基于UML的回归测试软件测试方法的研究与应用

李 杨,张春海,张美玲
(中国海洋大学 信息科学与工程学院,山东 青岛 266000)

摘 要:回归测试在软件的版本升级中起着至关重要的作用,它的测试效率直接取决于软件升级中对软件修改部分的测试实例的选择。提出了一种基于UML的回归测试方法,它采用UML中类图和状态机来确定软件升级中的修改,它将修改分为两类:一种是类驱动修改;一种是状态驱动修改。通过类图和状态机图的改变选择确定有效的测试用例,提高了回归测试的测试效率。将测试用例划分为无用的、可重用的、需重新测试的测试用例三类,并将此方法应用到一个实例中,验证了它的可行性和有效性。

关键词:基于UML的回归测试;类图;状态机图
中图分类号:TP311.5 **文献标识码**:A **文章编号**:1673-629X(2010)09-0235-04

Research and Application of Software Testing Based on UML-Based Regression Testing

LI Yang, ZHANG Chun-hai, ZHANG Mei-ling
(Dept. of Information Science and Engineering, Ocean University of China, Qingdao 266000, China)

Abstract: Regression testing is an important activity that ensures the reliability of evolving software. One of the major issues in this type of testing is the selection of test-cases to test the affected portion of the software. Present a UML-based selective regression testing strategy that uses state machines and class diagrams for change identification. Identify the changes using the UML semantics of state machines and class diagram. The changes are classified as class-driven (obtained from class diagram) and state-driven (obtained from state machine). With the help of the identified changes, classify the test cases of the test suite as obsolete, reusable, and retestable. Apply the approach on a case study to demonstrate its validity.

Key words: UML-based regression testing; class diagram; state machine diagram

0 引 言

随着软件复杂度越来越高,软件升级对于软件测试人员提出了极大的挑战,如何保证软件升级后的软件质量,如何测试升级软件的有效性已经成为软件测试人员必须要面临的难题。回归测试是一种非常有效的测试方法,回归测试确保了软件修改后的可用性[1,2],文中的目标是利用UML中的状态机图和类图来实现回归测试的测试实例的选择,提高回归测试的效率。

在面向对象的软件测试中对象的状态行为是主要的研究问题,UML状态机是对对象状态行为的建模手段,有很多测试方法利用状态机图来描述对象的状态行为,当软件状态发生改变后,对修改部分的测试就变得非常重要[3,4],这种情况下有两种选择,一种选择是将所有的测试实例重新测试,对于复杂的软件系统这种方法显然是不可行的,因为测试实例非常多,这将增加不必要的测试开销[5,6];另一种选择就是只对修改部分进行测试[7],但是对于测试用例的选择就变得非常重要了[8]。软件测试用例的选择直接决定了软件测试的效率,文中提供了一种基于UML图的回归测试软件测试方法的选择策略,利用UML的状态机图和类图来产生和选择回归测试的测试实例,并且将它应用到一个实例中,以验证它的可行性。

1 具体描述

在UML建模过程中,各种具体的图是相互联系的,在某个图中的改动可以影响到其它图的改动,但是却不能被明确地反映出来,例如,类图的变动会导致顺序图的变动,但是在顺序图中不能直接反映出来,因此通过对类图的研究发现改动就变得非常重要了,同样

收稿日期:2010-01-23;**修回日期**:2010-04-11
基金项目:青岛市科技计划项目(08-1-3-2-jch)
作者简介:李 杨(1985-),男,山东德州人,硕士研究生,研究方向为数据库理论;张春海,教授,研究方向为数据库理论。

