

# 软件全球化测试技术的研究与实现<sup>①</sup>

刘雅君 徐进 (北京理工大学 计算机科学技术学院 北京 100081)

**摘要:** 针对软件测试工具中全球化功能测试的问题,对软件产品全球化功能的基本概念及由此带来的测试技术难点给出了论述,讨论了 selenium-java 工具在全球化功能测试上的局限性。在此基础上,提出适应全球化功能测试的软件测试工具改进方案,并给出了改进的详细实现方法。

**关键词:** 软件测试 软件全球化 软件测试工具

## Study and Implementation of Globalization Software Test Technology

LIU Ya-Jun, XU Jin

(School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

**Abstract:** The paper focuses on the issues about the functional globalization testing using software automation testing tools. It introduces the concepts of globalization testing and the difficulties of test performing induced by it. Then, it concludes on the limitations of selenium-java, a popularly used automation testing tool on globalization testing. To address them, it gives an solution.

**Keywords:** software testing; Globalization of software; software automation testing tools

## 1 引言

为了使产品能够支持全球客户,软件开发者开始开发出支持全球化的应用。在支持全球化的软件开发技术引起广泛关注的同时,如何高效地进行软件全球化测试是具有挑战性且极具现实意义的软件工程问题。

### 1.1 软件全球化(Globalization)

软件全球化(Globalization)<sup>[1]</sup>,通常缩写为 G11N,是指包括软件进行正确的国际化设计,软件本地化集成,以及在全球市场进行的市场推广、销售和支持的全部过程。软件设计的国际化和软件产品的本地化是软件全球化的主要内容:

#### 1.1.1 软件的国际化(Internationalization)

软件的国际化(Internationalization),通常缩写为 I18N,是指不对软件系统使用者做任何假设的开发方法,这包括了使用者的语言、文字编码方式、习惯数据表达方式、书写方向、宗教信仰等。经国际化方法开发的软件可以按特定的地域环境要求改变自身的

#### 1.1.2 软件的本地化(Localization)

软件的本地化(Localization),通常缩写为 L10N,是将一个软件产品按特定国家/地区或语言市场的需要进行加工,使之满足特定市场上的用户对语言和文化特殊要求的软件生产活动。

为了满足全球化软件世界范围内发布的需要,全球化软件的开发过程中,软件的国际化设计和本地化工程处理是两个重要的步骤。与开发过程相对应,作为软件质量保证过程的全球性软件测试过程,包含一系列相互关联的测试技术和流程。在广泛采取了全球化开发模式的基础之上,如何提高这些测试的效率更是值得进一步研究。

### 1.2 全球化测试

全球化测试<sup>[2]</sup>通过测试软件的全球通用性,以确保测试软件在任何支持的区域性或区域设置中都能正常运行,可以分为功能性能测试、国际化测试、本地化能力测试和本地化测试等阶段。

<sup>①</sup> 收稿时间:2009-03-04

### 1.2.1 功能性能测试

功能性能测试是全球化软件测试的第一个测试过程,其对象是基本语言版本软件进行的。测试的目的是保证软件的功能和性能符合软件设计说明书和用户的使用需求。只有通过这项测试,才可以进行其他种类的全球化测试。

### 1.2.2 国际化测试

国际化测试的目的是测试软件的国际化支持能力,发现软件的国际化的潜在问题,保证软件在世界不同区域中都能正常运行。国际化测试使用每种可能的国际输入类型,重点测试软件接收、处理和发送不同字符集的能力,针对任何区域性或区域设置检查产品的功能是否正常。

### 1.2.3 本地化能力测试

本地化能力是指不需要重新设计或任何代码修改的过程,而将程序的用户界面翻译成各种目标语言的能力。这种能力越高的软件越容易地实施本地化处理。本地化能力测试将验证软件的本地化能力,发现软件在本地化过程中可能出现的各种问题。这也意味着只有通过这项测试,才能进行软件的本地化测试。

