

The Design and Implement of the Automation Test Based on Vehicle Certification System

Hengguo Zhuang, Jiansheng Guo, Xuebin Shao

China Automotive Technology & Research Center, Tianjin
Email: zhuanghenguo@catarc.ac.cn

Received: Oct. 6th, 2016; accepted: Oct. 23rd, 2016; published: Oct. 26th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

With the accelerating pace of the information technology, as the important guarantee for handling business and providing service, the development tasks of software are becoming increasingly arduous. All kinds of management software have become increasingly complex, which puts forward higher requirements for software testing. In order to save time and expenses, we must use automated testing technology instead of inefficient manual testing for saving software testing costs and shortening the release cycle of the product. Selenium is relatively mature in the automated testing tools. Especially after Selenium 2.0, the integration of Webdriver API significantly enhances its performance. Based on the software in the automotive sectors, this paper applies Selenium 2.0 to it with TestNg (Testing Next Generation) framework, which not only enhances the reusability and efficiency, but also improves the stability and reliability during system updating.

Keywords

Automated Test, Selenium, TestNg

基于车型认证系统的自动化测试设计与实现

庄恒国, 国建胜, 邵学彬

中国汽车技术研究中心, 天津
Email: zhuanghenguo@catarc.ac.cn

收稿日期：2016年10月6日；录用日期：2016年10月23日；发布日期：2016年10月26日

摘要

随着信息化步伐不断加快，软件作为处理业务和提供服务的重要保障，其开发任务日趋繁重，各类管理软件也变得越来越大和复杂，这就对软件测试提出了更高的要求。为了节省大量的时间和开支，必须使用自动化测试技术代替效率低下的手工测试，节省软件测试成本，缩短产品发布周期[1]。Selenium是众多功能自动化测试工具中较为成熟的一个产品。当集成了WebDriver API，进入了Selenium 2.0时代后，其能力和影响力得到极大增强。本文研究了基于Selenium 2.0的自动化测试技术，对TestNg (Testing Next Generation)框架进行封装，并结合汽车行业软件申报零件多且重复性高的特点，在实际汽车软件平台中加以应用，不仅提升了软件测试工作的积累和复用度，而且有效保障了系统升级过程中的持续稳定性。

关键词

自动化测试， Selenium， TestNg

1. 引言

传统的手工测试，因为其成本高、执行繁琐、效率低的特点，正在逐渐向自动化测试进行转变。特别是最近五年，伴随着敏捷开发模式的流行，自动化测试技术以其快速迭代、快速回归、快速响应等优势，在测试工作中作用愈发突显并得以迅猛发展。开源的测试组件 Selenium 2.0 的出现，让软件自动化测试技术克服了早期使用录制回放的不足，使得测试工作更加经济、有效、便于实施和维护。

当前，汽车企业在机动车产品型式认证过程中面临着管理混乱、成本高等诸多问题，为了帮助企业建立一个统一的型式认证平台，从而提高认证参数的一致性管理水平，实现企业内部认证工作的电子化，加快在汽车生产企业中运用电子信息技术，中国汽车技术研究中心的数据资源中心开发了“机动车型式认证管理系统”。为了响应众多车企的“机动车型式认证管理系统”上线部署要求，并考虑产品运维期间的测试工作，引入了基于 Selenium2.0 的功能自动化工具，结合 TestNg、Maven 等技术，并根据车企产品的自身特点，搭建了完整的自动化测试框架。

2. 研究背景

2.1. 背景

软件开发是一个持续集成和改进的过程，而每一次修改都有可能产生错误。因此，当软件产品的部分、全部、或者应用环境被修改时，都需要对软件产品重新进行测试，其目的是验证修改后的系统或者产品的质量是否符合规格说明。对于产品型的软件，每发布一个新的版本，其中大部分功能和界面都和上一个版本相似或完全相同，这部分功能特别适合于自动化测试[2]。

