

一种关键字驱动的自动化测试框架

接 卉, 兰雨晴, 骆 沛

(北京航空航天大学 软件工程研究所, 北京 100083)

摘 要: 为了提高软件自动化测试脚本的可复用性, 提出了一种基于关键字驱动的自动化测试框架(LKDT)。首先对目前现有自动化测试框架进行了归纳分析; 然后提出了 LKDT 的系统架构, 给出了其中所包含的关键技术; 最后以实际的 Linux 桌面应用程序为例对 LKDT 框架与现有框架进行实际对比验证, 结果表明该框架具有降低测试脚本规模、提高整体测试效率等优越性。

关键词: 关键字驱动; 自动化测试; Linux GUI 测试

中图分类号: TP311.5 **文献标志码:** A **文章编号:** 1001-3695(2009)03-0927-03

Keyword driven automated testing framework

JIE Hui, LAN Yu-qing, LUO Pei

(Software Engineering Institute, Beihang University, Beijing 100083, China)

Abstract: In order to improve the reusability of the test scripts during the automation test procedures, this paper proposed a keyword driven automated testing framework (LKDT). At first, it summarized and analyzed the recent automated testing frameworks; then it proposed the LKDT's system architecture, and also presented the key technology details of the framework. At last, this paper compared LKDT with the recent frameworks by the actual desk programs in Linux, which proved that LKDT had superiority in reducing the scale of test scripts, raising the overall efficiency of testing and so on.

Key words: keyword driven; test automation; Linux GUI testing

0 引言

软件测试是保障和提高软件质量的重要手段^[1]。软件开发者和使用者必须对软件进行充分的测试, 以确保其正常工作。统计表明, 在典型的软件开发项目中, 软件测试工作量往往占软件开发总工作量的 40% 以上^[2,3]。因此提高软件测试过程的效率至关重要。

为了提高测试效率, 越来越多的测试工作引入了自动化测试的思想和方法。实践证明, 软件自动化测试技术提高了软件测试的速度和效率, 节省了软件测试成本, 缩短了产品发布周期。同时, 自动化测试技术也完成了许多手工测试无法实现的工作。所以, 采用自动化测试方法和相应的测试框架成为了软件开发组织测试工作的重要支撑手段。例如, 采用自动化测试工具能在测试活动中减少一部分开销, 同时, 有些测试活动是靠手工方式难以实现和度量的; 自动化测试框架能够提高测试效率, 快速定位测试软件各版本中的功能、性能缺陷。

目前的自动化测试框架主要可以分为三种: 录制/回放测试框架、数据驱动的自动化测试框架、关键字驱动的自动化测试框架。通过比较, 发现关键字驱动的测试框架较前两者具有更好的可复用性, 而且随着测试量的增加, 测试代码的不断累积, 其优越性愈加明显。但当前基于关键字驱动的测试框架研究尚未给出直接可用于 Linux 桌面测试的框架。而 Linux 下支持 GUI 自动化测试的测试框架对于 Linux 的应用程序的丰富与成熟具有至关重要的保障作用。因此为了弥补理论和实践

的差距, 本文给出了一种实用、有效的关键字驱动的测试框架 LKDT, 并验证了 LKDT 在自动化测试领域的实用性和优越性。

1 自动化测试框架概述

1.1 自动化测试框架定义

一个自动化测试框架就是一个由假设、概念以及为自动化测试提供支持的实践的集合^[4]。自动化测试框架可以减少测试脚本实现和维护的成本, 使测试人员把精力集中在测试用例的设计上。自动化测试框架的好坏直接影响到自动化测试的成功与否^[2]。

