文章编号:1671-4598(2020)10-0072-04 DOI:10.16526/j.cnki.11—4762/tp.2020.10.015 中图分类号:TP311.1 文献标识码:A

基于组件的测试软件定制开发方法

秦振汉，林 渊，胡广明，郭双红
（航天科工惯性技术有限公司，北京 100074）

摘要：在特定领域的测试软件项目中，普遍存在着整体上基本相同或相似，但在个体上存在差异的情况；目前，在软件开发中使用的各种重用技术和方法，能够解决很多普遍性问题，但在面对个性问题时，开发效率下降，无法快速响应用户的个性化要求；为此，提出了一种基于组件的测试软件定制化开发方法；通过组件重用，解决项目中的共性问题；针对差异性需求，提供了标准化的个性组件，其内部预设了可变点，并提供了相应的变化机制和常用的实现方案；开发人员以个性组件为原型，通过对可变点的功能扩展，开发出全新的定制化组件；实际应用表明，在针对用户的多样性需求时，该方法既保证了开发的稳定性，同时更为敏捷和高效，满足软件产品的定制化开发需要。

关键词：组件；软件定制；可变点；测试软件

Customizable Development of Test Software Based on Component

Qin Zhenhan, Lin Yuan, Hu Guangming, Guo Shuanghong
(Aerospace Science & Industry Inertial Technology Co., Ltd., Beijing 100074, China)

Abstract: Here is a general situation that the whole is basically the same or similar but there are individual differences during the development of test software. The kinds of technologies and methods used in the software development could approach the common problems although lack of quick responsibility due to the drop efficiency when faced with personality problems. Paper put forward a component-based customized development method. Common problems can be solved by component reuse. The standardized personality components with variable points inside are offered according to the different requirements, in the meantime the corresponding change mechanisms and common implementation schemes are provided. Developers develop new customized components by extending the functions of variable points based ontaking personality components as prototypes. The practical application shows that this method not only ensures the application software with good reusability and development quality, but also with the flexibility to meet the needs of test software products customization development.

Keywords: component; software customization; variation point; test software

0 引言

测试软件是自动化测试系统的核心组成部分之一，与测试设备硬件开发的工业化、集成化相比，测试软件的开发中普遍存在开发周期长、重复开发、复用性差、可维护性和适应性低等问题[1-2]，不能很好地满足用户需求变化和个性化开发的需要。尤其是当被测产品处于研发、试验或小批量试制阶段时，情况更为明显。此外，在软件的全寿命周期中，不同的用户，包括产品设计人员、产线人员、工艺人员、相关管理人员等，都会根据各自的使用背景，提出不同的软件开发、改进和升级要求。

目前，测试软件开发中普遍使用了诸如开发平台、架构重用、组件复用等技术和方法，可以快速搭建测试软件的基础原型，能够解决开发中遇到的很多普遍性问题。但在解决个性化需求时，已有的平台开发、组件开发等方式显得过于僵化，灵活性不足。为实现这些特定需求，开发人员需要不断修改已有的软件产品，甚至大幅调整现有的体系结构，给后续的修改和维护带来了很多隐患。实际工程经验表明，在很多项目中，大部分时间都消耗在针对少数几个定制需求所进行的软件开发和调试工作上，同时，软件交付后暴露的问题也往往集中在这些特殊需求上。

本文提出了一种基于组件的测试软件定制化开发方法。通过对实践经验的总结，建立针对定制开发的原型化组件，在内部引入“可变点”的概念，标明了组件内部的变化热点，并提供了标准的开发流程和变化机制。通过这种方式，使得原型组件具有了很强的适应能力，可依据不同的软件需求，灵活选择必要的可变点并进行针对性开发，重构出满足要求的定制化组件，实现软件的快速开发。

