

基于 Robot Framework 的 MVC GUI 模块自动化测试框架^①

梁思秋, 葛华勇

(东华大学 信息科学与技术学院, 上海 201620)

摘要: 基于 Robot Framework 提出了一种面向 MVC GUI 模块的自动化测试框架. 该框架将 GUI 自动化测试分为四大方面: 用户界面功能测试、信息设置功能测试、信息查询功能测试和特殊项测试. 整个框架将上层 GUI 和底层 Database 进行结合测试, 使 GUI 自动化测试范围得以全面覆盖; 并根据不同系统环境, 编写 Python 脚本, 灵活调用底层数据; 而选用黑盒与白盒有机结合的测试方式, 使测试者可以迅速锁定代码缺陷, 保证了开发程序的正确性和完整性. 同时, 就 OpenWRT LuCI 模块作为具体实例, 验证了本框架的适用性.

关键词: MVC; GUI 自动化测试; Robot Framework; OpenWRT LuCI 模块; Python

Automation Test Framework for MVC GUI Based on Robot Framework

Liang Si-Qiu, GE Hua-Yong

(Communications and Electronics School of Engineering, Donghua University, Shanghai 201620, China)

Abstract: This article proposes an automated test framework for MVC GUI module, based on Robot Framework. The GUI automation test framework is divided into four aspects: user interface function testing, information set function testing, information query function testing, and special items testing. The whole framework, making the GUI test and the Database test combined together, fully completes the coverage of GUI automation test. According to different system environment, by writing Python scripts, the underlying data can be flexibly called. The combination of black-box and white-box test makes the tester quickly focus the code defect, and guarantees the correctness and completeness of the development program effectively. At the same time, the applicability of this test framework has been verified, as OpenWRT LuCI module specifically addressed.

Key words: MVC; GUI automation test; robot framework; OpenWRT LuCI; Python

MVC 是当今最经典的软件设计模型之一, 其三层结构为: 模型(model)、视图(view)和控制器(controller). MVC 模式能够成功实现 Web 系统的职能分工^[1,2]. 这一特点使它在 GUI 结构设计方向上被广泛应用.

目前, 对于 MVC GUI 模块的测试仍没有统一的测试框架和标准, 大多停留在手动测试和小规模的单元测试^[3,4]. 传统的手动测试不仅费时费力, 还不方便排错, 属于完全的黑盒测试. 手动测试人员即使检测出了问题, 也很难快速将问题与某一段错误代码联系

起来. 而小规模单元测试虽然可以克服不易排错这一缺点, 但由于该测试代码是直接就开发代码进行编写, 只能保证代码的正确性, 无法保证代码功能的完整性, 这也就是白盒测试的缺点.

根据当前测试现状, 本文以 OpenWRT 中 LuCI 模块作为验证实例, 提出一种基于 Robot Framework 的 MVC GUI 自动化测试框架, 并将该测试框架分为四大方面: 用户界面功能测试、信息设置功能测试、信息查询功能测试和特殊项测试. 该框架的实现, 使 GUI

^① 收稿时间:2014-11-13;收到修改稿时间:2015-01-22

自动化测试得以全面覆盖,使其底层数据得以灵活调用,使其代码缺陷得以迅速锁定,充分保证了开发程序的正确性和完整性.

1 MVC模块结构

MVC 是一个设计模式,它强制性地 将应用程序的输入、处理和输出分离.这一特点使得 MVC 应用程序被分成了三个核心部件:模型、视图和控制器.具体流程如图 1 所示.

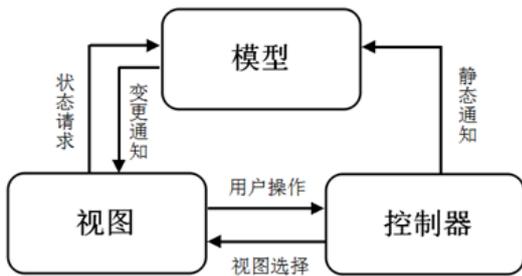


图 1 MVC 结构流程图

三大部件中,模型是最为关键的,它主要用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法,对数据有直接访问的权力;而视图主要用于对用户有目的地显示数据,在视图中一般没有程序上的逻辑;控制器则是起到不同模块间的组织作用,用于控制应用程序的流程,处理事件并作出响应.

