

文章

开源自动测试框架Tellurium

作者 张凯峰 发布于 2009年6月11日 下午8时51分

社区 .NET, Agile 主题 工件和工具, 敏捷技术, 单元测试 标签 测试, 测试驱动开发, 采访



Tellurium自动测试框架是一款针对web应用、基于UI模块的自动测试平台。UI模块是由一组复合的UI对象以嵌套的形式组成，比如，Google的搜索UI模块可以表示成：

```
ui.Container(uid: "GoogleSearchModule", clocator: [tag: "td", group: "true"]){
  InputBox(uid: "Input", clocator: [title: "Google Search"])
  SubmitButton(uid: "Search", clocator: [name: "btnG", value: "Google Search"])
  SubmitButton(uid: "ImFeelingLucky", clocator: [value: "I'm Feeling Lucky"])
}
```

Tellurium框架还定义了一套全新的领域特定语言来进行web测试，比如对于Google搜索模块，你可以使用下面的DSL来完成一次搜索测试：

```
type "GoogleSearchModule.Input", "Tellurium test"
click "GoogleSearchModule.Search"
waitForPageToLoad 30000
```

目前Tellurium已经发布0.6.0版本，InfoQ中文站就Tellurium的方方面面，特地邮件采访了Tellurium的创始人方剑先生：

1、请介绍一下您自己，以及所从事的工作？

我的名字是方剑，曾经在上海读书和工作多年。2000年在美国佐治亚理工（Georgia Institute of Technology）求学。毕业后在一家美国公司做软件开发工作，从事企业级应用（Enterprise Applications）开发，主要负责服务器端框架的设计和开发，商业应用服务（Business Services）的开发，和一些软件规范的制定。此外，我还有很强的人工智能和计算机网络方面的研究背景。

2、您开始做Tellurium这样一个自动化测试框架，是基于怎样的考虑呢？我看到有特色的两点是使用UI module-based这样的描述块来定义待测的UI，以及使用DSL来表述测试代码，还有哪些与以往框架不同的设计思想，缘起是什么呢？

在2007年，我们公司开始注意到Selenium测试框架。由于我们用敏捷开发方法（Agile development），在Scrum队伍中，每个人的角色开始变得多样性了。我有个工作（Task）就是用Selenium去测试我们的一个应用程序（是用Dojo Javascript框架写的）。应该说在当时，Selenium是一个开创性的框架，如果用他们的复制和重播模式（record and replay）很容易产生测试脚本。但一旦我开始用Selenium，我发现它还是有一些缺点和不便之处。主要测试脚本是对代码的更新很脆弱（Fragile）。但是在一个敏捷开发的环境下，一般都有阶段性的用户接受测试（User Acceptance Test），由于程序一直更新，Selenium测试脚本的维护就变成一个很头痛的问题。而且，对于很多企业级应用，复制和重播模式本身就不太适用，比喻我们有很多数据格（Data Grid），它的内容本身就是动态的。其他的问题包括Selenium测试脚本不是结构化的，你可以看到有XPath，这更增加了维护的困难。因此，我们决定开发一个在Selenium之上的框架来解决这些问题。我做的第一个版本是通过Spring框架和物体工厂（Object Factory）来产生UI元素（Element），使得框架能把UI元素的表达和测试代码分开，自动处理Javascript事件，并在一定程度上能处理动态网页内容。

由于这个框架用XML来配置UI元素，使用起来并不是很方便。在2008年，我用Groovy重写了整个框架并变成一个开源项目（open source project）。新的版本主要有三个大的改变。首先是采用UI Module（UI模块）描述块来定义待测的UI。这样做的好处是系统自动生成运行时的Locator，即使你改变了其中的一些元素，框架本身会生产新的Locator来适应这种改变。而且框架侧重一个集合的元素，而不是单个元素，这样使得我们可以利用元素之间的关系来帮助我们定位他们在DOM中的位置。此外，这样也增加了重用性，比如，我们可以定义一些Tellurium Widget，在你的测试代码中可以直接定义这个Widget，而不需要重新定义一个个的单个元素。第二个大的变化是用DSL来写测试代码。这样增加了表达性使得用户很容易写和维护测试代码。Tellurium测试代码可以用Java，Groovy，或纯DSL脚本来写。Tellurium本身也支持JUnit和TestNG测试框架。另一个大的变化是开始用UI模板（UI templates）来表述动态网页内容，例如数据格（Data Grid）。这样使得Ajax应用程序的测试变得可行和容易。

3、Tellurium主要有那些子项目构成？

Tellurium主要是由Tellurium Core, Tellurium Engine, Tellurium Widget Extensions, 和Tellurium UI Module Plugin（TrUMP）子项目构成。Tellurium Core主要是处理DSL和动态生成Locator。Tellurium Engine是测试驱动模块，目前还是利用Selenium Core。Tellurium Widget Extensions包括一些DOJO和ExtJS Javascript框架的可重用Widget模块。这些Widget被编译成一个jar文件方便用户调用。TrUMP是一个Firefox plugin来自动生成UI Module。

另外，Tellurium还提供两个参考子项目（Reference Projects），分别为JUnit和TestNG项目，来给用户示范如何创建Tellurium测试项目和如何使用Tellurium的各种功能。

除此之外，Tellurium还提供了Tellurium Maven Archetypes，使得用户可以用一个Maven命令就可以创建自己的Tellurium测试项目。