相比手工测试，自动化测试具有诸多优势：① 无人值守。利用非工作时间或服务器空闲时，自动执行测试并收集结果，实现 24 小时办公；② 解放人力。测试程序按照预设策略，自动测试现有功能，使测试人员可以专注新功能的测试设计；③ 安全可靠。自动化测试脚本或程序，每次运行时都会准确执行相同的操作，避免了人为错误；④ 降低成本。通过程序反复测试，可以有效避免新功能上线对原有功能影响的隐患，大幅降低了成本；⑤ 提高效率。自动化测试执行速度比人工快，极大提高了测试

效率[3]。

2.2. 国内外研究现状

二十世纪八十年代初，美国北卡罗来纳大学召开了第一届软件测试技术会议，这是软件测试技术史上的一个重要里程碑，软件测试作为软件工程的一个重要的分支开始凸显其作用和地位。一批经典的软件测试模型和流程规范不断涌现，测试技术不断发展。IEEE 及 ACM 等标准化组织陆续定义了一系列的软件测试标准，众多的国外知名大学及研究所也对此进行了大量的基础和应用化研究，如 Carnegie Mellon 大学开展了基于关键字驱动的自动化测试和基于 C/S 架构的测试技术研究[4]。

虽然软件测试行业在国外得到了很大的发展，但是对于国内的软件行业来说，软件测试仍然是一个相对新兴的行业[4]。与国外相比，国内的软件测试从技术到规范，从基础理论到实际应用，都有着很大的差距。近年来，随着我国综合国力的上升，软件行业尤其是软件测试行业，在应用技术上得到长足发展，大量的企业开始采用开源或自主研发的自动化测试工具及框架，从 2007 年开始，如交通银行、浦发银行、中国人寿、平安保险、淘宝、百度等金融、保险、互联网公司大规模使用自主研发的自动化测试框架或工具，其研发基础基本采用的是开源框架。

3. 设计与实现

3.1. 车型认证系统简介

现阶段我国汽车产品型式认证呈现多头管理的状态，形成了由国家发展和改革委员会、国家工业和信息化部、国家质量监督检验检疫总局及其所属国家认证认可监督管理委员会和国家环境保护部四大政府机构共同承担管理职能的局面，同时公安部、商务部、财政部、海关总署、交通和运输部等也涉及汽车行业相关环节的管理。这种监管模式造成针对个别参数企业要进行多次填报和审批。

对于企业而言，为了应对多头管理的认证工作现状，企业需要同时应对不同认证项目，在相关的认证平台上进行数据的填报，由于行业内暂无统一申报平台，而每个项目都有大量的数据需要进行填报。为了应对这样的情况，企业通常需要组织大量的人员进行型式认证工作，且在这个过程中很容易造成企业内部管理成本与人工成本的提高，以及管理的混乱。为了帮助企业改变通过人工手段应对型式认证多头管理的被动局面，在企业建立统一的型式认证平台。

车型认证系统帮组企业解决了多次填报、审批参数的问题，实现 CCC、北环、国环等认证工作的自动填报，全面提高企业型式认证工作能力。但是对于车型认证系统的测试验证工作中，面对大量参数的反复申报，人工测试显然低效、枯燥、乏味的。根据大数据量，反复使用的特点，中国汽车技术研究中心·数据中心引入 Selenium2.0 测试套件。

3.2. Selenium 简介

Selenium 是 2004 年 ThoughtWorks 公司的 Jason Huggins 专门设计开发的自动化测试工具套件，目的是实现多平台 Web 应用程序测试[5]。它经历了两个版本，Selenium 1.0 和 Selenium 2.0。Selenium 不是由单独一个工具构成的，而是由一些插件、类库组成，每个部分都有其特点和应用场景，主要用于 Web 应用程序的自动化测试，并支持所有基于 Web 的管理任务自动化，具有开源、多浏览器支持、多平台支持、多语言支持及简单灵活等特点。Selenium 1.0 由一些插件、类库组成，主要包括 Selenium IDE、Selenium Grid 和 Selenium RC 等，Selenium 1.0 家谱，如下图 1 所示。

Selenium 1.0 版本与 WebDriver 结合发展为当前的 Selenium 2.0 版本，为用户提供了最优秀的测试框架并被广泛应用。本文主要讨论的是 Selenium 2.0 在汽车软件平台的应用。