1 基于组件的定制化开发方法

组件技术从根本上改变了软件的开发方式，可以使用一些预先构建的组件，将它们组装在一起，形成具有全新功能的软件产品，而不用从头开发[2-4]。大量经过验证的、

收稿日期:2020-02-27； **修回日期**:2020-03-22。

作者简介:秦振汉(1981-),男,黑龙江绥化人,硕士,工程师,主要从事自动测试系统方向的研究。

成熟的组件，是测试软件能够进行定制开发的基础。

1.1　组件的划分

这些组件按照复用性质，分为共性组件和个性组件，两者遵循统一的接口定义规范和开发准则，但实现方法和开发目的各不相同。

1.1.1　共性组件

反映的是软件开发中的共性内容，为测试软件开发和运行提供了必须的、带有公用性质的基础材料。共性组件包括：统一定义各种组件接口的接口定义库；用于创建组件实例的类厂组件；定义了常用的数据结构、概念模型的公共数据模型库；数据访问组件库；数据模型组件库；在应用工程中提取的、具有复用价值的测试业务组件；基础界面组件等。

1.1.2　个性组件

个性组件反映了软件开发中的差异性内容，在应用软件开发时，需要进行针对性调整。例如，与用户交互的显示界面、总线通讯数据的解析、业务逻辑中的时序控制、数据分析算法等，经常会因为需求的变化而频繁修改。为此，将这些带有差异性的内容单独提取出来，封装为独立的原型组件。该类组件采用设计延迟方式，预先提供了若干个功能扩展点，并将组件的具体实现推迟到应用软件开发中。

共性组件和个性组件的划分并不是绝对的。随着被测产品的不断发展，其测试方法、测试流程、数据分析方法等也在不断变化，导致已有的共性组件失去了公用性质，转换为个性组件；同样，之前只在特定产品测试中出现的个性组件，成为了当前软件开发的标准配置，差异性消失，从而转换为共性组件。

1.2　定制开发方法

测试软件定制化开发的基本思路如图 1 所示。针对不同的软件任务时，对于共性需求，选择需要的共性组件；对于个性需求，以个性组件为模板，定制开发新的应用组件。通过对这些组件的灵活调配，建立满足特定需求的定制软件。

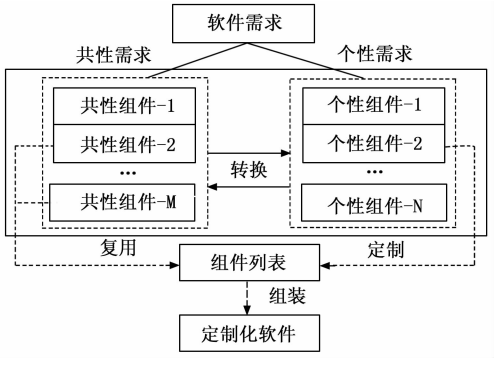

图 1　测试软件定制化开发方法示意图

该方法的本质是通过对共性组件的选择和个性组件的重构，将测试软件的定制化开发问题转换为对组件的复用问题，利用各种组件的大规模复用所带来的规模化效应，快速解决大部分共性问题，并保证开发的稳定性；然后，将工作的重心集中在个性问题的定制化开发上，使软件开发具备了更高的灵活性。

2　个性组件的设计与开发

在定制化开发过程中，主要是通过对个性组件的二次开发，衍生出新的应用组件，实现用户的定制需求，其原理如图 2 所示。个性组件来源于特定领域的实践经验，在设计中贯彻“开发以复用”的思想，专门用于后续的功能扩展。

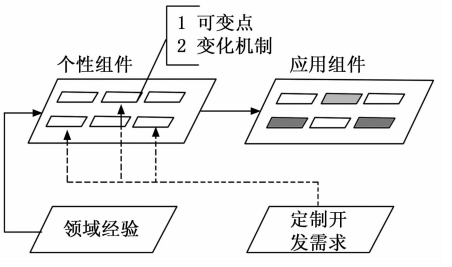

图 2　以个性组件为模板的定制化开发

在个性组件设计时，首先需要明确两个问题：哪些内容可以定制；如何进行定制。为此，引入了可变点和变化机制，用以标明定制开发的内容和实现方法。

2.1　可变点

“可变点”的概念来自软件产品线，用于标识各个项目差异性所在的位置[5]。在个性组件设计时，可变点代表了组件内部的变化热点。在软件开发时，如果发现各项目间存在差异，需要频繁修改，就可以通过设置可变点的方式，将其单独标注出来。设置可变点时应考虑以下因素：