### 1.2.4 本地化测试

本地化测试的目的是测试特定目标区域设置的软件本地化质量,测试内容包括软件翻译的语言质量、软件界面的布局和针对目标区域的使用习惯保持一致的能力等等。其中最后所指的能力包括很多,例如时间、日期、货币的表达方式、阅读顺序(从左到右或从右到左)等等。

## 2 软件测试工具全球化支持的难点

根据现代软件工程的软件测试方法,软件测试贯穿于软件开发的各个阶段,需要反复进行。全球化软件的测试也不例外,它也必须能支持软件开发的全过程。长时间手工执行测试用例容易使测试者感到疲劳和厌倦,降低测试效率。而自动化测试将大大降低系统测试、维护升级等阶段的成本。通过分析软件全球化测试的特征及需求,总结自动化测试工具支持全球化测试所需要实现的技术难点有以下几点:

### 2.1 脚本的复用

针对全球化测试的自动化测试的一般做法为:在基本语言版本软件的测试脚本的基础上进行扩展,为

每一个语言版本建立自动化测试脚本。这样的重复开发、调试和维护脚本做法仍然效率很低。因此,为保证全球化测试质量,提高测试效率和加强测试过程的管理,在针对全球化测试技术和策略中实现自动化测试脚本的复用就变得尤为重要。

### 2.2 测试环境的语言及字符集设置的选择

全球化测试的重要任务是验证在功能上对该特定区域设置的支持。在测试过程中,为了测试软件对显示、接收、处理和发送多种语言及其不同字符集的能力,需要正确设置测试环境,这里可能涉及多处的区域设置:服务器端的区域设置、数据库端的区域设置、客户端的区域设置和浏览器端的区域设置等等。如果缺乏这种选择或者错误地进行了选择,这可能在具体测试过程中带来系统由于码制的切换错误而出现乱码的问题。例如:在 US7ASCII 环境设置下,ZHS16GBK 的数据显示为乱码,这并非测试软件的错误。

在实际的全球化软件测试过程中,针对世界软件市场的语言优先级和软件测试的基本策略,一般要选择最具市场代表区域的文字作为要测试的目标版本软件的语言,例如选择德文和日文版本的软件进行测试。由于日语作为东亚双字节字符的典型语言,德语作为西欧高 ASCII 码的典型语言。其中:双字节语言还包括中文、韩文、日文等东亚语言,而高 ASCII 码语言是指德语、法语或意大利语等包含特殊字符的西欧语言。当然,全球化测试应尽可能涵盖软件支持的所有语言。因为在更全面的条件下,全球化测试可以产生更全面的结果。

在这样的背景下,根据测试需求所指定的一系列语言和字符集,在测试过程中依次选择语言和字符集,正确初始化对应测试用例的测试环境。

### 2.3 多种语言显示的测试

全球化软件必须能够动态检测出用户的区域设置,然后根据设置显示出正确的信息格式和数据。在国际化开发模式,取决于特定区域设置的元素是不能写死在代码中,而是需要动态选取的。这类元素包括页面上的字符、图片内的文字、以及提示或出错信息等等。在逻辑代码内,这类元素不会使用具体的语言文字,通常是定义成常量。在对应的翻译文件中,根据基本语言版本,将其翻译为对应文字。为了降低本地化能力测试的成本,提高测试效率,本地化能力测

试通常不必真正本地化产品,而是在软件的伪本地化版本上进行。软件的伪本地化是指将软件中需要本地化的基本语言文本,使用其它本地化的文本替换,模拟本地化版本的过程。但是等到进行本地化测试时,需要将待测试的软件版本翻译为特定目标语言。

对这些翻译质量的检查,是本地化测试的重要方面。软件的翻译质量包括翻译的准确性、完整性、一致性,以及特定区域市场的文化、传统、习俗等等内容。如果要对翻译的检查,需要根据所测语言而准备相应的翻译文件进行对比。当测试多种语言时,这样会增加测试人员的负担。

