



计算机科学

COMPUTER SCIENCE

一种基于 UIA 接口的 RPA 系统设计方法

王岩松, 秦云川, 蔡宇辉, 李肯立

引用本文

王岩松, 秦云川, 蔡宇辉, 李肯立. 一种基于 UIA 接口的 RPA 系统设计方法[J]. 计算机科学, 2022, 49(8): 225-229.

WANG Yan-song, QIN Yun-chuan, CAI Yu-hui, LI Ken-li. Design and Implementation of RPA System Based on UIA Interface[J]. Computer Science, 2022, 49(8): 225-229.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于双向蚁群算法的网络攻击路径发现方法](#)

Network Attack Path Discovery Method Based on Bidirectional Ant Colony Algorithm

计算机科学, 2022, 49(6A): 516-522. <https://doi.org/10.11896/jsjcx.210500072>

[模型驱动开发工具的自动化测试技术研究](#)

Research on Automatic Testing Technology of Model Driven Development Tools

计算机科学, 2021, 48(6A): 568-571. <https://doi.org/10.11896/jsjcx.201000139>

[面向 64 位 RISC-V 的基础数学库自动化移植](#)

Automatic Porting of Basic Mathematics Library for 64-bit RISC-V

计算机科学, 2021, 48(6): 41-47. <https://doi.org/10.11896/jsjcx.201200058>

[基于多 Agent 联合决策的队组协同攻击规划](#)

Team Cooperative Attack Planning Based on Multi-agent Joint Decision

计算机科学, 2021, 48(5): 301-307. <https://doi.org/10.11896/jsjcx.200800174>

[面向 Java 的 Randoop 自动化单元测试生成工具性能分析](#)

Performance Analysis of Randoop Automated Unit Test Generation Tool for Java

计算机科学, 2020, 47(9): 24-30. <https://doi.org/10.11896/jsjcx.200200116>

一种基于 UIA 接口的 RPA 系统设计方法

王岩松 秦云川 蔡宇辉 李肯立

湖南大学计算机科学与工程学院 长沙 410082

(wangys1997@hnu.edu.cn)

摘要 机器人流程自动化(Robotic Process Automation,RPA)是当今的研究热点之一,RPA 主要使用 Win32 接口对窗体进行自动化操作,该方法需要对 API 进行封装,开发成本高,且只能识别基于 Win32 API 设计的 UI 元素,同时基于 Win32 接口实现自动化需要模拟键盘和鼠标操作,由于该模拟操作基于一种广播消息机制,导致操作的响应时间较长,因此提出了一种基于微软的 UIA(UI Automation)技术,用于构建 RPA 流程自动化应用的方案。该方案采用 UIA 方法进行 UI 元素的自动化操作,适应的 UI 程序框架范围广,包括 Win32,WPF,QT,Silverlight 等;而且开发成本低。同时,该方案将消息与 UI 元素绑定,避开了低效的广播消息机制,提高了 RPA 的执行效率。实验结果显示,与模拟键盘鼠标方法相比,所提方法的执行时间平均可缩短 55.67%。

关键词:UIA 技术;自动化;RPA

中图法分类号 TP312

Design and Implementation of RPA System Based on UIA Interface

WANG Yan-song, QIN Yun-chuan, CAI Yu-hui and LI Ken-li

School of Computer Science and Engineering, Hunan University, Changsha 410082, China

Abstract Robotic process automation(RPA)is one of the current research hotspots. RPA mainly uses Win32 interface to automate the operation of windows. This method requires to encapsulate the API, which is expensive to develop and can only identify UI elements designed based on Win32 API. At the same time, automation based on Win32 interface needs to simulate keyboard and mouse operations. Because these operations are based on a broadcast message mechanism, the response time is long. This paper proposes a solution for building RPA process automation application based on Microsoft's UIA technology. The solution uses UIA methods to automate UI elements. It can adapt to a wide range of UI program frameworks, including Win32, WPF, QT, Silverlight, etc., and the development cost is low. At the same time, this method binds the message to the UI element, avoids the inefficient broadcast message mechanism, and improves the execution efficiency of RPA. Experimental results show that the execution time can be shortened by 55.67% on average compared with the keyboard and mouse method.

