

# 基于 AutoIt 和 Flow Chart 的测试用例生成与管理

李龙澍, 刘磊

(安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039)

**摘 要:**随着软件规模的日益扩大,模块间的复杂性也逐渐增多,使得测试用例的生成和管理成为一个重要而难以解决的问题,快速有效地生成测试用例并对其进行有效的管理将对提高实际测试的效率上有较大的改变。文中采用 AutoIt 语言开发一个测试用例管理工具,通过 Flow Chart 的方法快速生成测试用例,并把用 AutoIt 语言编写的自动化测试用例脚本存储于结构化的 XML 文件中,通过这种方法使得更加容易地管理和维护测试用例,在实际的项目测试中有着很好的实践效果,极大地改进了测试的效率和成本。

**关键词:**自动化测试;测试用例;测试脚本;AutoIt;测试用例管理

中图分类号:TP311.5

文献标识码:A

文章编号:1673-629X(2010)07-0056-04

## Generation and Management of Test Case Based on AutoIt and Flow Chart

LI Long-shu, LIU Lei

(Ministry of Education Key Lab. of Intelligence Computing and Signal Processing at  
Anhui University, Hefei 230039, China)

**Abstract:** As the size of the software increased more and more, the complexity of the modules has also increased obviously, so the generation and management of test case become an important problem which is also difficult to resolve. It has got a larger change for improving test efficiency in generation and managing test case fast and effectively. In this paper, use the AutoIt language to develop a case management tool and use the Flow Chart method to generate the test case fast. And also it will be so easy to manage and maintain the test case by store the automated test script written by AutoIt language into structured XML file. So it has a good effect in the actual test project and greatly improved the test efficiency and cost.

**Key words:** automation test; test case; test script; AutoIt; case management

### 1 概述

随着信息技术的飞速发展,使得软件产品应用到社会的各个领域,软件产品的质量成为人们共同关注的焦点。软件测试是保证软件质量最重要和最有效的方法<sup>[1]</sup>。但是由于软件项目越来越庞大,功能越来越复杂,为了保证软件产品的高可靠性和稳定性,很多软件开发机构将软件测试投入最大的研究时间和精力,

同时也给测试工作带来了很多问题。自动化测试虽然改变了手工的重复性测试的低效性,也减少了人为因素的随机性和冗余性,但是随着软件的模块功能越来越复杂,随之产生的测试用例也越来越庞大,因此大量的自动化测试用例的生成和管理成为一个显著而极其重要的制约自动化测试效率的重要因素。

当前,国内外已有的自动化测试方案采用数据库和测试脚本相结合的方式,并采用一种自动化测试框架<sup>[2]</sup>,有效地提高了自动化测试的效率,但是由于软件单个模块的复杂性,采用复杂的算法虽然能够自动产生相应 case,但是却不能够考虑到充分的路径,同时随着软件需求的不断增加,从而产生的测试用例的增多,使得查找和修改 case 的难度越来越大,也越来越不方便。因此,自动化测试效率受到了很大的影响,常常导致查找修改和维护测试用例的时间远远超过实际运行测试

收稿日期:2009-11-07;修回日期:2010-02-10

基金项目:国家自然科学基金项目(60273043);安徽省自然科学基金项目(050420204);安徽省教育厅自然科学基金项目(2006KJ098B);安徽省高校拔尖人才基金项目(05025102);安徽大学研究生创新基金项目(20073055)

作者简介:李龙澍(1956-),男,安徽亳州人,教授,博士生导师,研究方向为不精确信息处理技术和智能软件开发技术。

用例的时间,甚至不如手工的效率<sup>[3]</sup>。

文中通过 AutoIt 语言开发实现了 case 管理系统,采用测试脚本和 XML 相结合的方式,通过把自动化测试脚本存储于 XML 中的方法以及通过窗口化的操作界面使得更加容易地管理和维护 case,尤其在查找 case 和修改存储 case 时非常的方便。以后还可以对软件进行改进,通过导出 XML 节点中的 case 脚本为 au3 文件,从而可以批量运行 case,可以显著地提高效率。

