针对软件系统功能的自动化测试工具设计①

张志敏 周四望 (湖南大学 软件学院 湖南 长沙 410082)

摘 要: 通过分析目前常用测试工具的应用范围和一些优缺点,设计一种操作界面友好型、易用性、功能针对性强的自动化测试工具 openATF, 提出一种支持多方法步骤的测试用例生成和自动执行方案,用户可以方便通过 web 界面输入用例数据和用例预期结果,并且在广州地铁 AFC 系统功能测试工作中的应用,结果表明能很大程度上的提高测试质量和效率, 节省人力资源并确保可信度。

关键词: 测试用例; 测试用例集; 模拟器; 软件测试

Design of an Automatic Testing Tool for Software System Functions

ZHANG Zhi-Min , ZHOU Si-Wang

(College of Software, Hunan University, Changsha 410082, China)

Abstract: This paper analyses the applications, advantages and disadvantages of current testing tools. It designs an automatic testing tool named open ATF, that is web-UI friendly, easy to operate, and feature-targeted. It also presents a testing scheme about generating test and executing automatically case with multi-method and multi-step. Users can conveniently input the testcase data and expect result. Through applying it in testing the system of Guangzhou Metro AFC, the results show that it can improve the quality and efficiency of testing to a large degree, and save human resources and gurantee reliability.

Kewords: testcase; testsuite; simulator; software testing

1 引言

随着信息技术的飞速发展、软件产品越来越多的应用到社会的各个领域，对软件的测试工作提出更高的要求和效率，软件测试中可以采用手工测试，但是重复性有限，而且需要大量的人员和设备，特别是大量的测试工作需要在短时间内完成，若没有大量人力资源，靠手工测试是无法开展的，由于手工测试的局限性，多种自动化测试工具[1]应运而生。

通过对软件产品功能进行需求分析[2]总结，明确软件功能再进行测试。采用自动化测试工具主要具有如下的优势:

(1) 在较短的时间内进行更多更加频繁的测试，脚本可以重复利用，重复运行，降低开发成本，这是手工测试所不能做到的。

(2) 可以执行一些手工测试比较麻烦的或不可能进行的测试。比如，对于大量数据的输入的测试，但是却可以通过自动化测试模拟同时有很多用户或者产生很多数据，从而达到测试的目的。

(3) 便于程序的回归测试。在程序修改比较频繁时，效果是非常明显的。由于回归测试的动作和用例是完全设计好的，方便用户的操作，测试期望的结果也是预先得知的，将回归测试自动执行，可以极大提高测试效率，缩短时间。

(4) 更有效的利用资源。友好的操作界面、用例和用例数据的管理可以提高测试人员的积极性和测试结果准确性，测试技术人员测可以投入更多精力设计更好的测试用例。

(5) 使得测试具有一致性和可重复性。由于测试是自动执行的，每次测试的结果和执行的内容的一致性是可以得到保障的，从而达到测试的可重复的效果。

① 基金项目:湖南省自然科学基金(09JJ3123)
收稿时间:2009-09-21;收到修改稿时间:2009-11-09

© 中国科学院软件研究所 http://www.c-s-a.org.cn

(6) 测试的复用性。由于自动测试通常采用脚本技术，可以导入导出用例和用例数据，做很少的修改或者不做修改，可以达到在不同的测试过程[3]中使用相同的用例。

目前国内外的测试管理工具有如下几种，下面是对几种常用的工具进行比较，列出其中的优缺点，给文中设计的 openATF 自动化测试工具提供一定的依据。

(1) TestDirector

优点：①Web-UI 形式管理，界面友好；②有测试用例执行跟踪的功能；③和自身的缺陷管理工具紧密集成。

缺点：①快速执行测试用例的时候，不能看到测试用例内容；②每个项目同时操作的人数有限。

(2) TestManager

优点：①强大测试用例的管理功能；②可以和 Rational 的测试工具 robot、functional 相结合；③有测试用例执行的功能，但必须先生成对应的手工或自动化脚本。

缺点：①没有权限管理功能 51Testing 软件测试网□G;E,O6cb(r□K*_,s□o；②支持的数据库类型少，不能支持太多的访问并发用户数；③测试用例不太稳定，有时会造成用例丢失；④要安装客户端才可使用，所以造成交流不方便；⑤测试用例的展示形式单一。