1.2 自动化测试框架开发原则

一个优秀的自动化测试框架应该满足以下特点:

a) 测试框架与被测应用程序独立。虽然测试的应用程序不一样, 但被测应用程序之间却会有相同的地方, 测试框架应聚焦在不同测试应用程序中共同的部分, 把与具体应用程序有关的部分从框架中移除。

b) 测试框架应易于扩展、维护。测试框架应被高度模块化, 这样可以提高框架的维护性。各个模块之间相互独立, 对模块内部的修改不应该影响其他模块。除此之外, 系统应该有充足、详细的文档, 与软件开发一样, 这也是必不可少的。设计文档可以帮助开发人员扩展、维护测试框架, 而使用文档则可以告诉用户要怎么使用该框架。

c) 测试脚本所使用的测试语言应该是与框架独立的。不

收稿日期: 2008-06-17; 修回日期: 2008-08-17

作者简介: 接卉(1980-), 女, 山东烟台人, 硕士, 主要研究方向为计算机软件理论、软件测试(jiehui@cse.buaa.edu.cn); 兰雨晴(1969-), 男, 内蒙人, 副教授, 博士, 主要研究方向为需求工程、项目质量管理、质量保证等; 骆沛(1983-), 男, 四川成都人, 博士研究生, 主要研究方向为计算机软件与理论。

同的测试框架可能在不同的应用领域有不同的表现,有些适用于 Java 应用程序的测试,有些可能适用于 Web 应用程序的测试,如果测试脚本所采用的语言是私有的、与测试框架绑定的,那么当需要从一个测试框架迁移到另外一个测试框架时,所有的测试脚本都需要重写。

d) 测试框架不应该让框架的复杂性影响到测试人员。在大多数情况下,测试人员就是测试人员而不是开发人员,甚至有的时候,他们不是专业的测试人员,可能只是具有很少软件开发经验的某个应用领域的专家。对于这些使用者来说,测试框架的使用要简单、测试语言要易于理解,这样可以使他们专注于业务相关内容的编写。

2 常用的自动化测试框架

2.1 录制/回放测试框架

录制/回放测试框架所采用的原理是通过录制应用程序产生的线性脚本进行回放从而达到自动化测试的目的。其优点是简单,通过录制就可以得到所需脚本。但同时也有很大的缺点,它不具有逻辑判断的能力,可维护性差,效率低下。

2.2 数据驱动的自动化测试框架

该种框架的原理是采用了数据驱动脚本进行测试,数据驱动脚本是将数据输入存储在独立的数据文件中,脚本只存放控制信息,测试时输入直接从文件中读取,这样同一脚本可以运行于不同的测试用例中,实现了脚本与数据的分离。其优点是可以快速增加相似测试,测试者增加新的测试不必掌握测试工具语言,对此后的类似测试无额外维护开销;缺点是初始建立测试脚本的开销较大,进行数据扩展的脚本需要针对相同的测试内容并具有相同的测试逻辑。

2.3 关键字驱动的自动化测试框架

关键字驱动(表驱动)是对数据驱动的逻辑扩展,它的核心思想可以概括为三个分离。

1) 界面元素名与测试内部对象名的分离 在被测应用程序和录制生成的测试脚本之间增加一个抽象层,它可以将界面上的所有元素映射成相对应的一个逻辑对象,测试针对这些逻辑对象进行,界面元素的改变只会影响映射表,而不会影响测试。

2) 测试描述与具体实现细节的分离 把测试描述和测试的具体实现细节分离开来。测试描述只说明软件测试要做什么以及期待什么样的结果,而不管怎样执行测试或怎样证实结果。这样做是因为测试的实现细节通常与特定的平台以及特定的测试执行工具有着密切的联系。这种分离使得测试描述对于应用实现细节是不敏感的,而且有利于测试在工具和平台间的移植。

3) 脚本与数据的分离 最后,可以把测试执行过程中所需的测试数据从脚本中提取出来,在运行时测试脚本再从数据存放处读取预先定制好的数据,这样脚本和数据可以独立维护。

以上这三个分离各司其职、互相独立,最大程度地减少相互之间的影响。从关键字驱动的思想可以看出,该种测试框架不仅实现了将数据和脚本相分离,而且实现了测试逻辑和数据的分离,大大提高了脚本的复用度和维护性,从而更大幅度地实现了测试工具的自动化。

3 用于 Linux 下桌面应用程序的关键字驱动的自动化测试框架

当前 Linux 下的测试工具主要是针对内核和服务端方面进行测试,对桌面应用程序的测试基本不支持,开发一个支持 Linux 下桌面应用程序测试的测试框架对于解决 Linux 下应用程序的发展与成熟具有积极的促进和保障作用。本文根据测试自动化框架开发原则,结合软件测试的实际需求,提出一个可行的自动化测试框架: LKDT (an automated testing framework of keyword driven for Linux)。