Keywords UIA technology, Automation, RPA

1 引言

机器人流程自动化能够根据相应规则替代或辅助人工操作,具备高效、准确、无人值守的优点^[1-2]。近年来,RPA 已在各个行业得到广泛应用,如金融业、工业、教育机构、人工智能等^[3-13]。Sutipitakwong 等研究了 RPA 在微小繁琐工作方面的用途和重要性,研究表明 RPA 可以以非常高的准确度管理大容量文档^[3]。Yokoyama 等研究了 RPA 可以应用的编程学习领域,研究表明将 RPA 与教育相结合,将对提高学习动机产生巨大影响^[4]。Chen 等基于 RPA 对自动化财务报销审计方法进行了优化,实验结果表明,基于 RPA 机器人的财务审计错误率明显低于传统方法^[5]。Wewerka 等评估了

RPA 在工业中的可用性,并推导出了设计 RPA 机器人用户界面的 7 项准则^[6]。Wewerk 等开发了一个模型来评估 RPA 的用户接受度以及影响它的变量,为 RPA 机器人的设计和实施提供了具体建议^[7]。Baidya 将 RPA 与 AI 相结合,使用机器学习模型来处理正式文档,并结合深度学习算法作为分析方法,提出了与非结构化数据相关挑战的解决方案^[8]。文献^[9-10]表明,RPA 将是第四次工业革命的突破性解决方案,并分析了 RPA 的发展前景。Ketkar 等提出了一种利用 RPA 特性从源数据收集数据的方法^[11]。Issac 等分析了 3 个领先的 RPA 平台,即 UiPath Studio, Automation Anywhere 和 Blue Prism 各自的优越性^[12]。MARTINS 等将 RPA 与深度学习技术相结合,实现实时动态检测应用程序界面,该方法

到稿日期:2021-11-03 返修日期:2022-03-14

基金项目:国家重点研发计划(2020YFB2104002)

This work was supported by the National Key Research and Development Program of China(2020YFB2104002).

通信作者:秦云川(qinyunchuan@hnu.edu.cn)

不具有出色的准确性,且能够采取动态行动^[13]。

近年来,基于 Windows 开发的应用软件愈来愈多且风格迥异,将 RPA 技术应用到这些软件上是非常困难的。传统的 Win32 方法不仅无法识别许多新框架窗体内容,又因为其工作于操作系统底层,需要对 API 进行额外封装,开发成本高,同时实现自动化需使用模拟键盘鼠标方法,响应时间长。

针对以上缺点,本文提出了一种基于微软的 UIA 技术,用于构建 RPA 流程自动化应用的方案,对这些功能不同且风格迥异的窗体实现自动化操作。

本文第 2 节介绍了 RPA 和 WindowsUIA;第 3 节介绍了基于 UIA 的流程自动化应用;第 4 节给出了实验结果;最后总结全文。

2 相关技术介绍

2.1 RPA

RPA 即机器人流程自动化,指一种新型的人工智能虚拟流程自动化机器人。RPA 的核心是通过智能化、自动化技术替代人工进行重复性、低价值、无需人工决策等固定的流程化工作^[14]。

RPA 最显著的优势是降本增效^[14]。首先,RPA 可以代替或者辅助人工处理枯燥费时且重复性高的流程自动化任务,为企业节约大量成本。其次 RPA 相当于一个或者任意多个虚拟员工,它可以快速继承人在各个业务领域所积累的经验,在指令下达后,可快速进入工作状态,不知疲倦,效率更高,且错误率接近“0”。

在实际流程应用中,RPA 机器人通常承担以下基础功能^[15]:数据检索和记录,RPA 机器人可以跨系统完成数据检索、数据迁移和数据录入;图像识别和处理,利用 OCR(光学字符识别)技术,RPA 机器人可以完成信息识别,并在此基础上检查和分析文字;数据存储和整理,包括数据查询、数据分析、数据整理、数据存储和数据校验。

2.2 Windows UIA

UI(User Interface)是用户操作界面的简称,是用户与软件系统进行交互的接口。UIA 可以理解为开发一个应用程序,其能够模拟用户的输入和操作,并且得到预期结果^[16]。因此,其过程包括两个步骤:拾取应用软件界面上的 UI 元素以及模拟用户的操作。

在 UIA 中,每一个被公开的 UI 元素都可以看作是一个自动化元素(Automation Element),所有自动化元素都被包含在一个树结构中,Windows 操作系统的桌面就是这棵树的根节点^[17]。