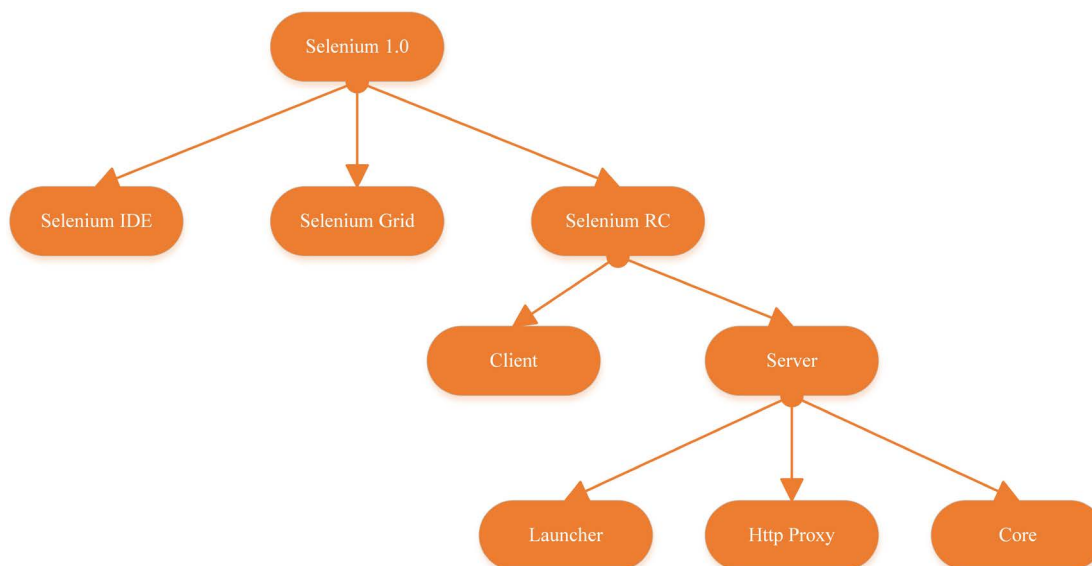


Figure 1. Selenium 1.0 genealogy

图 1. Selenium 1.0 家谱

3.3. Selenium 2.0

Selenium 2.0 支持 WebDriver API 及其底层技术，同时在基于 WebDriver API 基础上，通过 Selenium 1.0 技术对测试代码的移植，提供了极大的便利。此外，为了向后兼容 Selenium 2.0 仍然使用 Selenium 1.0 的 Selenium RC 接口。

WebDriver 是通过原生浏览器支持或者浏览器扩展来直接控制浏览器。WebDriver 针对各个浏览器而开发，取代了嵌入到被测 Web 应用中的 JavaScript。与浏览器的紧密集成，使其支持创建更高级的测试，避免了 JavaScript 安全模型导致的限制。除了来自浏览器厂商的支持之外，WebDriver 还利用操作系统级的调用，模拟用户输入。

Selenium 与 WebDriver 早先属于两个不同的项目，WebDriver 的创建者 Simon Stewart 早在 2009 年 8 月的一份邮件中解释了项目合并的原因。主要原因是 WebDriver 解决了 Selenium 存在的缺点，例如能够绕过 JavaScript 沙箱；还有部分原因是 Selenium 解决了 WebDriver 存在的一些问题，例如对多浏览器的兼容等。

3.4. TestNG

TestNG 是一个测试框架，主要是根据 JUnit 和 NUnit 的思想而构建的，利用了注释来强化测试功能，它既可以用来做单元测试，也可以用来做集成测试[6]。TestNG 与 Selenium 结合使用，可以实现其他测试框架无法实现的功能，如使用依赖项进行测试，重新运行失败了测试，以及使用单独文件中定义参数进行参数化测试。上述特性使其在众多 Web 应用程序测试框架中脱颖而出。

在测试自动化中，测试代码不仅包含测试逻辑，还包含许多其他代码，如 URL 拼接、Html/xml 解析、访问 UI 控件等。若把测试逻辑与这些无关的代码混在一起，测试逻辑将很难理解，也不容易维护。TestNG 采用分层结构很好地解决了这一问题，分层的测试框架如图 2 所示。