3.1 LKDT 总体架构

LKDT 主要由测试用例管理、测试计划、测试脚本编写、测试执行和测试结果收集等组件构成。其总体架构逻辑图如图 1 所示。

测试用例管理组件采用与测试用例管理工具,如目前常用的 test runner 管理工具集成的办法,提供对 test runner 中已有测试用例的导入、导出功能,从而方便测试人员对于测试用例的统一管理。

测试脚本编写组件允许采用两种脚本编写方式: a) 测试人员通过录制工具产生测试脚本,即测试人员根据测试用例中的步骤,对应用程序进行操作,录制工具将这些动作捕获下来,生成脚本文件; b) 测试人员通过关键字的方式手动进行编写,即在该框架中测试人员可以通过拖放关键字来进行脚本的编写。

测试计划组件的主要作用即允许测试人员对测试用例的执行方式(顺序、并行、条件、分支、循环、分布式等)、执行时间、日志级别等进行定制。

测试执行组件是对测试用例或者测试计划所关联的测试用例按照要求进行执行,该过程为自动执行过程,无须人工进行干预;如果需要手动控制测试脚本执行,需以 debug 的方式执行。

测试结果组件收集测试结果信息,包括一个测试结果总结,如测试执行了几个测试用例、执行时间以及成功、失败的用例数;还包括一个失败的测试用例执行的详细信息,包括失败的测试脚本所属的测试计划、所属的测试用例、在测试脚本中的位置以及出错过程中的截图。

3.2 关键字驱动模块

关键字驱动模块是 LKDT 的核心模块,在 LKDT 框架中起着不可或缺的作用。该模块主要解决了脚本关键字的定义、关键字解析等问题。

关键字驱动的测试脚本为测试数据关键字和测试逻辑关键字的有机组合。数据关键字相对简单,与数据驱动技术类似,一个数据关键字代表一个测试脚本中的测试数据。在脚本运行时只需通过框架用真实的测试数据对数据关键字进行替换。

逻辑关键字的定义与测试活动的结构密不可分。在真实的测试活动中,一个测试目标需要有一组测试用例来支持,而每个测试用例又由一系列的测试步骤完成。这是一个三层结构,测试步骤为测试活动中的最小单位,测试目标为最大单位,测试用例居中。逻辑关键字基于测试活动的这种结构即可分为高层脚本、中层脚本和底层脚本三层,分别对应于测试中的

测试目标、测试用例和测试步骤。高层次脚本由逻辑关键字代表的低层次脚本,加上数据关键字代表的测试数据组合而成。其组合关系如图2所示。

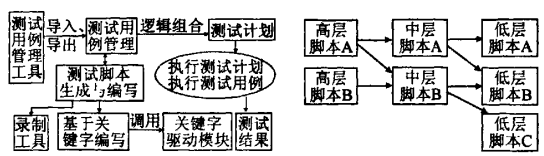


图1 LKDT总体架构逻辑图

图2 关键字脚本的组合关系

根据上述分析可以看出,该模块主要由核心模块、脚本语言解释器模块和支持库模块组成。

3.2.1 核心模块

该模块主要作用是对测试脚本中的关键字信息进行解析,并控制脚本的执行。它由数据解析器、脚本解析器、脚本执行器、中间层四部分组成。其中,数据解析器负责数据关键字的解析;脚本解析器负责解析脚本中的逻辑关键字;脚本执行器负责脚本的执行;中间层负责测试库调用。

3.2.2 脚本语言解释器模块

脚本语言解释器模块包括三个部分,即词法分析器、语法分析器和解释器。词法分析器负责将输出的字符流解析为一个一个的词^[5-7];语法分析器负责将词序列解析为具有语义的语句^[8];解释器负责语义的翻译。

3.2.3 支持库模块

支持库包括两个部分:a)所有测试可以共用的库,包括日志库和测试支持库。日志库负责向测试人员提供日志记录的功能;测试支持库则提供所有测试能共用的功能。b)GUI测试库主要作用是支持Linux下对桌面应用程序的测试。其基本结构如图3所示。

