

GUI 软件测试文档辅助工具的设计与实现

隋颖¹, 于秀山², 杨豹²

(1. 中国人民解放军理工大学 指挥自动化学院, 江苏 南京 210007;

2. 中国电子系统设备工程公司研究所, 北京 100141)

摘 要: GUI 软件的应用越来越广泛的同时, GUI 软件的测试任务变得也越来越重, 如何减少 GUI 软件测试文档的编写负担成为一个值得关注的问题。文中介绍了一款自主研发的测试文档辅助工具 AutoDoc 的设计和实现, AutoDoc 通过调用 Windows 底层的 Hook 消息处理函数, 来监视用户操作及系统响应的各种消息, 达到录制的目的。文中着重说明了工具实现过程中的解决方案, 结合例子对工具进行实验, 实验证明本工具可以有效地提高测试文档编写的效率。

关键词: AutoDoc; 图形用户界面; 测试文档; 录制

中图分类号: TP311.56

文献标识码: A

文章编号: 1673-629X(2009)12-0146-04

Design and Implementations of Tool for Aiding Document of Test

SUI Ying¹, YU Xiu-shan², YANG Bao²

(1. Institute of Command and Automation, PLA University of Science and Technology, Nanjing 210007, China;

2. Institute of China Electronic Systems Equipment Company, Beijing 100141, China)

Abstract: The using of application of GUI software becomes increasingly wide, the task to test it become arduous. How to effectively reduce the burden of testing the GUI software has become a matter of concern. The article introduces the design and implementation of a self-developed tool named AutoDoc for aiding document of test, highlighted introduce the solution of implementation process and check it with two examples. This tool can greatly improve the efficiency of the preparation of test documentation effectively.

Key words: AutoDoc; GUI; document of test; record

0 引言

软件测试作为软件质量的保障手段^[1], 越来越受到人们的重视, 作为测试结果的书面表现, 测试文档的重要性不可言喻。对于广泛使用的图形用户界面(GUI), 测试文档编写任务更是非常繁重。在测试文档编写过程中, 测试人员需要根据特定的模板, 将测试步骤等内容填入相应的模板中, 这需要测试人员返回测试界面以及查看测试用例相结合才能完成填入内容工作, 耗费了测试人员大量的精力。再者相对测试工作来说, 书写测试文档有其滞后性, 很难保证测试文档与实际测试过程完全一致^[2]。三是由于文档书写的遗漏, 使得测试完毕后错误难以再现。据统计, 在 GUI 软件测试过程中, 测试文档的编写约占整个项目时间的 70%, 这就迫切需要一款能够简化这个过程并且客

观记录操作流程的测试文档辅助工具。

AutoDoc 是一款针对 GUI 软件的测试文档辅助生成工具, 该工具可以自动捕获测试人员在软件测试过程中所做的操作以及系统响应信息。测试人员只需根据规定的文档格式, 通过复制、粘贴或简单编辑等操作, 就可以完成测试文档中测试步骤的填写工作, 从而减少测试文档编写时间, 提高测试效率和准确性。

1 总体设计

1.1 设计要求

根据实际测试中的经验, 要设计一款实用的测试文档辅助工具, 需要包含以下几点设计要求:

第一: 启动之后不影响被测软件的正常使用, 这是设计 AutoDoc 最基本的要求。

第二: 对操作的控件进行识别。不同的控件取其属性文本的方法是不同的, 在 GUI 软件窗体中, 除了按钮、文本框、标签、下拉框、选择栏等标准控件外, 还有列表框、工具栏按钮、菜单等非标准控件, 还要尽可能的对开发平台所有的控件进行识别处理, 才会对各

收稿日期: 2009-03-23; 修回日期: 2009-06-11

基金项目: 军队 916 工程项目(编号: 略)

作者简介: 隋颖(1982-), 男, 天津人, 硕士研究生, CCF 会员, 研究方向为软件工程、软件测试; 于秀山, 博士后, 研究员, 研究方向为软件测试。