的对于其它图的改动也将会影响到状态机图[9]。文中提出的基于 UML 的回归测试软件测试方法分为以下几个步骤：

(1)确定新版本软件的修改元素。

(2)分析确定间接受影响的元素。

(3)分析原始测试套件中修改元素和间接受影响元素的测试用例，并对它进行分类为无用的测试用例、需重测的测试用例、新增测试用例。

(4)舍弃无用的测试用例，增加新测试用例，重新测试需重测的测试用例。

具体测试步骤如图 1 所示。

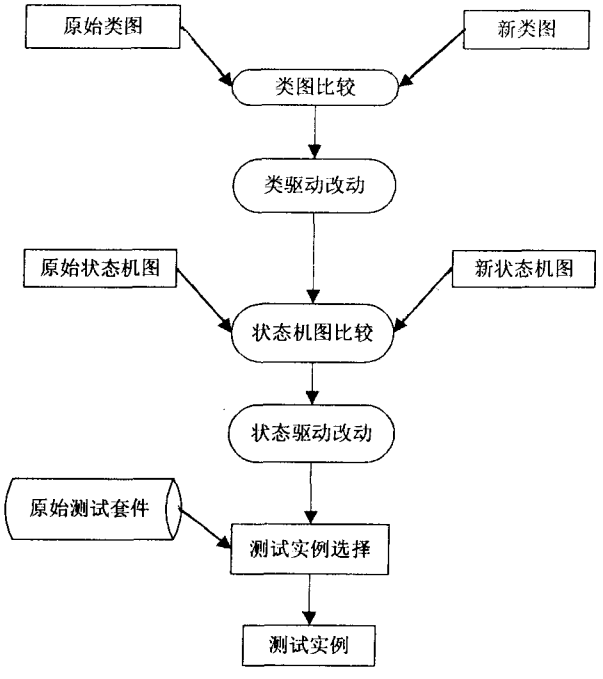

图 1 基于 UML 的回归测试的方法

在 UML 中存在两种类型的状态机图，一种是行为状态机[10]，另一种是协议状态机，在文中只是应用行为状态机图。在 UML 中对状态机图影响最大的是类图，其它的例如顺序图等，也会对它产生影响，文中只关注类图对于状态机图的影响。Hsia 将面向对象编程的改动划分为对不同类的对象行为[11]、数据成员、成员函数、成员函数的行为、关系的改动，在文中提出的方法中，通过状态机图来描述对象行为的改动，通过类图来描述数据成员、成员函数、成员函数的行为、关系的改动。

1.1 类驱动改动

一个类图的改动是指对它的属性、成员函数，或关系的改动[12]，接下来定义了几种类驱动改动：

表达式改动：在 UML 中表达式是指在上下文中代表一系列值的树形结构，可以通过两个步骤确定表达式的改动，首先将表达式作为纯文本进行比较，如果比较结果不同则认为此表达式改动了，如果相同再进行第二步，分析表达式的类间的关联，通过它进行表达式改动的确定。

属性改动：将原始类图的属性集定义为 A_0，新类的属性集定义为 A_n，改动的属性集定义为 A_m，一个属性被定义为修改的属性，当且仅当在 A_0 中存在相同名字的属性，但是属性类型却不相同。

界值改动：界值改动是指某数值的上界和下界的改动。

操作符改动：如果某操作符的可视权限被改动或者操作符的附带异常被改动，就认为该操作符被改动，另外如果某操作符的属性被改动，它也被认为是操作符改动，操作符属性包括，isLeaf，isStatic，isQuery 三种属性，若其中任何一个改动了都被认为是操作符改动。

关联性改动：关联性是指其值为类型实例的元组的集合，如果关联终结点发生了改动，则被认为是关联性的改动。

成员函数参数改动：如果一个成员函数的参数的默认值被修改或者方向(流入，流出，返回)被改变，则认为这个成员函数的参数被改动。

成员函数改动：一个成员函数被改动分两种情况：

(1)成员函数的访问权限被改动或者成员函数的异常被改动(新增或者删除异常)。

(2)成员函数的限制条件被修改，一个成员函数有两种限制条件，操作前条件、操作后条件。操作前条件是指在操作调用前的条件，操作后状态是指在操作调用后的状态。

增加删除关联性：增加删除的关联性可以在相应的状态机图的转化中反应出来，状态机图的转化为动作、事件关联性的连接。

增加删除属性：在 UML 类图中，删除的属性是指在新的类图中没有存在的相同的属性名，增加的属性是指在新类图中存在但在原始类图中没有的属性，如果相应的属性在状态机图中被用到，这也将在状态机图中反应出来。

