效率, 而且还确保了针对管理员的执行动作所产生的测试用例的正确性.

关键词: 测试用例; 命题投影时序逻辑; 模型检测; 覆盖准则

中图分类号: TP311 **文献标志码:** A **文章编号:** 0253-987X(2011)08-0018-06

Test Case Generation Based on UML2.0 Models

ZHANG Chen^{1,2}, DUAN Zhenhua^{1,2}

(Institute of Computing Theory and Technology, Xidian University, Xi'an 710071, China;

2. State Key Laboratory of Integrated Service Networks, Xidian University, 710071 Xi'an, China)

Abstract: To improve the test automation in software development process, following the researches on test case generation technology from models, a novel method is proposed based on UML2.0 sequence diagram (SD) with use case description, and event deterministic finite automata (ETDFA) are employed to describe the SD models of system interaction. By the model checked with propositional projection temporal logic (PPTL), the correctness of ETDFA is verified. The achieved accurate ETDFA models and use case description are regarded as the origin of test cases. The testing strategy derives the test cases according to the event and full path coverage criterion. The experimental analyses of library management system (LMS) show that this approach enables to improve software testing efficiency, and guarantee the accuracy of the test cases for the administrator's action.

Keywords: test case; propositional projection temporal logic; model checking; coverage criteria

随着信息技术的持续发展, 软件的规模与复杂度也不断增加, 如何保证并提高软件的质量已成为软件工程中最为关心的问题之一. 软件测试能够有效地发现软件中的缺陷和故障, 是确保软件质量的关键技术. 随着面向对象软件开发方法的广泛应用, 以及对软件测试自动化的要求, 基于统一建模语言 (UML) 模型的测试方法逐渐得到了开发和测试人

员的认可与接受. 它是一种新兴的测试用例生成技术, 其模型以定义良好、功能强大、普遍适用的优点, 为基于模型的测试提供了非常好的条件. 同时, 降低了在面向对象软件的设计与实现过程中, 封装、继承、多态、绑定等特性对测试工作所带来的复杂性.

在软件测试过程中, 测试用例的选择决定了测试的有效性, 是测试工作的关键和难点. 因此, 本文

收稿日期: 2011-01-11. 作者简介: 张琛(1981—), 女, 博士生; 段振华(联系人), 男, 教授, 博士生导师. 基金项目: 国家重点基础研究发展规划资助项目(2010CB328102); 国家自然科学基金资助项目(60433010, 60873018, 60910004, 91018010, 61003078, 61003079); 教育部高等学校博士学科点专项科研基金资助项目(200807010012); 中央高校基本科研业务费专项资金资助项目(JY10000903004).

提出了一种使用 UML2.0 模型自动生成测试用例的方法. 将 UML2.0 序列图与用例描述共同作为产生测试用例的源, 所生成的测试集能够覆盖各类操作错误、交互错误以及场景错误. 本文采用事件确定有限自动机(ETDFA)来描述序列图, 对使用命题投影时序逻辑^[1](PPTL)描述的序列图的性质进行验证, 通过遍历正确的 ETDFA 得到的场景信息与用例描述中的信息, 来共同产生测试用例的集合.

文献[2]提出了一种通过将控制流和数据流分析技术相结合的基于状态图的测试用例生成方法, 该方法主要针对类测试, 不适合于集成测试与系统测试, 且缺少对状态图正确性的有效判断, 因此无法保证所生成的测试用例建立在对系统正确描述的基础之上. 文献[3]通过将状态图转化为展平正规式(FREE)模型, 然后按照 FREE 模型图的测试覆盖准则产生复合迁移序列, 并生成测试用例. 与文献[2]相同, 通过文献[3]所示方法产生的测试用例适用于面向对象的类测试, 但同样缺少对系统状态图正确性的有效验证, 即无法保证所产生测试用例建立在对需求正确表达的模型基础之上. 文献[4]介绍的基于 UML 协作图的集成测试用例生成方法, 将表示设计的协作图作为测试模型, 通过遍历每条消息的直接后继来识别协作图中所有可能的场景路径, 然后在遍历每条场景路径的过程中获取相应的协作执行信息, 从而得到覆盖场景路径的测试用例, 并用于测试一个协作场景路径上的交互行为. 协作图与序列图虽都用于描述对象之间的交互, 但序列图的重点是放在消息序列上, 而协作图则强调了空间. 因此, 基于序列图产生的测试用例更加适用于检查系统的操作错误、交互错误以及场景错误.