4、介绍一下Tellurium的代码贡献者们吧？

我主要是负责Tellurium的整体设计和很大一部分的代码编程。除我之外，现在还有四个来自美国和英国的队友（team members）。Vivek Mongolu主要负责TrUMP的UI设计和实现。Matt Senter主要负责Maven支持，包括代码的编译，发布，和Maven Repository的维护。Haroon Rasheed参加了Tellurium参考项目（Tellurium Reference Projects）的开发，Selenium Grid的支持和其他的维护工作。Mikhail Koryak参与了TrUMP的开发工作，他是jQuery方面的专家，负责Tellurium的jQuery支持。现在参与Tellurium Engine的开发。

5、你了解在自动化测试工具这个领域，有着哪些和Tellurium类似的竞争对手吗？比如ThoughtWorks的Twist？与它们相比，Tellurium的优势在什么地方？有什么劣势吗？

Tellurium脱胎于Selenium，它的主要竞争对手还是Selenium，毕竟Selenium已经推广好几年了。要用户用一个新的框架是要花一定的时间的。此外Canoo WebTest也是一个比较流行的网页测试框架（Framework）。但是Tellurium还是有它本身的优势的，如UI Module的概念，鲁棒性好，可重用性好，表达性好（Expressiveness）。用Tellurium写的测试代码的结构性好，比较容易维护。

毕竟Tellurium还是一个新的框架，到现在只有一年多的开发时间。有些特色还有待成熟。此外，Tellurium要用到Groovy动态语言，尽管用户并不一定需要了解Groovy才可用Tellurium，但由于Groovy相对比较新，所以不少人还是有疑虑的。

ThoughtWorks的Twist基本上是Selenium + GSpec，就是在Selenium之上增加了行为测试（Behavior Driven Test）的DSL。其实我很早就考虑到对行为测试的支持，但精力有限，Tellurium目前还没有这方面的实现。将来会增加的，可以和EasyB框架结合起来支持行为测试，或直接实现对行为测试的支持。

6、现在Tellurium的应用情况是怎样的？来自使用者的反响如何？

现在已经有不少Tellurium用户，主要来自美国，印度和欧洲。由于Tellurium本身的特色，如易用，可维护性好，新的功能如对jQuery Selector的支持，使用者的反应还不错。甚至有的用户在自己的公司里给同事作Tellurium的培训。当然，Tellurium还很年青，而且前一段时间由于全球性经济危机的影响，我们对Tellurium的推广还做得很不够。以后会加大对Tellurium的推广。

7、Tellurium在社区采用了哪些方式和开发者们进行交互呢？

我们有自己的用户组Tellurium user group，用户提出问题往往能很快地得到解答。我们也有LinkedIn用户组让用户之间能更好地交流。此外我们还鼓励用户参加Tellurium的设计讨论和推广。我们会每年从用户中推选出一个最活跃用户（most active user）和一个最有价值用户（most valuable user）。Tellurium将来会设推广队（Evangelism Team），如果某人对Tellurium做了很多推广工作，他/她也可以成为Tellurium正式成员（team member）。

8、Tellurium未来的发展有怎样的规划吗？有计划推出中文的社区及文档吗？

尽管Tellurium和Selenium在概念上有很大的不同，一直到Tellurium 0.6.0，我们还是依赖Selenium Core作为底层的测试驱动Engine。Tellurium 0.7.0将成为Tellurium发展史上的一个重要里程碑，我们将开发自己的测试驱动Engine使得Tellurium能更好，更有效地支持UI Module，同时可以进行UI Module的缓存（Caching）以增加其可用性和提高测试速度。并可对UI Module进行部分匹配以增加其鲁棒性。在新的Tellurium Engine的支持下，Tellurium widget将变得更容易，更实用，和更有效。其他的发展规划包括TrUMP的改进，行为测试（Behavior Driven Test）的支持，功能测试（functional test）的增强支持，和IDE的支持。

毕竟我来自中国，当然希望得到更多的来自自己国家的用户的参与和支持。现在我们已经开通中文社区和文档项目，会有相关的中文文档逐渐添加进来，我们也热烈欢迎更多国内的开发者能加入进来，帮助支持和推广Tellurium，谢谢。

9、能给国内的用户一个快速的开始吗？怎么立刻感受到Tellurium带来的好处？

我们提供了一份[中文版的Tellurium QuickStart](#)，还有一个短小的演示：[十分钟感受Tellurium](#)（10 minutes to Tellurium）。它包括利用Tellurium Maven archetype去建立一个新的Tellurium测试项目，再用 Tellurium Firefox plugin TrUMP去自动生成一个UI Module，然后再写自己的Tellurium测试代码。

如果用户不熟悉Maven，他可以下载[Tellurium参考项目](#)（Tellurium Reference Project）。这个项目是我们用来测试Tellurium项目网页的。包括了各种例子，可以直接运行。

更多有关Tellurium的信息，请参考DZone上的[相关资料](#)和Tellurium的[Wiki](#)。敬请期待Tellurium在InfoQ中文站的更多技术文章。

给InfoQ中文站投稿或者参与内容翻译工作，请邮件至editors@cn.infoq.com。也欢迎大家加入到[InfoQ中文站用户讨论组](#)中与我们的编辑和其他读者朋友交流。

加入书签 鲜果+, digg+, reddit+, del.icio.us+, dzone+ slashdot+

1 条回复

关注此讨论 回复

不错 发表人 zzeric lee 发表于 2009年6月12日 上午12时5分

不错
2009年6月12日 上午12时5分 发表人 zzeric lee

支持国货，等0.7后一定要试一下

回复