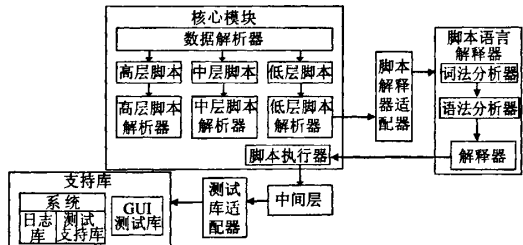


图3 关键字驱动模块逻辑结构图

4 LKDT 功能验证

目前,该工具已应用于中标软件公司测试部,主要对Linux下的GUI应用程序进行功能测试。本章以Linux下的文本编辑器Gedit的编辑功能测试为例,简要介绍进行实际测试的有效性。

首先对测试目标进行分析,为其制订两个测试用例,如表1所示。

根据测试步骤,可知需要创建三个关键字,即“打开文件”“编辑文件”和“保存文件”。

根据创建测试脚本的向导创建底层测试脚本,该框架提供了一个支持拖拽的图形化编辑器,用户可以按照自己的需要把相应的控件拖放到编辑器上,从而得到三个低层脚本(“打开文件脚本”“编辑文件脚本”“保存文件脚本”)、两个中层脚本(“新建文件编辑脚本”“已有文件编辑脚本”)、一个高层脚本(“Gedit编辑测试脚本”)。在该框架上运行高层脚本即可进

行相应测试。

表1 测试用例图表

测试用例	测试步骤
1. 新建文件编辑	1) 打开 Gedit 2) 输入“Gedit 编辑测试,新建” 3) 点击“另存为” 4) 输出“新建文件编辑.txt” 5) 点击“保存”
2. 已有文件编辑	1) 打开 Gedit 2) 点击“打开” 3) 选择“新建文件编辑.txt” 4) 点击“打开” 5) 修改文件为“Gedit 编辑测试,修改” 6) 点击“另存为” 7) 输出“修改文件编辑.txt” 8) 点击“保存”

通过测试发现,上述例子中若用其他测试方式则需要编写两个测试脚本,分别包含测试用例中的每个步骤,共需要16条指令;而采用LKDT以后,测试人员需要编写三个底层脚本,共8条指令,以及几个关键字调用。由于测试脚本编写的复杂度主要集中在脚本步骤的编写中,关键字的调用几乎可以忽略不计。因此可以看到,使用了关键字技术后,减少了约一半的脚本编写量。而且随着测试软件规模的扩大,脚本本身的复杂度也会随着增加,关键字所起到的作用就会变得愈加明显。

5 结束语

实践充分表明,依靠传统的自动化工具来完成Linux下的GUI测试非常困难。一是因为捕捉产生的脚本维护困难;二是由于录制的测试脚本与测试数据耦合得太紧密导致录制生成的测试脚本可复用性较差,生存期短。而LKDT框架采用的关键字驱动则是一种截然不同的思想,它把传统测试脚本中的变化与不变的东西进行了分离,这种分离使得分工更明确,并且避免了它们相互之间的影响。

当然对于该框架仍有许多工作需要探讨。例如如何使测试框架提供分布式测试支持以及需要考虑到Linux操作系统中用户权限问题等,通过后续研究以期使之成为一个更加简单、易用、高效的自动化测试框架。

参考文献:

- [1] BENNETT J P. Introduction to compiling techniques-first course using Ansi C, Lex and Yacc[M]. [S.l.]: McGraw Hill Book Co, 1990.
- [2] FEWSTER M, GRAHAM D. 软件测试自动化技术与实例详解[M]. 舒智勇,包晓露,焦跃,等译. 北京:电子工业出版社,1999: 333-342.
- [3] PATTON R. 软件测试[M]. 北京:机械工业出版社,2002:1-50.
- [4] 朱鸿,金凌紫. 软件质量保障和测试[M]. 北京:科学出版社,1997.
- [5] DUSTION E. The automated testing life-cycle methodology[C]//Proc of International Conference on Practical Software Assurance Techniques and Practical Software Test Techniques. 2000:1-16.
- [6] YU Yi, XU Zhong-wei, WU Fang-mei. A strategy based on a knowledge base for black box software testing[C]//Proc of International Symposium on Test and Measurement (ISTM/99). Xi'an: [s.n], 1999:206-210.
- [7] 张克东. 软件工程与软件测试自动化教程[M]. 北京:电子工业出版社,2002.
- [8] 朱菊,王志坚,杨雪. 基于数据驱动的软件自动化测试框架[J]. 计算机技术与发展, 2006, 16(5): 68-70.

