

面向对象软件的测试

陈文宇*

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】 以软件工程中面向对象软件开发模式为参考,分别在面向对象分析,面向对象设计,面向对象编程的三个阶段中,依据各阶段的地位、作用、实现目标,具体阐述测试目的和应注意的测试点。依照传统的三个测试步骤:单元测试、集成测试、系统测试,借鉴传统测试方法有用的部分,提出有效的对面向对象软件进行测试的方法。结合传统的测试技术,针对面向对象技术新特性在测试中所引发的问题,提出一种测试模型。该模型简便、实用性强。

关键词 软件开发; 软件测试; 面向对象; 对象设计

中图分类号 TP311.56

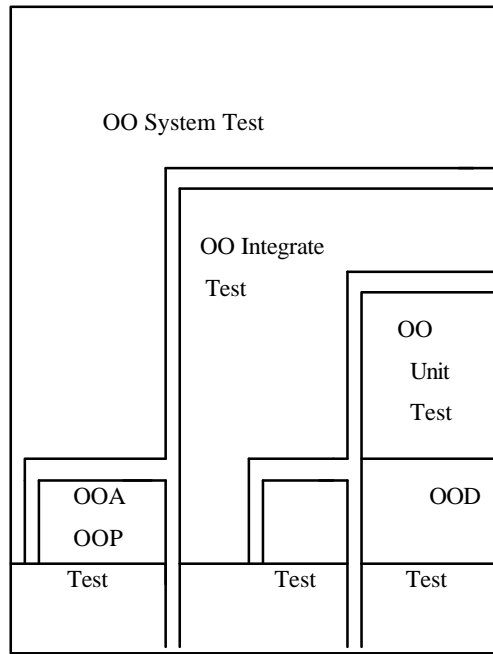
对面向对象技术所引入的新特点,传统的测试技术已经无法有效地进行测试。对面向对象软件的测试,其测试策略或方法都需作相应的变革或更新。广义的软件测试是由确认、验证、测试三个方面组成^[1]。面向对象技术是一种全新的软件开发技术,正逐渐代替广泛使用的面向过程开发方法,被看成是解决软件危机的新兴技术。面向对象技术产生更好的系统结构,更规范的编程风格,极大地优化了数据使用的安全性,提高了程序代码的重用。尽管面向对象技术的基本思想保证了软件应该有更高的质量,但实际情况却并非如此,因为无论采用什么样的编程技术,编程人员的错误都是不可避免的,而且由于面向对象技术开发的软件代码重用率高,更需要严格测试,避免错误的繁衍,故软件测试并没有因面向对象编程的兴起而丧失其重要性。

面向对象程序的结构不再是传统的功能模块结构,作为一个整体,原有集成测试所要求的逐步将开发的模块搭建在一起进行测试的方法已不适用。而且面向对象软件抛弃了传统的开发模式,对每个开发阶段都有不同以往的要求和结果,已经不能用功能细化的观点来检测面向对象分析和设计的结果。针对面向对象软件的开发特点,本文提出一种新的测试模型。

1 面向对象测试模型

面向对象的开发模型突破了传统的瀑布模型,将开发分为面向对象分析(OOA),面向对象设计(OOD)和面向对象编程(OOP)三个阶段。分析阶段产生整个问题空间的抽象描述,在此基础上,进一步归纳出适用于面向对象编程语言的类和类结构,最后形成代码。由于面向对象的特点,采用这种开发模型能有效地将分析设计的文本或图表代码化,不断适应用户需求的变动。针对这种开发模型,结合传统的测试步骤的划分,提出一种在整个软件开发过程中不断测试的测试模型,使开发阶段的测试与编码完成后的单元测试、集成测试、系统测试成为一个整体,测试模型如图 1 所示。

OOA Test 和 OOD Test 是对分析结果和设计结果的测试,主要是对分析设计产生的文本进行测试,是软件开发前期的关键性测试。OOP Test 主要针对编程风格和程序代码实现进行测试,其主要测试内容在面向对象单元测试和面向对象集成测试中体现。面向对象单元测试是对程序内部具体单一的功能模块的测试,如果程序是用 C++ 语言实现,主要就是对类成员函数的测试。面向对象单元测试是进行面向对象集成测试的基础。面向对象集成测试主要对系统内部的相互服务进行测试,如成员函数间的相互作用,类间的消息传递等。面向对象集成测试不但要基于面向对象单元测试,更要参见 OOD 或 OOD Test 结果^[2]。