3.5. 框架设计

在机动车型式认证管理系统自动化测试平台中，测试框架如图 3 所示，包括构建层、执行层和展示

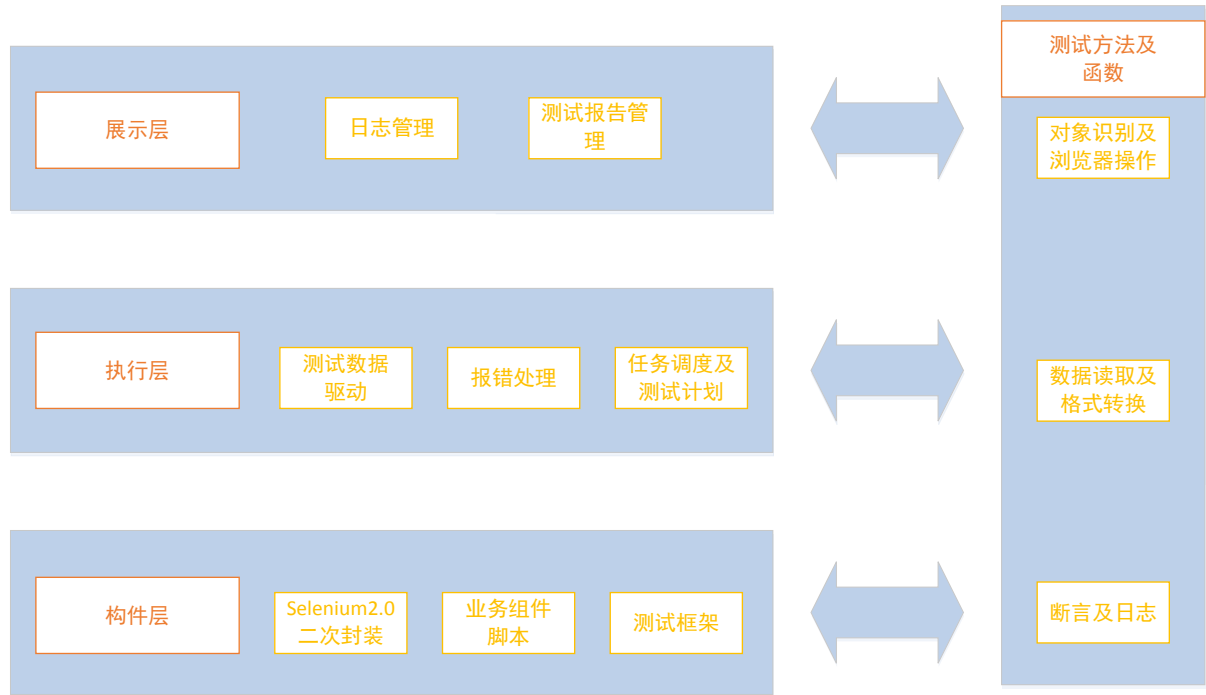


Figure 2. Test framework diagram
图 2. 测试框架图

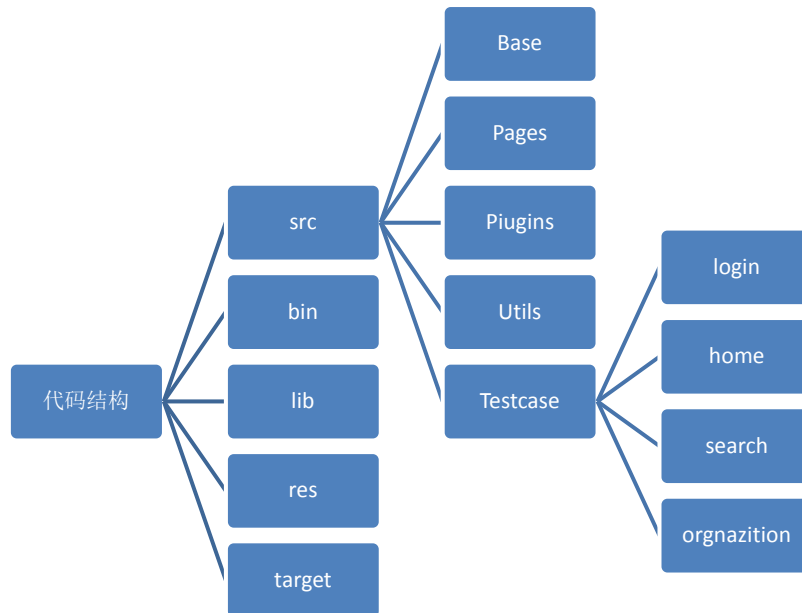


Figure 3. Code frame implementation diagram
图 3. 代码框架实现图

层三部分。各个框架及工具所实现的功能分别为：① 构建层是对开源 Selenium 2.0 封装开发，实现对于 WebBrowser 的控制功能；② 控制层集成 TestNG，根据测试用例执行数据驱动测试，实现测试执行逻辑的控制，通过扩展实现测试失败跳出执行、截图等策略；③ 展示层通过开源的 Selenium 插件展示测试结果，对其进行修改定制，以满足结果展示需求；④ 测试用例采用 Excel 文件维护。

3.6. 代码框架实现

在代码框架实现中，首先创建基类。目的是为了使框架中所有的基本操作都从基类中去调用，实现统一性，同时，由于测试平台多样，对应的浏览器又不同，通过创建的基类能够解决测试平台和测试浏览器选择的问题。其次，对页面元素的操作单独抽象出来，通过页面对象模型(Page Object)实现操作的统一管理。不仅使测试脚本有更高的可维护性，减少了重复的代码，而且帮助缓存远程，避免出现元素过期的问题。此外，对工具包进行了封装，封装了 Selenium 的 API、读取数据类等。图 3 为代码框架实现图的具体内容。

在测试框架代码结构中，根据不同功能设置了 base、pages、plugins、testcases 及 utils 等多个子目录(子包)，各子包具体功能介绍如下。

- ① base: 存放基类，做预处理和释放资源的职能。