一种关键字驱动的自动化测试框架

作者: 接卉, 兰雨晴, 骆沛, JIE Hui, LAN Yu-qing, LUO Pei
 作者单位: 北京航空航天大学, 软件工程研究所, 北京, 100083
 刊名: 计算机应用研究 **ISTIC PKU**
 英文刊名: APPLICATION RESEARCH OF COMPUTERS
 年, 卷(期): 2009, 26(3)
 引用次数: 0次

参考文献(8条)

1. BENNETT J P [Introduction to compiling techniques-first course using Ansi C, Lex and Yacc](#) 1990
2. FEWSTER M, GRAHAM D, 舒智勇, 包晓露, 焦跃 [软件测试自动化技术与实例详解](#) 1999
3. PATTON R [软件测试](#) 2002
4. 朱鸿, 金凌紫 [软件质量保障和测试](#) 1997
5. DUSTION E [The automated testing life-cycle methodology](#) 2000
6. YU Yi, XU Zhong-wei, WU Fang-mei [A strategy based on a know-ledge base for black box software testing](#) 1999
7. 张克东 [软件工程与软件测试自动化教程](#) 2002
8. 朱菊, 王志坚, 杨雪 [基于数据驱动的软件自动化测试框架\[期刊论文\]-计算机技术与发展](#) 2006(5)

相似文献(10条)

1. 期刊论文 朱宏峰 [基于关键字驱动的Web自动化测试的研究 -硅谷2009\(24\)](#)
 研究与设计以Selenium测试工具为基础, 运用关键字驱动技术, web应用的自动化测试框架.

2. 学位论文 曾北溟 [自动化测试框架的研究与实现](#) 2004

软件测试工作在软件开发的过程中占有极其重要的位置. 为了克服手工测试的一些困难, 提高软件质量和测试效率, 测试过程的自动化被广泛地引入进来. 它以其自动化程度高、实用性等特点, 引起了人们的广泛重视, 成为软件测试的发展方向. 自动化测试框架产品的出现表明软件测试自动化技术正在趋于成熟. 早期使用录制回放和脚本工具的不足正在被克服, 使得自动化测试更加经济、有效, 更加有利于实现和维护. 一个充满希望的信息是, 随着在开发和维护脚本上的时间越来越少, 更多的时间可用于提高测试的覆盖范围 and 产品质量, 从而在自动化上的投资能够更快地得到证明. 本文结合对自动化测试框架的实际应用详细阐述了自动化测试的理论、ATLM方法学, 分析讨论了自动化测试框架方法以及实现. 论文内容共分四部分: 第一部分是自动化测试概述, 介绍自动化测试的引入及内容概要; 第二部分是自动化测试内容的详细介绍, 包括自动化测试面临的问题、实施的原则以及自动化测试框架的提出; 第三部分介绍了几种自动化测试框架, 对它们进行了比较; 第四部分以定义开发的一种自动化测试框架WRAP说明自动化测试框架的应用.

3. 学位论文 崔丽 [GUI功能自动化测试框架的设计及GC PACS系统应用](#) 2006

软件测试自动化, 已经成为国内软件工程领域一个众所周知的课题. 自动化测试框架产品的出现更是表明软件测试自动化技术正在趋于成熟, 它提高了测试脚本的可重用性和维护性, 使非编程人员也能加入到自动化测试的开发过程中. 本论文设计和实现了一种复用性强、可扩展、以及易于维护和使用的功能自动化测试框架(STAF), 它将数据驱动脚本与关键字驱动框架相结合, 主要应用于对被测应用程序进行系统测试、回归测试等的自动化测试开发, 通过与Rational Robot自动化测试工具的结合, 实现了该框架在GC PACS系统上进行自动化测试开发的成功应用.