增加删除操作符：某操作符被认为删除当且仅当在原类图中存在的操作符没有出现在新类图中，两个操作符被认为相同(当且仅当它们有相同的名称、返回值类型、参数个数)。删除的操作符能够反映在状态机图中(如果相应的操作在状态机图中被用作调用事件或调用动作)。某操作符被认为增加当且仅当在原类图中不存在的操作符出现在新类图中，两个操作符被认为相同当且仅当它们有相同的名称，返回值类型，参数个数。增加的操作符能够反映在状态机图中(如果相应的操作在状态机图中被用作调用事件或调用动

作)。

增加删除成员函数:如果在新类图中不存在同样的操作名、参数个数、参数类型及名称、返回值,那么这个成员函数被认为是删除的,如果删除的成员函数在状态机图中被用到,那么将会影响到状态机图。如果在新类图中存在某一成员函数,它在原始类图中不存在相同的成员函数名、参数个数、参数类型及名称、返回值等,则这个成员函数将被认为是增加的成员函数。

1.2　状态机驱动改动

在 UML 中状态机图由区域组成,区域由状态、转化和一些点组成[9,13],本方法中假设状态机图在单一的区域中,并且忽略了组合状态和子状态,因为这样的平板状态机可以减少状态追踪和状态转化的开销。本方法只关注对状态和转化的改动,以下描述中将用 S_m 代表原始状态机,S_m' 代表新状态机,原始状态机中的状态集合用 S 表示,转化集合用 T 表示,新状态机中的状态集合用 S' 表示,转化集合用 T' 表示。

状态机驱动改动可分为以下几种:

增加删除状态:状态的增加和删除总会导致状态转化的增加和删除,所以本方法只关注状态转化的增加和删除,通过转化的增加和删除确定状态的增加和删除。

状态机改动:一个状态 $S \in S'$ 将会被认为改动 当且仅当 S 的状态常量,状态行为及状态上下文被改变,则此状态被改动。

增加删除状态转化:用 T_a 代表增加状态转化的集合,用 T_d 代表删除状态转化的集合,一个状态转化 $t \in T'$,$t \in T_a$ 当且仅当 $t \in T'$ 但 $t \notin T$,一个状态转化 $t \in T$,$t \in T_d$ 当且仅当 $t \in T$ 但 $t \notin T'$。

状态转化改动:将状态转化改动集定义为 T_m',一个状态转化 $t \in T'$,$t \in T_m'$ 当且仅当 t 包含改动的事件、改动的动作。

1.3　测试套件分类

以上两节分别讨论了类驱动改动和状态机驱动改动的定义,通过定义获得类驱动改动和状态驱动改动的更改集合,接下来就是应用状态机驱动改动对原始套件进行分类,根据 Leung 和 White 的研究[1],一个可维护的测试套件需要区分为三种测试用例:无用的、可重用的、需重新测试的测试用例。无用的测试用例是指对修改后的新软件系统没有用的测试用例,可重用的测试实例是指对应原始软件没有被修改部分的测试实例,它们是有效的但是却不需要被重新测试,重新测试的测试用例是指对应原始软件的修改部分,需要被重新测试的测试用例。测试套件的任何一个测试用例将代表 S_m 中状态转化的顺序,用 Ts_o 代表无效的测试用例,用 Ts_{ru} 代表可重用的测试套件,Ts_{rt} 代表需重新测试的测试套件,一个测试用例属于 Ts_o 当且仅当某一转化 $t \in T_d$,一个测试用例 $t_s \in T'$,若 $t_s \in Ts_{rt}$ 当且仅当某一状态转化 $t_i \in t_s$,$t_i \in T_m$ 但是 $t_i \notin T_d$,其它测试用例属于 Ts_{ru}。

2　实验设计及分析

2.1　实验设计

为了验证文中提出的基于 UML 的回归测试软件测试方法的可行性,将它应用于一个线程池的试验项目中,此项目主要目的是用一个线程池来实现对异步操作线程的管理和应用,主要功能包括管理初始线程个数、增加个数、缩减个数等。它的操作就像队列,当外部应用程序将它们的任务传输到线程池中并请求处理时,线程池将选择某个空闲线程完成此请求,此外还包括如何处理异常情况,如在实现某请求时,发生了异常,则线程池有相应的操作进行处理,线程池还应该有接受任务、处理任务、任务完成后的清理工作,它可以为任何应用程序提供轻量级的异步处理,例如可以将它应用于聊天程序中,用它来管理从发送者到一个或多个接收者的消息处理。它的类图和状态图如图 2 和图 3 所示。