文献[5]介绍了一种从序列图中生成测试用例的方法, 将序列图转换为顺序图(SDG). 遍历使用来自用例模板、类图、数据字典等信息扩展后的 SDG 生成测试用例, 得到的测试用例适用于系统测试, 能够检测交互错误与场景错误. 文献[6]给出了由用例图和 UML2.0 序列图共同生成测试用例的方法, 从用例图和序列图中分别得到的用例依赖图和并发控制流图来产生测试序列, 用于集成测试与系统测试, 所得测试用例适用于检测用例与消息的同步与依赖、检测对象交互以及操作错误. 文献[5-6]均以序列图作为产生测试用例的基础, 但两者都未对序列图的正确性以及需求的等价性进行说明.

序列图强调了消息的时间顺序, 更适合作为测试用例生成的依据, 但从测试的角度观察, 该图中可

能存在一些错误, 其中包括约定的冲突、不能创建正确的对象、图中不存在关系的发送者和接收者之间的消息传递, 以及序列图是否满足需求说明等问题. 针对上述问题, 本文主要研究了通过模型检测的手段, 在获得正确表达 UML2.0 序列图语义模型的基础上, 使用经过验证正确的与序列图等价的自动机生成测试用例的方法.

1 序列图的有穷状态自动机模型

UML2.0 序列图添加了 loop、opt、alt、break、par、neg、ref 等多种组合片段^[7], 增强了在面向对象系统交互需求分析与设计中的建模能力. 结合 UML2.0 序列图和形式化方法的优点, 将非形式化的 UML2.0 序列图转换为具有精确语义定义的形式化规范, 在非形式化的图形表示与形式化定义之间建立映射关系, 从而为序列图的分析 and 验证奠定了基础.

1.1 UML2.0 序列图的语法

为了能够准确地描述序列图的动态交互过程, 抽象对象生命周期内事件发生的序列, 本文采用了 UML2.0 序列图的形式化定义, 其中包括事件之间的偏序关系, 以及序列图所增加的新特征.

定义 1 序列图 D 用一个十三元组表示, 即

$$D = (P, E, S, R, M, F, O_P, C, m_s, o_b, f_r, \rightarrow, <);$$
$$E = S \cup R, \quad S \cap R = \emptyset$$

式中: P 是有穷对象的集合; E 是有穷事件的集合; S 是发送事件集; R 是接收事件集; M 是有穷消息的集合; F 是组合片段的集合; O_P 是组合片段操作域的集合; C 是组合片段执行条件的集合; m_s 是从 E 到 M 的一个函数关系; o_b 是从 E 到 P 的一个函数关系; f_r 是从 E 到 F 的一个函数关系; $\rightarrow$ 是 M 上的一个全序关系; $<$ 是发送事件与接收事件之间的一个二元关系, 在 2 个对象之间, 发送消息事件先于接收消息事件发生. 图 1 给出了一个包含组合片段 opt 的序列图, 其形式化定义如图 2 所示.

1.2 序列图中单个对象的形式化定义

序列图表达了多个对象的消息交互, 将单个对象的形式化定义如下.

定义 2 序列图中对象 P_i 由六元组表示, 即

$$P_i = (E, F, O_P, C, n, f_r)$$

式中: n 是事件发生顺序的标号, 将第 1 个发生的事件标记为 1, 第 2 个发生的事件标记为 2, 并依次进行递推.

来验证序列图的正确性.

用 P_r 表示原子命题的集合,将 PPTL 公式归纳定义为

$$X ::=$$

$x \mid X_1 \vee X_2 \mid \neg X_1 \mid \bigcirc X_1 \mid (X_1, \dots, X_m) p_r Y$
其中 $X \in P_r, X_1, \dots, X_m$ 和 Y 均为 PPTL 公式. PPTL 公式包括的时序操作符有: $\bigcirc$ (next), (always) , $\Diamond$ (sometimes), $\odot$ (weak next), $;$ (chop), $(\text{chop in the past})$, prj (projection);常用的推导公式有: $\text{empty} \equiv \neg \bigcirc \text{true}$, $\text{more} \equiv \bigcirc \text{true}$, $\Diamond X \equiv \text{true}; X$, $\Diamond X \equiv X$; $\text{true}, \Box X \equiv \neg \Diamond ? \neg X$, $\odot X \equiv \text{empty} \vee \bigcirc X$, $\text{true} \equiv X \vee \neg X$, $\text{false} \equiv X \wedge \neg X$ 等.