4. 学位论文 陆栋 [自动化软件测试框架分析及应用](#) 2008

随着计算机应用日益普及和深化, 现代软件的规模越来越庞大. 以前用手工作坊式方法开发出来的许多大型软件, 由于没有进行软件质量管理, 因此几乎无法维护, 致使项目报废, 造成大量人力、物力浪费. 如何提高软件质量, 保证软件安全性是一个涉及面广、难度很大的课题. 超高质量软件的开发技术将是打开21世纪高技术市场的钥匙, 而软件测试则是软件质量保证中的关键技术. 一个大型应用软件系统在开发过程中以及研制完成后, 如何检验系统是否可靠、是否达到预期目标, 成为人们越来越关注的问题. 软件测试一般分为手工测试和自动化测试, 我国在软件测试方面起步较晚, 并且主要在项目组内部进行手工测试, 近几年才逐渐对软件自动化测试引起重视. 但如何进行软件测试, 特别是如何进行自动化软件测试, 则是项目管理层和软件测试人员急需了解的内容[1]. 因此, 如何构造并建立一个软件自动测试机制, 是一个很有实际意义的研究课题. 本文针对当今世界最先进的几类软件自动化测试框架进行了细致和深入的分析, 并且针对软件自动测试技术中的回归测试构造了自动测试环境, 通过规定测试用例的书写方式以及编写关键字驱动脚本, 实现了某类金融软件的自动化回归测试, 提高了软件测试的效率. 本文通过这个成功案例阐明了在金融及相关软件行业实施自动化测试软件系统的必要性和可行性. 并提出了在目前流行的自动化测试框架系统基础上, 结合关键字智能化提取改进原有系统框架依赖性过强的想法. 并给出了初步的结合持续集成技术的系统自动化整合测试设计.

5. 学位论文 侯勇 [关键字驱动的自动化测试系统的研究](#) 2006

软件测试是软件开发的关键过程之一, 对于确保软件产品的质量也有着十分重要的作用. 软件自动化测试技术研究的目的是在无人干预的情况下自动测试软件, 提高整个软件测试的效率. 对于当前功能强大的软件系统, 人们希望通过自动化测试技术来显著提高软件测试的效率, 因此对自动化测试技术的研究是具有实际意义的. 正是在这个前提之下, 本文首先介绍了软件测试的研究背景、国内外发展状况和自动化测试发展历程. 接着阐述了自动化测试的基本理论和自动化测试脚本技术. 针对当前所使用的数据驱动自动化测试系统中脚本无法灵活适应软件业务变化的缺陷, 论文提出了基于关键字驱动脚本技术的自动化测试系统, 并对其进行了详细的研究. 文中通过实例对关键字驱动自动化测试系统的测试流程做了较完整的描述, 并以自主研发的自动化测试平台为依托, 应用关键字驱动脚本技术进行自动化测试. 通过自动化测试效率的对比可以预见, 由于关键字驱动自动化测试系统适应了软件业务的发展, 其必将带来测试效率的进一步提高.

6. 期刊论文 侯勇, 张海林, Hou Yong, Zhang Hailin [自动化测试中的关键字驱动脚本技术 -电子科技2006\(2\)](#)

主要介绍了自动化测试中的乙一种扩展, 它不仅实现了数据和脚本的分离, 同时也实现了业务同脚本的脱离, 文中对脚本技术进行了对比, 并阐述了关键字脚本技术的实现.