(3) Test Runner

优点：①web-UI 形式的管理；②自动邮件提醒；③有测试用例执行管理；④测试用例分优先级。

缺点：①测试用例的编写不方便；②安装配置比较繁琐。

(4) Testlink

优点：①web-UI 形式管理，界面友好；②可以自定义和其他缺陷管理工具的整合；③具有需求管理的功能。

缺点：①不具备自动测试功能；②测试用例结果统计不完善。

全文组织如下：在第二部分将以工具的易用性、高效性和可重复性等特点设计自动化测试工具框架[4]；第三部分通过面向 web-UI 操作，设计出一种新的编写测试用例方案[5]，并且定义好用例数据的结构，能完成输入数据、输出数据、自动获取结果、和预期结果自动对比等功能；第四部份试验与分析，通过实验结果证明该工具能达到预期效果和设计目标；最后总结和展望。

2 自动化测试工具框架设计

文中设计的测试框架是针对面向消息型、基于请求/响应的功能性自动化测试。框架主要分成三大部分：openATF 管理端，模拟器，被测实体。采用这种框架的优点在于 openATF 端的友好的操作界面使得测试人员可以更加专注于测试用例的编写、测试的自动执行、测试结果的分析统计尽早的发现缺陷，不必关注模拟器端如何实现，模拟器将由专门的模拟器编写人员来维护，任务分工明确。如图 1 所示：

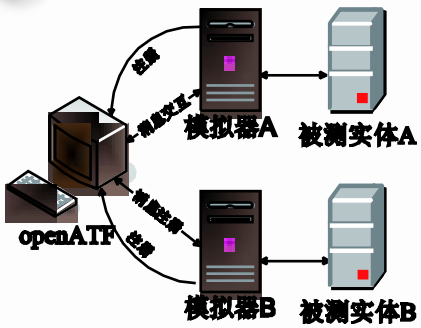

图 1　OpenATF 框架

整个测试过程就是，由 openATF 端通过在浏览器上生成测试用例[6],启动被测实体、模拟器，模拟器启动时向 openATF 发送注册消息，此消息中包含有本模拟器提供给外部调用的方法，openATF 收到模拟器的注册信息后将其存储在 xml 文件中。当测试用例执行时，openATF 通过调用模拟器提供的方法，以 http+xml 向模拟器发送数据,从图中看出，模拟器收到 openATF 发过来的信息，调用被测实体提供的接口，来驱动测试，从被测实体返回 xml 数据到模拟器端，再有模拟器发送给 openATF,openATF 再将被测实体返回的结果与预期结果进行比较，然后得出最后结果，通过或者失败。在此过程中用户，只需要点击执行，就可以方便的进行测试。

openATF 管理端的作用：①生成测试用例，②给用例添加方法，③添加测试数据，④添加预期结果，⑤执行自动测试，⑥将被测实体返回结果与预期结果对比得出最后结果。

模拟器在 openATF 自动化测试框架当中，发挥

© 中国科学院软件研究所 http://www.c-s-a.org.cn

至关重要的作用，它是真正和被测实体进行交互的对象和通信的桥梁，openATF 通过控制模拟器，从而控制用例的执行逻辑。

模拟器的具备功能：①提供交互命令，给测试人员使用，测试人员可以输入测试指令进行测试；②它既是服务器也是客户端；③将 openATF 端发送来的 xml 格式数据转换后，通过服务端发送到被测实体；④当被测实体有数据返回的时候，将数据换为 xml 格式数据,模拟的客户端接受后将其进行转换再发送给 openATF 的服务器；⑤模拟器具有 GUI 界面，可以看到交互的信息；⑥模拟器启动时要主动向 openATF 发送注册消息，将自己可以提供的方法告知 openATF，以便调用；⑦模拟器要持续向 openATF 发送心跳信号。

用户通过 web-UI 生成测试用例一直到查看用例的测试结果，其活动图如图 2 所示。

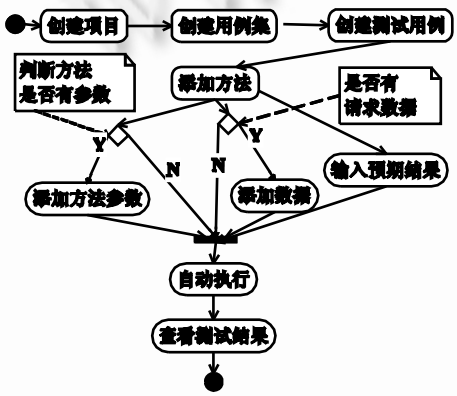

图 2　测试流程

从图 2 可以看到用例执行到输出测试结果仅在一个基于浏览器的应用中便可完成，而不需要每个客户端都安装一套客户端程序。这样可以方便测试人员操作和加快测试的速度。该自动测试系统的自动执行不仅只对单个用例可以执行，还可以执行用例集或者项目，操作灵活，加快测试速度，系统将以报表的形式进行结果统计，方便用户查看。