OOP:面向对象编程

OOA Test:面向对象分析的测试

OOP Test:面向对象编程的测试

OO Integrate Test:面向对象集成测试

OOD Test:面向对象设计的测试

OO Unit Test:面向对象单元测试

OO System Test:面向对象系统测试

图 1 测试模型

面向对象系统测试是基于面向对象集成测试的最后阶段的测试,主要以用户需求为测试标准,需要借鉴 OOA 或 OOA Test 结果。

2 面向对象分析的测试

传统的面向过程分析是一个功能分解的过程,是把一个系统看成可以分解的功能的集合。这种传统的功能分解分析法的着眼点在于一个系统需要什么样的信息处理方法和过程,以过程的抽象来对待系统的需要。而面向对象分析(OOA)是把 E-R 图和语义网络模型,即信息造型中的概念,与面向对象程序设计语言中的重要概念结合在一起而形成的分析方法,最后通常是以得到问题空间的图表的形式描述^[3,4]。

OOA 直接映射问题空间,全面地将问题空间中实现功能的现实抽象化。将问题空间中的实例抽象为对象,用对象的结构反映问题空间的复杂实例和复杂关系,用属性和服务表示实例的特性和行为。对一个系统而言,与传统分析方法产生的结果相反,行为相对稳定,结构则相对不稳定,这更充分反映了现实的特性。OOA 的结果是为后面阶段类的选定和实现,类层次结构的组织和实现提供平台。因此,OOA 对问题空间分析抽象的不完整,最终会影响软件的功能实现,导致软件开发后期大量可避免的修补工作;而一些冗余的对象或结构会影响类的选定、程序的整体结构或增加程序员不必要的工作量。因此,对 OOA 的测试重点应该放在完整性和冗余性方面。OOA 阶段的测试划分为以下五个方面:1)对认定的对象的测试;2)对认定的结构的测试;3)对认定的主题的测试;4)对定义的属性和实例关联的测试;5)对定义的服务和消息关联的测试。

3 面向对象设计的测试

通常结构化的设计方法是用面向作业的设计方法,它把系统分解以后,提出一组作业,这些作

业是以过程实现系统的基础构造,把问题域的分析转化为求解域的设计,分析的结果是设计阶段的输入。

而面向对象设计(OOD)采用“造型的观点”,以 OOA 为基础归纳出类,并建立类结构或进一步构造成类库,实现分析结果对问题空间的抽象。OOD 归纳的类可以是对象简单的延续,也可以是不同对象的相同或相似的服务。OOD 是 OOA 的进一步细化和更高层的抽象。所以,OOD 与 OOA 的界限通常难以严格区分。OOD 确定类和类结构不仅能满足当前需求分析的要求,更重要的是通过重新组合或加以适当的补充,能方便实现功能的重用和扩充,以不断适应用户的要求^[5,6]。因此,对 OOD 的测试,建议针对功能的实现和重用以及对 OOA 结果的拓展,从如下三方面考虑:

- 1) 对认定的类的测试;
- 2) 对构造的类层次结构的测试;
- 3) 对类库的支持的测试。

其中对构造的类层次结构的测试通常基于 OOA 中产生的分类结构的原则来组织,着重体现父类和子类间一般性和特殊性。在当前的问题空间,对类层次结构的主要要求是能在解空间构造实现全部功能的结构框架。为此,测试如下几个方面:

- 1) 类层次结构是否包含了所有定义的类;
- 2) 是否能体现 OOA 中所定义的实例关联;
- 3) 是否能实现 OOA 中所定义的消息关联;
- 4) 子类是否具有父类没有的新特性;
- 5) 子类间的共同特性是否完全在父类中得以体现。