- ② pages: 存放页面元素类，根据 PO 模型统一管理对应页面的所有元素。
- ③ plugins: 主要存放的是第三方插件。
- ④ testcases: 用于存放测试用例，如登录模块，首页模块等。
- ⑤ utils: 封装了各种工具类，包括读取 excel、Selenium api 封装类，读取数据库类，读取属性文件类和选择浏览器类等。

此外，项目编译生成的 class 文件存放在 bin 目录中。框架类所需的所有属性文件存储在 config 目录中。各模块需要用到的测试数据以 Excel 文件形式存放在 data 目录下，每个模块对应一个 Excel 文件，每个 sheet 对应一个测试用例。框架的说明文档、编码规范等一些项目文档存放于 doc 目录。chromedriver 和 iedriver 等驱动程序存放在 res 目录中。测试框架运行测试用例生成的报告(包含 log, 截图等)存储在 result 目录下。

3.7. 测试结果

机动车型式认证管理系统对法规认证涉及的车型认证 2600 多项参数，进行收集、审批、变更和上报；根据 48 家车企、16 种车型的要求，对约计 200 万条数据进行多次迭代测试。应用基于 selenium 自动化的测试解决方案后，有效地提高了测试人员对产品质量的测试效率，降低了功能测试的时间成本与测试报告分析成本，表 1 是实验数据抽样统计结果。

在自动化测试实施前，对手工测试和自动化测试的时间成本进行推演，发现手工测试在用例设计阶段的成本要小于自动化测试的成本，但在单次用例执行阶段，自动化测试要明显少于手工测试时间。手工测试与自动化测试的时间成本推演结果见表 2。

测试结果的展现通过开发插件实现，该插件基于 TestNG 进行开发，既可以用在基于 TestNG 的自动化测试框架中，也可以收集测试结果并存储在 Excel 中，进而直观的了解项目各个模块用例的执行条数、结果、通过率及耗时。通过各个模块的详细日志和截图可以精准快速的定位问题所在。