1）每个可变点实现的功能应单一、明确，对应一个完整的功能需求。

2）可变点不是孤立存在的，在设置时应明确前置条件、后置条件、限定信息等。

3）位置信息，包括可变点所在的层级和具体位置。当系统较为复杂时，分层结构便于职责定义，也利于可变点的管理。

4）为可变点提供若干种可选的实现方案。定制开发人员可根据被测产品的具体要求，选择不同的实现方式。

5）变化机制。定义了标准的开发流程，用于指导定制开发人员进行后续的重构开发。

可变点的设置是一个不断完善的过程，随着对该领域问题认识的逐渐清晰和明确，可变点所在的位置、功能职责、实现方案等同步渐进明确，不断更新和完善现有的个性组件。

2.2　变化机制

变化机制定义了可变点的具体实现方法。本文从工程

应用出发，结合已有的研究成果[6-7]，提供了如下几种常用的变化机制。

2.2.1 代码层面的变化机制

为实现功能扩展，在代码层面的修改和开发是必须的。在面对定制化需求时，无法对所有细节进行标准化处理，进而通过参数配置、组件替换等方式实现所需的功能，很多情况下还是需要通过修改代码的方式实现。

该层面的变化机制包括：

1）使用面向对象编程中的继承、重写方式，重新实现父类定义的方法[8]。

2）使用桥接、装饰、模板、类厂等设计模式[2,9]，将抽象与实现解耦、或者延后实现。

3）通过预处理器、宏定义等方式，在代码编译时调整具体实现细节。

2.2.2 组件层面的变化机制

对于每个可变点，提供多种可直接复用的功能组件，代表了该可变点上预期的各种扩展选项。开发人员可根据定制需要，从中选择需要的组件，快速实现功能扩展。

该层面的变化机制包括：

1）使用新的组件替代已有的组件，两者的接口定义一致，但组件内部的具体实现不同。这是一种静态替换方式。

2）使用动态创建机制，通过组件类厂，采用反射机制，完成组件的实例化操作。这是一种动态替换方式，减少组件间的直接耦合关系。

2.2.3 软件体系结构层面的扩展要求

组件不是独立存在的，必须通过预先设计的软件体系结构，将其组合在一起，构成具有完整功能的应用软件。为满足体系结构扩展方式的要求，个性组件需满足：

1）设计和开发时必须严格遵循接口定义、接口规范、交互机制等。

2）个性组件在设计时要有合理的划分，以能够完整地实现一项相对独立的软件功能为标准。因为功能相对独立，所以修改时避免了连锁反应，提高软件体系结构的可修改性[10]。

3）在交互方式设计时，对于下层组件，可使用接口直接调用；对于上层组件，采用事件方式的隐式调用，减少组件间的直接关联。

2.2.4 配置文件

将具体业务的实现细节从组件内部剥离，形成了标准化的配置参数。改变配置文件中的对应参数，从而影响组件对象内部的运行状态，实现对需求变化的快速响应。

3 组件定制开发实例

本文以某类电子产品的性能测试功能为例，描述了个性组件的设计和开发过程。性能测试是该类电子产品的必备测试项目，用以完成被测产品的各项功能与性能指标的检测，确定是否满足指标要求。

3.1 差异性分析

不同型号被测产品的性能测试功能具有很强的相似性，其测试场景、测试流程、激励信号的形式和种类、数据处理的实现过程等宏观方面基本一致，在测试软件开发时，可以将这些共性内容封装在组件内部，作为定制化开发的基础。但不同产品在某些细节方面，存在明显的差异，主要体现在以下几个方面：