根据以上，自动化测试框架分成三大部分，openATF 与模拟器之间消息交互采用 http+xml 方式。模拟器注册、消息交互等都是建立在 http 协议基础上，附带 xml 实体，openATF 和模拟器都需要具有收发 xml 实体、解析 xml 的功能。

在通信方式上，无论是 openATF 还是模拟器在收到 http 消息时都要返回一个 200 OK 的响应。对于 http 消息里的 xml 实体内容，分为请求消息和响应消息。所以在某一方收到一个请求消息时，除了返回一个 200 OK 的响应外，在处理完毕后还要针对此请求返回一个 http 响应的 xml 实体给对方。

表 1 是经过多次讨论和试验得出的消息格式，其中各个节点的含义：

表 1　请求和响应消息格式

A：请求消息格式

```
<RequestHeader>
    <type>send</type>
    <version>1.0</version>
    <sender>172.16.53.146:8001</sender>
    <receiver>172.16.53.99:8002</receiver>
    <messageid>SC-SIMULATOR</messageid>
    <transid>0x1111</transid>
<RequestBody>
  <methods>
      <method>
      <name>getDeviceState</name>
      <param>String,int,boolean</param>
      <description>.....</description>
      <datas></datas>
   </method>
</methods>
</RequestBody>
</RequestHeader>
```

B：响应消息格式

```
<ResponseHeader>
      <type>send</type>
      <version>1.0</version>
      <sender>172.16.53.99:8002</sender>
    <receiver>172.16.53.146:8001</receiver>
    <messageid>SC-SIMULATOR</messageid>
    <transid>0x1111</transid>
<ResponseBody>
    <results>
    </results>
</ResponseBody>
</ResponseHeader>
```

注：1.<RequestHeader>标识请求消息头，2.<ResponseHeader>标识响应消息头，3.<type>标

© 中国科学院软件研究所 http://www.c-s-a.org.cn

识消息的类型，目前可用的有：register，send，general，error， 4.<version>标识消息的版本，5.<sender>标识消息发送方，以 IP+端口号方式提供， 6.<receiver>标识消息的接收方，以 IP+端口号的方式提供， 7.<messageid>：标识模拟器的名称， 8.<transid>标识请求和响应的事务，请求与响应的 transid 是相同的。

3　用例编写和用例数据结构定义

由于在当前的软件开发过程中，一份好的测试用例对测试有很好的指导作用，能够及时的发现问题，测试人员在测试过程中一直在积累编制测试用例[7]经验，争取编制出好的测试用例来指导测试，可以使测试人员理清头绪，避免做重复多余的测试；分清楚测试工作的重点和优先级；每个测试用例都已经写好测试步骤和预期结果，不需要太多思考，照做就可以；可以知道那些测试用例测出的 bug 的几率比较大。

该自动化测试工具的测试用例是在已知被测实体的执行方法的情况下，给测试用例按顺序添加被测实体的方法和方法的参数，如果该方法需要数据则添加测试数据，预期结果是每个方法都有的，数据都支持 xml 格式的导入和导入，采用这种自定义标准的数据驱动模式关键在于：第一，可以降低增加测试用例的时间成本，不需要重复而繁琐的输入数据，第二，降低用户的二次开发成本，第三，便于维护和使用，第四，保证测试用例的延续性和扩展性。

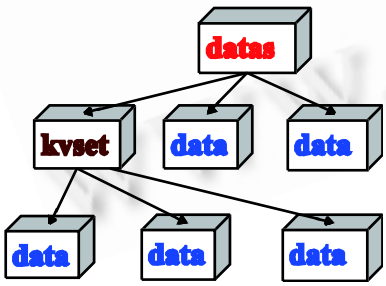

图 3　用例数据结构定义

注：其中根节点的 datas 和 kvset 元素不含任何属性；真正用到的数据是 kvset 下面的 data 元素，kvset 节点命名是根据 java API 中 Map 类中的键值对，形如 key-value。其中 key-value 中的 key 与 data 元素节点中的属性 key 对应，value 与 data 元素中的 value 属性相对应，data 元素中的 type 属性为 value 值的类型，如：String、int、long 等等。

在图 3 中 datas 节点为根节点与表 1 中的请求消息格式中的 datas 元素是对应的，datas 元素下的结构与响应消息中的 results 元素下的结构是一种形式，因为一份请求数据对应一份返回结果。xml 数据的解释将由模拟器来完成。

用例数据和预期结果都采用这种形式的定义，树形展现为无限极的，结构清晰，由于数据结构一样，比较以后得出的测试结果信任度提高。

图 3 所示结构对于下的 xml 格式定义。预期结果和被测实体返回结果的数据结构在 xml 中都是按照这种方式所定义。

```
<datas>
  <kvset>
    <data type="string" key="n1" value="v1"/>
    <data type="string" key ="n2" value="v2"/>
   <kvset>
    <data type="string" key ="n3" value="v3"/>
    <data type="string" key ="n4" value="v4"/>
    <data type="string" key ="n5" value="v5"/>
  </kvset>
 </kvset>
</datas>
```