## 2 AutoIt 脚本

AutoIt 是用以编写脚本程序的自动化测试工具,它被设计用来在 Windows GUI 中进行自动操作<sup>[4]</sup>。通过它可以组合使用模拟键击、鼠标移动和窗口/控件操作等来实现自动化任务。

AutoIt 包括解释和运行脚本程序的核心文件以及 AutoIt 窗口信息工具 AU3Info<sup>[5]</sup>。其中 AU3Info 可以获取指定窗口或控件的相关信息,例如 Class, Position, Size, Text, Hwnd, ClassNameNN, ID, Style 等等。

AutoIt 脚本是指用 AutoIt 脚本语言编写的自动化程序,它可以编译成可执行文件后然后运行,从而实现软件的自动化操作。

一般自动化测试用例的 case 可以用很多种语言编写,形势也不受拘束,各有特点。由于 AutoIt 脚本语言通俗易懂,因此本篇的 case 就是采用 AutoIt 脚本编写的,case 管理工具也是采用 AutoIt 脚本语言编写,因此非常的方便。

## 3 Flow Chart 生成测试用例

自动化测试中需要相关的测试用例。为了发现软件中的错误需要在软件的输入域中选择相应的数据(测试用例)来进行测试<sup>[6]</sup>。而软件的大多数模块的输入域是无限的<sup>[7]</sup>,因此通过某种方法选择最适合的测试用例是至关重要的<sup>[8]</sup>。

当前,已有很多生成优秀测试用例的方法,如遗传算法,模拟退火算法,基于 UML 状态图,基于 Z 语言与状态图,迭代松弛法<sup>[9]</sup>等等。这些方法各有所长,但实现较为复杂,因此这里采用 Flow Chart 和“头尾消减”的方法使得生成测试用例更加的简单和快速。