#### 2.4 多语言和字符集的测试数据的自动生成

在开发过程中,软件完全可以使用 Unicode 进行文本编码。但是程序与每种可能的国际输入类型交互时,需要调用适当的转换处理。通过测试数据的输入,可能检测到软件在这些处理方面存在的潜在问题:软件处理后全部或部分显示为问号(如????)而不是文本,这表示软件在字符集转换处理中出现了问题;软件处理后显示随机的字符(如"\*、#、%等)而不是正常文本,这表示字符集选择存在问题,可能设置了错误的区域或显示字符集;软件处理后出现象形字(如口、|等),表示所选字体无法显示某些字符。

为了测试软件对接收、处理和发送不同字符集的能力,需要实现针对不同语言和字符集数据的自动产生。测试的数据一般包含东亚语言字符、高 ASCII 码语言字符、复杂脚本字符和英语字符(可选);其中复杂脚本字符指阿拉伯语、希伯来语、泰语。

总之,伴随对软件测试的日益重视以及自动化测试的逐步引进,各种自动化测试工具相继推出。但是各种自动化测试工具的可应用性程度和支持侧重有所不同,并且广泛缺乏对全球化测试的支持。

### 3 对软件测试工具全球化支持的扩展及实现

针对以上论述的全球化软件测试的特征和技术实现难点,结合具体测试项目对采用的 Selenium RC-java 工具进行必要的分析和全球化支持的扩展。

Selenium是由 ThoughtWorks 团队开发的 web 应用程序自动化测试的工具,适合进行功能测试、验收测试。该工具主要分为三个部分: Selenium Core、

Selenium IDE、Selenium Remote Control(简称为 Selenium RC)。Selenium RC 工具具有以下几点优势:是完全开源的测试工具;支持一些主流编程语言编写标本,包括 Java, .NET, Perl, Python 和 Ruby 等;允许在任何支持 JavaScript 的浏览器中进行测试,如 Internet Explorer、Firefox 等;支持多操作系统(Windows, Mac OS 和 Linux)<sup>[3]</sup>。Selenium RC 基本测试流程如图 1 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。

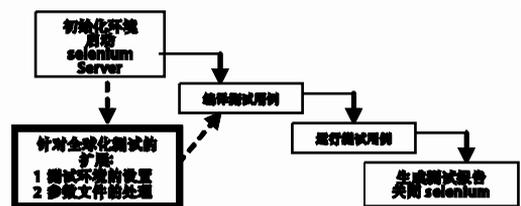


图 1 基本测试运行流程和针对全球化测试扩展的添加

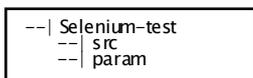
但是如第二部分所述, Selenium RC 和大多数的自动化测试工具一样,也在很多方面缺乏对全球化测试支持:在没有指定浏览器参数文件的情况下, Selenium Server 会启动一个系统默认设置的浏览器执行测试,默认设置语言为当前浏览器版本语言;操作函数直接使用大量页面元素作为参数,对这类数据取值的存在依赖,使得 Selenium 测试变得脆弱且需要针对测试脚本中同一页面元素文本属性值进行重复修改;不能提供多语言和字符集的测试数据等等。

针对 selenium 工具的上述不足,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。具体做了如下方案的设计和实现。

#### 3.1 将特定语言的测试元素独立于测试脚本代码,支持多语言测试的脚本复用

为了达到在多国语言测试时复用整套脚本的目的,将所有取决于特定语言文化的测试元素独立于测试脚本代码,存储在参数文件内。测试脚本文件结构

设计如下所示:



其中, `src` 目录中存放测试用例的源代码文件, `param` 目录下存放参数文件。参数文件均为 `xml` 格式, 根据参数的用途和与语言的关系分为下列三类。

