

文章编号: 2095-4980(2013)05-0797-05

## 基于脚本的自动化软件测试框架

张杲轩, 张 谊, 李春雷

(中国工程物理研究院 电子工程研究所, 四川 绵阳 621999)

**摘要:** 为缩短测试周期, 提高软件自动化测试的效率, 设计了一种自动化测试模型, 在模型框架内根据测试场景定制自动化测试程序。通过框架内组件的复用, 达到快速开发自动化测试程序的目的。该自动化软件测试框架模型采用了脚本技术、关键字驱动测试技术以及可扩展标记语言(XML)技术, 有效提高了测试效率, 减轻了测试人员的工作压力, 提高了测试资产的利用率, 增强了脚本的可维护性, 缩短了自动化测试的准备时间。基于该框架的自动化测试程序已经在一些小型项目中得到初步应用, 能够覆盖基本的功能测试需求。

**关键词:** 软件测试; 自动化测试; 测试框架; 关键字驱动; 脚本

**中图分类号:** TP311.55

**文献标识码:** A

**doi:** 10.11805/TKYDA201305.797

## Software testing automated framework based on script

ZHANG Gao-xuan, ZHANG Yi, LI Chun-lei

(Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang Sichuan 621999, China)

**Abstract:** An automated software testing model is designed in order to shorten the test cycle and improve the efficiency of software automation test. Automated test procedures can be customized within the framework according to the test scenarios. The automated test process can be rapidly developed by reusing the components in the framework. The script technology, keyword-driven technology and eXtensible Markup Language(XML) technology are adopted in the automated software testing framework. The scratch can improve testing efficiency and reduce the working pressure of the test personnel. It can improve the utilization of test assets, enhance the maintainability of the script, and reduce the preparation time of the automated testing. The automated testing program based on the framework has been initially applied to some small projects, which can cover the basic functional testing needs.

**Key words:** software testing; automated testing; testing frame; keywords-driven; script

一般而言, 自动化测试是指<sup>[1]</sup>: 把以人为驱动的行为转化为计算机依据一定规则与设计自动执行测试行为的一种过程。通常, 在设计了测试用例, 而且通过评审之后, 由测试人员根据测试用例中描述的规程一步步进行测试, 得到实际结果与期望结果进行比较。在这个过程中, 测试执行等部分工作具有机械性强、重复性高的特点<sup>[2]</sup>。这些工作可以程序化后由机器自动完成。与手工测试相比, 测试自动化的优势明显。首先自动化测试更便于测试资产的数字化管理, 使得测试资产在整个测试生命周期内可以得到复用, 这个特点在功能测试和回归测试中尤其具有意义; 其次自动化测试可以提高测试效率, 使测试人员更加专注于新的测试模块的建立和开发, 从而提高测试覆盖率; 此外, 测试流程的自动化管理可以使机构的测试活动开展更加过程化, 符合软件能力成熟度模型集成过程改进的思想。实施测试自动化是一个不可逆转的趋势, 如果在这个领域走在了前列, 无论是单位的核心竞争力还是个人的工作技能, 都有很大的优势<sup>[3]</sup>。

### 1 自动化测试技术研究

#### 1.1 自动化软件测试的依据

在软件测试过程中, 并不是所有的测试过程都适合自动化测试。要采用自动化测试方法, 需要研究测试内容,

对涉及到的技术要求和测试步骤进行仔细的研究分析,确定测试过程中包含能够进行自动化测试的内容,再选择成熟的自动化测试的工具或者自行开发合适的自动化测试工具。自动化测试的引入有相关标准,根据成熟度分为 5 个级别,如表 1 所示。

表 1 自动化测试技术成熟度分级表

Table1 Classification about mature degree of automated software testing		
level	tech. trait	application field
I	record & play	Tested software has no change or has few changes.
II	record, edit, play	(a) recursive testing (b) Tested software has less changes.
III	program & play	(a) large-scale automated software testing (b) Lots of testing modules have been developed, executed and maintained.
IV	automated software testing using data-driven technology	(a) large-scale automated software testing (b) Lots of testing modules have been developed, executed and maintained.
V	automated software testing using keywords-driven technology	(a) large-scale automated software testing (b) Lots of testing modules have been developed, executed and maintained.