一般来说,模型、视图和控制器三大部件是分工独立的,而它们的交互也是按既定的程序运行.当应用程序初始化时,视图会在所需监视的数据模型中进行注册操作.当运行中模型数据发生变化时,视图通过一种刷新机制了解在数据模型上发生的改变并作出响应.当用户在视图进行操作时,视图会将用户操作反应给控制器,由控制器安排下一步流程.若用户进行的是页面切换操作,控制器会进行页面选择和页面切换;若为数据配置操作,控制器则会将数据更新发送给模型层,令其进行数据修改.

本文将 MVC GUI 模块的整体功能分为四大方面:

① 用户界面基本功能: 主要涉及视图和控制器.包括界面中各文字与控件的显示,各页面间的切换,页面的刷新等.

② 用户界面信息设置功能: 涉及模型、视图和控制器三层.包括界面设置是否成功,底层数据库是否更新等.

③ 用户界面信息查询功能: 涉及模型、视图和控制器三层.包括界面查询是否成功,界面中的相关显示是否与底层数据库中相关数据一致等.

④ 用户界面确认/警告功能: 主要涉及视图和控制器.针对界面中非固定出现的文字或控件,包括用户误操作后的警告框,用户输入非法数值后的提示文字等.

2 MVC GUI测试结构的设计

为了提高 GUI 自动化测试的覆盖率、灵活性和锁定代码缺陷的速度,本文将基于 Robot Framework^[5]的 MVC GUI 自动化测试框架根据功能模块分为四大方面:

① 用户界面功能测试: 检验界面中各项显示及各项值是否正确,界面基本操作(例如: 页面切换)是否运作良好.

② 信息设置功能测试: 检验设置后返回值是否正确,设置结果是否保存到了数据库.

③ 信息查询功能测试: 检验信息查询时的返回值是否正确,查询到的信息是否与数据库中的一致.

④ 特殊项测试: 检验例如界面弹出信息、向导信息、错误提示信息等特殊项是否显示正确.

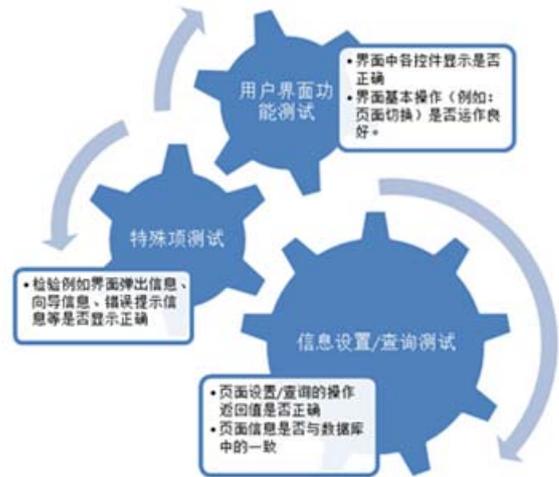


图 2 OpenWRT LuCI 测试框架

用户界面功能测试,是就视图部分进行的黑盒自动化测试.其主要测试目的在于:通过检验界面中各控件显示及其值是否正确,界面基本操作(例如: 页面切换)是否运作良好,判断 MVC GUI 模块中用户界面基本功能是否实现.

信息设置/查询测试,是就视图部分和模型部分相结合进行的白盒自动化测试.其主要测试目的在于:通过检验发生数据变化后,用户界面与底层数据库的信息是否具有-致性,判断 MVC GUI 模块的信息设置/查询功能是否实现.