### 3.1.1 存放固定参数的参数文件

在这类文件中, 存放针对任何语言版本测试过程中均不需改变的参数数据。这些固定参数可以是测试用例运行过程中输入的字符串, 也可以是界面元素的属性值, 例如测试机 IP 地址、界面上按钮的 `id` 值、界面上表格的 `xpath` 值等等。

### 3.1.2 根据测试语言和字符集, 存放输入的随机数据的参数文件

根据测试语言和字符集要求, 分析涉及直接或间接处理字符串输入/输出的测试案例, 抽取其中可以被替换为任意国际输入类型的随机数据。在执行测试时, 通过替换这些输入数据, 以测试软件对处理不同语言和字符集的能力。例如, 在执行新建用户这个测试用例时, 需要输入一个新建用户的用户名。在设计中, 这个用户名可以为任意国际输入类型的数据。为了测试软件对语言为英文, 字符编码为 `ISO-8859-1` 的输入数据 `UserName` 的处理能力, 将该输入数据作为参数存储在参数文件进行测试。

### 3.1.3 根据测试语言, 存放界面元素的文本属性值的参数文件

在 `Selenium` 工具包中, 操作函数直接使用界面元素的属性值作为参数, 而这些属性值的类型包括 `name`、`id`、`xpath`、`txt` 等等。当测试的目标语言发生改变时, 界面元素的文本属性值也根据目标语言对应的翻译文本而发生改变。如果这类参数与测试语言不相匹配, 测试会因为界面文本数据不合法而失败。因此, 将这类参数存储在单独的参数文件中, 以方便对其灵活处理。例如, 在测试英文版本软件时, 需要点击一个命名为 `Apply` 的按钮, 将其名字作为参数存储在参数文件中。

## 3.2 动态选择语言和字符集, 支持多种语言测试过程

以某具体 `web` 产品的全球化测试项目为例, 测试项目全过程是通过使用 `ant` 这个构建部署工具作为流程脚本引擎, 自动调用程序而完成: 首先, 初始化正确

的测试环境。然后, 通过网络下载构建成功的被测软件, 修改相应的配置文件, 部署到相应的测试环境中去。在被测软件部署成功后, 编译运行用 `java` 书写的脚本。最后, 运行完成之后, 通过用 `Ant` 自带的方法来生成测试报告, 并通过邮件发送报告给相关人员。整个过程可以全部自动化, 测试人员只需输入简单的配置命令。当然在每日构建的基础上, 还可以不用输入而通过定时任务的设定实现每天自动执行回归测试。

为了满足这种全球化回归测试的要求, 首要条件就是正确初始化测试环境——需要添加对测试环境语言和字符集的设置。这种语言及字符集设置的选择, 不仅包括对当前所测软件语言版本的选择, 还需要包括当前操作系统等等环境语言与字符集设置的选择。在项目中, 主要任务是实现对客户端操作系统 (`Redhat4.0`) 和浏览器 (`Firefox3.0`) 的语言与字符集设置的选择。其中浏览器区域设置就代表了用户当前使用的区域设置。

在没有指定浏览器参数文件的情况下, `Selenium Server` 只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。特定浏览器语言和参看字符编码的设定主要通过以下方法实现: 首先, 根据测试人员的选择, 将需要测试的语言和字符集存储在配置文件中。然后, 根据测试要求的浏览器语言和字符集设置, 在 `Firefox` 的参数文件 `prefs.js` 中添加为用户语言首选项。最后, 在启动 `Selenium Server` 时, 添加“`firefoxProfileTemplate`”选项并指向已修改的 `Firefox` 的参数文件目录。例如选择中文和 `GB2312` 的字符编码, 将其作为用户选项(如下列语句)加入到指定的 `Firefox` 的参数文件 `prefs.js` 中。

```

user_pref("intl.accept_languages", "zh-cn");
user_pref("intl.charsetmenu.browser.cache", "
GB2312 ");

```

客户端操作系统(`Red hat4.0`)的语言和字符集, 可以通过修改文件 `/etc/sysconfig/i18n` 或者通过设置 `LC_ALL` 和 `LANG` 两个参数实现:

```

export LC_ALL=zh_CN.gbk
export LANG=zh_CN.gbk