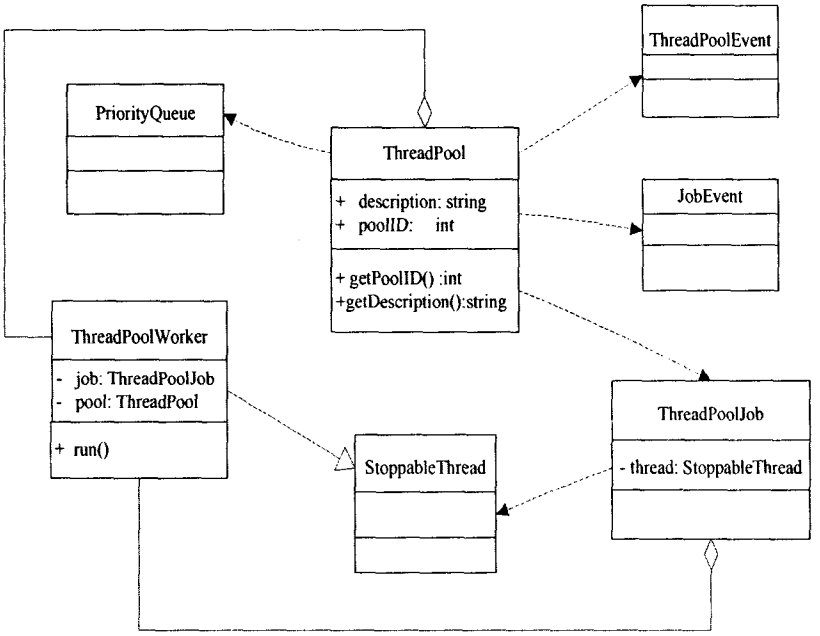

图 2　线程池类图

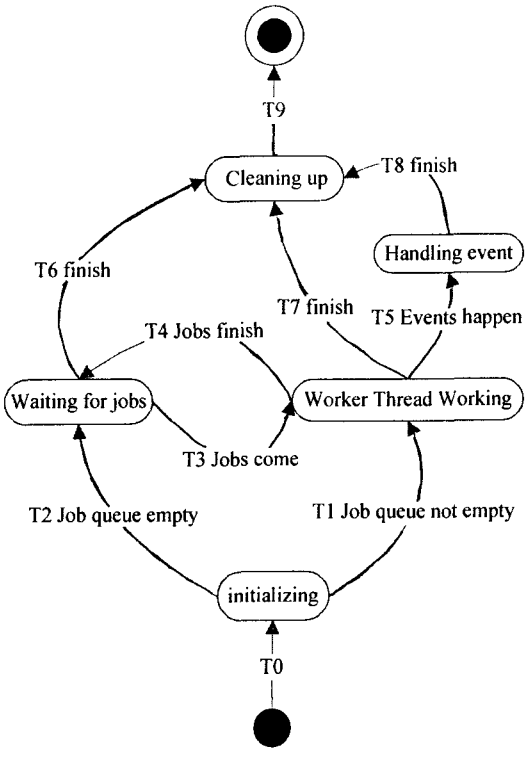

图 3 线程池状态图

根据文中提出的方法，首先确定类图和状态机图的改动，包括在类 ThreadPool 中，增加了 description 属性，将 poolID 属性类型由 int 改为 string，状态机图增加了 Cleaning up 状态，相应地增加了转换 T_7，T_8，T_9，根据图中的改动对测试用例分类，其中测试用例 4,5,7,14 为无用测试用例，3,9,12,16,18 为可重用测试用例，其它的测试用例 1,2,6,8,10,11,13,15,19,20 为需重测的测试用例，表 1 为所有的测试用例。根据以上分类，在对当前项目进行回归测试时只需对需重测的测试用例进行测试，大大提高了测试效率。

