

现代应用软件的测试策略与工具*

周之英 钱 岭

(清华大学计算机系 北京 100084)

摘要 现代应用软件有许多新的特性,这样就对软件测试提出了新的要求,如要求测试工作更有效,进行面向用户的测试等.这些要求是传统的测试策略所不能解决的.本文提出了一种基于应用操作覆盖的测试方法来解决这些问题.这种方法的基本分析工具是导航图和事件记录序列.本文最后还给出了相应的自动测试工具.

关键词 软件测试,导航图,逻辑覆盖,(应用)操作覆盖,事件记录序列,测试工具.

如同任何生产中都离不开产品质量检查一样,软件测试是计算机软件开发过程中必不可少的一步.软件测试做的怎样,决定着软件质量的好坏.据 Boehm 的统计,软件开发总成本中,用在软件测试上的开销要占到 30~50%,而如果讨论整个软件生存周期,即把维护也考虑在内,则有人估计达 50%,甚至 50%以上的成本要花在测试上;对于人命攸关的软件,如飞行控制或反应堆监测软件,其测试费用可高达总费用的 75~85%.然而软件测试的重要性及其基本方法也是逐渐为人们所认识的.在 50、60 年代,软件开发人员就是测试人员,而且将测试和调试混为一谈,实际上调试就是排错(Debug),是软件测试的一个环节;在 70 年代,由于“软件危机”的影响,软件工程这一学科产生并发展起来,这时人们对软件测试的认识更广泛也更深刻了,许多关于测试的理论被提出并完善,测试人员可以在测试方法的指导下,用自动工具进行测试.

时至今日,由于计算机软、硬件的不断发展以及计算机的普及,计算机软件的开发呈现出许多新的特性,如规模更大、集成性更好、可维护性更强等.这就使传统的如基于输入/输出的逻辑覆盖测试方法不能满足新的需要,为使软件测试能跟上软件开发的变化,必须有新的测试策略.本文首先分析了现代应用的新特性;基于这些讨论我们在第 2 节提出了一种基于应用操作覆盖的测试策略,其基本分析工具是导航图和事件记录;其对应的测试工具表现出它的优越性和实用性.

1 现代应用的新特性

与传统软件相比,现代应用有许多新的特点:

(1) 就应用本身而言

* 作者周之英,1939年生,教授,主要研究领域为软件工程.钱岭,1972年生,博士生,主要研究领域为软件工程.

本文通讯联系人:周之英,北京 100084,清华大学计算机系

本文 1996-01-30 收到修改稿

① 规模更大. 硬件的发展使软件可能以更大的介质来发行, 例如 CD-ROM; 由于软件开发商间的激烈竞争和对旧成果的利用, 开发商们有必要、也有可能使某种产品的功能无所不包, 从而提高它的竞争力; 另外, 由于运行环境的关系, 应用本身要求有更多的代码…总之, 应用的规模被成倍的扩大了.

② 集成性更大. 由于用户/市场的要求以及硬件和系统软件的有力支持, 一个成功的应用软件中往往包含了许多技术要点, 如数据库技术、多媒体技术、数据可视化技术、网络通信技术等等. 这在以往是不可能的.

③ 可维护性更好, 要求软件的生命周期更长. 对商业化软件来说, 能否不断的升级不仅反映出开发商的实力, 而且是软件生命力的体现; 而对面向企业的软件来说, 由于企业需求的不断变化, 要求软件能不断的改进以适应这些变化.

④ 开发时间缩短. 开发商之间的竞争日趋激烈, 要求开发商在更短时间内完成软件的开发任务, 以抢先占有市场. 这不仅减少了软件开发的时间, 而且缩短了软件测试的时间, 要求在更短的时间内能有效的完成测试.

(2) 就开发/运行环境而言

① 可移植性好. 应用往往要求跨多种软、硬件平台, 由于平台间的不兼容性, 应用在设计时就要考虑到留有不同的系统接口. 另外, 应用还经常要和各种外设打交道, 因此其内部有更多的控制和分支.

② 良好的可操作性. 应用一般都要求在图形界面环境下运行, 在图形界面中有各种标准资源, 如菜单、窗口和对话等. 它们之间也往往要求有相互控制的要求, 即可能有操作上的时序关系, 而用户使用时的操作却是随意的. 这就要求应用有很好的可操作性和健壮性, 不仅能防止用户的不当动作, 也应能在不当动作之后保持系统的正确性.

(3) 就开发人员而言

① 许多开发人员协同工作. 现代应用的开发早已不是独门独户的个体劳动, 它要求将大量的开发者组织起来共同开发. 这就使应用中包含各种不同风格代码和复杂的接口关系. 应用的出错机会增大了.