以上这些设计都是经过在实际项目中不断的进行反复测试和验证的基础上修改，从而达到了产品设计的目标。符合测试人员的习惯。更好的利用了公司的资源，很方便的执行回归测试，只有经过大量测试用例测试过的软件版本才是最可靠的，而只有只用自动化测试工具才能保证在短时间内完成大量的测试工作。

4　应用实例总结

实验测试环境：SC(Station Computer) Simulator,即：车站计算机模拟器，主要用于实现 Open-ATF 与 LCC(Line Central Computer)之间的通信；

© 中国科学院软件研究所 http://www.c-s-a.org.cn

被测实体 LCC，即：线路计算机，主要用于设备状态监视、启动降级、紧急模式等等；数据库使用 sybase，openATF 端服务器使用 jboss。

实验目标：测试 openATF 的最后测试结果是否正确，因为测试结果是已知的手工测试结果，如果和手工测试结果完全吻合的，说明工具的有效性，能保证测试质量和效率。

在 web-UI 界面中，用户编辑好测试用例，然后选择测试用例添加被测实体要执行的方法，如果方法有参数就输入方法参数，方法的用例数据，还有返回的预期结果。准备以上条件以后开始执行测试用例，测试用例的执行方案灵活，可以选择单个用例、用例集或者一个项目，使得用户可以有针对性的进行测试。在执行过程中可以查看执行相关信息，包括用例数据的发送和接受，模拟器端的情况等等信息，这些信息都将保存在日志里面，以便测试人员的查看。

自动执行完以后测试结果的展现界面，openATF 采用的测试结果统计方式有分为：全部测试测结果展现(其中也包括未测的用例)、失败用例、成功用例、锁定用例、未测试用例，如图 4 展示的是全部的测试结果，只是一个项目中的一部分测试结果。用户如果需要看测试详细信息可以通过点击用例名查看。

属于产品【地铁LCC功能测试】----全部测试用例统计结果

序号	用例集	用例名	状态
1	1 参数管理	1.1 参数版本报告	通过
2	2 状态监控	2.1 车站进入降级模式	通过
3	2 状态监控	2.2 车站取消降级模式	通过
4	2 状态监控	2.3 设备进入初始关闭状态	失败
5	2 状态监控	2.4 设备进入脱机状态	失败
6	2 状态监控	2.5 设备正常工作状态	通过
7	2 状态监控	2.6 设备进入故障状态	通过
8	3 交易	3.1 AGM进站交易	通过
9	3 交易	3.2 TVM发售交易	通过
10	4 控制	4.1 控制LCC发送降级模式控制命令	锁定

图 4 测试结果展现

通过多次的自动化的测试试验和在 AFC 系统中的应用，该 openATF 自动化测试工具达到了产品的预期的效果和设计目标。

5 结语

软件测试的目的不仅仅是为了找出错误，是通过它发现错误、分析错误、找到错误的分布规律，从而帮助项目管理人员发现在软件开发过程中的缺陷，以便改进；实践证明该工具能针对性的测试，提高测试的有效性，友好的操作界面，用例编写和数据的操作更加灵活保证软件测试的质量和满足公司的实际需要。

参考文献

1 Ciupa I, Leitner A. Automatic Testing Based on Design by Contract. Proc. of Net.ObjectDays 2005. 545－557.

2 付剑平,陆民燕,阮镰.软件测试用例生成中的前置条件分析.计算机应用研究, 2007,24(3):103-105.

3 Marinov D, Khurshid S. TestEra: A Novel Framework for Automated Testing of Java Programs. Proc. 16th IEEE International Conference on Automated Software Engineering (ASE), 2001. 22－34.

4 赫建营,晏海华,刘超.基于本体的软件测试知识管理模型研究.计算机科学, 2007,34(10):280－284.

5 江曼,王天青,潘金贵.一个面向对象软件自动测试框架的设计和实现.计算机应用与软件, 2007,24(3):69－71.

6 Cheon Y, Kim MY, Perumandla A. A Complete Automation of Unit Testing for Java Programs. Proc. of the 2005 International Conference on Software Engineering Research and Practice(SERP'05), Las Vegas, 2005. 290－295.

7 Tonella P. Evolutionary testing of classes. In:International Symposium on Software Testing and Analysis (ISSTA'04).Boston,Massachusetts, USA: ACM Press, 2004. 119－128.

© 中国科学院软件研究所 http://www.c-s-a.org.cn