下面通过一个简单的例子来说明。

根据 Windows 中文件的删除功能可以做出以下所示的 work flow(见图 1)。

该 Flow Chart 用微软作图工具 visio 绘制而成,椭

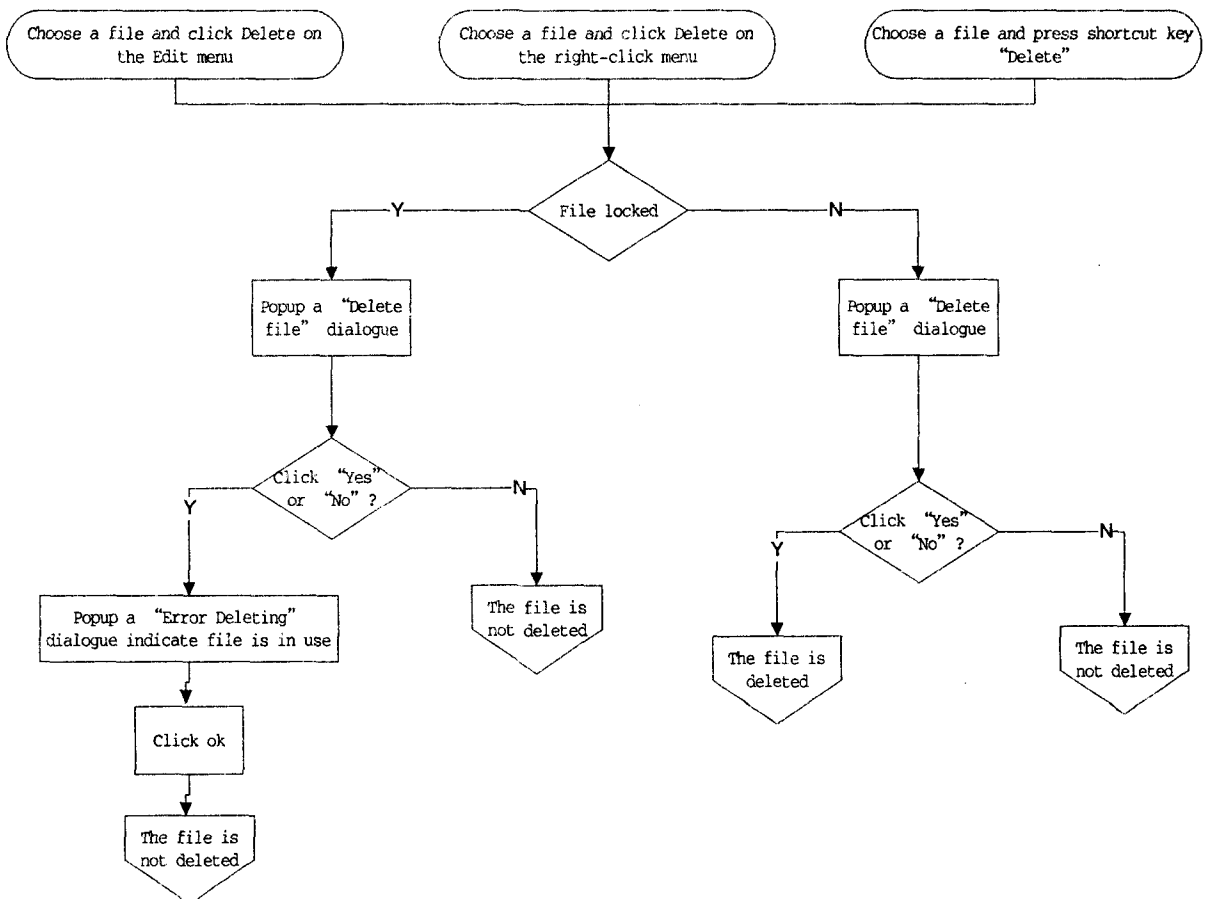


图 1 删除文件的 Flow Chart

圆表示入口,五边形表示出口,菱形表示路径分支,方框表示用户的操作或系统产生的自动行为过程。

首先,所谓“头尾消减”的原则,就是把开始的四个入口单独提取出来并选择一个出口。

其次,选择一个入口和出口覆盖各个路径,最后,对功能设计相应的用例。

用以上的方法并简化,最后得到的 6 条测试用例如图 2 所示。

Title	Steps and Expected Results
Remove a file from File menu	Steps: 1. Select a file and select "Delete" from File menu 2. Delete file confirmation dialog appears, click "Yes" Expected Results: 1. Can remove the file
Remove a file from right-click menu	Steps: 1. Select a file and select "Delete" from right-click menu 2. Delete file confirmation dialog appears, click "Yes" Expected Results: 1. Can remove the file
Remove a file by shortcut key "Delete"	Steps: 1. Select a file and press "Delete" in the keyboard 2. Delete file confirmation dialog appears, click "Yes" Expected Results: 1. Can remove the file
Remove an opened file	Steps: 1. Select a opened file and select "Delete" from right-click menu 2. Delete file confirmation dialog appears, click "Yes" Expected Results: 1. Error deleting dialog appears: You cannot operate a file that is currently in use. Try again when this file is no longer in use. 2. The file cannot be removed.
Cancel to remove a file by clicking "No"	Steps: 1. Select a file and select "Delete" from right-click menu 2. Delete file confirmation dialog appears, click "No". Expected Results: 1. Can cancel to remove a file
Cancel to remove a file by clicking "X"	Steps: 1. Select a file and select "Delete" from right-click menu 2. Delete file confirmation dialog appears, click "X". Expected Results: 1. Can cancel to remove a file

图 2 删除文件的测试用例

由此可见,生成的测试用例能覆盖主要功能,因此具有很好的测试效果。

#### 4 case 脚本的设计

用 AutoIt 脚本语言对上述的删除文件的案例的

case 2 设计自动化脚本如下(其他类似):

```
;Case2 title: Remove a file from right-click menu
```

```
ShellExecute(@DesktopCommonDir);打开桌面目录所在的路径
WinActivate("桌面");激活桌面窗口
```

```
MouseClick("right",645,555,1);单击坐标位置的文件
```

```
;通过发送键盘命令找到删除选项
```

```
Send("{Up}")
```

```
sleep(2000);停止 2 秒
```

```
Send("{Up}")
```

```
sleep(2000)
```

```
Send("{Up}")
```

```
sleep(2000)
```

```
MouseClick("left");点击删除
```

```
sleep(2000)
```

```
Send("{enter}");在弹出的对话框中选择确定从而删除文件
```