② 开发人员要适应开发环境的变化. 由于系统软件的升级换代, 使开发人员经常要在各种不同的环境下工作, 难保开发人员尚未充分了解环境就开始编程. 由于对系统的不了解, 可能写出不可靠的代码.

(4) 就软件用户而言

① 用户有更高的要求. 这在前面的分析中已经体现出来.

② 由于计算机的不断普及, 软件将面向更多不熟悉计算机的用户, 要求软件有更充分和细致的文档, 无形中加大了开发商的负担. 而且不了解计算机的用户会有开发人员料想不到的操作, 使软件发生故障.

另外, 还有许多新特性在这里没有提到, 例如软件开发技术的改进(应用底层多采用事件机制来通信; OO 技术在开发中的推广) 等.

从上面的讨论中可以看出现代应用的复杂程度加大了, 无论是内部实现还是外部界面(规模、集成度、可维护性和图形界面…); 而且出现了不少传统的基于输入/输出逻辑覆盖的测试策略不能解决的问题, 如事件驱动的特性和测试的有效性. 但同时应用对测试的要求却

提高了.这就要求有一种适合这些新特性的测试策略.在下面的论述中我们将提出一种基于对应用使用方式进行覆盖的测试策略,即操作覆盖.

2 一种现代应用的测试策略—操作覆盖的策略

从上面的分析中可以看出,现代应用对其测试策略的要求有了新的要求:① 测试策略要求更有效.测试时间缩短;发现更多的错误.② 测试要面向用户使用.模拟用户的使用经验,找出不当操作的操作序列.③ 测试要求适应软件开发过程中的变化.如OO技术的使用等.④ 测试要求适应在测试中考虑用户对对象的操纵.即有如图1的测试模型.

2.1 传统测试策略的危机

传统的测试策略是不能解决上面这些问题的.前面在第1部分中已经说到传统的测试策略是基于输入/输出数据驱动的测试策略,它根据测试中是否考虑程序的内部结构而分为2类:1)不考虑程序内部结构和特性的黑盒测试.它以各种输入数据的组合进行边值分析和等价类划分来设计测试用例;2)使用可以分析的程序内部逻辑路径构造来设计测试用例的白盒测试,以覆盖程序内部的各种执行路径,而不考虑程序的功能.各种不同级别的覆盖决定了测试的彻底性.然而,在现代应用的测试中它们都不再适应了.

对黑盒测试来说,由于用户的操作和以往相比有了很大的变化,不像传统的计算机软件那样,以不同的数据输入来区分不同的执行路径,而是由用户操纵界面上的不同对象来决定系统的执行方式.这些操纵方式是:① 更多且更复杂;② 是随机的而不是顺序的;③ 对程序的运行结果影响更大.这样单纯的考虑数据的输入来进行测试就不能充分的找出应用中的错误,不能反映出现代应用的特点.

对白盒测试来说,由于程序结构方面出现了许多新的特征:① 程序复杂度更高;② 程序模块间不是直接调用的关系;③ 主程序被最小化了,往往只有一条语句表示进入事件循环;④ 对于多进程的应用,往往有并发执行的部分.这些都使程序不易分解,或是不能分解,那样就不易或是不能从程序结构中分析得出程序的执行路径.传统的白盒测试策略也就不起作用了.

因此,我们提出了一种基于应用操作覆盖的测试策略,简称为操作覆盖.

2.2 操作覆盖的测试策略

(1) 操作覆盖的含义和特点

操作覆盖的策略主要是通过对用户界面上各种对象的操作顺序的覆盖来设计用例,所谓操作顺序就是用户为完成某种功能而对应用的使用顺序,导航图是描述这些顺序的基本工具.导航图是应用需求说明书中用户界面设计部分对应用功能的图示化.它用树的方式将应用的界面功能表示出来,从树的根到树的叶的一条路径就对应了一个操作顺序.图2是导航图的一个例子.(选自IMDSS的数据查询子系统的概要设计说明书P16页).

