

# 基于组件的软件系统自动化测试研究

赵再军

(台州职业技术学院 台州 318000)

**摘要** 本文重点研究基于组件的软件系统自动化测试问题。在对现有自动化测试框架进行总结的分析的基础上,提出了一个基于组件的软件系统自动化测试过程模型,将软件的自动化测试过程有机融入软件开发过程中,分析了每个测试阶段的主要任务和应该注意的问题,实现了不同测试框架的整合应用。

**关键词** 组件,软件系统,测试,过程模型

## Study on the Automatic Testing of Software System Based on Components

ZHAO Zai-Jun

(Taizhou College of Vocational Technique, Taizhou 318000)

**Abstract** In this paper, the automatic testing of software system based on components is studied. A new procedure model of automatic testing of software system based on components is proposed, with summarizing the current automatic testing frames. This testing procedure is merged into the development of software system and the main tasks of every stage are analyzed. Using this model, the integrated application of various testing frames is complimented.

**Keywords** Components, Software system, Testing, Procedure model

继面向对象的设计方法之后,基于组件的软件开发是近年来兴起的一种新的软件开发方式。组件是近乎独立的、可替换的、满足一定功能的模块,通常是经过编译的二进制代码或机器码,组件之间通过接口进行通信。它具备以下特点:组件是封装了数据和实现方法的软件模块,只有接口是可见的,只能通过接口与外界交互;组件可以增加新的接口,但原有接口的定义不能改变;组件是经过编译的二进制代码,对使用者源代码是不可见的;系统中的组件是可替换的,不需要重新编译。由于组件的这些特点,使得基于组件的软件开发更具优势。基于组件的软件就是由各类组件组成,通过相互协同、合作一起实现特定功能的软件系统。在实际应用中,由于对软件系统的弹性及可维护性的要求越来越高,基于组件的软件开发方式有着开发成本较为低廉、开发周期短、维护简便等优点,因此得到了软件开发业的重视。

### 1 基于组件的软件系统测试

任何软件系统在开发完毕正式投入使用之前不可避免地要涉及到测试的问题,基于组件的软件系统也不例外。软件测试就是在软件投入运行前,对软件需求分析、设计规格说明和编码的最终复审,是软件质量保证的关键步骤。软件测试是为了发现错误而执行程序的过程。或者说,软件测试是根据软件开发各阶段的规格说明和程序的内部结构而精心设计一批测试用例(即输入数据及其预测的输出结果),并利用这些用例去运行程序,以发现程序错误的过程。软件开发的过程是一自顶向下,逐步细化的过程,而测试过程则是依相反的顺序安排的自底向上,逐步集成的过程。软件测试的直接目的是发现软件中存在的缺陷,进而改正软件中的错误、弥补缺陷、完善功能,从而保证开发软件的质量、性能。

随着软件组件技术的发展,人们意识到基于组件的软件

产品的质量是依赖于软件组件的质量和有效的软件测试过程。组件所具有的特点决定了基于组件开发出来的软件系统适合用各种自动化测试方法进行软件测试。按照组件化程序开发的思想,复杂的应用程序被设计成一些小的、功能单一的组件模块,这些模块可以运行在同一机器上,也可以运行在不同的机器上,每台机器的运行环境可以不同,甚至可以是不同的操作系统。为了实现这样的应用软件,组件需要一些细致的规范,只有组件程序遵守这些共同的规范,组件软件才能正常运行。组件的可测试性是决定组件质量的一个重要因素,建立具有良好可测试性的程序和组件总是可以简化测试操作,减少测试成本并提高软件质量。一个理想的可测试软件组件不仅是可配置的和可执行的,而且支持标准的组件测试设备。可测试的组件必须有一组内置的接口,便于和一些定义明确的测试设备交互。这样可以节约测试人员花在组件测试方面的时间精力,并且有助于控制组件的可跟踪性。

### 2 软件系统的自动化测试框架简述

在目前的软件测试领域,自动化测试手段已经逐渐取代手工方式,成为主流测试方法。这主要是由于自动化测试方法具有效率高、成本低、效果好、可以复用等优点。综观学术界对软件自动化测试领域的研究情况,目前已经形成了几种颇有代表性的自动化测试框架。

#### (1) 基于数据驱动的自动化测试框架

基于数据驱动的自动化测试框架是从某个数据文件中读取输入、输出的测试数据,然后通过变量传入事先录制好的或手工编写的测试脚本中。其中,这些变量被用作传递(输入输出)用来验证应用程序的测试数据。在此过程中,数据文件的读取、测试状态和所有测试信息都被编写进测试脚本里;测试数据只包含在数据文件中,而不是脚本里,测试脚本只是一个

