

扩展 UML 状态图在软件测试中的应用

徐博¹, 樊晓光¹, 田涛²

(1. 空军工程大学空军工程学院航空电子工程系, 西安 710038; 2. 空军装备软件测评中心, 北京 100076)

摘要:在可测试性方面, UML 状态图模型存在不完整性和二义性及结构复杂不便于状态分析等缺陷。该文讨论了基于扩展 UML 状态图模型的测试用例生成技术, 分析了基于扩展 UML 状态图模型的测试用例生成技术。针对 UML 状态图的不完整性和二义性进行扩展, 以提高其可测试性。

关键词:扩展 UML 状态图; 可测试性; 测试用例

Application of Extended UML Statechart in Software Test

XU Bo¹, FAN Xiao-guang¹, TIAN Tao²

(1. Department of Avionics and Electronic Engineering, Engineering Institute, University of Air Force Engineering, Xi'an 710038;

2. Airforce Software Testing and Evaluating Center, Beijing 100076)

【Abstract】 Generation Technology of executable tests from behavioral models such as UML Statecharts offers benefits such as systematic test and test adequacy. This paper attempts to lay a new foundation for UML Statechart based test generation by introducing extended UML Statechart model that can be used by testers in the test phases just as the conventional UML Statecharts are required during the design and development phases.

【Key words】 extended UML statechart; test adequacy; test case

UML状态图用于描述一个特定对象的所有可能状态以及由于各种事件的发生而引起状态之间的转移。大多数面向对象技术都用状态图来表示单个对象在其生命周期中的行为^[1-2]。UML状态图来源于David Harel的Statechart,本质上是一个扩展有限自动机。它与传统自动机的区别在于, UML状态图支持并发、层次化、事件等特性, 它强调对复杂系统进行系统建模, 提供了层次状态机的框架, 即一个单独状态可以扩展为更低级别的状态机, 并提供了并发机制的描述, 因此, UML使用状态图作为对单个类的行为建模。由于有限状态自动机的相关理论非常成熟, 因此基于UML状态图的测试可以借鉴这些成熟的技术。但是, 在可测试性方面, UML状态图还存在不足之处。UML的灵活性使得模型中可能存在不完整性和二义性。例如, 对状态图信息的定义不够精确, 不能表达状态、属性、方法之间的关系; 缺乏必要的测试约束信息等。在软件测试中, 如果不对状态进行精确定义, 则无法对结果状态进行检查^[3-8]。

1 UML 状态图的扩展

为了解决 UML 状态图存在的问题, 需要扩展 UML 状态图中的图形元素描述信息。UML 中提供了丰富的可视化的模型元素和图形表示方法来描述软件系统, 这足以满足大多数软件系统的要求。但是, 有时用户需要使用一些新的模型元素和表示法或为模型添加一些非语义信息, 例如, 在输入年月日时, 要考虑闰年, 2月29日等条件是否可行, 因此, UML使用扩展机制来解决这一需求。UML 扩展机制允许建模者在不用改变基本建模语言的情况下做一些通用的扩展, 它提供了新的模型元素以及可以附加在模型元素上的各种信息。这些扩展机制已经被设计好, 以便在不理解全部语义的情况下可以存储和使用。

本文引入的 UML 扩展机制包括: 事件, 约束, 状态迁移时的任务和测试说明。事件表示在某一特定的时间或空间出现的能够引发状态改变的运动变化, 如接收到的从一个对象对另一个对象发送的信号、某些值的改变或一个时间段的终结。约束是用文字表达式表示的语义限制。它是一些规则, 表示模型元素中的语义联系, 用对象约束语言(OCL)表达一些限制条件的文本, 规定某些条件必须为真, 否则该模型表示的系统无效。

每个表达式有一种隐含的解释, 这种语言可以是正式的数学符号; 或是一种基于计算机的约束语言; 或是一种编程语言, 如 C++; 或是伪代码或非正式的自然语言。约束可以表示不能用 UML 表示法来表示的约束和关系。当陈述全局条件或影响许多元素的条件时约束特别有用。任务和测试说明是对状态的详细描述, 当测试选择状态转移时, 会根据任务和测试说明选择有效的, 可执行的路径进行测试, 避免了无效的测试, 减轻了工作量。在扩展 UML 状态图中, 事件可表示为 $E(event)::xxx$; 约束可用 $G(guard)::xxx$ 表示; 而任务和测试说明用 $S(statement)::xxx$ 表示。

图1是ATM取款机的扩展UML状态图模型, 本文以其为例介绍测试用例的生成过程。通过对图1的分析, 可以简单描述一下ATM取款机的交易过程: 当磁卡插入ATM机后, 输入密码验证身份, 如果密码正确, 则进行存款, 取款或查询余额的操作; 如果密码不正确, 则提示重新输入密码, 如果3次输入密码错误, 则退出磁卡, 取消交易。

作者简介:徐博(1982-), 男, 硕士研究生, 主研方向: 分布式系统与智能检测; 樊晓光, 教授、博士; 田涛, 工程师、博士
收稿日期: 2006-12-30 **E-mail:** jazz33@sohu.com