UIA 定义了读写 UI 元素的 UIA 属性(Properties)^[18],例如 Name,Class Name,Control Type,Automation Id 等,利用这些属性可以快速定位到相应的 UI 元素。

UIA 还定义了 UI 元素的 UIA 行为模式(Patterns)^[18],可以通过使用这些函数接口,实现对不同类型 UI 元素的控制。例如,输入框的自动赋值、按钮的自动点击、下拉框的自动选择等。

通过 UIA 技术实现计算机窗体的自动化具有如下优势^[18]:1)对于 Windows 系统平台上的应用程序都有很好的

支持性,UIA 不仅可以支持 WPF 和 Silverlight,还可以支持 Win32 和 Win Form 程序;2)UIA 的树结构、UIA 的自动化元素属性和控件模式属性以及 UI 元素支持区域化和条件化搜索等都为实现自动化提供了坚实的基础;3)UIA 完成了方法的封装,开发成本低;4)与 Win32 事件不同,UIA 事件不基于广播机制,而是通过窗体注册特定事件并将其加入到它们的处理程序中,响应时间短。

本文利用 UIA 方法制定规则去自动化所有运行在 Windows 操作系统上的窗体,构建 UIA 与 RPA 相结合的流程自动化应用。

3 基于 UIA 的流程自动化应用

若要实现计算机窗体的自动化,则需要识别窗体中的 UI 元素并操控它^[19]。例如实现自动保存一个文件,相应的 RPA 流程规则如下:首先设置保存文件的名称;其次选择保存文件的类型;最后点击“保存”按钮,实现文件保存到本地系统。利用 UIA 技术可以实现以上操作。

本文基于 UIA 构建了 RPA 流程自动化应用,用于实现 Windows 系统下窗体的自动化。

3.1 架构设计

通过对窗体自动化应用的需求分析,明确了其需要实现的功能,设计了两大模块,即元素拾取模块和自动化操作模块^[20],其总体架构如图 1 所示。

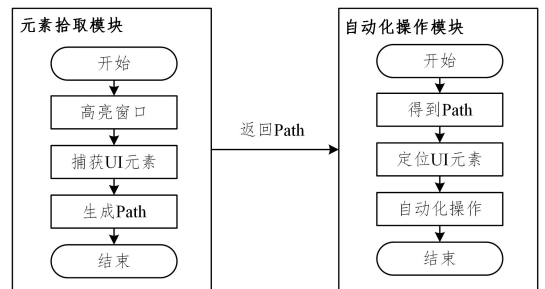


图 1 Windows 应用自动化平台的架构图

Fig. 1 Architecture diagram of Windows application automation platform

Path 信息记录目标 UI 元素在 UIA 树结构中的路径,如图 1 所示,Path 信息对于两个模块都非常重要。元素拾取模块最终得到目标 UI 元素的 Path 信息,自动化操作模块会利用 Path 信息快速准确定位目标 UI 元素。

图 2 给出了由元素拾取模块生成的一个 Path 示例,用于存放每层级节点的信息。

```
'Path': [{ 'type': 'Window', 'name': 'MyMFCApp', 'pname': 'MFC-CAppl.exe' },
          { 'type': 'Edit', 'num': '1' } ]
```

图 2 Path 示例

Fig. 2 Example of Path

3.2 元素拾取模块

元素拾取模块是整个自动化应用的基础,主要用于展示窗体中所有 UI 元素的详细信息,包括句柄(handle)、名称(name)、类型(class_name)、控制类型(control_type)、坐标信息(rectangle)等。同时选择想要自动化操作的 UI 元素,返回

此 UI 元素的必要信息。元素拾取模块的流程如图 3 所示。

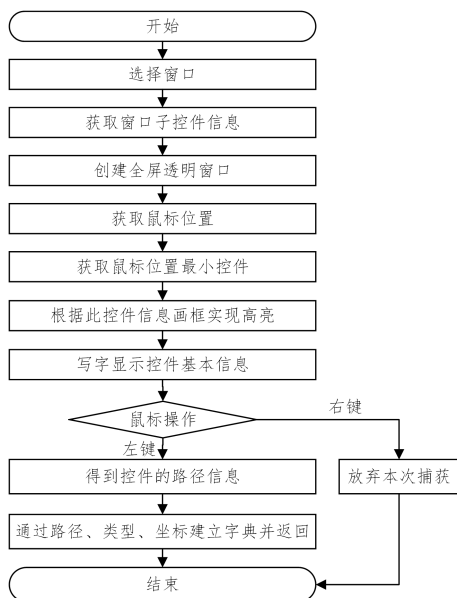


图 3 元素拾取模块流程图

Fig. 3 Flow chart of element picking module

模块执行后,首先选择要拾取 UI 元素的窗体,利用 Win32 API 得到目标窗体的句柄;然后将此句柄作为参数传入算法 1 所示的函数中,该函数的作用是利用 UIA API 得到窗体中所有 UI 元素的信息,并存入字典。函数的具体实现如算法 1 所示。

