

软件回归测试技术^①

刘雅君

(西安理工大学, 西安 710043)

摘要: 对回归测试的基本概念和回归测试现状给出论述, 分析 Selenium 开源工具在回归测试方面的缺陷和不足, 研究并实现适应回归测试的自动化测试方案, 并给出详细的实现方法。

关键词: 软件测试; 回归测试; 软件测试工具

Regression Test Technology

LIU Ya-Jun

(Xi'an University of Technology, Xian 710043, China)

Abstract: The paper introduces the concept of regression testing and the situation of regression testing using software automation testing tools. Then, it introduces the limitations of selenium, a popularly used automation testing tool. To address these limitations, it gives the solution.

Key words: software testing; regression testing; software automation testing tools

1 引言

软件测试是比较直接地发现软件缺陷的手段, 所以也就成为了软件缺陷控制的主要手段。回归测试作为软件测试中的重要环节, 是为保证软件在不断改进中的正确性测试技术。

1.1 回归测试的基本概念

回归测试是一种用来保证各种针对软件的改动不会带来不可预测的行为或者另外的错误的测试活动^[1]。所指的改动主要包括发展性的改动和修正性的改动:

1.1.1 测试中的发展性改动

针对发展性的改动的回归测试, 是集成测试的重要手段。每当一个新的模块被当作集成测试的一部分加进来的时候, 软件就发生的变化。新的数据流路径建立了起来, 新的 I/O 操作操作可能也会出现, 还有可能激活了新的控制逻辑。这些改变可能会使原本工作的正常的功能产生错误。在集成测试策略的环境中, 回归测试是对已经执行通过的测试案例再重复执行某些子集, 以保证上述改变不会传播无法预料的副作用。

根据不同的集成测试策略, 回归测试技术也不同。在自顶向下测试中, 需要对当前待测模块进行必要的

条件假定, 假定当前模块所涉及的所有相关模块都是正确无误的, 因此需要设定桩模块, 即修改它们为空模块。在之后的通过测试中, 再逐次地修改替换它们为真实的, 一个个地通过替换后得测试用例, 以确定它们都是正确的。在这种策略下, 针对已经通过用例的模块, 当加载新替换的桩模块后, 可能会引起路径、I/O 操作和控制逻辑等的变化, 使原来正确地模块受到影响。因此, 需要重新执行原来已经执行过的测试用例。自底向上的测试由于无需编写桩模块, 因此, 没有回归测试问题。

1.1.2 测试中的修改性改动

修正性的改动, 可能源于发现了错误并做的修改, 也可能是出于满足客户的需求或适应应用的要求对现有功能进行的修改, 以及其他情况下的对软件进行的改变。每当软件被修正性的改变后, 软件配置的某个方面(程序、文档或者数据)都有可能被修改了, 需要重新执行已经执行过的测试用例, 为确保修改后的软件成分对旧的测试用例仍然可行。

这种情况下的回归测试, 是用来保证修正性的改动不会带来不可预料的行为或者另外的错误的活动。针对修正行改变的回归测试, 有 2 种用例选取方案:

① 收稿时间:2010-10-19;收到修改稿时间:2010-12-04

完全重复性回归测试和选择性回归测试。

回归测试的价值在于它是一个能够检测到回归错误的受控实验。我们需要根据项目实际情况，选择正确的策略进行有效的测试，达到平衡。

1.2 回归测试现状

回归测试是软件测试的重要组成部分，占有很大比例。对于一个软件开发项目来说，项目的测试组在实施回归测试的过程中会将所开发的测试用例保存到“测试用例库”中，并对其进行维护和管理。当得到一个软件的基线版本时，用于基线版本测试的所有测试用例就形成了基线测试用例库。在需要进行回归测试的时候，就可以根据所选择的回归测试策略，从基线测试用例库中提取合适的测试用例组成回归测试包，通过运行回归测试包来实现回归测试^[2]。在实际工作中，为了保证回归测试质量，提高测试脚本复用质量、效率和加强测试过程的管理是至关重要的。