2.2 验证步骤

使用模型检测器来验证序列图是否满足系统属性,需要将待验证性质用模型检测器可接受的形式表示,如果模型检测器得到该属性的反例,说明模型和性质是不一致的. 根据反例的路径可以找出序列图在建模过程中存在的问题,如果没有得到反例,则说明模型可满足序列图所表达的性质. 模型检测器作为一种自动化验证工具,具有验证速度快、效率高,以及可得到违反属性的执行路径的优点,具体验证步骤如图 4 所示.

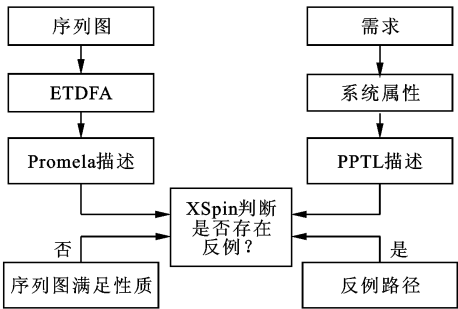


图 4 序列图验证流程

- 步骤 1 通过 ETDFA 构造算法建立序列图的自动机模型.
- 步骤 2 将 ETDFA 使用 Promela 语言描述.
- 步骤 3 序列图属性使用 PPTL 来描述.
- 步骤 4 将步骤 2、3 产生的模型与性质描述共同作为模型检测器的输入,使用已开发的基于 PPTL 的验证工具 XSpin^[9]输出验证结果.

3 测试用例生成

序列图表示了对象间存在的各类交互,因此测试用例集要求能够发现对象之间的方法调用和消息

传递顺序中可能出现的错误. 通过验证所得到的 ETDFA,若能满足性质要求,即可作为生成测试用例的依据. 遍历 ETDFA,得到从初始结点到终止结点的全部路径,该路径能够覆盖所有的交互以及消息序列.

3.1 覆盖准则

序列图描述了用例图中的不同事件流,包括正常流和异常流,每个事件流称为一个场景. 本文从可能存在的各对象上的操作错误、对象之间的消息交互错误,以及场景错误出发,使用事件覆盖准则与全路径覆盖准则,从遍历经过验证得到的 ETDFA 中获得相应的测试用例集.

定义 5 事件覆盖准则. 给定测试用例集 T 和 UML2.0 序列图 D ,使得 T 能够满足 D 中对象 P_i 上发生的事件至少执行一次.

定义 6 全路径覆盖准则. 给定测试用例集 T 和 UML2.0 序列图 D ,使得 T 能够满足 D 上产生的消息路径序列至少执行一次.

针对系统设计中隐含的操作错误,为了能够产生满足事件覆盖准则的测试用例,需要遍历序列图中对象所对应的 ETDFA,得到从初始结点到各个终止结点的所有路径,访问每条路径,获得相应的测试场景. 针对系统设计中可能存在的交互与场景错误,为了得到满足全路径覆盖准则的测试用例,遍历序列图各对象 ETDFA 的积,得到从初始结点到各个终止结点的所有路径,访问每条路径,获得相应的测试场景.

3.2 用例描述和生成算法

用例图是由参与者、用例以及它们之间的关系构成的图形化表达方式,而对于每个用例,可以通过用例描述所包含的信息对其进行补充与完善. 用例描述通常包含多方面信息,表 1 给出了一些常用的内容. 在生成测试用例的过程中,场景执行的前、后置条件以及期望的输出结果均来自用例描述.

表 1 用例描述

前置条件	执行用例之前系统必须要满足的条件
成功的输出	导致用例执行成功的输出结果
失败的输出	导致用例执行失败的输出结果
主要、次要参与者	在用例中扮演主要、次要成员的角色
后置条件	用例一旦执行后系统所处的状态
主要事件流	用例执行的主要流程
扩展事件流	用例执行过程中扩展的流程

在测试用例自动生成过程中,ETDFA 与用例

描述具有非常重要的作用,它们包含了产生测试用例所需要的信息.ETDFA 描述了场景与事件发生的过程,而用例描述则给出了场景发生的前置、后置条件以及期望的执行结果.

定义 7 测试集生成算法.令事件确定有限自动机 $A_t=(Q, C, E, \Sigma, \delta, q_0, F_s)$,测试用例集为 T .