## 1.2 基于脚本的自动化测试的实现方式

基于脚本的自动化测试是自动化测试技术的主要方向之一,其优势在于脚本是非编译语言,测试数据与测试逻辑可以很容易分离,同时在测试过程中也可以修改测试数据和测试流程,复现复杂场景,提高测试的自由度<sup>[4]</sup>。基于脚本的自动化测试通常有以下 2 种实现方式:

1) 数据驱动测试:将数据从测试脚本中提取出来,测试脚本从外部获得测试数据,一个脚本可以针对多组数据执行多组相似功能的测试。

2) 关键字驱动测试<sup>[5]</sup>:数据驱动方式的变种,将关键字原语也从测试脚本中提取出来,更加细化了可重用的粒度。

上面的 2 种测试方式各有长处,但还存在以下缺点:

- 1) 不能使用模板化的方式开发测试用例;
- 2) 不能对所有的测试用例进行统一管理,让各个测试用例自动执行;
- 3) 新功能的扩展比较困难。

自动化测试框架(Testing Automation Framework, TAF)可以弥补以上缺陷。TAF 如同一个测试控制器,它可以自动收集框架内已经被规范化的测试用例,然后按指定顺序执行。在测试用例运行的过程中,TAF 会处理测试用例抛出的各种异常,以便未被执行的测试用例能顺利运行。在每个测试用例运行结束后,TAF 会保存其运行的结果。在测试的最后阶段,TAF 会针对所有测试用例的运行情况生成最终的测试报告。

## 2 自动化软件测试框架设计

### 2.1 自动化测试框架模型

测试框架描述了自动化测试程序的工作方式,但没有细化具体组件设计方法。因为软件的种类和用途千差万别,很难用一种固定的模式构建完美的自动化测试程序。设计目标是提供一种程序工作模型,在该模型的框架下,根据测试场景,快速灵活地定制自动化测试程序。

基于以上研究,自动化测试框架应具备以下主要要素:

- 1) 测试框架能够生成、记录和收集测试数据,实现自动化测试的基本功能;
- 2) 所有测试数据需遵循统一格式录入脚本文件,便于测试数据的分析和管理的<sup>[6]</sup>;
- 3) 测试框架能够支持必要的接口,如控制器局域网络、串口、以太网等,实现与被测软件的数据交互;
- 4) 测试脚本可以采用关键字驱动和数据驱动混合的方式,既可以是数据,也可以包含指令,测试脚本能够匹配正则表达式,以适应大量测试数据和复杂逻辑的场景;
- 5) 测试框架要具备可扩充性,可以在以后的工作中支持更多种类的软硬件接口;
- 6) 测试框架应具备跨平台性,用于满足多种平台的软件测试需求。

本文设计的基于脚本的自动化测试框架(Script Based Test Automation Framework)分为 3 个层次:脚本层、核心层和接口层,如图 1 所示。自动化测试框架在设计上采用 python 语言编写脚本解释器,通过添加接口模块,

配合 python 语言模块库中的日志模块、时间模块和其他功能模块实现完整的测试功能。框架以 XML 脚本文件作为输入，构建闭环的测试体系。在框架的构建中涉及到几个关键问题：脚本文件的构建方式、接口模块技术、核心模块的设计。

### 2.2 脚本文件的构建方式

所有的脚本文件构成了脚本文件集，脚本文件集是测试框架的数据来源。其中每个脚本文件是对不同测试行为的脚本化描述，它将测试数据和测试行为有机地组织起来。根据测试的需要，每个脚本文件可以包含一个或多个测试条目。每个测试条目是对测试用例的抽象。测试条目包含有指令(关键字)、数据、时间约束、执行参数等信息。为了便于脚本解释器解析，同时又符合规范的文件记录方式，脚本文件采用 XML 方式进行记录。XML 脚本文件不易直接编写，需要按照格式将测试数据和关键字等信息一起写到测试用例表中，再由测试框架完成测试用例表到脚本文件的转换。脚本文件结构如图 2 所示。

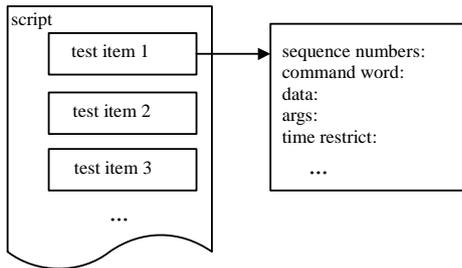


Fig.2 Structure about script  
图 2 脚本文件结构

### 2.3 核心模块的设计

核心模块的作用是将测试脚本转化成具体的测试行为。核心模块作为框架的主控器，负责解析 XML 脚本文件，运行测试用例，记录和处理测试过程中的异常，保存测试结果以及生成测试报告。脚本解析器的主要工作是解析脚本文件的结构，关联脚本文件中关键字和测试数据，重构测试逻辑。考虑到测试时可能有时序要求，解析器需要和时间模块协同工作。核心模块的工作流程如图 3 所示。