#### 5 用 XML 存储并管理 case

由于测试功能的复杂,有的模块的 Flow Chart 变得非常的复杂,呈现多入口和多路径的状态<sup>[10]</sup>。因此,通过 Flow Chart 设计后将会产生大量的 case。在这些大量的 case 中,存储问题和管理问题就变得格外的重要。大型数据库可以很好地存储和管理这些 case<sup>[11]</sup>,但是需要占用大量的空间资源,excel 表格也可以少量存储 case,但是大量的 case 就会很难查找,甚至有可能误操作,导致文件丢失。另外,软件的一些模块功能需求不断的变化,从而导致在这些大量的 case 中很难修改和维护管理。

XML(可扩展标记语言)作为存储数据和展示数据的一个非常优秀的方式,它能很好地解决以上问题。用 XML 存储 case 如下所示:

```
<Case Management>
```

```
<CaseVersion Version="1.0"/>
```

```
<SuiteCases>
```

```
< Case CaseID="number 1" CaseDescription="
.....">
```

```
< NameSetting>
```

```
< module name="module01"/>
```

```
</NameSetting>
```

```
</Case>
```

```
< Case CaseID="number2" CaseDescription=".....">
```

```
< NameSetting>
```

```
< module name="module02"/>
```

```
</NameSetting>
```

```
</Case>
```

```
...
</SuiteCases>
</Case Management>
```

其中 suitecases 是一个测试集,其中可以存放大量的 case,每条 case 有对应的 ID 和 Description 以及这条 case 相对应的模块。

这里采用 AutoIt 脚本开发设计了一个可视化的 case management。它的主要功能是通过读取存储 case 节点的 XML 文件,并查找和替换相应的 case,从而实现管理 case 的目的。