“驱动”，或者说是一个传送数据的机制。数据驱动的自动化测试利用相同的测试过程测试不同的输入、输出组合。它将测试输入、预期输出组织为表，表中的一行对应一个测试。然后创建一个从表中逐行读入的自动化测试过程，执行每个输入步骤，并检验预期结果。当把数据驱动测试过程放在一起后，就可以反复使用该过程来执行新测试。这种方法对于具有很多不同数据选项的应用系统来说是非常有效的。

### (2) 基于功能驱动的自动化测试框架

基于功能驱动的自动化测试框架是将整个系统的所有功能划分成为边界清晰、功能独立的应用模块。这些应用模块将是自动化测试实现首先要自动化的部分。所有这些划分出来的应用模块将被测试脚本语言实现为一个个的步骤，而所有这些自动化的测试仅仅是可实现的测试载体。代表着一个个独立的测试用例的测试体将被实现为一个包含测试相关信息的测试函数，这个测试函数将根据测试的具体需要去调用代表一个个独立功能模块的应用模块，读取驱动这些应用步骤所需要的测试数据。所有和系统界面相关的操作和输入，都被封装到应用模块中，而这些应用模块，由于是与系统的独立功能紧密相联，因此维护和开发都相对独立。在整个基于功能驱动的自动化测试框架中，只有应用模块库中的一些实用脚本真正和应用系统发生了交互。其他的一些控制脚本或者驱动脚本，都是独立于待测系统而独立存在的。从另外一个角度说，就是所有与待测系统有关的系统行为和属性，在该框架中被应用模块脚本所封装。这样做的一个显而易见的好处就是，当待测系统发生了界面的改变，测试人员仅仅需要去修改被影响应用模块的脚本，其它的所有脚本可以保持稳定。

### (3) 基于指令驱动的自动化测试框架

学术界普遍认为，到目前为止，比较理想的自动化测试框架是基于指令驱动的自动化测试。基于指令驱动的自动化测试的整个过程所包含的功能都是由指令驱动的，指令控制了整个测试过程。它是基于数据驱动自动化测试框架的改进，可支持由不同序列或多个不同路径组成的测试。基于指令驱动的自动化测试框架建立在数据驱动方法之上，但表中包含指令，而不只是数据。它通常独立于应用程序的自动化测试框架，可以用来处理我们的自动化测试，并且也可以适应手工测试。这些测试被开发成使用指令的数据表，它们独立于执行测试的自动化工具，是对数据驱动的自动化测试的有效改进和补充。

## 3 基于组件的软件系统自动化测试过程模型

上述三种自动化测试框架分别从不同的角度提出了软件自动化测试的基本思想，也都有各自的适用范围。但现有的这些测试框架忽视了软件测试是一项复杂的活动这一事实，仅把测试工作视为编码之后的工作，并对整个测试采用串行方式进行，效率很低。因此，要想减少测试在开发过程中所占的时间，提高测试的效率和质量，测试方案的设计应该在需求阶段就开始，而不是在传统的编码之后，并且软件测试进程应该是和软件开发的开发进程并行进行的。对于基于组件的软件系统来说就更是如此，因为组件之间不是孤立的，而是具有或紧或密的联系的。也就是说，组件之间是耦合在一起的。在这种情况下，无论是数据驱动还是功能驱动、指令驱动都会受到一定的限制，从而影响测试效果。事实上，基于组件的软件开发过程可以抽象为根据目标功能选择各类组件——对组件进行装配——子模块集成——系统集成等几个步骤。基于

此，本文提出了一种基于组件的软件系统自动化测试过程模型，将组件以及软件系统的测试过程有机融入系统的开发过程中，达到软件测试与软件开发同设计、同实施、同验收的目的。此过程模型的示意图如图1所示。

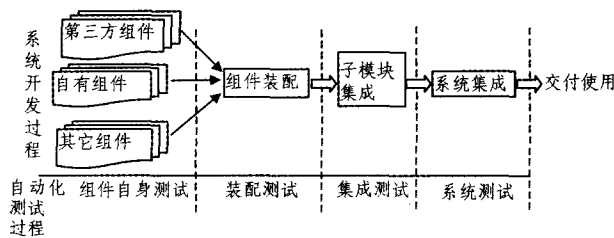


图1 基于组件的软件系统自动化测试过程模型

### (1) 组件自身测试

该测试是针对软件设计的最小单位——组件，进行正确性检验的测试工作，其目的在于发现各组件内部可能存在的各种差错，确保每个组件都能有效实现自己的预期功能。这一步是为了在选择组件阶段把好组件“准入”关，无论是自己开发组件、第三方组件或是其它来源的组件，都要对每一个组件进行测试。组件自身测试是典型的黑盒测试，即在不考虑代码内部结构的前提下，根据需求对应用程序进行测试。它的任务是验证组件的功能及其它特征是否与预期的要求一致。组件测试时应充分考虑到该组件可能涉及到的尽可能多的内外环境因素。