```

同时, 上述方法也可以扩展到多个语言和字符集

环境的设置的情况：首先，将所有需要测试的语言的字符集存储在配置文件中。然后，串行或者并行地在测试机上进行测试。其中，串行执行是指依次正确初始化测试环境，循环执行测试流程；而并行执行是通过 socket 通信将信息传递到多台目标测试机，并行初始化测试环境，执行全部测试流程。

### 3.3 通过对 XLIFF 格式的翻译文件的查找，实现测试页面上显示的翻译

本项目中的翻译文件是采取 XLIFF 文件格式。XLIFF 是参与翻译项目的各方之间，用来交换本地化数据的标准格式。它最初是由 Oracle、Sun 等公司开发的。现已成为本地化的行业标准，目前由 OASIS 组织负责开发。XLIFF 是一种 XML 应用，每个文本片段保存的一个翻译单元中(<trans-unit>)[4]。如示例一所示，提纲中使用的标记可以作为翻译单元的属性，源语言对应的翻译字符串作为 source 属性，而目标语言对应的翻译字符串作为 target 属性。因此，target 标记和 id 标记的映射关系，可以将 XLIFF 文件格式的翻译文件作为查找对照表。

```
<xliff version="1.1" xmlns="urn:oasis:names:tc:xliff:document:1.1">
  <file original="demo.txt" source-language="en"
  target-language="zh_CN" datatype="plaintext">
    <body>
      <trans-unit id="User Tabel">
        <source>User Tabel</source>
        <target>用户表</target>
      </trans-unit>
    </body>
  </file>
</xliff>
```

示例 1 XLIFF 文件

通过查找相应测试语言的 XLIFF 文件，实现对前述的第三类参数文件的正确替换，以支持对不同语言版本显示的翻译的测试。如示例二和三所示，通过查找英文和中文对应的 XLIFF 文件，将参数文件 Test\_gui.params 中的页面元素参数(可为任意语言，以英文为例)替换为测试所指定语言(简体中文)对应的页面元素参数。

```
<?xml version="1.0" encoding="UTF-8">
<parameters>
<param name="colUserTblName">User Tabel</param>
.....
</parameters>
```

示例 2 替换前的 Test\_gui.params

```
<?xml version="1.0" encoding="UTF-8">
<parameters>
<param name="colUserTblName">用户表</param>
.....
</parameters>
```

示例 3 替换后的 Test\_gui.params

### 3.4 利用正则表达式，处理包含特殊字符的翻译单元

在上述自动化的参数文件替换过程中，需要对大量的 XML 格式的翻译文本数据进行查找。但是，有一些特殊字符在写入到 XML 文件时，会被实体引用所替换。所以，目标界面上显示的字符串，可能出现和写入在 XLIFF 翻译文件中存储的翻译字符串不一致的现象。

针对上述这类翻译字符串与显示字符串不一致的情况，可以通过正则表达式而解决。正则表达式提供了功能强大、灵活而又高效的方法处理文本数据。利用正则表达式的模式匹配表达法，对字符串进行比较，匹配，从而判断测试结果是否正确。例如：软件界面上存在一个按钮，英文版本的按钮显示的字符串为“About(A)”而简体中文版本的按钮显示的字符串为“关于(A)”。然而，写入在 XLIFF 文件时，英文对应的翻译字符串应为“About(&Amp;A)”，简体中文对应的翻译字符串应为“关于(&Amp;A)”。因此，在进行简体中文本地化测试时，首先需要利用正则表达式符号\*替换“&Amp;”，通过“About\*”查找 XLIFF 文件。然后，将查找结果“关于(&Amp; A)”处理为“关于\*”，再进行界面翻译测试。

表 1 在翻译文件中被替换的字符

界面显示的字符	XLIFF 格式的翻译文件中替换为
&	&Amp;
'	&Apos;
▪	&Aquot;
<	&Acl;
>	&Agt;
©	&Acopy;
.....	.....