4 面向对象编程的测试

典型的面向对象程序具有继承、封装和多态的新特性,这使得传统的测试策略必须有所改变。封装是对数据的隐藏,外界只能通过被提供的操作来访问或修改数据,这样降低了数据被任意修改和读写的可能性,降低了传统程序中对数据非法操作的测试。继承是面向对象程序的重要特点,继承使得代码的重用率提高,同时也使错误传播的概率提高。继承使传统测试遇到了一个难题,即:对继承的代码究竟如何测试,多态性使得面向对象程序对外呈现出强大的处理能力,但同时却使得程序内“同一”函数的行为复杂化,测试时不得不考虑不同类型具体执行的代码和产生的行为。

面向对象程序是把功能的实现封装在类中,能正确实现功能的类,通过消息传递来协同实现设计要求的功能。正是这种面向对象程序风格,将出现的错误能精确的确定在某一具体的类。因此,在面向对象编程(OOP)阶段,忽略类功能实现的细则,将测试集中在类功能的实现和相应的面向对象程序风格,主要体现为以下两个方面:

- 1) 数据成员是否满足数据封装的要求;
- 2) 类是否实现了要求的功能。

5 面向对象的单元测试

传统的单元测试是针对程序的函数、过程或完成某一定功能的程序块。沿用单元测试的概念,实际测试类成员函数。一些传统的测试方法在面向对象的单元测试中都可以使用。如等价类划分法,因果图法,边值分析法,逻辑覆盖法,路径分析法,程序插装法等等。

用于单元级测试进行的测试分析(提出相应的测试要求)和测试用例(选择适当的输入,达到测试要求),规模和难度等均远小于后面将介绍的对整个系统的测试分析和测试用例,而且强调对语句应该有 100%的执行代码覆盖率。在设计测试用例选择输入数据时,可以假设:

- 1) 如果函数(程序)对某一类输入中的一个数据正确执行,对同类中的其他输入也能正确执行;
- 2) 如果函数(程序)对某一复杂度的输入正确执行,对更高复杂度的输入也能正确执行。在面向对象程序中,类成员函数通常都很小,功能单一,函数的间调用频繁,易出现一些不宜发现的错误。

因此,在做测试分析和设计测试用例时,应该注意面向对象程序的这个特点,仔细地进行测试分析和设计测试用例,尤其是针对以函数返回值作为条件判断选择,字符串操作等情况。

面向对象编程的特性使得对成员函数的测试,又不完全等同于传统的函数或过程测试。尤其是继承特性和多态特性,使子类继承或过载的父类成员函数出现了传统测试中未遇见的问题。需做以下的考虑:

1) 继承的成员函数是否都不需要测试

对父类中已经测试过的成员函数,两种情况需要在子类中重新测试:继承的成员函数在子类中做了改动;成员函数调用了改动过的成员函数的部分。

2) 对父类的测试是否能照搬到子类

只需在父类测试要求和测试用例上添加对子类函数的新的测试要求和增补相应的测试用例。

多态有几种不同的形式,如参数多态、包含多态和过载多态。包含多态和过载多态在面向对象语言中通常体现在子类与父类的继承关系,包含多态虽然使成员函数的参数可有多种类型,但通常只是增加了测试的繁杂。对具有包含多态的成员函数测试时,只需要在原有的测试分析和基础上扩大测试用例中输入数据的类型。

6 面向对象的集成测试

传统的集成测试是由底向上通过集成完成的功能模块进行测试,一般可以在部分程序编译完成的情况下进行。而对于面向对象程序,相互调用的功能是散布在程序不同的类中,类通过消息相互作用申请和提供服务。类的行为与它的状态密切相关,状态不仅仅是体现在类数据成员的值,还包括其他类中的状态信息。由此可见,类相互依赖极其紧密,根本无法在编译时对类进行测试。所以,面向对象的集成测试通常需要在整个程序编译完成后进行。此外,面向对象程序具有动态特性,程序的控制流往往无法确定,因此也只能对整个编译后的程序做基于黑盒子的集成测试。

面向对象的集成测试能够检测出相对独立的单元测试无法检测出的那些类相互作用时才会产生的错误。基于单元测试对成员函数行为正确性的保证,集成测试只关注于系统的结构和内部的相互作用。面向对象的集成测试可以先进行静态测试,再进行动态测试。