此外有必要对被测软件的运行时间、内存使用、模块调用、测试结果等数据进行记录。这些数据对于判定软件测试结果和分析软件可靠性都非常重要，因此数据记录模块和日志模块都是不可或缺的。最后，将最终的数据分析整理，合并输出到文档中。对于新出现的测试场景，可能需要定制新的关键字以表示新的测试行为。因此，

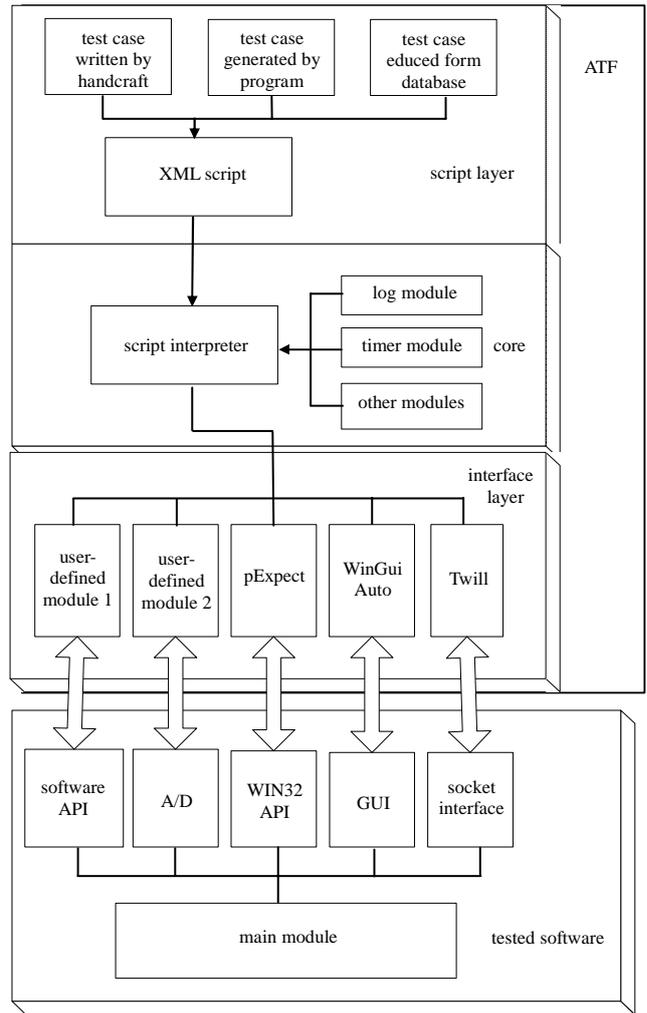


Fig.1 Software TAF  
图 1 自动化软件测试框架

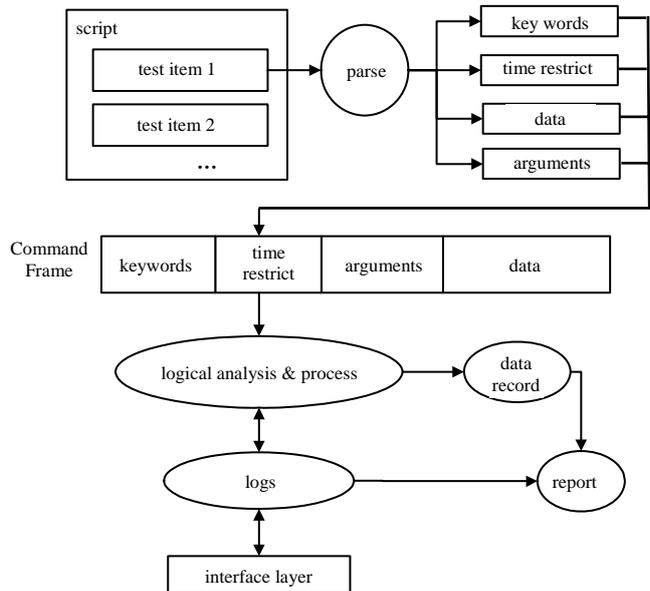


Fig.3 Work flow of kernel module  
图 3 核心模块的工作流程

关键字集合会不断地被扩充。在设计上将解析过程和逻辑处理过程也做成框架结构，扩充关键字时只需添加新的处理函数，从而保证了核心模块的可扩展性。

### 2.4 接口模块技术

接口模块是脚本解析器和被测软件的中间件，实质是一个测试函数库。在接口模块中，测试函数间的关联性、时序和优先级关系可通过信号量和软件中断的方式来解决<sup>[7]</sup>。测试函数中包含的进程调用、软件接口调用、硬件接口访问等各种操作可以从操作系统 API 中获得支持。如果测试函数的执行涉及到硬件，还需要往测试系统中插入对应的硬件以获得支持<sup>[8]</sup>。每个接口模块只面向特定接口或对象，以 python 语言编写的测试工具为例，twill 模块负责浏览器和网络模块的数据交互，WinGuiAuto 模块(一种 GUI 测试模块)负责人机界面交互。因为每个模块只与特定的协议或接口相关，在缺少某些接口模块的情况下，只需要添加或编写对应接口模块，就能保证测试框架的正常工作，这种方式从设计上保证了框架的可扩展性。

### 3 自动化测试框架的初步应用

按照测试框架，开发了一个自动化测试工具的原型版本，初步构造了一个关键字集合以及脚本结构。这个工具

在某项目的配置项测试中进行了应用。该项目中需要对一个网络通信软件进行测试，软件使用 UDP 协议，自定义了数据帧格式。数据帧定义了帧头、命令字、数据位、帧尾等，数据帧的大体结构如表 2 所示。

该软件的主要功能是作为服务器接收数据帧，并统计数据帧的数量和丢包数，响应指令字。指令字有开始、结束、发送统计信息等。设计一个主要测试过程(如图 4 所示)，覆盖上述全部功能。