种操作做到信息捕获。

第三:生成的文件可以直接或稍加编辑后为测试文档所利用。为了做到这点,除了尽多的识别控件外,还需要对获取的控件属性文本进行规范化描述。

第四:自动记录下被测软件错误提示,手动通过快捷键截取错误窗口,直接保存到记录文件中。这方面要求便于测试完毕后使测试人员和被测软件开发方进行错误跟踪,共同解决问题所在。

第五:便于扩展新的开发平台开发的软件。

1.2 Auto 的设计流程

图 1 为 AutoDoc 的设计流程图,其中第五点为 AutoDoc 的内部执行流程,AutoDoc 在系统中安装的自定义 Hook 在这一点起关键作用,系统根据用户操作调用相应 Windows 底层 API 函数,完成特定的功能。

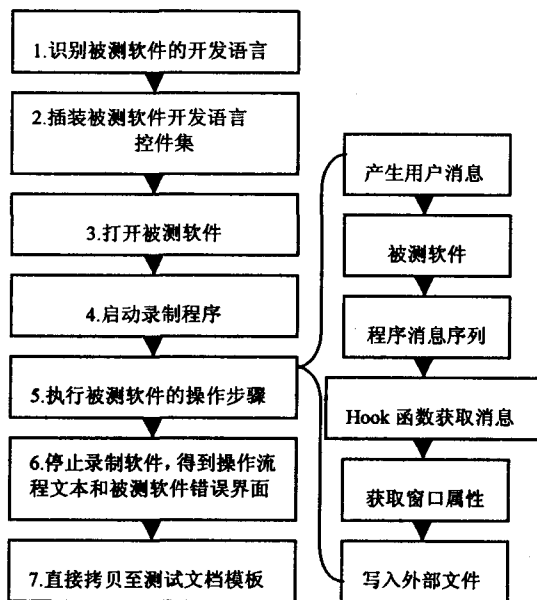


图 1 AutoDoc 执行流程图

2 AutoDoc 的实现

AutoDoc 是基于 .net 平台,运用 C# 语言开发的。为了完成特定的功能,需要调用大量的 Windows 底层 API 函数,这些 API 函数在 .net 平台上调用起来非常方便。结合 AutoDoc 的实现,下面将 AutoDoc 的基本原理、实现手段以及应用实例作以下表述。

2.1 AutoDoc 的基本原理

AutoDoc 主要采用的是录制技术,与现在大多数自动化测试软件中录制/回放中录制的核心技术相同。虽然现在自动化测试工具的研究已经取得了很多的成果,不乏有如 Winrunner, QTP 等优秀的自动化测试工具,在录制/回放方面做的非常成功^[3],但它们的录制操作是为回放之用,生成的是脚本语言,不容易让人理解,文档编写时测试人员只能适当借鉴,不能对测试

文档的编写提供多大的帮助。

录制过程主要用的是 Windows Hook 技术,Windows 系统建立在事件驱动的机制上,整个系统都是通过消息的传递来实现的^[4],消息的工作机制如图 2 所示。Hook 是 Windows 系统中非常重要的系统接口,通过安装 Hook 子程序就可以来监视被测程序触发事件的传递,并且在消息到达目的窗口过程之前对消息进行处理^[5]。AutoDoc 安装了鼠标 Hook 和键盘 Hook,它的挂钩就是通过 Windows SDK 的 SetWindowsHookEx 函数安装线程挂钩过程,利用 Hook 捕获 Windows 发给应用程序的消息,也就是鼠标、键盘操作和系统响应,调用 API 函数获取操作控件属性文本,再按照固定格式输出,形成事件序列,从而形成规范的操作流程描述。

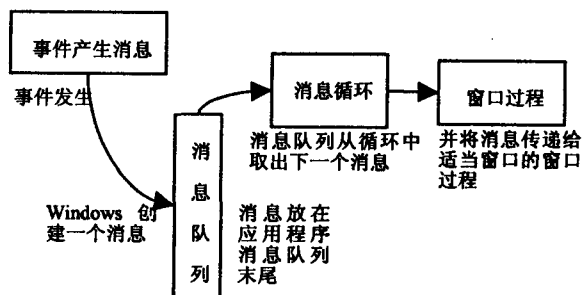


图 2 Windows 消息机制

2.2 实现手段

为了使工具达到以上的设计要求,在软件的开发过程中,主要采用了以下几种手段。

(1) 控件识别。

对于操作过程中控件属性的识别主要是调用 Windows API 的底层函数,Windows API 中的所有函数都包含在 DLL 中,其中三个最重要的 DLL 分别是 KERNEL32.dll,它包含用于管理内存、进程和线程的各个函数;USER32.dll,它包含用于执行用户界面任务(如窗口的创建和消息的传送)的各个函数;GDI.dll,它包含用于画图和显示文本的各个函数。AutoDoc 的代码编写主要调用的就是它们封装的函数^[6]。