对导航图上的对象进行操作覆盖可以代替传统的黑盒测试.而对传统的白盒测试,我们提出覆盖事件记录序列来代替它.事件记录序列也是一种描述用户操纵的方法,由于不同的操作会引发不同的事件,一个事件将引发系统中相应的功能调用,一个事件序列就唯一对

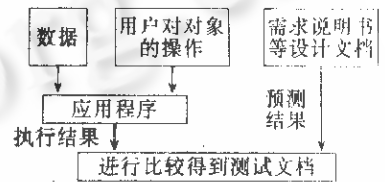


图1 测试模型

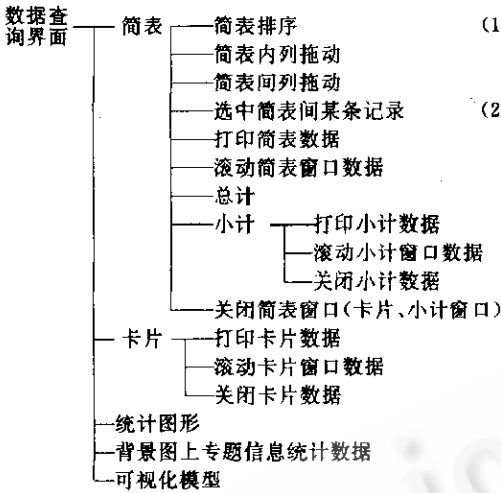


图2 导航图的例子

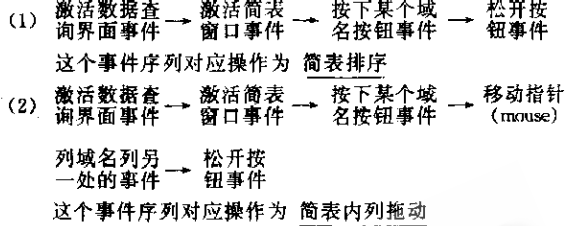


图3 事件序列的例子

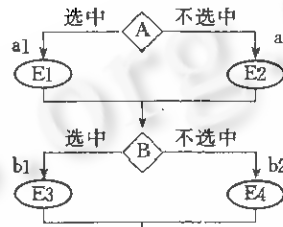


图4 操作关系图(其中E为执行)

应了系统的一次执行过程. 根据图 2, 给出 2 个事件序列的例子(如图 3).

注意, 这里后一种方法虽与程序的调用关系有关, 但它并不关心程序的内部实现.

操作覆盖测试策略是一种较高层次的策略, 它不考虑软件内部的实现方法, 而是以用户接口为测试对象, 封装了内部的复杂性; 同时它也是面向用户的策略, 从用户的角度来看待应用, 完成测试; 另外, 这种策略还有简洁、直观的特点, 是一种有效的策略, 可以提高测试的速度.

(2) 操作覆盖的不同级别

操作覆盖有各种不同的覆盖级别, 我们将它分为 3 个层次:

1) 对象覆盖: 这里对象不是 OO 中的对象, 它是指界面上各种供用户操作的资源. 如菜单、按钮等. 对象覆盖是指将应用界面上的各个对象都用到 1 次, 这种覆盖是级别最低的一种覆盖方式. 由于只是独立的覆盖各个对象, 所以其覆盖面是不完全的. 例如应用中有一个选择操作, 那么选中和不选中在对象覆盖的用例集中就是 2 个覆盖用例.

2) 分支覆盖: 它针对对象覆盖的不足之处, 要求对各个选择操作的不同选择分支作出不同的覆盖用例, 显然它做到了对象覆盖, 其覆盖程度应强于对象覆盖. 但它仍可能有遗漏的地方. 例如, 有 2 个二值的选择操作 A, B, 有如图 4 的操作关系图.

那么如果对路径 a1b2, a2b1 设计 2 个用例, 就可以达到分支覆盖的要求, 因为它覆盖了全部的路径. 但显然 a1b1, a2b2 这 2 条路径没有被测试到.

3) 组合覆盖: 它对各种不同的分支路径和它们的组合进行全部覆盖. 这是一种最完全的覆盖. 在上例中, 它要求对 a1b1, a1b2, a2b1, a2b2 作出测试用例. 然而这种覆盖会造成用例数目的极度膨胀. 在操作序列中每多一种 K 值选择, 就会使用例数目要乘以 K, 用例数目于操作序列中的选择的数目成指数关系. 因此, 如果测试时间要求比较短, 那么只能从中选出有代表性的或用户经常使用的路径组合来设计用例.

(3) 操作覆盖的用例生成

前面已经提到有 2 种描述用户操作的工具: 导航图和事件记录序列, 根据这 2 种工具可

以完成测试用例的设计。

导航图方法(黑盒方法):在这种方法中,每条由树根到树叶的路径就对应了一个测试用例,所有路径的集合就对应了测试用例集。对树的不同遍历方法就对应了覆盖的不同级别。另外,由于导航图中不包括非法的操作,因此还要针对应用设计一些异常情况下的用例。

事件方法(白盒方法):在这种方法中,每个事件序列就对应了一个测试用例,所有事件序列的集合对应了测试用例集,事件序列集的大小反映了覆盖的不同级别。因为事件序列可以包含各种各样的事件,通过分析就可以设计出异常或合法的用例。