2.2 实验分析

通过以上实验，在对线程池项目的回归测试中，文中提出的方法仅仅对需要重新测试的测试用例进行测试，在以上项目中只对 1,2,6,8,10,11,13,15,19,20 测试用例进行测试，而不需要对其它测试用例进行测试，减少了不必要的测试花费，由此可以分析得出，在对软件的回归测试中，文中提出的方法可以大大提高测试效率。

3 结束语

文中提出了一种基于 UML 类图和状态图的回归测试测试用例选择的方法，该方法致力于发现类图和状态机图的改动，从而有针对性地选择测试用例，文中定义了类驱动改动和状态机驱动改动，并将测试用例划分为无用的、可重用的、需重新测试的测试用例。最后将此方法应用于线程池项目中，证明了它的可用性。

表 1 测试用例

1	$T_0T_1T_7T_9$
2	$T_0T_2T_6T_9$
3	$T_0T_2T_3T_5$
4	$T_0T_1T_5$
5	$T_0T_2T_3$
6	$T_1T_5T_8T_9$
7	$T_0T_2T_3T_4$
8	$T_2T_3T_4T_6T_9$
9	$T_0T_2T_3T_4T_3T_5$
10	$T_2T_3T_5T_8T_9$
11	$T_2T_3T_4T_3T_4T_6T_9$
12	$T_0T_2T_3T_4$
13	$T_2T_3T_4T_3T_5T_8T_9$
14	$T_0T_2T_6$
15	$T_0T_1T_5T_8T_9$
16	$T_0T_1T_5T_8$
17	$T_0T_1T_6$
18	$T_0T_2T_3T_4T_7$
19	$T_0T_1T_4T_3T_7T_9$
20	$T_0T_1T_5T_8T_9$

基于现有工作下一步的研究方向是通过将此方法应用于大型工程项目中，进一步验证它的可用性和高效性，另外正在进行此方法的全自动实施。

参考文献：

[1] Leung H K N, White L. Insights into regression testing of software testing[C]//Proceedings of Conference on Software Maintenance. [s. l.]:[s. n.],1989:60 - 69.

[2] Mellor S J. Introduction to executable and translatable UML [M]. New York, NY: Van Nostrand Reinhold,2005:43 - 56.

[3] 屈 波，聂长海，徐宝文. 基于测试用例设计信息的回归测试优先级算法[J]. 计算机学报，2008,31(3):431 - 439.

[4] Bellur U, Vallieswaran V. On OO design consistency in iterative development[C]//Third International Conference on Information Technology. [s. l.]: New Generations, 2006:46 - 51.

[5] 伦立军. 软件测试中路径覆盖自动生成技术研究[J]. 哈尔滨工业大学计算机学报，2003,10(3):52 - 58.

[6] 卢炎生，王 曦. 基于依赖性分析的 UML 状态图切片技术[J]. 计算机工程，2006,32(15):81 - 83.

[7] 仲晓芳，张春海，李 杨. 基于回归测试的软件测试方法的研究与应用[J]. 计算机技术与发展，2010,20(1):155 - 158.

（下转第 242 页）

义短时傅里叶变换谱图的瞬时频域相关函数为：

$$R_{stft}(t,\tau)=\int_{-\infty}^{+\infty}|STFT_{x_1}(t,\omega)|^2|STFT_{x_2}(t-\tau,\omega)|^2d\omega \quad (7)$$

(7)式是谱图 $S_x(t,\omega)$ 在频域的一维相关，取不同 τ 值计算频域相关系数 $R_{stft}(t,\tau)$，$R_{stft}(t,\tau)$ 的大小表明 $x_1(t)$ 和 $x_2(t)$ 谱图中每个瞬时 t(对于 $x_1(t)$)和 $t-\tau$(对于 $x_2(t)$)的频域能量之间的相关性，当 $\tau=D$ 时经频域相关处理的信号能量集中，获得最大的 $R_{stft}(t,\tau)$。这样基于短时傅里叶谱图瞬时频域相关的估计方法就是寻找在信号脉冲宽度内瞬时频域相关的最大值，即：

$$R_{stft}(t,\tau)=\int_{-\infty}^{+\infty}|STFT_{x_1}(t,\omega)|^2|STFT_{x_2}(t-\tau,\omega)|^2d\omega\leqslant R_{stft}(t,D) \quad (8)$$