1）通讯端口的种类、数量上的差异。被测产品必须含有通讯端口（最少为 1 个），但不同产品的端口的数量、种类是不确定的。

2）测试信号上差异。该类产品测试中涉及的信号主要包括数字量、模拟量、脉冲量、计数器信号、陀螺仪信号、波形信号、光纤陀螺信号等，但不同产品的具体情况各不相同。

3）通讯格式上的差异。既包括通讯帧在协议格式上的差别，也包括通讯帧在数据构成、类型上的差别。前者决定了能否实现数据的正常收发，后者决定了该产品测试参数的种类和存储形式。

4）测试数据分析与处理上的差异。很多情况下，测试数据的分析处理算法需要在与被测产品对接调试后才能彻底固化下来，软件可以提供一般性的标准流程，但无法准确描述全部细节。

5）数据显示、保存方式的差异。

3.2 组件设计实例

通过对性能测试功能的共性和差异性分析，建立了面向电子产品性能测试的个性组件。图 3 使用统一建模语言（unified modeling language，UML），以组件图的形式，描述了该组件模型的顶层拓扑结构。该组件实现了测试流程控制、数据收发与解析、测试结果的分析与处理、数据保存、数据显示等一系列功能，是一个功能相对独立的业务组件。组件通过 3 个对外端口，完成与测试仪器、数据访问、上层业务逻辑的交互。在组件内部，共设置了 6 个可变点，实现具体业务功能的扩展。

V1：位于测试组件核心业务类中，主要职责是根据被测产品信息（如型号、个数、产品编号等），建立被测产品业务执行类的实例对象，实现与外部组件的交互等。该类的职责限定在宏观控制上，不涉及执行细节，各步骤高度程式化，因而采用了配置参数方式实现。

V2：该可变点位于被测产品业务执行类中，测试流程可以划分为初始化、前置操作、测试执行、后置操作等。其中，“测试执行”实现不同产品的业务流程控制，差异性较大，其功能包括：利用通讯测试接口，完成不同通讯端口的流程控制与同步；利用硬件交互端口，完成测试信号的输出和时序控制。此处采用模板设计模式，开发人员根据业务需求，修改该操作的具体实现。

V3：该可变点位于组件界面的具体实现。界面部分是

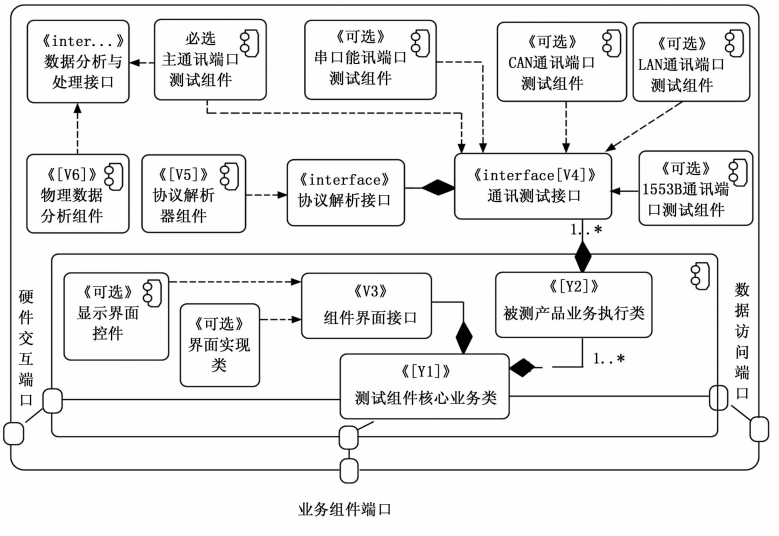

图 3　性能测试组件的顶层拓扑结构（组件图）

变化比较频繁的部分，界面显示的内容、界面风格、样式等需要根据不同用户的要求进行调整。在代码层面，通过开发单独的界面实现类实现；在组件层面，开发独立的显示界面组件，通过组件替换方式实现，但两种方式必须遵循组件界面接口定义。