算法 1 获取所有 UI 元素信息算法

输入:窗体的句柄

输出:一个字典,字典中存放所有 UI 元素的信息

1. 定义一个空字典:domWindow
2. 使用传入句柄创建 UIA 对象
3. 使用 UIA API 得到对象包含的所有 UI 元素
4. 对于每一个 UI 元素
 - 4.1. 使用 UIA API 得到它的句柄、名称、类型、控制类型、坐标信息
 - 4.2. 将以上信息加入字典
5. 返回字典

接下来创建一个全屏透明窗口作为实现高亮的窗口。遍历算法 1 得到的字典,寻找到包含当前鼠标位置的最小 UI 元素,在相应位置动态绘制一个不同颜色的矩形,达到高亮效果,并且将该 UI 元素信息显示在透明窗口上。

鼠标左击想要拾取的 UI 元素,返回此 UI 元素的必要信息,同时销毁高亮窗口,退出模块;如果是鼠标右击,代表放弃此次拾取,同时销毁高亮窗口并退出模块。

元素拾取成功后,会返回 UI 元素的必要信息。自动化操作模块会利用返回的信息重新定位目标 UI 元素。一个复杂的窗体上,可能存在上百个 UI 元素,甚至更多,返回的信息确保模块可以准确定位到目标 UI 元素;当 UIA 树太过庞大时,定位 UI 元素的时间消耗也会增加,返回的信息确保模块可以快速定位到目标 UI 元素。

最直观的方法是,返回 UI 元素所有可以得到的信息,根据这些信息在 UIA 树结构中逐个节点地去匹配,该方法虽然可以准确定位到目标 UI 元素,但时间开销非常大。

Path 生成算法如算法 2 所示,其思路是记录目标节点在

UIA 树结构中的路径,如果在路径中只包含每一级节点的名称,且在同一级中存在名称相同或者名称为空的节点,则会造成不可预计的错误。解决重名节点问题的方法是同时记录节点的控制类型;解决节点名称为空这一问题的方法是为节点分配索引号,用其表示同类型子节点的编号。

算法 2 Path 生成算法

输入:目标 UI 元素 aimdom

输出:目标 UI 元素的路径信息 path

1. 定义一个空的列表 path
2. 定义一个空的字典 pathInfo
3. 使用 UIA API 得到当前 UI 元素的控制类型,并将其加入字典 pathInfo["type"]
4. 如果当前 UI 元素的名称不为空
 - 4.1. 得到当前 UI 元素的名称并加入字典 pathInfo["name"]
5. 如果当前 UI 元素的名称为空
 - 5.1. 得到索引并加入字典 pathInfo["num"] //索引代表同类型子节点的编号
6. pathInfo 加入列表 path
7. 重复步骤 3 一步骤 6,直到当前 UI 元素的父节点是 UIA 树结构的根节点,最终返回的必要信息包括:目标 UI 元素的路径、控制类型和坐标信息。

3.3 自动化操作模块

自动化是通过程序代替或者辅助用户操作,需要被自动化的目标软件是由一个个界面组成的,每个界面上都有不同的 UI 元素,通过操作这些 UI 元素,就可以实现对应用软件的自动化^[21]。

如图 4 所示,自动化操作模块的具体实现步骤包括:利用路径信息准确快速定位到目标 UI 元素;利用 UIA API 或者模拟鼠标键盘的方法自动化操作该 UI 元素。

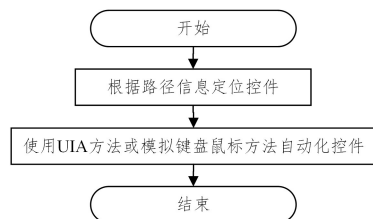


图 4 自动化操作模块流程图

Fig. 4 Flow chart of automated operation module

利用元素拾取模块得到的 UI 元素 Path 准确快速定位到目标 UI 元素,具体步骤是利用 Path 列表并遍历 UIA 树,一层一层地去获得目标 UI 元素。