### 3.5 采用简单替换方式，自动生成指定语言字符集的测试数据

在执行全球化测试时，需要测试软件对处理不同语言和字符集的能力。这种能力，可以通过简单的预处理进行测试。即简单替换前述的第二类参数文件。

具体步骤如下：首先，需要维护一个包含所有待测语言和字符集的字符库。然后，根据参数文件中参数的字节数，从字符库中抽取制定测试语言和字符集的字符进行替换。如下列示例四和五所示。

为了实现自动生成指定语言字符集的测试数据，上述方法是相对简单且易行。但是伴随测试语言和字符集范围的扩展，字符库需要覆盖的语言和字符集也需要相应增加。从而，势必带来巨大的维护工作量。另外，如果需要字符库覆盖较全面的东亚语言字符，也将带来一定的工作量。

```
<?xml version="1.0" encoding="UTF-8"?>
<parameters file="">
  <param
    name="CreateUserFamilyName">UserFamilyName</param>
  .....
</parameters>
```

示例 4 替换前的 Test\_data\_user.params

```
<?xml version="1.0" encoding="UTF-8"?>
<parameters file="">
  <param name="CreateUserFamilyName">示无尚第
子苏斯</param>
  .....
</parameters>
```

示例 5 替换后的 Test\_data\_user.params

## 4 结论

上述 5 项扩展方案的设计和实现，已经应用于作者参加的某公司实际测试项目中。该项目针对 web 技术的大型全球化应用软件的测试，主要使用 Selenium RC-java 工具开发测试脚本。项目要求同时实现在 linux+Firefox3.0 环境下，对软件的九种语言版本进行全球化测试。

在实际的测试工作中，我们运用了测试元素独立、动态选择语言字符集、根据标准化 XLIFF 格式转化三项测试软件结构改造，以及特殊字符翻译处理、测试数据生成两项实际技术，对 Selenium RC-java 工具进行了必要的扩充和修改，实践验证了对于全球化测试的支持，并取得了良好的效果。

伴随全球化软件发展的必然的趋势，支持全球化的软件开发技术越来越引起业界广泛地关注，同时，全球化测试的研究也会得到必要的重视。本文讨论了关于自动化测试工具的全球化支持特性的扩展方案，并讨论了对 selenium-java 工具进行的改造。

然而，全球化测试技术所涉及的问题很多，本文仅从实践的角度，对其进行了初步的探索和研究。就本文所涉及的问题来看，从实际的应用出发也还存在较多的问题，例如：针对翻译单元的替换规则具有大的局限性，需要进行优化，才能支持更复杂的情况；还有，在文中已经论述过的，对于指定语言和字符集的测试数据生成问题，需要研究自动算法问题，以支持更大规模的语言覆盖测试，等等。这些问题，随着对全球化测试技术的进一步深入研究和学习，相信在今后的技术实践中能够得到必要的解决

## 参考文献

- 1 于千里..net 平台的软件全球化解方案. 中国农业大学学报, 2005, 6: 14.
- 2 Lu LM, Wang RS, Li WG. A target detection method in range-Doppler domain from SAR Echo data. Proc. of the 16th Int'l Conf. on Pattern Recognition (ICPR). 2002, 1: 91 - 94.
- 3 崔启亮, 胡一鸣. 全球化软件测试. 北京: 电子工业出版社, 2006.
- 4 Selenium. <http://www.openqa.org/selenium/>
- 5 Rodolfo M. Raya. XML 本地化: 使用 XLIFF 翻译文档. <http://www.ibm.com/developerworks/cn/xml/x-localis2/index.html>
- 6 全球化测试. <http://www.sortol.com/projects/2007-12-25/article23063.shtml>
- 7 尤金鑫. 国际化测试的研究与实现[学士学位论文]. 北京: 北京理工大学, 2008.