录制最主要的是对用户操作的控件进行识别。操作系统为每个 WINDOWS 控件提供了一个句柄,句柄是窗体和控件的唯一标识。安装 Hook 程序后,通过鼠标点击控件获取它的句柄,再根据句柄判断类名或类型,然后依据控件类型调用 API 函数获取属性文本,获取的属性文本是规范化描述处理的基本要素。

(2) 规范化处理。

录制过程获取的是控件属性文本,为了直接或者通过简单的编辑就可以在测试文档中使用,需要制定一个操作控件的描述规范用于 AutoDoc。控件包括按

钮、文本框、标签、选择列表框、菜单项等控件,对部分控件所进行的规范化描述如表 1 所示。

表 1 控件属性的规范化描述

序号	控件名称	控件属性	属性描述
1	BUTTON	确定	在“***”窗口内点击“确定”按钮
2	LISTBOX	管理员	在列表框中选择“管理员”项
3	LISTVIEW	我的电脑	在列表框内点击“我的电脑”列表项
4	COMMANDBAR	保存	在工具栏中点击“保存”按钮
5	EDIT	张三	在“用户名”文本框中输入“张三”
6	MENU	文件 打开	点击菜单项“文件——>打开”

(3) 其他解决方案。

对于一些比较特殊的控件和一些特殊情况,在开发的过程中也进行了相应的处理:

1) 文本框、下拉框控件。

文本框和下拉框控件基本都有标签说明,如果单把文本框控件属性值取出,测试人员不会清楚它实际代表的意思,所以在录制程序开发的过程中,需要对文本框和下拉框等类似控件进行标签匹配(如表 1 第 4 项),使得测试人员更好地理解这类控件代表的意义。

2) 菜单。

用户取菜单项时首先点击菜单栏,展开菜单后才能点击菜单项。例如操作记事本程序时,若再点击“打开”菜单项,必须首先点击顶级菜单“文件”。在点击“文件”时捕获触发的是一个菜单事件,AutoDoc 不对这项操作做任何动作,直至点击至最底级菜单项时,AutoDoc 调用 API 函数取出菜单项的整个路径“文件——>打开”。

3) 自动获取错误提示与截取屏幕实现。

在测试过程中如果遇到软件错误时,一般会弹出错误提示窗口,窗口中包含文字错误提示。窗口的类型可以通过调用 API 函数来判断,这种情况下 AutoDoc 首先判断窗口的类型是否为错误提示窗口,然后再保存窗口的错误提示信息并记录下来;另外一个重要方面是为了保留“事故现场”,AutoDoc 通过键盘 Hook 设置快捷键将当前窗口进行截屏,然后保存至录制文件,把测试场景以及场景发生时间记录下来,这样才有利于将来发现缺陷,追溯问题,直观地查看错误出现时的一系列场景。

2.3 应用实例

AutoDoc 通过调用 Windows 底层应用程序接口 API 获得操作信息,因此 AutoDoc 的适用环境也限于标准 Windows API 环境下开发的软件。经过对 C++、C#、.NET 平台下开发的软件进行实验,AutoDoc 可以对用户的操作准确录制。