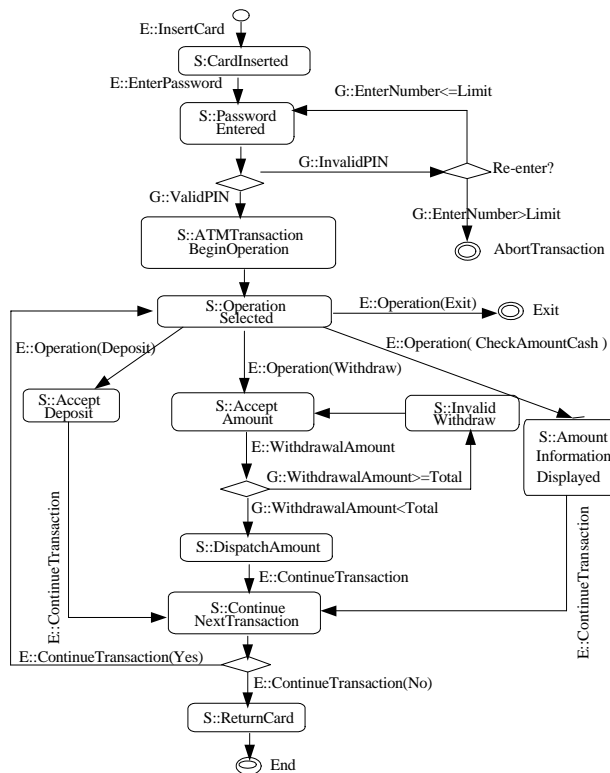


图1 ATM取款机的扩展UML状态图模型

2 测试用例生成

由于有限状态机可以较为精确地刻画软件系统或其子系统的行为，因此它被广泛应用于许多领域的应用系统建模，例如：通信协议，实时系统，面向对象软件种类的行为及其交互等。同时，在面向对象涉及领域已经被广泛使用的UML中的状态图与FSM十分相似，可以对已经很成熟的FSM技术进行调整用于状态图的分析。因此，根据图1的扩展UML状态图模型，可用事件，约束，任务和测试说明等扩展机制来生成有限状态机。有限状态机模型如图2所示。表1、表2说明了各状态迁移含义。

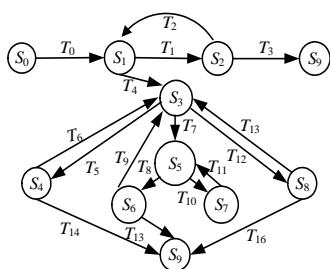


图2 ATM取款机的有限状态机模型

表1 状态说明

状态	形式化说明
S ₀	插入磁卡
S ₁	输入密码
S ₂	密码输入错误
S ₃	密码输入正确，选择操作
S ₄	存钱
S ₅	取钱
S ₆	取款金额小于帐户余额
S ₇	取款金额大于帐户余额
S ₈	查询余额
S ₉	退出磁卡

表2 全局状态转移说明

状态迁移标号	状态迁移说明
T ₀	输入密码
T ₁	密码不正确
T ₂	输入密码错误次数小于最大值
T ₃	输入密码错误次数大于等于最大值
T ₄	密码输入正确
T ₅	存钱
T ₆	钱已存入，继续其他交易
T ₇	取钱
T ₈	取款小于最大限额
T ₉	钱已取出，继续其他交易
T ₂₀	取款大于等于最大限额
T ₂₂	返回重新取款
T ₁₂	查询余额
T ₁₃	余额查询完毕，继续其它交易
T ₁₄	存款完毕，结束交易
T ₁₅	取款完毕，结束交易
T ₁₆	查询余额完毕，结束交易

生成有限状态机模型后，定义测试覆盖准则：

- (1) 状态覆盖要求产生的测试用例能够测试每一个状态；
- (2) 状态 - 事件对覆盖测试时组合每个状态和事件，这些事件还包括非法事件，这种覆盖能有效检测出被测实现是否接受了非法消息；
- (3) 状态 - 转换覆盖要求测试用例满足：每个状态至少都要被测试一次；每个转换都要求最少被执行一次。

通过有限状态机模型和测试覆盖准则，可以产生测试路径，如图3所示。

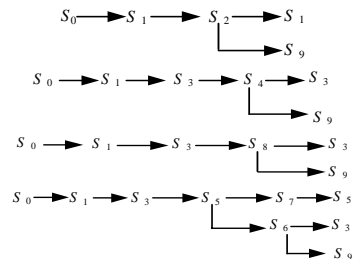


图3 测试路径

本文的测试用例产生过程主要是通过对扩展UML状态图模型分析，生成有限状态机，然后产生测试路径和路径覆盖准则，最后生成测试数据。

3 结束语

软件测试作为软件质量保证的重要环节，其作用受到越来越多的重视，测试用例是软件测试的关键，为提高软件生产效率，开发自动化软件测试工具势在必行，因此，本文将扩展UML状态图模型与有限状态机相结合来生成测试用例。但该方法还存在一些问题需要改进和完善，例如：描述大型复杂系统功能的能力有限；测试用例生成还不能完全实现自动化，必须有人工参与等。

参考文献

- [1] 王少峰. 面向对象技术[M]. 北京: 清华大学出版社, 2004.
- [2] 李刚, 朱关铭. 形式方法与面向对象方法的结合探讨[J]. 计算机工程, 1998, 24(1): 13-16.
- [3] 兰毓华, 毛法尧, 曹化工. 基于Z规格说明的软件测试用例自动生成[J]. 计算机学报, 1999, 22(9): 963-969.
- [4] 李留英. UML测试技术的研究与实现[D]. 长沙: 国防科技大学, 2000.

(下转第82页)