界面设计见图 3。

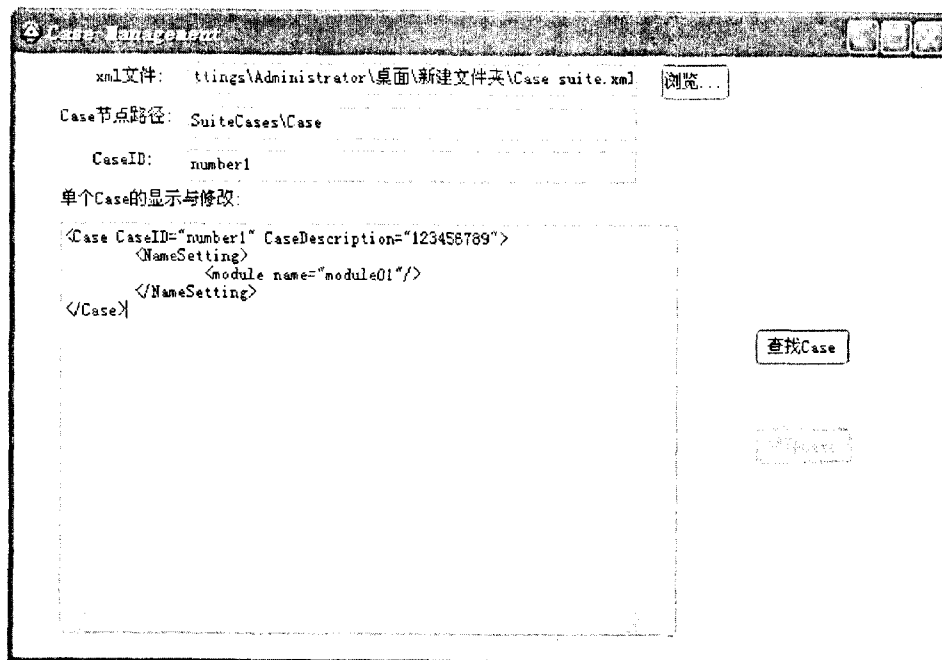


图 3 Case Management

这个工具用 AutoIt 脚本语言开发,其中查找 XML 中存储的 case 的核心代码如下:

```
Func Find_xml( $ Searchnode)
Dim $ re[2] $ Tmp_CasePath= GUICtrlRead( $ Input2)
$ CasePath= StringSplit( $ Tmp_CasePath, "\ ")
For $ n=1 To $ CasePath[0]
    If $ Searchnode. nodeName = $ CasePath[ $ n] Then
        ContinueLoop
    EndIf
    If $ Searchnode. childNodes. length <= 0 Then
        MsgBox(4096, $ Title, "Case 节点路径出错!" & $ Searchnode.
        nodeName & "没有子节点.")
        ExitLoop
    Else
        For $ i = 0 To $ Searchnode. childNodes. length-1
            If $ Searchnode. childNodes. item( $ i). nodeName = "Case"
Then
                If $ Searchnode. childNodes. item( $ i). attributes. get-
```

```
NamedItem("CaseID"). value= GUICtrlRead( $ Input1) Then
    $ re[0]= $ Searchnode. childNodes. item( $ i)
    Return $ re
EndIf
ElseIf $ Searchnode. childNodes. item( $ i). nodeName =
$ CasePath[ $ n] Then
    $ re[1]= $ Searchnode. childNodes. item( $ i)
    $ Searchnode = $ Searchnode. childNodes. item( $ i)
    ExitLoop
EndIf
If $ i = $ Searchnode. childNodes. length -1 Then
    MsgBox(4096, $ Title, "Case 节点路径出错! 找不到节点:"
    & $ CasePath[ $ n])
    Return "null"
EndIf
Next
EndIf
Next
EndFunc
```

从图 3 中可以看出,首先选择存储了大量 case 的一个 XML 文件(如果 case 达到一定的数量,也可以分为几个文件存储),然后输入 case 节点的路径和 CaseID,点击查找 case 就可以找到相应的 case 了,其中的 CaseDescription 就是具体 case 的内容,这时可以编辑修改,最后点击保存就可以

保存修改后的 case 到 XML 文件中。

## 6 结束语

文中借助 AutoIt 语言和 Flow Chart 实现测试用例的生成并编写测试用例脚本,然后通过把 case 和 XML 文件的结合以及通过窗口化的操作界面使得更加容易地管理和维护 case。这样大量的 case 无需借助数据库就可以存储于若干个 XML 文件中,因此不仅方便 case 的查找修改和维护,而且使 case 得到了统一的管理。

今后为了实现测试用例的自动化执行,可以通过导出 XML 中节点的 case 脚本为一个或多个可以在 AutoIt 环境下运行的 au3 文件,从而可以批量运行 case。

## 参考文献:

- [1] 万松松,薛锦云,谢武平.最小测试用例集生成方法改进及  
(下转第 64 页)

viceMethod 是 WebService 提供的方法, TargetControlID 是关联的 TextBox 的 ID。

## 8 系统特点

青岛市数字城建档案馆系统以电子档案的接收、录入、管理、借阅业务流程为主线,覆盖了档案馆内包括财务、办公、公文管理等各方面的业务,基本满足了正常的工作办公实际需要。系统具备以下几个特点:

(1) 基于三层架构的 Web 模式开发。服务器端主要基于 Asp.net 2.0 开发。Asp.net 采用 code-behind 技术,将页面控件和后台逻辑代码分离。在后台代码中,调用业务逻辑、数据访问等组件,形成软件分层。在客户端,大量采用了 Asp.net Ajax 技术,减少了与服务器端的交互,提高了系统性能,提供了更好的用户体验。

(2) 提供了电子文件档案、图片档案的批量上传、打印、在线浏览等功能,方便用户的借阅使用,提高了档案的使用率。

(3) 流媒体技术。声像档案是城建档案的重要组成部分,具有数据量大、形象直观等特点。系统采用了当前较流行的流媒体服务器 Microsoft 公司的 Windows Media,它支持 MPEG 等多种影像格式,结合 JavaScript 实现了嵌入式页面播放器,能够支持在线浏览和在线下载。