下面通过测试 Windows 系统下的两款比较有代表性的 GUI 软件进行分析。

(1) 记事本字体设置示例。

首先执行的是对记事本工具的字体窗口进行设置的任务,图 3 是打开的“记事本字体”窗口,执行的是选择“字形”列表框中的斜体,然后点击“确定”操作。

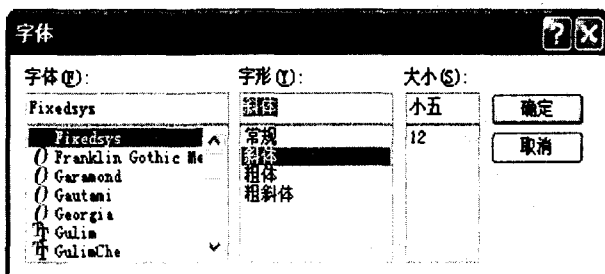


图 3 记事本字体窗口界面

图 4 所示的是录制的结果,通过测试,可以看出 AutoDoc 可以准确记录对菜单、列表框、文本框和标签匹配以及默认值自动获取的操作。

```

录制开始时间: 2009-2-26 9:34:49
任务名称: 设置字体字形用例
您的操作是:
打开“测试-记事本”
选择菜单项: “格式->字体”
弹出“字体”对话框,
字形--选择--“斜体”,
在“字体”对话框内点击“确定”按钮
--点击按钮“确定”后,各默认内容为
【字体】的内容为【Fixedsys】
【字形】的内容为【斜体】
【大小】的内容为【小五】
【字符集】的内容为【CHINESE_GB2312】
点击“stop”按钮
录制结束时间: 2009-2-26 9:35:15
    
```

图 4 记事本设置字体记录结果

(2) WinRAR 压缩设置示例。

下面执行的是对 test.txt 文件进行压缩的任务,图 5 所示的是打开的“压缩文件名和参数”窗口,用户执行的是选择“压缩文件格式”的 RadioButton 项“ZIP”,选择“压缩选项”的 CheckButton 项“压缩后删除源文件”,在“压缩方式”下拉框中选择“存储”,最后点击“确定”按钮。

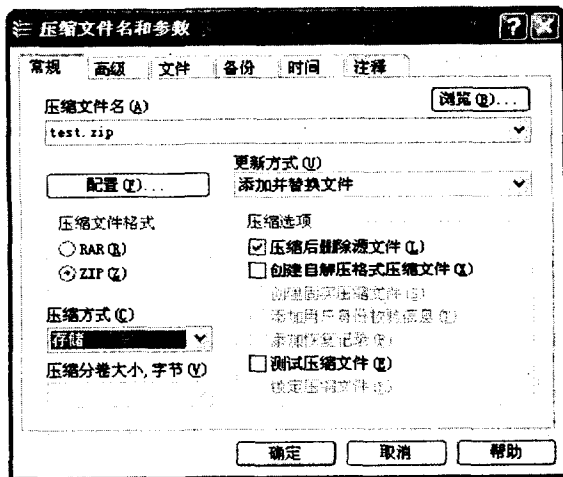


图 5 WinRAR 软件“压缩文件名和参数”窗口界面

图 6 所示为 AutoDoc 录制的结果,通过测试,AutoDoc 可以准确记录对工具栏按钮,RadioButton,

CheckBox,下拉框等控件的操作。

```

录制开始时间: 2009-2-26 10:19:59
任务名称: 文件压缩任务
您的操作是:
列表框选择的内容--test.txt
选择工具栏“添加”按钮
弹出“压缩文件名和参数”对话框;
“压缩后删除源文件”--设为--ON
“压缩方式”选择“存储”
“压缩文件格式”--ZIP--设为--ON
在“压缩文件名和参数”对话框内点击“确定”按钮
**点击按钮--确定--后,各控件默认内容为
【压缩文件名】的内容为【test.zip】
【压缩方式】的内容为【存储】
【压缩分卷大小,字节】的内容为【】
【更新方式】的内容为【添加并替换文件】
点击“stop”按钮
录制结束时间: 2009-2-26 10:20:33

```

图6 WinRAR 压缩设置记录结果

通过实验,两实例录制的结果符合图1中测试文档中的规范要求,完全可以直接用于测试文档的编写,可以复制需要的部分或全部内容,快速客观完成测试文档中测试步骤的填写工作,对于GUI测试文档的编写提供了极大的方便。