V4：位于通讯测试接口的不同实现方式上。该可变点提供了多种通讯测试方案，其中，主通讯端口测试组件是必选项，其它的端口测试组件均为可选项。该可变点采用了组件的动态创建机制，可根据被测产品的端口信息，在运行时完成扩展组件的实例化。

V5：位于协议解析器组件中，该组件负责通讯数据的解帧和组帧。在特定领域中，总线通信协议的格式固定，在此限制条件下，当针对不同的被测产品时，只需调整各通讯帧的格式。该可变点采用了配置文件方式。

V6：位于总线通讯数据的物理数据分析组件中，负责将接收的数据帧解析为物理数据，并对其进行分析，确定测试指标是否合格，其核心在于针对不同的产品，采用不同的处理方式和分析算法。该可变点采用了模板设计模式，将共性处理方法封装在抽象基类中，将特定算法的实现细节放到子类中实现。

3.3　实例结果分析

通过个性组件的开发实例，得到以下结论：

1）通过个性组件，为该类问题提供了一个完整的、标准化的解决方案。面对同样问题时，可以直接复用该组件，快速搭建应用软件的基础。

2）针对项目间的差异性问题，提供了可变点和变化机制，开发人员按图索骥，精准定位，通过选择、编辑、修改等方式，在已有的软件基础上，完成特定问题的定制化开发。

3）组件设计的质量取决于对领域开发经验的总结，重点是对各项目间差异性的准确识别和描述，对开发人员的经验要求较高。

4　结束语

本文提出了一种基于组件的测试软件定制开发方法。通过对个性组件内部预设可变点的灵活配置，实现对需求变化和用户个性需求的快速响应，开发出更加贴合用户要求的应用软件。

该方法已经在多种测试软件开发项目中得到应用。从使用情况看，该方法尤其适用于特定领域内中小规模软件的定制开发，此时问题的范围相对固定，项目间的共性内容占多数，差异性的需求只占少数，并且基本控制在可预估的范围内。在扩展可变点的功能时，多采用参数配置、组件替换等方法，只在少数情况下才会使用代码层面的扩展方式。本方法保留了基于组件开发方式在提高复用性、维护性方面的优势，同时，在针对用户的多样性需求时，实现上更为敏捷和高效，便于应用软件的定制化开发。

参考文献：

[1] 张卫民. 航天测控领域软件构件设计 [J]. 测控技术，2000，19 (10)，38－49.

[2] 尹禄高，刘旺开，沈为群. 基于组件与设计模式的测控系统 [J]. 计算机测量与控制，2010，18 (2)：360－362.

[3] 刘福军,汤宫民,孙香冰,等.基于构件的开放式网络化自动测试技术研究[J]. 计算机测量与控制,2016,24(8):12－15.

[4] 郭素敏，徐克宝，吴　波，等. 基于构件技术的通用自动测试系统软件设计 [J]. 计算机测量与控制，2013，21 (10)：2859－2861.

[5] 李伟刚，李　易. 软件产品线工程－原理与方法 [M]. 北京：科学出版社，2015.

[6] 刘吉伟，毛新军. 软件运行时可变性动态演化机制研究 [J]. 计算机学报，2016，39 (11)：2216－2235.

[7] 邓小娥，葛孝堃. 软件产品线可变性的分析和实现技术研究 [J]. 计算机应用与软件，2011，28 (9)：134－184.

[8] 祝家意，彭　鑫，赵文耘. 基于 OOP 和 AOP 的软件产品线实现技术研究 [J]. 计算机科学，2009，36 (7)：120－123.

[9] 李晓伟，徐冰霖，张银发，等. 设计模式在测控通信构件设计中的应用 [J]. 飞行器测控学报，2012，31 (6)：63－67.

[10] 原雅茹，谢斌红，潘理虎，等. 面向 SCADA 系统的产品线体系结构设计 [J]. 计算机工程与设计，2019，40 (1)：103－115.