特殊项测试,是就视图部分中的显示特殊项进行验证测试.用户图形界面中的特殊项,是指界面中非固定出现的文字或控件.这些特殊项只在特定情况下,出现数秒或等待至用户确认后关闭,例如:用户输入非法时弹出的错误提示信息、进入页面时的向导信息、用户操作时弹出的确认/警告框等.

四大测试方向的融合与协作,实现了 MVC GUI 的自动化功能测试全覆盖;而选用黑盒与白盒有机结合的测试方式,更使代码缺陷可以被迅速锁定,保证了开发程序的正确性和完整性.

2.1 用户界面功能测试的设计

MVC 的视图部分,实质上就是一个用户图形界面(GUI).而用户界面功能测试的实质就是针对 GUI 的一套测试方案,该方案完全采用 Robot Framework 的自带关键字库 Selenium2Library 进行黑盒测试,就测试对象是静态的还是动态的分为两大部分.

就测试对象为静态的情况,例如:页面提示文字,固定文本框、按钮等控件,主要是利用 Selenium 2Library 中的关键字判断在某页是否存在某段提示文字或某些固定控件.

就测试对象为动态的情况,例如:页面切换,刷新页面等操作,自动化测试的主要步骤如下:

首先,获得页面各项控件的定位符,建议用浏览器 Firefox 自带的 Firebug 插件进行 Locator 的获取;然后,根据页面中获取的定位器调用 Selenium2Library 关键字库,编写测试用例.如果要进行页面切换操作,则可用关键字 click link 锁定选中 locator 进行切换,并用关键字 wait until page contains 查看页面是否在未超时的情况下成功显示某些特定文字或控件,以检验页面是否切换成功;如果要进行页面刷新操作,则可用 click button 锁点击 locator 指定的刷新按钮,并用关键字确定页面是否刷新成功.

2.2 信息设置及信息查询功能测试的设计

信息设置功能测试和信息查询功能测试,这两块的测试思路十分一致,并可相互结合.对它们的自动化测试,需要将 Python 脚本白盒测试和 Selenium2

Library 关键字黑盒测试相结合,以进行全面的自动化功能测试.就测试层次将该测试框架大致分为两大部分:对 View 层的验证以及对 Model 层的验证.

对 View 层验证的主要步骤为:

首先获得页面各项控件的定位符;然后,利用 Selenium2Library 中关键字在 View 界面内对系统进行自动化设置(例如:输入 IP, 点击按钮,勾选选项等),并获得返回值;最后,检验该返回值是否正确,判断是否在 View 层设置成功.与此同时,可分别进行正确和错误的设置并进行相应检验,以完成一个全面的自动化测试框架.

对 Model 层验证的主要步骤为:

首先获得页面各项控件的定位符;然后,利用 Selenium2Library 中关键字在 View 界面内对系统进行自动化设置(例如:输入 IP, 点击按钮,勾选选项等);接着,就 Model 层(数据库)内容编写 Python 测试脚本,并调用该脚本编写测试用例,获取数据库中的配置信息;最后,将 Model 层获得的配置信息与界面设置值进行对比,检验是否一致,判断在 View 层的设置是否被保存到了 Model 层.相反的,若界面操作为非法时,检验是否不一致.

2.3 特殊项测试的设计

特殊项测试的进行,通常穿插在各部分功能测试之中.本文根据特殊项的不同显示形态,将该测试大致分为两部分.

第一部分的测试是针对特殊项以文字形式在页面中显示的情况,例如:在对 View 层的信息设置测试中,当操作或输入非法时弹出的文字提示信息.该测试内容主要包括:

验证是否有错误提示信息弹出,并验证该信息是否在显示数秒消失,或是否在操作/输入合法后消失(根据测试对象的不同情况,特殊项的消失方式可能略有不同).

其自动化测试的主要步骤如下:

首先利用 Selenium2Library 中关键字在 View 界面内对系统进行自动化的非法操作/输入;然后,利用关键字 wait until page contains 查看页面中是否在未超时情况下弹出了提示文字信息;最后,根据测试特殊项不同的消失方式,使用关键字验证页面中该文字是否在操作/输入合法后消失,或在 GUI 系统代码设定的时刻消失.