2 对测试工具回归测试支持的改造及实现

针对上述的回归测试的特征和技术向实现按点，结合具体项目对采用的 selenium RC-java 工具进行必要的分析和扩展。

2.1 针对回归测试 Selenium 工具的缺陷

Selenium 是 ThoughtWorks 专门为 Web 应用而开发的自动化测试工具，适合进行功能测试、验收测试。Selenium 主要分为三个部分：Selenium Core、Selenium IDE、Selenium Remote Control。选择此工具，主要是因为其具有以下几点优势：是完全开源的测试工具；支持一些主流编程语言编写测试脚本，包括 java, .net, Perl 等；允许在任何支持 JavaScript 的浏览器中进行测试，例如 IE, Firefox 等；支持多种操作系统 (Windows, Max OS 和 Linux 等)^[3]。但是，Selenium 工具在对自动化回归测试方面也存在许多缺陷：

1) 大量使用 name、id、xpath 等页面元素。无论是功能修改、UI 重构还是交互性改进都会影响到这些元素，这使得测试变得脆弱。

2) 过于细节的页面操作不容易体现出行为的意图，一段时间之后就很难真正把握测试原有的目的了，使得 Selenium 测试变得难于维护。

3) 对具体数据取值的存在依赖，当个别数据不再合法的时候，测试就会失败，但这样的失败并不能标

识功能的缺失，这使得 Selenium 测试变得脆弱且难以维护。

4) Selenium 是轻量的测试框架，脚本所处理的测试用例构成简单，其实质就是通过 HTTP 协议，发送请求来完成测试用例，所以很困难处理业务逻辑关系强的测试用例。

5) Selenium 和大多数的自动化测试工具一样，也在很多方面缺乏对全球化回归测试支持：在没有指定浏览器参数文件的情况下，Selenium Server 会启动一个系统默认设置的浏览器执行测试，默认设置语言为当前浏览器版本语言；操作函数直接使用页面元素作为参数，需要根据软件不同语言版本对测试脚本中同一页面元素文本属性值进行重复修改，使得回归测试变得脆弱；不能提供多语言和字符集的测试数据等等。

2.2 对 Selenium 工具和脚本的改进

针对上述缺陷，将尝试对 Selenium 工具和脚本进行分析并进行改进：

2.2.1 使用 xml 格式的参数文件，方便参数的维护

在 Selenium 工具包中，操作函数直接使用大量页面元素作为参数，而且对这类数据取值的存在依赖。无论是功能修改、UI 重构，还是交互性改进都会影响到这些元素，例如页面元素 name、id、xpath 等属性值的修改。当个别页面元素数据不再合法的时候，执行回归测试就会失败。但是这样的失败并不能标识功能的缺失，使得 Selenium 测试变得脆弱且需要针对测试脚本中同一页面元素属性值进行重复修改。

为了提高测试脚本的维护性，针对重复修改问题，编写程序实现页面元素参数的提取，实现参数的替换。页面参数文件定义为 xml 格式，并且依据所在模块命名。如用户管理模块，将参数文件命名为 test_gui_user.params。基本参数文件格式如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<parameters file="">
  <param name="NewUserButton">_sBox:b5c
109e:0_1:b5c10a1</param>
  .....
</parameters>
```

为了提高程序的健壮性，尽量使用 xpath 等相对稳定和标示性好的页面元素属性作为页面参数。同时，为了增加判断逻辑，通过多次判断页面元素确认页面正确与否。

2.2.2 改善超时机制，提高回归测试的健壮性

进行回归测试时，测试结果和测试机器性能息息相关。很多时候，仅仅因为个别数据传输延迟就造成测试失败，但是这样的失败同样不能标识功能的缺失。为了提高程序健壮性，改写基本页面元素判断函数，添加超时功能。

首先，在参数文件中，定义长短不同的 3 种命名为 INT_TIME_LOW, INT_TIME_MED, INT_TIME_HIGH 的等待时间参数。然后，在调用的等待时间参数内，每 300 毫秒执行一次对页面元素的判断。一旦判断为页面元素显示正确，则跳出循环执行后续脚本，在规定时间内，仍没有显示正确才返回超时错误。所修改函数如下：