7. 学位论文 高深 一种关键字驱动的Tuxedo测试工具的设计与实现 2007

Tuxedo是一个C语言开发的企业级事务处理中间件，主要应用于高端的分布式事务处理领域。它的测试面临多平台、多资源、复杂的测试环境，还需要满足高稳定性和高效率的要求。显然手工测试繁琐的配置带来的巨大的人力，物力开销是不能够满足需求的，而原有的测试工具MegaTest1.0已经很难满足Tuxedo的分布式和特殊复杂环境测试的需求。本文鉴于Tuxedo测试的实际需求，在已有的MegaTest1.0的基础上，重新优化设计实现了一个由关键字驱动的，自动化程度更高的Tuxedo测试工具—MegaTest2.0。MegaTest2.0的设计主要是基于两方面的实际需求：1、提高测试用例的开发效率

；2、提高测试执行的效率。在设计中使用了基于关键字驱动的自动化测试工具开发思想，抽象并提取出了高内聚，低耦合的公共接口测试场景MSF文件，用以驱动和控制整个测试流程。在MSF文件中对测试用例所要执行的测试动作或者测试过程进行抽象描述，使用一系列关键字指定测试中要执行的任务，这些任务的具体实现过程和它们的描述进行分离。测试工具通过对不同的关键字进行解析，找到关键字所对应的实际要执行的动作，进而执行这些动作，来完成测试。这样做使得测试逻辑与它的具体实现发生分离，测试数据和测试脚本发生分离，减少了测试用例的冗余，实现了测试用例的定制。

与手工测试和原有的测试工具相比，MegaTest2.0不仅提高了自动化测试的程度，使添加测试用例变得相对简单和快捷，在很大程度上提高了测试的效率和质量，同时也提供了良好的可复用性和健壮性。

8. 学位论文 卞斐 基于关键字驱动的WMM自动化测试的研究和实现 2008

软件测试是软件开发中的一个重要过程，受软件开发日期的影响，自动测试成为首选的解决办法。无线接入点(AP, AccessPoint)的自动化测试首先需要解决的是其自动配置问题，自动配置的健壮性决定软件是否可以连续、正确的运行，本文在自动配置AP及其它模块采用关键字驱动的思想，对各个模块的实现过程进行介绍，并提出基于框架的改进。使得在配置AP过程更加易于操作和稳定。AP测试环境的特殊性决定软件必须通过主

从控制端的方法来实现，本文采用串行通信，借鉴TLV(T—Type, L—Length, V—Value)数据格式并考虑到主从端通信的特殊性提出了

LVS(S—Synchronization, 同步)的数据格式来解决主控和被控的通信和同步问题。

当AP及STA自动配置完成后利用Ixia公司提供的接口编写自动

发送不同或相同接入类型的数据包，并记录转发速率、分析测试数据、保存测试波形图。

最后通过对比WMM(Wi—FiMulti—Media)和普通性能测试

的异同处分析了WMM自动化测试在本平台上应用的可行性，详细介绍了WMM功能在本自动化测试软件上的三层菜单编辑、脚本编辑、运行过程、错误处理以及结果分析所遇到的问题和解决办法。

9. 学位论文 张磊 UI软件自动化测试方法和应用的研究 2008

软件自动化测试研究在无人干预的情况下自动测试产品应用。UI软件自动化测试在整个自动化测试领域占了相当大的比重，利用UI自动化测试技术提高软件测试的效率是软件研发企业的普遍追求。论文首先介绍了软件自动化测试的研究背景、国内外发展状况，并系统讨论了自动化测试的基本理论和用户界面测试的基本理论。

针对当前所使用的数据驱动自动化测试系统中脚本无法灵活适应软件业务变化的缺陷，论文对基于关键字驱动的UI自动化测试系统，进行了比较深入、细致的研究；并结合工程实践，系统论述了关键字驱动的UI自动化测试系统的测试流程。论文以自主研发的自动化测试平台为依托，建立了一种新颖的脚本的自动化组织和维护机制，有效的改进了测试效率。

10. 期刊论文 徐振良. 樊滨温. 王志鹏. Xu ZhenLiang. Fan BinWen. Wang Zhipeng 关键字驱动技术在SAFS中的研究 - 微计算机信息2006, 22 (15)

本文以在软件测试中引入自动化测试为切入点, 简要地介绍自动化测试及其测试工具. 重点的分析关键字驱动技术和新一代自动化测试框架

SAFS(Software Automation Framework Support)中应用以及相互关系. 针对实际被测软件, 借助Rational公司系列测试工具, 使自动化测试和手工测试达到了很好的结合.

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjyyj200903039.aspx

下载时间: 2010年4月20日