静态测试主要针对程序的结构进行,检测程序结构是否符合设计要求,提供一种称为"可逆性工程"的功能,即通过原程序得到类关系图和函数功能调用关系图,将"可逆性工程"得到的结果与OOD的结果相比较,检测程序结构和实现上是否有缺陷,即通过这种方法检测 OOP 是否达到了设计要求。

动态测试设计测试用例时,通常需要上述的功能调用结构图、类关系图或者实体关系图为参考,确定不需要被重复测试的部分,从而优化测试用例,减少测试工作量,使得进行的测试能够达到一定覆盖标准。测试所要达到的覆盖标准可以是达到类所有的服务要求或服务提供的一定覆盖率,依据类间传递的消息,达到对所有执行线程的一定覆盖率,达到类的所有状态的一定覆盖率等。同时也可以考虑使用现有的一些测试工具来得到程序代码执行的覆盖率。

值得注意的是设计测试用例时,不但要设计确认类功能满足的输入,还应该有意识的设计一些被禁止的例子,确认类是否有不合法的行为产生,如发送与类状态不相适应的消息,要求不相适应的服务等。根据具体情况,动态的集成测试,有时也可以通过系统测试完成。

7 面向对象的系统测试

通过单元测试和集成测试,仅能保证软件开发的功​​能得以实现。但不能确认在实际运行时,它是否满足用户的需要,是否大量存在实际使用条件下会被诱发产生错误的隐患。为此,对完成开发的软件必须经过规范的系统测试,即开发完成的软件仅仅是实际投入使用系统的一个组成部分,需要测试它与系统其他部分配套运行的表现,以保证在系统各部分协调工作的环境下也能正常工作。

系统测试应该尽量搭建与用户实际使用环境相同的测试平台,保证被测系统的完整性,对临

时没有的系统设备部件, 也应有相应的模拟手段。系统测试时, 应该参考 OOA 分析的结果, 对应描述的对象、属性和各种服务, 检测软件是否能够完全"再现"问题空间。系统测试不仅是检测软件的整体行为表现, 从另一个侧面看, 也是对软件开发设计的再确认。系统测试需要对被测的软件结合需求分析做仔细的分析, 建立测试用例。

8 结束语

本文依照传统的单元测试, 集成测试, 系统测试三个测试步骤, 借鉴传统测试方法有用的部分, 提出有效的对面向对象软件进行测试的方法, 得到了一个模型, 该模型简便, 实用。

参 考 文 献

- 1 蔡希尧 陈平. 面向对象技术. 西安: 西安电子科技大学出版社, 1993
- 2 Coad, P, Yourdon, E. Object-Oriented Analysis, Yourdon Press, 1999
- 3 汤庸. 软件工程方法学及应用. 武汉: 中国三峡出版社, 1998
- 4 郑人杰. 计算机软件测试技术. 北京: 清华大学出版社, 2000
- 5 Su Sen, Tan Xuefei, Liu Jinde. Object-oriented interoperability technology. Journal of University of Electronic Science and Technology of China 1998, 27(1): 90~94 [苏森, 唐雪飞, 刘锦德. 面向对象的互操作技术. 电子科技大学学报 1998, 27(1): 90~94]
- 6 Ding Zhiqinag. A reuse approach based on object data flow diagram. Journal of University of Electronic Science and Technology of China 1999, 28(3): 306~310 [丁志强. 基于对象数据流图的可复用方案. 电子科技大学学报, 1999, 28(3): 306~310]

Object-Oriented Software Test

Chen Wenyu

(College of Computer Science and Engineering, UEST of China Chengdu 610054)

Abstract This paper describes and discusses about how to test the object-oriented software at the three phases of object-oriented analysis(OA), object-oriented design(OOD) and object-oriented programming(OOP) while taking the object-oriented software development model in software engineering as reference. Following and taking some useful methods of the traditional three testing steps, unit test, integrated test and system test, an effective method for object-oriented software test is proposed. In associate with the traditional testing technology, and as a solution for the problems caused by the new properties of object-oriented technology in testing, a new testing model is also presented. The new testing model is convenient and functional for software test.

Key words software development; software test; object-oriented