汇总页效果如图 4 所示，其中，模块中的蓝色字体可以点击，点击之后可以跳转到对应的模块中，汇总页面中的所有输出都是程序自动计算的结果，无需人工进行任何操作。

通过测试报告的结果展示，可以发现所有的测试用例均已被覆盖到，但是失败情况还需要人工进行校验判断，这是以后要持续研究改进的地方。

4. 结束语

本文通过对基于 Selenium 的自动化测试技术进行研究和应用，采用自动测试工具 Selenium 2.0 进行回归测试，设计和实现了持续集成自动化测试方案。自动化测试的目的是节省成本，除了用于回归之外，

Table 1. Experimental data sampling statistics**表 1.** 实验数据抽样统计结果

	Selenium	
	使用前	使用后(单测试机)
测试时间成本 (case 执行时间)	3000 条/约 64 hours	3000 条/约 15 hours
测试报告分析成本	8 hours	2 hours

Table 2. The results of the automated test time cost**表 2.** 自动化测试时间成本推演结果

测试方式	用例设计	单次执行时间	备注
手工测试	127 人天	76 人天	自动化测试可横向扩展机器减少执行时间,同时可增加执行轮次,来提高软件质量保证。
自动化测试	389 人天(一次性投入)	4.5 机器天	

自动化测试结果					
运行日期	2016/4/22	项目名称	型式认证参数管理系统测试项目		
开始时间	9:11:00	结束时间	12:55:35	耗时	3:44:35
用例总数	1636	通过用例总数	1293	失败用例总数	343
测试功能列表					
序号	模块名称	用例总数	成功条数	失败条数	通过率
1	数据应用模块	230	189	41	82.17%
2	系统管理模块	234	198	36	84.62%
3	综合查询模块	349	278	71	79.66%
4	业务基础数据管理模块	186	162	24	87.10%
5	流程管理模块	513	379	134	73.88%
6	公告申报模块	124	87	37	70.16%

Figure 4. Test result display diagram**图 4.** 测试结果展示图

还可以用于兼容性测试、稳定性测试、单元测试,并有效提醒研发人员在开发或修改 Bug 阶段构造前置业务数据,实现真正意义的“持续集成”[7]。研究表明,采用 Selenium 2.0 + Testng 搭建持续自动化测试框架,能自动化完成测试工作,无需人工干预,既节省了重复手工测试成本,又保证了测试案例的一致性,同时测试目标达到被验证的目的,提高了开发和测试效率,有效保障了软件测试的质量。

软件开发过程的持续集成与自动化测试的发展是必然的,不同的技术和开发环境对测试自动化有不同的要求,还有很多值得研究的地方,下一步将持续改进测试框架,深入研究测试方法。一套成熟的自动化测试框架是需要的项目实践中持续优化的,只有不断实践,才能发现问题、解决问题、积累经验、逐步完善[8]。

参考文献 (References)

- [1] 杨晔. 基于 BDD 的自动化测试框架研究[J]. 电脑与电信, 2016(6): 30-33.
- [2] 刘纯录, 主编. 软件评测师教程[M]. 北京: 清华大学出版社, 2005: 59.
- [3] 梁向阳, 高丽萍. 一种基于 Selenium 的 Web 应用软件自动化测试平台设计与实现[J]. 科研信息化技术与应用, 2014, 5(6): 44-52.

- [4] 曹磊, 董科军, 袁博文. 基于 Selenium 的 Web 测试研究及应用[J]. 电脑编程技巧与维护, 2015(21): 5-9.
- [5] 何焯. 通过 Selenium 实现 Web 自动化测试的研究[J]. 电子技术与软件工程, 2016(7): 65.
- [6] 张慧琳, 李威, 佟秋利, 等. 基于 Selenium 和 TestNG 的集成自动化测试平台设计[J]. 实验技术与管理, 2015, 32(9): 153-159.
- [7] 冯楠, 陈有为, 张静姝, 等. Web 自动化测试框架的设计, 实现和应用[J]. 自动化与仪器仪表, 2016(6): 226-228.
- [8] 樊付星, 黄大庆, 周末, 等. 基于 Web 的自动化测试框架的研究与实现[J]. 电子设计工程, 2012, 20(1): 126-128.

期刊投稿者将享受如下服务:

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: sea@hanspub.org