(4) 高安全性。系统对用户身份验证、授权、电子档案数据的安全性提供了保障,采用了基于 XML 的动态生成菜单的技术,控制用户的访问权限,提高了系统安全性。

(5) 各个业务之间采用模块设计方法,提高了系统的可扩展性、灵活性、开放性。

## 9 结束语

目前,“青岛市数字城建档案馆”系统已基本开发完毕试运行。实践证明,系统规范了城建档案馆内的工作流程,提供了快捷、形象的电子档案在线浏览,提高了档案利用检索的效率。

### 参考文献:

- [1] 黄宏. 档案数字化在数字城建系统中的开发与利用[J]. 今日科技, 2007(6): 43-44.
- [2] 樊海斌. 基于 ASP.NET 技术的办公自动化系统设计[J]. 软件导刊, 2007(5): 85-86.
- [3] Martin R C. 敏捷软件开发原则、模式与实践[M]. 孟岩, 译. 北京: 清华大学出版社, 2006.
- [4] Shaw M, Clements P. The Golden Age of Software Architecture[J]. IEEE Software, 2006, 23(2): 31-39.
- [5] 庄亮, 王丽芳, 蒋泽军, 等. .Net 平台下基于 Facade 模式的软件 N 层架构的设计与实现[J]. 微电子学与计算机, 2006, 23(7): 75-77.
- [6] Lhotka R. Expert C# 2005 Business Objects[M]. 王鑫, 译. 北京: 电子工业出版社, 2007.
- [7] 李晓华. 城市规划档案信息系统的架构优化分析与设计[J]. 计算机系统应用, 2006(9): 2-5.
- [8] 奚江华. 圣殿祭司的 ASP.NET 2.0 开发详解: 使用 C# [M]. 北京: 电子工业出版社, 2008.
- [9] 阮军, 杨春金. 基于模式的多层分布式软件系统架构的设计[J]. 计算机工程, 2006, 32(14): 57-59.
- [10] 魏洪涛, 李群, 王维平. 基于 Web 的管理信息系统的安全模型设计[J]. 计算机应用, 2003, 23(5): 77-80.
- [11] Frydenberg M. Slickr: A Multi-Tiered Web Development Capstone Project Using Databases, Web Services, and AJAX [EB/OL]. 2008-03-18. [http://isedj.org/6/37/ISEDJ.6\(37\).Frydenberg.pdf](http://isedj.org/6/37/ISEDJ.6(37).Frydenberg.pdf).

(上接第 59 页)

- 应用[J]. 计算机技术与发展, 2008, 18(10): 181-183.
- [2] Kelly M. Choosing a test automation framework[EB/OL]. 2003-11-20. <http://www.ibm.com/developerworks/rational/library/591.html#N10223>.
- [3] Pettichord B. Success with Test Automation[EB/OL]. 1996-05. <http://www.io.com/wazmo/succpap.htm>.
- [4] Memon Atif M, Xie Qing. Studying the fault-detection effectiveness of GUI test cases for rapidly evolving software[J]. IEEE Transactions on Software Engineering, 2009(5): 31-38.
- [5] Dustin E, Rashka J, Paul J, et al. Automated software testing - introduction, management, and performance[M]. [s.l.]: Addison-Wesley Professional, 1999.
- [6] 罗文飏, 周伯生, 王慧, 等. 基于切片技术获取系统测试用例的方法[J]. 计算机工程, 2008(13): 13-18.
- [7] Mosley D J, Posey B A. 软件测试自动化[M]. 邓波, 黄丽娟, 曹青春, 等译. 北京: 机械工业出版社, 2003.
- [8] 马菁, 顾景文. 决策树在软件测试用例生成中的应用[J]. 计算机技术与发展, 2008, 18(2): 66-69.
- [9] Mansour N, Bahsoon R. Reduction-based methods and metrics for selective regression testing[J]. Information and Software Technology, 2002, 6(4): 132-133.
- [10] Hermadi I, Ahmed M A. Genetic algorithm based test data generator[J]. Evolutionary Computation, 2003, 2(2): 36-41.
- [11] 马庆利, 王澜. 自动生成 XML 测试脚本的类测试[J]. 计算机技术与发展, 2006, 16(11): 145-148.