为了描述测试逻辑，测试脚本构造了关键字 sendUDP。同时，对脚本解析器进行修改，添加对这个关键字的处理逻辑。为描述复杂的测试逻辑，将测试脚本分解为多个测试条目，每个测试条目描述一个测试行为。测试条目采用键—值结构，如表 3 所示。

表 2 一种数据帧结构

Table 2 A structure of data frame

frame header	command word/Byte	data/Byte	frame tail
0101H	4	1 024	CRC16

表 3 一种测试条目的结构

Table 3 A structure of test items

key	value
sequence numbers	integer number
keywords	sendUDP
argument 1	(IP:PORT)
argument 2	data package numbers
time	time restrict value count in millisecond
frame data 1	frame header
frame data 2	command word
frame data 3	1 024 Byte communication data
frame data 4	CRC16 for frame data1-frame data3

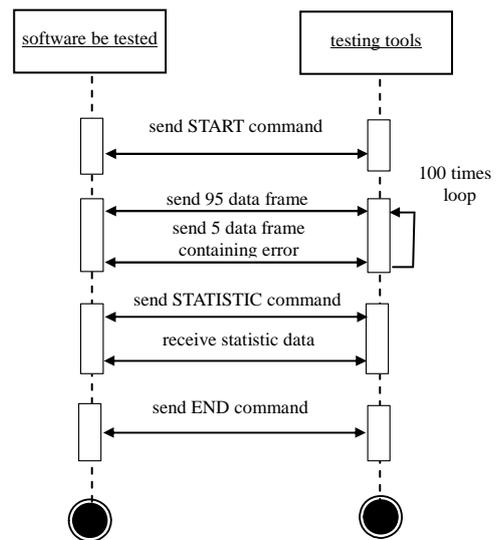


Fig.4 Main testing flow

图 4 主要测试流程

核心模块根据测试脚本的描述信息重构测试逻辑和测试数据，完成整个测试流程。测试报告根据预定标准判断通过与否，数据记录信息和日志信息可以使测试人员清楚地定位测试中存在的异常。

测试框架在该项目的应用中，对测试脚本的结构做了微调，增加了关于数据帧的描述。脚本解析器也做了相应的修改。但测试框架中其他组件得到复用，达到了在测试框架内快速定制测试工具的设计目标。经过仔细的测试设计和使用自动化测试工具，覆盖了被测软件的主要功能，对被测软件造成了一定的负载，仿真了被测软件在真实环境中运行的场景。并由此发现了无法通过手工测试发现的软件缺陷，证明了测试工具的有效性。测试框架不断地应用于测试项目，积累了测试素材，扩充了工具组件，渐进地修正了测试框架的设计方案。