算法:测试集生成算法.
输入:事件确定有限自动机 ETDFA.
输出:测试用例集 T .

步骤:①枚举所有从 ETDFA 初始结点到终止结点的路径,限定循环至多发生一次,每条路径对应于一个测试场景;②对任意场景,测试用例为前置条件,事件,事件约束,后置条件,期望的输出结果;③对于每个测试用例,场景发生的前、后置条件与期望的输出结果来自用例描述,事件以及事件约束对应于各状态上发生的事件和组合片段的执行条件.

4 实例分析

本节以图书管理系统(LMS)中部分功能模块片段的概要设计为例,来说明测试用例的生成过程.

4.1 LMS 的序列图描述

图书管理系统包括了用户管理、图书管理、借阅管理等多个功能模块.图 5 仅给出了该系统中个别功能模块设计的序列图.

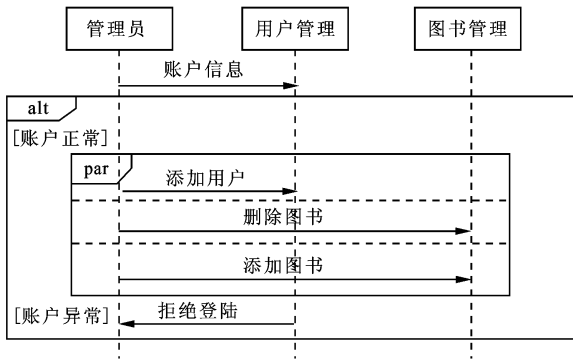


图 5 添加用户、删除图书及添加图书序列图

4.2 LMS 的自动机表示与验证

图 6 依次给出了图 5 所示序列图中管理员、用户管理、图书管理 3 个交互对象所对应的有限自动机.采用 PPTL 描述待验证的性质,如管理员正常登录后才可进行用户与图书的添加、删除操作,并且这些操作的执行可并行完成.下面,以 LMS 不允许用户在正常登录系统前处理其他操作为例进行说明.

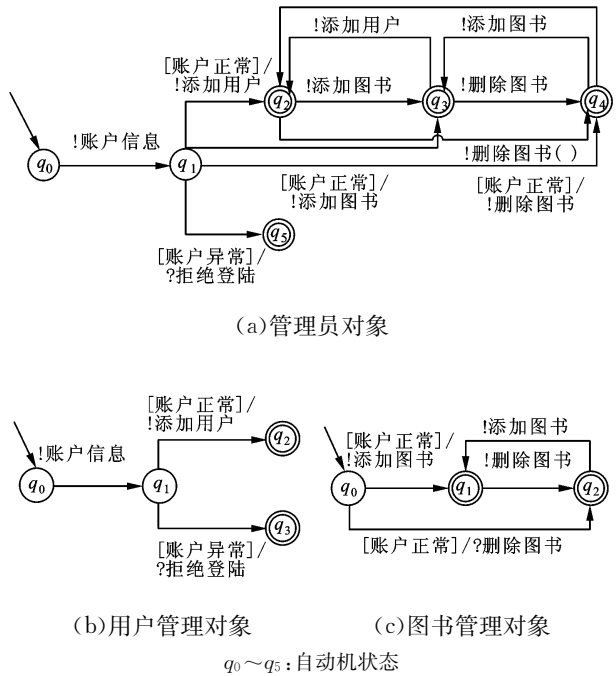


图 6 管理员、用户管理、图书管理对象分别对应的 ETDFA

PPTL 公式: $((\neg x) \rightarrow \bigcirc(y \wedge \text{empty}))$
其中 x 为真表示用户账户有效, y 为真表示系统拒绝登录.

将上述自动机的 Promela 描述与 PPTL 公式输入 XSpin 之后,可得到错误为“0”的输出结果,表示序列图满足该性质.在确认自动机正确表达了序列图所描述的系统属性后,方可使用自动机来完成测试用例的生成工作.

4.3 测试用例

针对序列图中管理员对象进行测试,说明路径、测试场景、测试用例的产生.按照事件覆盖准则,遍历图 6a 所示的 ETDFA,得到 7 条路径:①! 账户信息. [账户正常]/! 添加用户. ! 添加图书. ! 删除图书;②! 账户信息. [账户正常]/! 添加用户. ! 删除图书. ! 添加图书;③! 账户信息. [账户正常]/! 添加图书. ! 添加用户. ! 删除图书;④! 账户信息. [账户正常]/! 添加图书. ! 删除图书. ! 添加用户;⑤! 账户信息. [账户正常]/! 删除图书. ! 添加用户. ! 添加图书;⑥! 账户信息. [账户正常]/! 删除图书. ! 添加图书. ! 添加用户;⑦! 账户信息. [账户异常]/? 拒绝登录.