### (2) 装配测试

在组件自身测试的基础上，需要将部分组件按照功能设计要求装配成子系统时进行阶段性测试。此步骤的主要内容就是充分利用组件的接口功能，实现两两组件之间的协同工作。这是一个组件由少到多的聚集过程，当有新组件加入到子系统中时，必须继续进行装配测试。由于组件之间的耦合性，装配测试可能要反复多次进行。

### (3) 集成测试

集成测试是在形成子模块的过程中进行的，将由已测试过的组件集成的子模块作为一个实体进行测试。此阶段的测试目的就是利用组件间的耦合关系来达到对子模块集成测试的目的。组件间的耦合关系主要产生于组件之间的依赖、通信以及组件之间的协作等。

### (4) 系统测试

系统测试是将通过子模块集成测试的软件，作为整个计算机系统的一个元素，与计算机硬件、外设、支持软件等结合起来，在实际运行环境下，对计算机系统进行一系列的综合测试。这也是自动化测试的最后步骤，将直接关系到软件系统的可用性。需要指出的是，对软件系统所作的任何修改以后都必须重新进行系统测试。

## 4 对基于组件的软件系统进行自动化测试需要注意的问题

### (1) 自动化测试系统的自身质量

自动化测试对待测系统的依赖性比较大，待测系统的一些微小改变可能使整个自动化测试系统失效。一个健壮和易维护的自动化测试模型的中心思想是以测试数据为核心，以数据驱动自动化测试为基础的一个自动测试框架。其基本前提是数据驱动测试技术的运用并体现出待测系统的特征。因此可以用一些自动化系统测试技术来建立更加健壮的自动化

系统测试,但比起手工测试,自动化系统测试更加依赖于软件系统。另一方面,自动化测试系统本质上只是一种工具,工具只能判断实际结果与期望结果之间的差别。自动化测试结束后,必须对自动化系统测试的输出结果进行分析,判断其正确性。因此,确保自动化测试系统自身的质量比自动化系统测试本身更加重要。

### (2) 测试过程中的安全性

自动化测试过程的安全性问题包括两个方面的内容。一是用来验证集成在系统内的保护机制是否能够在实际中保护系统不受非法的侵入。二是在测试过程中,由于错误的关联性,并不是所有的组件缺陷都能够得以修复。某些组件缺陷虽然能够修复但在修复的过程中可能会引入新的缺陷。很多组件缺陷之间是相互矛盾的,一个矛盾的消失可能会引发另外一个矛盾的产生,如我们在解决通用性的缺陷后往往会带来执行效率上的缺陷。更何况在缺陷的修复过程中,还要受到时间、成本等方面的限制,因此无法有效、完整地修复所有的软件缺陷。因此评估软件缺陷的重要度、影响范围,选择一个折中的方案或是从非软件的因素(比如提升硬件性能)考虑软件缺陷成为实施自动化测试过程中需要重点考虑的问题。

### (3) 测试用例的选择

测试用例是指为实施一次测试而向被测系统提供的输入数据、操作或各种环境设置。测试用例驱动着软件自动化测试的执行过程,它是对测试大纲中每个测试项目的进一步实例化。无论测试自动化做得怎样成功和出色,如果测试案例本身是失败的,那么测试结果也变得毫无意义。因此,慎重选择测试用例是直接关系到自动化测试能否取得预期效果的先决条件。选择测试用例时,应当首先关注其代表性,即能够代表各种合理和不合理的、合法的和非法的、边界和越界的,以及极限的输入数据、操作和环境设置等,一个好的自动化测试

用例必须具有较强的针对性,能够体现自动化测试的效率。其次要使测试执行结果的正确性是可判定的或可评估的,以后对同样的测试用例,系统的执行结果应当是相同的。

**结束语** 软件测试是一项复杂的工作,是一门综合的学科,需要不断地进行探索。由于基于组件的软件开发本身还存在一些未能解决的问题,需要不断发展和完善,因此对基于组件的软件测试这项工作更需要进一步研究。本文提出的基于组件的软件系统自动化测试过程模型充分考虑了软件开发过程与测试过程应该并行进行的思想,将不同测试阶段可以使用的各种自动化测试框架和方法整合起来,融入软件开发的全过程,有效地提高了组件软件系统的自动化测试效果。当然,组件软件测试标准虽然已经有了初稿,但其中的不完善之处是显而易见的,通用的组件软件测试方法也需要明确和进一步开发。所有这些问题决定了组件软件技术在展示它巨大魅力的同时,也向人们提出了一个一个的挑战。相信随着自动化测试和自动测试框架的成熟,开发和维护测试将节省出大量的时间用于提高测试覆盖率和测试本身的质量,从而大幅度提高基于组件的软件系统的运行质量。