### 4 结论

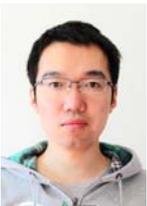
通过对现有的一些研究结论的总结和分析，结合实际工作经验，提出了一种轻量化的测试方案：基于脚本的自动化测试框架，并在日常测试工作中已得到初步的探索和应用。目前已将自动化测试技术初步应用到了单元测试、功能测试、性能测试中，整个测试过程能够按照预先设想的方式工作，验证了方案的可行性。但对于多个模

块的集成测试,甚至是多台设备协同工作的系统测试场景,设计的自动化测试方案尚显薄弱,有很多地方需要改进和完善:XML测试脚本的结构设计还不完善,驱动指令字的设计还需要优化;随着支持的驱动指令字增加,脚本解释器需要进一步完善;脚本解释器的工作效率需要提高,以适应性能测试的要求;接口模块也存在不足,目前只能针对几种特定的软硬件接口进行测试;尚未与测试管理工具进行整合。此外,从现阶段的工作中认识到,虽然自动化测试能大大减少手工测试工作,但无法完全取代手工测试。完全的自动化测试只是一个理论上的目标,实际上想要达到100%的自动化测试,不仅代价相当昂贵,而且操作上也几乎无法实现。

#### 参考文献:

- [1] 董娜娜,詹惠琴. 软件测试自动化技术应用研究[J]. 电子测试, 2010(11):47-50. (DONG Nana,ZHAN Huiqin. Research and application of software testing automation[J]. Electronic Testing, 2010(11):47-50.)
- [2] 郭德红,陈勇. 自动化测试框架底层驱动研究[J]. 电脑知识与技术, 2011,7(34):8871-8873. (GUO Dehong,CHEN Yong. Automation Test framework drive research[J]. Computer Knowledge and Technology, 2011,7(34):8871-8873.)
- [3] 莫熹,赵方. 一种数据模型驱动的软件自动化测试框架[J]. 计算机工程, 2009,35(21):78-81. (MO Xi,ZHAO Fang. Data model driven software automation test framework[J]. Computer engineering, 2009,35(21):78-81.)
- [4] 郝晓晓,张卫丰. 基于XML的SDK自动化测试框架的设计与实现[J]. 计算机技术与发展, 2010,20(4):101-104. (HAO Xiaoxiao,ZHANG Weifeng. Design and implementation of automatic SDK test framework based on XML[J]. Computer Technology and Development, 2010,20(4):101-104.)
- [5] 漆莲芝,谢敏,张军. VI仿真环境下嵌入式软件测试研究与应用[J]. 信息与电子工程, 2011,9(1):101-104. (QI Lianzhi, XIE Min,ZHANG Jun. Embedded software testing under VI simulation environment and its application[J]. Information and Electronic Engineering, 2011,9(1):101-104.)
- [6] 漆莲芝,张军,谢敏. 故障树分析测试用例生成技术研究与应用[J]. 信息与电子工程, 2010,8(5):594-597. (QI Lianzhi, ZHANG Jun,XIE Min. Research and application on FTA testing case generating technology[J]. Information and Electronic Engineering, 2010,8(5):594-597.)
- [7] 黄哲. 基于Windows的API自动化测试框架的设计与实现[J]. 科技传播, 2010(18):232,235. (HUANG Zhe. The Design and Implementation of API Test Automation Frameworks Based on Windows[J]. Public Communication of Science & Technology, 2010(18):232,235.)
- [8] 孙俊若. 正交法在雷达软件测试中的应用[J]. 信息与电子工程, 2009,7(6):501-505. (SUN Junruo. Test for radar software based on intersection method[J]. Information and Electronic Engineering, 2009,7(6):501-505.)

#### 作者简介:



张泉轩(1984-),男,四川省绵阳市人,硕士,助理研究员,主要研究方向为软件工程.  
email:zhangaox03@163.com.

张 谊(1974-),男,四川省井研县人,硕士,副研究员,主要研究方向为软件测试及可靠性.

李春雷(1981-),男,山东省青岛市人,工程师,主要研究方向为嵌入式软件测试.