

# 基于构件软件系统集成测试的初步研究

张书杰<sup>1</sup>, 于学军<sup>1</sup>, 阎健卓<sup>2</sup>, 于 华<sup>3</sup>

( 1.北京工业大学 计算机学院, 北京 100022; 2.北京工业大学 电子信息与控制工程学院, 北京 100022;  
3.长春大学 计算中心, 长春 130022 )

**摘 要:** 针对构件软件系统存在着构件源代码无法获得, 构件本身性能无法进行完全测试等问题, 提出了构件化软件系统集成测试模型, 并提出了运用统一建模语言(unified modeling language, 简称UML)建立多种视图(如顺序图、合作图等)实现组件依赖关系的建模, 从而为构件化软件系统级测试提供了可行的测试源。

**关键词:** 构件; 软件集成测试; 统一建模语言(UML)

**中图分类号:** TP 311.53

**文献标识码:** A

**文章编号:** 0254-0037(2004)02-0224-03

随着商用的中间件组件的大量发展, 使得组件技术进入了一个“可重用的应用框件和平台”阶段。其中包括 SUN Microsystems 的 EJB 模型, 微软公司的 COM+ 模型等等, 所有这些方法都提供了好的开发过程、分析和设计模型以及标准。在基于软件构件的软件系统构造过程中, 共有 3 种构件可能被用到: ①第三方组件; ②自己已研制的组件; ③新开发的组件。

正是由于上述 3 种组件的存在, 对这样的软件系统质量度量有很大的影响, 为构件化软件系统的测试提出了新的挑战: 能否使用现有的测试技术及模型完成可重用组件的验证? 如何验证这些可重用组件的易用性、交互性和可定制性等特性<sup>[1]</sup>? 如何验证这些可重用性组件的性能? 这些问题都将是基于组件的软件系统测试过程中要考虑的新问题。

## 1 基于构件的软件系统

截至 2003 年, 大约 70% 的软件系统都是基于构件技术的<sup>[2]</sup>。目前有多种关于构件的定义<sup>[3]</sup>: 1) 可重用组件是指表示单一逻辑的松散的模块——Gready Booch。2) 软件组件是指有明确的接口和上下文依赖关系描述的独立单元。一个软件组件可以被独立分发并被第三方使用。基于组件软件开发方法将一个软件系统的开发过程分为应用系统分析与设计、组件的开发以及组件的装配 3 个阶段。这样的软件系统在进行测试时, 常常会遇到很多的问题, 如: 这些组件由不同编程语言开发出来; 大多需要运行在不同的平台上; 许多组件的源码并不公开等等。因此, 对基于构件的软件系统进行充分的集成测试更为重要。

## 2 测试模型

现有的传统软件测试所采用的测试模型<sup>[4]</sup>(见图1)主要适用于从始至终进行的测试过程, 其中的主要的测试源为需求规格说明书及程序源代码。在基于组件化的软件开发过程中, 这种模型在某种程度上遇到了困难, 最典型的就是测试人员无法得到组件的源码是如何进行语句覆盖性测试的。因此, 需要将上述的测试模型加以调整以适应构件化的软件系统测试的要求。

在现有的传统测试模型的基础上结合基于组件的软件系统结构的构成特点, 笔者提出如图 2 所示的

基于构件的软件系统的测试模型。当进行构件化软件系统的软件测试时,假定每一个组件都经过了充分的软件测试,也就是说完成了组件级的测试以及完成了组件选择测试。

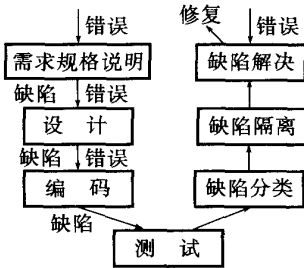


图1 现有的软件测试模型

Fig.1 Current software testing model

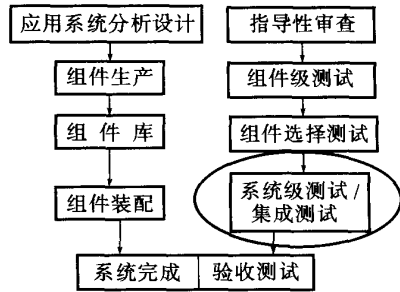


图2 基于构件软件系统的测试模型

Fig.2 Component-based software testing model

由于假设组件内部已经通过了测试,这样基于构件的软件系统的可靠性关键就在于这些构件的交互,也就是组件的装备。按照构件化软件系统的构成原理,构件之间的交互可以通过直接的方式也可以通过间接的方式。所谓直接交互就是通过接口(interface)实现构件之间的交互,而间接交互就是通过消息传递的方式来实现交互。因此,在考虑基于构件化软件的系统测试时,下面几个因素则不可避免地要考虑到:

- 1) 接口(interface)。大多情况下对组件的使用是通过接口完成的。因此,测试每个组件接口是非常必要的。为了充分地实现接口的测试工作,要求接口的设计是可测试并可追踪的。
- 2) 事件(events)。在不同的场景中,不同的组件可以有一个相同的接口,不同的组件激活一个相同的接口可能会得到不同的结果。与此同时,一些不是由接口触发的消息也可能对组件造成很大的影响,因此也要对这些事件进行测试。
- 3) 上下文关系(context-dependence relationship)。不同的上下文对于组件的使用和输出也是非常有影响的。因此,基于构件的软件的系统测试对不同上下文关系也要进行覆盖测试。当进行组件的集成时,程序员往往把精力主要放在了两个因素上,即接口和事件。但这些接口和事件将如何进行交互,在集成之后会有哪些潜在的风险,却往往被忽视。通过对前后依赖关系的建模研究,可以考察组件接口、事件的交互依赖关系。在实践过程中对组件依赖关系的建模可以利用UML(unified modeling language)的方法进行,如图3所示。图3完成的是学生注册系统的某一个成功场景,其中ICourseCatalog是一个用于连接系统外部组件的接口。

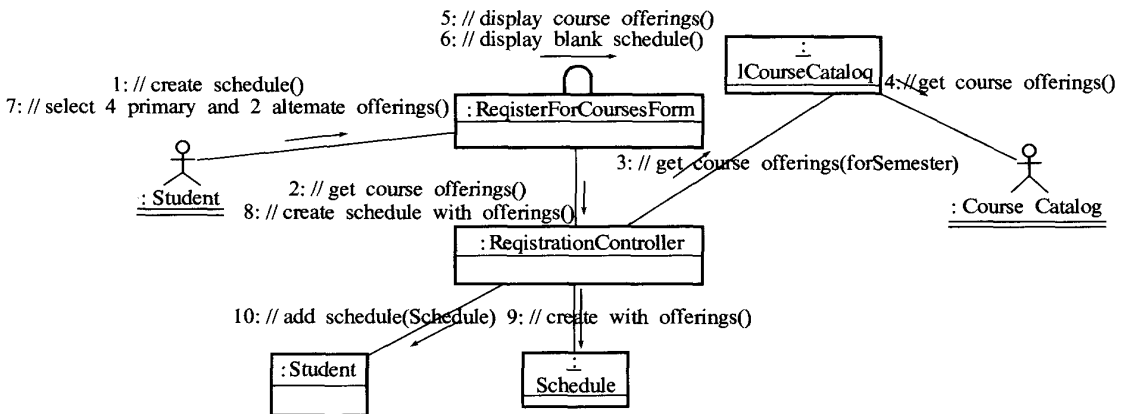


图3 利用UML合作图完成组件依赖关系的建模

Fig.3 Components dependency modeling by UML collaboration diagram

4) 数据依赖关系 (content-dependence relationship). 假如定义在接口 ( $I_1$ ) 中的一个方法中使用的数据和定义在另外一个接口 ( $I_2$ ) 中的一个方法中的数据存在依赖关系, 则两个接口的使用顺序不同也会影响到组件的使用结果.

### 3 结束语

作者就基于组件的软件在测试过程中要考虑的因素进行了阐述, 并介绍了运行 UML 的不同建模技术, 如采用顺序图和合作图可以对测试因素进行描述, 这些都有助于在测试过程中, 尤其在源代码无法获得的情况下进行全面测试以保证系统的质量. 在基于组件化软件系统测试方面还存在着很多有待解决的问题, 比如说如何对数据依赖关系建模, 如何对组件化软件系统的测试进行量化等等, 这些也是作者目前正在研究和考虑的问题.

#### 参考文献:

- [1] WEYUKER E J. Testing component based software: A cautionary tale[J]. Computer, 2001, 15(5): 54-59.
- [2] WU Ye, CHEN Mei-hwa, OFFUTT J. UML-based integration testing for component-based software[Z]. Ottawa: The 2nd International Conference on COTS-Based Software Systems (ICBSS), 2003.
- [3] LU Guo-jun. Communication and Computing for Distributing Multimedia Systems[M]. MA: Attach House, 1996.
- [4] JORGENSEN P C. 软件测试[M]. 第2版. 北京: 机械工业出版社, 2003.  
JORGENSEN P C. Software Testing[M]. 2nd ed. Beijing: China Machine Press, 2003. (in Chinese)

## Study on the System Integrating Testing of Component Software

ZHANG Shu-jie<sup>1</sup>, YU Xue-jun<sup>1</sup>, YAN Jian-zhuo<sup>2</sup>, YU Hua<sup>3</sup>

( 1.College of Computer Science, Beijing University of Technology, Beijing 100022, China;

2. College of Electronic Information and Control Engineering,  
Beijing University of Technology, Beijing 100022, China;

3.Computation Center, Changchun University, Changchun 130022, China )

**Abstract:** Owing to such difficulties as unable to achieve component source code and no way to test completely the component performance, the authors propose the software integrating testing model for component-based software system and use the unified modeling language to establish the model of kinds of UML views to reflect the component dependent relationship, such as sequence diagram and collaboration diagram. This model provides the testing means for component software system.

**Key words:** component; software integrating testing; unified modeling language(UML)