## 参考文献

- Marilyn L. A Fault Taxonomy for Component-Based software; [Technical report]. University of Study Milano-Bicocca Italy, 2002
- 郑人杰. 实用软件工程(第二版). 北京:清华大学出版社,2003
- 张克东,庄燕滨. 软件工程与软件测试自动化教程. 北京:电子工业出版社,2002
- 马瑞芳. 计算机软件测试方法的研究. 小型微型计算机系统, 2001,24(12):2211~2213
- King S, Hammond J, Chapman R. Is Proof More Cost Effective Than Testing? IEEE Transactions on Software Engineering, 2000,26(8):675~686
- International Conference on the Unified Modeling Language (UML '00), 383~395
- Li Liu-ying, Qi Zhi-chang. Test selection from UML statecharts technology of object oriented languages and systems [ C ]. In: 1999, TOOLS 31, Proceedings, Sept. 1999, 273~279
- Li Liu-ying, Wang Ji, Qi Zhi-chang. A test cases generation method for UML statecharts [ J ]. Journal of Computer Research and Development, 2001, 38(6):691~697 (in Chinese)
- Li Liu-ying. Research and implementation of testing techniques for UML; [Phd dissertation]. National University of Defense Technology, Changsha, 2000 (in Chinese)
- Kim Y G, Hong H S, Bae D H. Test cases generation from UML state diagrams. Software, IEE Proceedings-[ see also Software Engineering, IEE Proceedings ], 1999, 146(4): 187~192
- Chevalley P, Thevenod-Fosse P. Automated Generation of Statistical Test Cases from UML state Diagrams [ C ]. In: Proceedings of the 25<sup>th</sup> Annual International Computer Software and Applications Conference, 2002
- Gnesi S, Latella D, Massink M. Formal Test-case Generation for UML Statecharts. In: Proceedings of the Ninth IEEE International Conference on Engineering Complex Computer Systems Navigating Complexity, 2004
- Lee D, Yannakakis M. Principles and Methods of Testing Finite State Machines - A survey. Proceedings of the IEEE, 1996, 84(8):1099~1123
- The Agedis Project. The Agedis Home Page, 2003. <http://www.agedis.de/index.html>
- 张涌,钱乐秋,王渊峰. 基于确定有限状态机的测试输入序列的选取. 计算机研究与发展, 2002(9): 1144~1150
- 董威,王戟,齐治昌. UML Statecharts的模型检验方法. 软件学报, 2003(14):750~756
- 颜炯,王戟,陈火旺. 基于UML的软件Markov串使用模型构造研究. 软件学报, 2005(16):1386~1393
- 王林章,李宣东,郑国梁. 一个基于UML协作图的集成测试用例的生成方法. 电子学报, 2004(8):1290~1296

(上接第267页)

测试用例,如何减少测试用例生成的数量是测试方法主要研究的方面。另一方面,UML状态图也在不断发展,不断有新的元素加入进来,对这些新加入的特性也要进行测试,对测试技术提出了新的要求。

基于UML状态图的测试是测试技术的一个热点研究课题,在实际中有着广阔的应用,发展潜力巨大。对它的研究也在不断完善中。本文是对现有的研究成果和应用的一个综述,主要讨论了基于UML状态图测试的理论基础、现有的测试技术和测试工具的开发。

## 参考文献

- 单锦辉,姜瑛,孙萍. 软件测试研究进展. 北京大学学报, 2005, 41(1):134~145
- Chow T S. Testing software design modeled by finite state machines. IEEE Trans on Software Engineering, 1978, 4(3):178~187
- Bogdanov K, Holcombe M, Singh H. An automated Test Set Generation for Statecharts. In: Lecture Notes in Computer Science 1641. Berlin; Springer, 1999. 107~121
- Offutt A J, Abdurazik A. Generating tests from UML specifications [ C ]. In: Second International Conference on the Unified Modeling Language, UML '99, 1999
- Offutt A J, Xiong Y, Liu S. Criteria for generating specification based tests [ C ]. In: Proceedings of Fifth IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'99), Las Vegas, Nevada, USA, October 18221, 1999. 119~129
- Abdurazik A, Offutt J. Using UML collaboration diagrams for static checking and test generation. In: Proceedings of the Third