第二部分的测试是针对特殊项以确认/警告框形式弹出的情况. 该自动化测试的内容及主要步骤如下:

首先, 利用 Selenium2Library 中关键字查看是否存在确认/警告框弹出, 并正确显示; 然后, 运用 click 等关键字对弹出框进行确认/取消操作; 最后, 根据上一步的自动化操作, 验证是否得到合法的结果.

3 MVC GUI测试结构的实现

本文以 Linux 作为测试环境, 基于 Robot Framework, 对 OpenWRT LuCI 模块^[6,7]进行 MVC GUI 自动化测试结构的实现.

3.1 用户界面功能测试的实现

用户界面功能测试是为了检验界面中各控件显示是否正确, 界面基本操作是否运作良好. 本节以 OpenWRT LuCI 登陆界面为具体实例, 验证该功能测试框架的适用性, 其实现内容为: 登陆试图界面, 检验显示完整性, 填写用户名密码, 验证复位键, 点击登入键, 检验登入是否完成. 图 3 为登入界面.



图 3 OpenWRT LuCI 登陆界面

登陆视图界面的大致流程为: 启用 Firefox 浏览器, 在 URL 栏中填入视图所在 IP 地址, 登陆登入界面, 并进行窗口最大化. 若整个测试流程顺利进行, 说明 OpenWRT LuCI 模块的视图层拥有良好的用户接入能力.

在登入界面上, 主要控件包括文字、文本框和按钮等, 需要运用 Robot Framework 语句对这些控件显示的正确性进行检验.

控件除了显示的正确性之外, 其可用性至关重要. 本登入界面的逻辑流程如图 4 所示.

根据图 4 所示逻辑流程, 调用 Robot Framework 中 Selenium2Library 关键字库, 即可编写测试代码, 代码格式如图 5 所示.

在命令行中用 Pybot 命令运行该测试代码, 系统会自动显示自动化测试结果, 但仅显示各用例测试结果和总测试结果. 若想要查看各用例中具体日志报告, 可在本目录中找到日志报告文件进行查看. 图 6 为登入界面测试结果.



图 4 登入界面逻辑流程图

```
*** Test Cases ***
login web gui
set selenium speed .5 seconds

#Open Browser and go to login page
open browser ${WEBGUIURL} browser=${BROWSER}
maximize browser window
```

图 5 登入界面代码实例

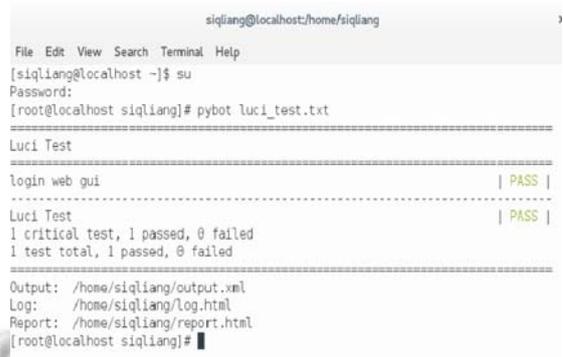


图 6 登入界面测试结果

3.2 信息查询/设置功能测试的实现

信息查询功能测试是为了检验查询到的信息是否与数据库中的一致.

本节以“接口-LAN”界面为例进行实现方法的阐述, 实现内容为: 获取用户界面中地址与子网掩码信息, SSH 至 OpenWRT 并获取相关信息, 对获取到的信息进行比较. 图 7 为“接口-LAN”界面.