通过对 STFT 谱图瞬时频域相关法与三种基于 WVD 时间延迟估计方法进行比较，仿真结果表明[12]，所提出的基于 STFT 谱图的瞬时频域相关法的时延估计可以达到克拉美－罗界，并可获得最佳效果。

通过上述方法，分别得到 τ_{12}，τ_{13}，τ_{14} 三个延迟估计数值。然后，由(1)、(2)、(3)式计算出目标位置的空间坐标。

3 实验分析

实验过程中，采用放置于高处的爆炸声信号作为目标源。声传感器阵列采用边长为 2m 的正方形四元面阵，用自己设计的 DSP 数据采集系统采用触发方式进行数据采集。当信号幅度达到一定量值即认为目标有声音信号，采集到的声信号数据通过 RS232 串口发送给 GPRS 模块，然后再通过 GPRS 无线网络传送到远程计算机处理系统。计算机系统对接收到的数据信号进行分解、预处理、特征提取，最后进行目标识别与定位。

实验表明，该系统能够实时、正确地将传感器获取的信号远程传输，并实现对爆炸源的空间位置进行计算，计算结果与实际位置误差在 10% 左右。

4 结束语

对基于 GPRS 的声目标定位系统的硬件和软件部分进行了设计，实验验证了系统的有效性和可行性。鉴于实验所采用目标的单一性，对于复杂的声目标进行识别与定位的算法还需进一步研究，定位精度有待于进一步的提高。伴随着新一代无线数据传输技术(如 3G 无线网络)的逐渐成熟与发展，开展基于 3G 的声目标识别与定位技术的研究将是下一步的工作。

参考文献：

[1] Namorato M V. A concise history of acoustics in warfare[J]. Applied Acoustics,2000,59:101－135.

[2] 靳 莹,杨润泽.声测定位技术的现状研究[J].电声基础,2007,31(2):4－8.

[3] 刘维群,凌风彩,匡国防.基于 GPRS 网络的远程数据采集系统及应用[J].微计算机信息,2009,25(6－1):135－137.

[4] 林志斌,徐柏龄.基于传声器阵列的声源定位[J].电声技术,2004(5):19－24.

[5] 陈华伟,赵俊渭,蔡宗义,等.两种声学阵列的定向精度分析与仿真[J].声学与电子工程,2001(3):6－11.

[6] 贾云得,冷树林,刘万春,等.四元被动声敏感阵列定位模型分析和仿真[J].兵工学报,2001,22(2):206－209.

[7] 王 昭,李 宏,赵俊渭,等. 空气声被动定位的误差分析[J].应用声学,2000,19(2):39－43.

[8] 吕永林.声目标识别技术研究[J].楚雄师范学院学报,2008,23(3):24－30.

[9] 吕永林,字正华. 基于 VC 与 Matlab 的声目标识别系统设计[J].计算机技术与发展,2009,19(9):207－210.

[10] 行鸿彦,刘照泉,万明习.基于小波变换的广义相关时延估计算法[J].声学学报,2002,27(1):88－93.

[11] 戴延中,李志舜.基于 Wigner－Ville 分布的宽带回波到达时刻估计方法[J].声学学报,2002,27(1):84－87.

[12] 字正华,石庚辰. 基于短时傅立叶变换谱图的非平稳信号时延估计方法[J].探测与控制学报,2007,29(6):19－23.

(上接第 238 页)

[8] Sajeev A S M, Wibowo B. Regression test selection based on version changes of components[C]//Tenth Asia－Pacific Software Engineering Conference. [s.l.]:[s.n.],2003:78－86.

[9] Liang H. Regression testing of classes based on TCOZ specification[C]//Proceedings of 10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS). [s.l.]:[s.n.],2005: 450－457.

[10] Beizer B. Software testing techniques[M]. New York, NY: Van Nostrand Reinhold,1990.

[11] Hsia P, Li X, Kung C T, et al. A technique for the selective revalidation of OO software[J]. Journal of Software Maintenance: Research and Practice,1997,9(4):217－233.

[12] 张志军.面向对象软件的回归测试策略研究[D].长沙:湖南大学,2004:32－35.

[13] 郭 宁.UML 及建模[M].北京:清华大学出版社、北京交通大学出版社,2006:52－59.