3 结束语

编写测试文档是测试过程中的一个重要的组成部分,目前市场上测试文档辅助工具的欠缺促成我们的研究。所实现的 AutoDoc 工具作为 GUI 软件测试文档编写的辅助工具,可使文档编写更加快捷方便、客观准确,可较好地解决测试人员的现实需求。下一步的研究将针对马里兰大学 Memon 博士的 GUI 软件逆工

程以及覆盖标准的基础上^[7,8],设计出手动测试和自动测试相结合的测试模型,使软件更加贴近实际应用。

参考文献:

- [1] 王泓. 软件测试文件的编制[J]. 电子产品可靠性与环境试验, 2005, 44(5): 63-64.
- [2] 李菲, 晏海华, 郝建营. 基于 XML 的通用文档生成方法研究[J]. 计算机工程与设计, 2007, 28(20): 4836-4838.
- [3] Steven J, Chanda P, Fleck B, et al. A Capture/Replay Tool for Observation-Based Testing[M]. [s.l.]: ACM, 2000.
- [4] 戴春达, 符红光. Win32 中钩子的实现技术及应用[J]. 计算机应用, 2002, 22(8): 72-74.
- [5] 陈浩, 邹北骥, 江文, 等. 基于消息机制的自动化黑盒测试方法研究[J]. 计算机应用研究, 2004(10): 33-35.
- [6] Petzold C. Windows 程序设计[M]. 北京博彦科技发展有限公司译. 北京: 北京大学出版社, 1999.
- [7] Memon A M. Reverse Engineering of Graphical User Interface for Testing[C]// Proceedings of the 10th Working Conference on Reverse Engineering (WCRE'03). [s.l.]: IEEE, 2003.
- [8] Memon A M, Soffa M L, Pollack M E. Coverage criteria for GUI testing[C]// ESEC/FSE-9: Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering. New York, NY, USA: [s.n.], 2001: 256-267.

(上接第134页)

关系,见(2)式。

$$\xi < \frac{\alpha_{\text{gate}}^{K_{\max}+1}}{A} \quad (\xi > 1, K_{\max} \text{ 为正整数}) \quad (2)$$

其中, A 代表了(1)式括号中的算式, α_{gate} 是 α 的取值门限, 当网络拓扑一定时, A 和 α_{gate} 为大于 1 的常量; 如果 α 的取值不超过此门限, 算法就不会落入陷阱。经测试同时可知, ξ 取值不宜过大, 否则会增加算法的运行时间。一般来说, 建议在链路代价数量级差别较小的网络中, ξ 取值只需比 1 稍大即可。

5 结束语

在网络信息量与日剧增的当代, 提高网络抗毁能力的重要性日益凸显。SRLG 分离的双路由选择策略为网络抗毁性问题提供了一个较为有效的解决方案。针对几种传统策略, 文章中提出的新策略简单灵活, 效率较高, 具有一定的实用价值。

参考文献:

- [1] 陈太一, 陈常嘉. 通信网的可靠性[J]. 电信科学, 1992, 8

(5): 18-25.

- [2] 冯海林. 网络系统中可靠性问题的研究[D]. 西安: 西安电子科技大学, 2004.
- [3] 张俊良. 复杂网络可靠性研究[D]. 大连: 大连理工大学, 2006.
- [4] 刘啸林. 网络抗毁性研究介绍[J]. 计算机应用与软件, 2007, 24(6): 135-137.
- [5] Mannie E. Generalized multi-protocol label switching (GMPLS) architecture[S]. IETF Network Working Group. Request for Comments: 3945 Standards Track, 2004.
- [6] Xu Dahai, Xiong Yizhi, Qiao Chunming, et al. Trap avoidance and protection schemes in networks with Shared Risk Link Groups[J]. IEEE Lightwave Technology, 2003, 21(11): 2683-2693.
- [7] Yen J. Finding the k shortest loopless paths in a network[J]. Management Science, 1971, 17(11): 712-716.
- [8] Oki E, Matsuura N, Shiimoto K. A disjoint path selection scheme with shared risk link groups in GMPLS networks[J]. Communications Letters, IEEE, 2002, 6(9): 406-408.
- [9] 谢希仁. 计算机网络[M]. 第4版. 北京: 电子工业出版社, 2003: 434-435.