```

public int waitForObjTimeOut(Selenium objSel,
String objApp, long timeOut) //等待对象函数
public int waitForObjTimeOut(Selenium objSel,
String strAppObj, String strTitleObj, long lngTimeOut)
//等待对象函数（重载形式）
public int waitForText(Selenium objSel, String txt)
//等待文本函数
public int waitForTitle(Selenium objSel, String title,
long lngTimeOut) //等待页面标题函数
public int waitForTableToRefresh(Selenium objSel,
String strFirstRowCellId, long timeOut) //等待表格刷新
函数
public int waitForWindowToClose(Selenium objSel,
String strWinLocator, long timeOut) //等待窗口关闭函
数
public int waitForWindow(Selenium objSel, String
strWinLocator, long timeOut) //等待窗口开启函数
public int selectWindowTimeOut(Selenium objSel,
String strWinLocator, long timeOut) //等待欲选定窗口
函数
public int waitForObjVisibility(Selenium objSel,
String objApp, long timeOut) //等待对象可见函数
public void waitForTime(long sec) //等待函数

```

2.2.3 过于细节的页面操作，创建小而独立的操作模块

过于细节的页面操作不容易体现出行为的意图，一段时间之后就很难真正把握测试原有的目的了，这使得 Selenium 测试变得难于维护。针对此问题，依据模块化测试脚本框架思想，创建小而独立的可以描述

的操作模块。将这些小脚本组合起来，就能组成能用于特定的测试用例的脚本。如编写删除用户脚本，可以使用登陆系统、初始化用户数据、添加用户、查找用户、删除用户五个小脚本组合起来。

2.2.4 添加截图，方便测试人员判断和调试

添加 addScreenShot () 截图函数，可以在测试脚本异常处、测试点等关键点处调用，方便调试程序和判断测试结果。

2.2.5 针对动态信息的判断，使用描述性编程方法

每次单击按钮都回新建一个窗体，窗体的名称可能动态产生的。通过脚本程序的界面操作，在回归测试时是肯定是要失败。这种情况，可以通过描述性变成解决。使用测试脚本中给出的对象描述来查找对象，查找的位置不是固定的参数，而是与测试程序运行时为其创建的临时测试对象版本进行匹配。把描述性编程中的所有“描述”控件属性的值当作正则表达式来处理，因此，替换成特殊字符例如“*”、“?”等在正则表达式中有意义的字符。

2.2.6 针对一些业务逻辑关系强的测试用例，使用 Java . awt . Robot 来执行真实的鼠标动作

Selenium 是轻量的测试框架，脚本所处理的测试用例构成简单，其实质就是通过 HTTP 协议，发送请求 (request) 来完成测试用例。因此，Selenium 命令会对一些重量级的组件存在没有响应的情况，而且针对一些业务逻辑关系强的测试用例处理困难。

针对这种情况，一般有两种解决方法：

第一种方法是改造模拟 Selenium 工具包中模拟 JavaScript 事件的代码。这种方法的缺点在于，测试人员可能不能从用户角度观察测试过程。

第二种方法是，使用 Java.awt.Robot 来执行真实的鼠标动作，这样需要重新编写动作代码，具体做法如图 1 所示：

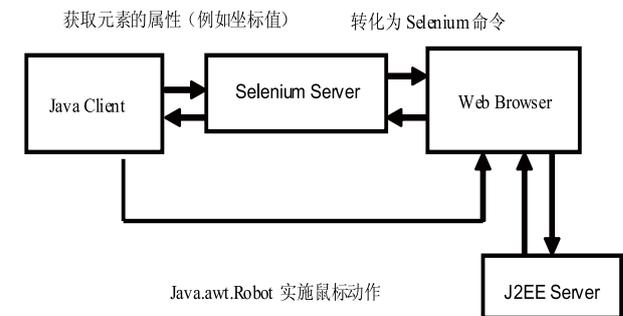


图 1 使用 Java.awt.Robot 的流程图