这 7 条路径对应于 7 个测试场景,以测试场景 1 和 7 为例说明相应的测试用例.场景 1:前置条件——管理员未登录;事件(发出)账户信息;事件约束——账户正常;事件(发出)添加用户;(发出)添加图书;(发出)删除图书;后置条件——可进行权限内

其他操作;期望的输出结果是系统正常,未出现不可执行的操作.测试用例场景7:前置条件——管理员未登录;事件(发出)账户信息;事件约束账户异常;事件(接收)拒绝登录;后置条件显示登陆提示信息;期望的输出结果是不可进行添加用户、添加图书、删除图书等操作.

5 结 论

通过形式化方法和技术对软件系统进行分析、验证、测试,是构造可信软件的一条重要途径,因此本文提出了一种基于 UML2.0 模型产生测试用例的方法,该方法具有如下优点.

(1)在采用基于 PPTL 的模型检测方法验证与序列图等价的自动机的过程中,通过验证设计模型与需求的一致性,提高了生成测试用例所需模型的正确性.

(2)通过测试集生成算法,改变了测试用例手动生成所造成的测试效率低的问题,从而提高了软件的测试效率.

(3)将系统设计模型作为产生测试用例的依据,可以把软件测试工作提前到开发过程的早期,在使用模型检测技术对序列图验证的同时,对需求的表达与设计的一致性进行有效确认,不仅提高了测试自动化的程度,而且也提高了所生成测试用例的可靠性.

(4)覆盖率是测试有效性的一个度量.本文从事件覆盖准则和全路径覆盖准则入手,采用序列图表达的消息交互,对应于自动机边上所发生的事件,使事件覆盖率等同于自动机上的边覆盖率.由于全路径覆盖率对应于自动机上的路径覆盖率,因此通过确定有限自动机能够简化测试场景的识别.

参考文献:

[1] DUAN Zhenhua. Temporal logic and temporal logic programming [M]. Beijing: Science Press, 2005: 6-93.

- [2] KIM Y G, HONG H S, CHO S M, et al. Test cases generation from UML state diagrams [J]. IEE Proceedings-Software, 1999, 146(4):187-192.
- [3] 缪淮扣,占学德,刘玲. 基于 UML Statecharts 的测试用例生成 [J]. 小型微型计算机系统, 2005, 26(4): 662-666.
- MIAO Huaikou, ZHAN Xuede, LIU Ling. Generating test case based on UML statecharts [J]. Journal of Chinese Computer Systems, 2005, 26(4):662-666.
- [4] 王林章,李宣东,郑国梁. 一个基于 UML 协作图的集成测试用例生成方法 [J]. 电子学报, 2004, 32(8): 1290-1296.
- WANG Linzhang, LI Xuandong, ZHENG Guoliang. An approach to generate integration test cases based on UML collaboration diagrams [J]. Chinese Journal of Electronics, 2004, 32(8):1290-1296.
- [5] MONALISA S, DEBASISH K, RAJIB M. Automatic test case generation from UML sequence diagrams[C] //Proceedings of the 15th International Conference on Advanced Computing and Communications. Piscataway, NJ, USA: IEEE Computer Society, 2007:60-65.
- [6] SANTOSH K S, DURGA P M, RAJIB M. Test case generation based on use case and sequence diagram [J]. International Journal of Software Engineering, 2010, 3(2):21-52.
- [7] KIM H, RUSSELL M. Learning UML 2.0 [M]. California, USA: O'Reilly, 2006:1-286.
- [8] DUAN Zhenhua, TIAN Cong, ZHANG Li. A decision procedure for propositional projection temporal logic with infinite models [J]. Acta Informatica, 2008, 45(1): 43-78.
- [9] 杨琛,段振华. PPTL 模型检测器实现的一个关键技术 [J]. 西安交通大学学报, 2010, 44(10):24-29.
- YANG Chen, DUAN Zhenhua. A key technique to develop the model checker for propositional projection temporal logic [J]. Journal of Xi'an Jiaotong University, 2010, 44(10):24-29.

(编辑 管咏梅 武红江)