图 7 接口-LAN 界面

首先,利用 Robot framework 中 Selenium2Library 关键字库,编写关于获取界面中 LAN 地址与子网掩码信息的测试代码.然后,使 Linux 虚拟机 SSH 至 OpenWRT 以获取相关信息.在这种情况下,可以直接利用 SSHLibrary 进行关键字代码编写,也可以编写个性化 Python 脚本,创建自己的关键字库,再进行关键字代码编写.图 8 为 Python 代码大致结构.

```
import sys;
import os;
import pexpect;
class ssh_roboter:
    def ssh_connect_addr(self,ip,name,pwd,cmd):
        ret = "failed";
        result = "";
        try:
            ssh = pexpect.spawn('ssh %s@%s "%s"' % (name,ip,cmd))
            ...
            ssh.expect('password: ')
            ssh.sendline(pwd)
            except pexpect.EOF:
                ret = 'ok'
            ...
        ssh.close()
        return ssh.before;
```

图 8 Python 代码结构

最后,将 GUI 中与命令行中获取的信息进行比较.若两值相等,则判断“接口-LAN”部分信息查询功能良好;若不等,则要进行进一步的错误排查.

信息设置功能测试在代码编写上,较信息查询功能测试而言,多了自动化 GUI 设置这一块内容,运用 Selenium2Library 中的关键字即可完成,在这里就不详细阐述了.

3.3 特殊项测试的实现

特殊项测试,通常穿插在各部分功能测试之中,以检验例如错误提示信息此类特殊项是否正常显示.图 9 为登入界面中的一种情况,当输入错误用户名或密码时,将有错误提示信息“用户名或密码错误,请重试”弹出.该情况的具体测试代码如图 10 所示.



图 9 登入界面-错误提示信息

```
#check whether there are error hints when input wrong password
input text      name=username      ${WEB_ADMIN}
input password  name=password      ${PASSWORD_WRONG}
click button    css=.cbi-button-apply
Page Should Contain      用户名或密码错误,请重试。
```

图 10 特殊项测试代码实例

4 结语

本文基于 Robot Framework 提出了一种面向 MVC GUI 模型的自动化测试框架,并以 OpenWRT LuCI 模块作为实例,进行了框架实现和适用性验证.

框架将 MVC GUI 模块分为四大部分进行全面功能性测试,分别涵盖用户界面基本测试、信息设置测试、信息查询测试以及特殊项测试.整个框架测试了 MVC 三大部件的所有功能,并将上层 GUI 和底层 Database 进行结合,完善了 GUI 自动化测试的覆盖性;根据不同系统环境,测试者可编写不同的 Python 脚本,灵活调用底层数据;而黑盒测试与白盒测试的有机结合,确保了模块代码的正确性和完整性.

较之于目前大多白盒与黑盒测试、手动与自动测试分离进行的 GUI 测试现状,本框架的提出不仅弥补了当前 MVC GUI 功能测试覆盖范围的漏洞,全面的自动化也提高了功能测试的效率,使代码缺陷得以快速锁定.

在未来的研究与实现中,还可以就整个系统(例如: OpenWRT)拓展出许多结合命令行的测试用例.例如,在 View 层设置固定 IP 正确后,可以到 Model 层调出信息核查,也可利用 OpenWRT 的命令进行验证等.

参考文献

- 刘欣.基于 MVC 模式的 Web 软件系统开发框架设计与实现[学位论文].济南:山东大学,2013.
- 李霞.MVC 设计模式的原理与实现[学位论文].长春:吉林大学,2004.
- 游泽青.面向 GUI 软件的自动化测试框架的研究与实现[学位论文].重庆:西南大学,2012.
- 张艳雪.复杂 GUI 应用自动化测试框架设计与实现[学位论文].北京:北京交通大学,2013.
- 黎守秀.基于 Robot Framework 的自动化测试研究及应用[学位论文].重庆:西南交通大学,2011.
- 危思思.基于 OpenWrt 开源平台的移动智能装备[学位论文].杭州:浙江大学,2014.
- 李俊灏.基于 OpenWRT 的多 WAN 口路由器.科技信息,2014,5:83,187.
- Xun Y, Cohen MB, Atif M. Memon. GUI interaction testing: incorporating event context. IEEE Trans. on Software Engineering, 2011, 37(4): 559-574.
- Alsmadi IM. Using Mutation to Enhance GUI Testing Coverage. IEEE Software, 2013, 30(1): 67-73.