在 Selenium 工具中,任何 HTML 元素都可以通过其 Xpath、DOM API、或者 plain ID 等属性值来定位。通过调用 Selenium 的函数可以返回所需 HTML 元素的 XY 坐标值。本来,Selenium Server 与浏览器使用 AJAX (XmlHttpRequest) 直接通讯。现在当返回 HTML 元素的 XY 坐标值后,直接调用 Java.awt.Robot 在浏览器上执行真实的鼠标动作。

2.2.7 增加针对全球化回归测试的支持

Selenium RC 基本测试流程如图 2 所示,它提供了一个 Selenium Server,可以自动开始/关闭/控制所支持的浏览器。Selenium Server 与浏览器使用 AJAX (XmlHttpRequest) 直接通讯,可以使用简单的 HTTP GET/POST 请求直接向服务器发送命令,即可以使用任何可以发出 HTTP 请求的编程语言在浏览器中自动执行 Selenium 测试。

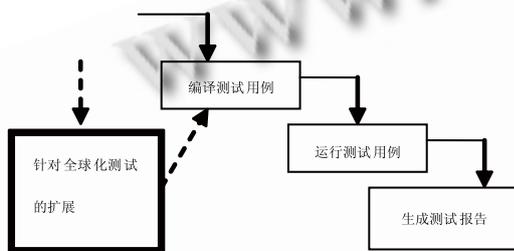


图 2 基本测试运行流程和针对全球化测试扩展的添加

如图 2 所示,针对 Selenium 工具的对全球化回归测试支持的不足,通过对测试脚本结构的设计,以及在 Selenium RC 封装的 Java 工具包中添加对环境设置的选择和参数文件的处理等改进,从而实现针对全球化测试的扩展任务。

3 结论

伴随渐进式或迭代式的面向对象开发技术不断发展的同时,回归测试的研究也需要更多重视。本文论述了回归测试概念、分类和现状。针对回归测试,分析了 selenium-java 开源工具在回归测试方面的缺陷和不足,研究并实现适应回归测试的自动化测试方案,并给出详细的实现方法。本文提出的已经应用于某公司的 web 产品的国际化测试中,并取得很好的效果。当然,回归测试涉及的问题还有很多需要继续深入研究,例如重量级回归测试框架的研究和全面的回归测试平台的研究等。

参考文献

- 1 Pressman RS. 软件工程--实践者的研究方法. 北京:机械工业出版社,1999.
- 2 李刚毅,金蓓红. 自动化回归测试的技术与实现. 计算机应用研究,2006,30(3):100-103.
- 3 Selenium. <http://wiki.javascud.org/display/SEL/Home>, 2002.

(上接第 128 页)

- 的移动终端设计与实现. 计算机应用与软件,2010,27(3).
- 4 陈科,刘宁,冯继宏,等. 基于 Socket 和多线程的远程医疗数据传输系统设计. 北京生物医学工程,2010,29(4).
- 5 杨立才,刘乃智,王德伟. 基于 SIP 的社区医疗网络系统设计. 现代电子技术,2008,31(8).
- 6 胡学芹. 远程会诊系统中实时通信子系统的设计与研究[学位论文]. 杭州:浙江大学,2007.
- 7 陈章样,郝继飞,张道明. 基于 Windows CE 的便携式心电监护系统. 工业控制计算机,2007,20(11):31-32.
- 8 Cabral JE, Kim Y. Multimedia systems for telemedicine and their communications requirements. IEEE Commun. Mag, 1996: 20-27.

- 9 熊利祥,吕锋. 基于 JMF 的远程视频监控系统的研究. 中国科技论文在线精品论文,2009,2(18):1-6.
- 10 孙一林,彭波. Java 网络编程实例. 北京:清华大学出版社,2003.219-277.
- 11 何详,吴庆宪. 基于 CS 与 BS 模式的远程控制实验系统. 电光与控制,2005,12.
- 12 朱晓蕾,于达夫. 基于客户/服务器模式的远程控制系统. 计算机工程与设计,2004,25(5):796-799.
- 13 Downing TB. Java RMI: Remote Method Invocation. IDG Books Worldwide, New York, 1998.
- 14 Weaver AC. Secure sockets layer. Computer, 2006, 39(4): 88-90.