但无论用哪一种方法,其生成的用例始终由一个操作序列构成。

(4) 进一步讨论

与测试过程的关系.软件测试基本上可以分为3个阶段:单元测试、集成测试和系统测试。它们之间有一种自底向上的构成关系。由于导航图是可分解的一张图,可以通过先对子图的测试完成较低层的测试,再将结果归纳完成对父图的测试,如此反复就完成了系统的测试。因此这种策略是可分解的。通过合理的分解使分解点就是各个测试阶段的衔接处,于是完成了各个测试阶段的自然过渡。

与数据分析的关系.由于这种策略的封闭了应用的内部实现,可能将错误也隐藏起来,所以单纯的操作覆盖还不是充分的,必须加入数据信息来检查其正确性。由操作序列和输入数据共同构成用例(参见图1)对于操作和数据的信息量都大的应用,用例数将非常大,这时可以用正交分析的方法来优化用例集合。

与设计方法的关系.为提高测试的效率,缩短测试时间,往往要求测试和开发并行进行。对操作覆盖的策略来说,如果能够尽早地作出导航图就能尽早的进行测试,这就要求有相适应的设计方法。快速原形(Rapid Prototyping)的方法能较早地确定用户界面是一种相适应的方法。

用例设计与界面资源的关系.操作覆盖的策略与应用设计中采用何种界面资源有很大的关系,因为各种资源对应的选择分支的数目是不同的。例如,一组5个的广播按钮(Radio Button)其分支数就是5。另外,用例设计中还要考虑到操作系统中的一些影响应用界面的系统功能,如窗口的关闭和缩放等。

软件测试中还要求能发现累积错误,这就必须在用例执行时重复执行一些用例。

3 测试工具的设计

测试工具带来的好处是人尽皆知的。对于现代应用,由于它的种种特点,使测试中要设计和执行的用例数大大多于传统软件,测试工具的重要性就更突出了。

基于前面讨论的操作覆盖的测试策略,现代应用的测试工具至少包含以下几个部分:

(1) 操作分析工具 A1 (包括 a. 导航图的绘制工具; b. 导航图到事件记录序列的转换工具; c. 生成事件序列) 由于导航图接近于需求分析说明书,利于书面表达测试;而事件序列接近于系统设计,利于实际使用。因此,如果能用导航图设计用例,用设计序列执行用例将是很理想的。

(2) 数据分析工具 A2 (准备测试数据)。

(3) 正交分析工具 A3 (完成数据与操作序列的正交处理,得到测试用例)。

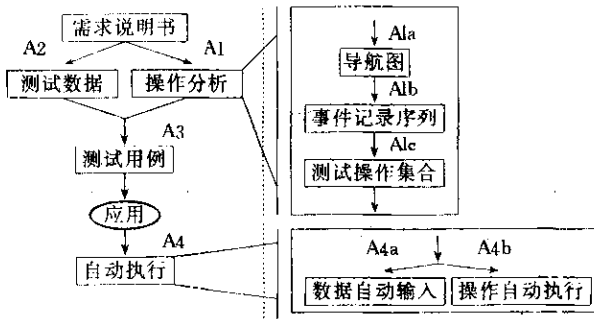


图5 工具间关系图(虚线右侧是注解部分)

演示和自动讲解工具.

4 结束语

现代软件的复杂性等特点决定了传统的测试策略,如白盒测试及黑盒测试已经不再适应了. 本文提出了一种面向操作的测试策略及相应的测试工具. 这些方法已在一个大型企业的一个复杂的多媒体、可视化、智能决策系统中应用,顺利地进行了测试计划的制定及测试工具的研制工作. 这种方法将使现代应用软件的测试更方便及更有效.

参考文献

- 1 Pressman Roger S. 软件工程. 北京:国防工业出版社,1988.
- 2 郑人杰. 计算机软件测试技术. 北京:清华大学出版社,1992.
- 3 栾军. 试验设计的技术与方法. 上海:上海交通大学出版社,1987.
- 4 玉溪卷烟厂信息管理与决策支持系统数据查询子系统概要设计说明书,1995.

TESTING STRATEGY AND TESTING TOOLS OF MODERN SOFTWARE

Zhou Zhiying Qian Ling

(Department of Computer Science Tsinghua University Beijing 100084)

Abstract There are many new features in modern software development. New demands are needed in software testing, such as more efficient, user-oriented etc. Traditional testing methods are unable to solve these problems. In this paper, the authors describe a new testing strategy to meet these demands, which is called application operating coverage based software testing(AOC). The fundamental analysis tools of AOC are navigation diagram and event record serials. At the end of this paper, they give an automatic testing tool for AOC.

Key words Software test, navigation diagram, logic coverage, (application) operation coverage, event record serials, testing tools.