利用图 2 所示的 Path 信息定位 UI 元素的过程如下:首先得到“桌面”根节点的直系后代节点,选择名称为“Qt-TestWindow”和控制类型为“Window”的节点作为新的父亲节点继续向下遍历;得到新节点的直系后代节点后,选择编号为 1 并且控制类型为“Edit”的子节点。至此 Path 遍历完成,最后得到的 UI 元素即为目标 UI 元素。函数的具体实现如算法 3 所示。

算法 3 目标控件定位算法

输入:路径信息 path

输出:目标 UI 元素 x

1. 使用“桌面”根节点创建 UIA 对象 x
2. for(i=0 to len) // len 代表 Path 列表长度

- 2.1. 获取 x 的直系后代节点列表 childlists
- 2.2. 如果 $path[i]$ 中包含 $name$ 属性
 - 2.2.1. 遍历 childlists 列表, 选择名称相同和控制类型相同的节点, 作为新的父亲节点 x
- 2.3. 如果 $path[i]$ 中包含 num 属性
 - 2.3.1. 遍历 childlists 列表, 选择编号为该数字的同类型子节点作为新的父亲节点 x
3. end for
4. 返回 x

算法 3 的时间复杂度分析如下: 算法第 2-4 行的执行次数为 $f(n) = f_2(n) + f_3(n) + f_8(n) = 2nm + n$; 第 5-6 行本身会执行 n 次, 但由于是在循环内, 因此根据乘法法则, 其执行次数为 $f(n) = f_2(n) + f_5(n) + f_8(n) = 2nm + n$; 同理第 8-9 行的执行次数为 $f(n) = f_2(n) + f_5(n) + f_8(n) = 2nm + n$ 。最后根据加法法则, 整个算法的执行次数如式(1)所示, 其中 n 表示列表长度, m 表示后代节点的个数。算法的时间复杂度如式(2)所示。

$$f(n) = f_2(n) + f_2(n) + f_5(n) + f_8(n) = 2nm + n \quad (1)$$

$$T(n) = O(f(n)) = O(2nm + n) = O(nm) \quad (2)$$

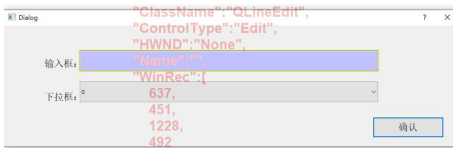
4 实验结果

第 3 节介绍了基于 UIA 技术构建 RPA 流程自动化应用的原理和算法, 本节使用这些算法对输入框、按钮、下拉框的自动化进行了实验, 以验证其可用性, 并与传统的 Win32 方法进行对比。

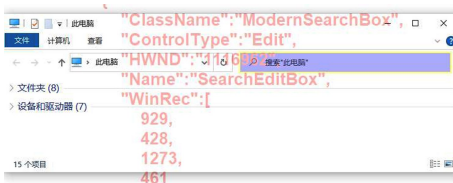
4.1 元素拾取模块实验结果

已经实现的自动化应用是支持重复使用的, 且适用于所有运行在 Windows 操作系统上的应用软件, 而不是针对某一个特定的软件设计的。因此, 在测试之前, 需要选择一个包含窗体的程序作为测试目标。

本次实验设计了两个测试案例, 一个测试案例是使用 QT 创建的窗体, 窗体包含的 UI 元素包括输入框、下拉框和按钮; 另一个测试案例是 Windows 操作系统的“此电脑”窗体。



(a) QT 窗体实验结果



(b) 此电脑窗体实验结果

图 5 元素拾取模块运行结果(电子版为彩图)

Fig. 5 Running result of element picking module

对上述测试窗口运行元素拾取模块, 运行结果如图 5 所示。当鼠标位置在目标 UI 元素上时, 该 UI 元素会变成紫色, 实现高亮效果; 同时, 屏幕上用红色的字体显示了该 UI 元素的基本信息。用鼠标左击该 UI 元素时, 会返回该 UI

元素的路径、类型和坐标信息。

4.2 自动化操作模块实验结果

使用元素拾取模块完成 UI 元素的拾取之后, 再使用自动化操作模块对该 UI 元素进行操作, 就完成了整个自动化流程。对于不同类型的 UI 元素, 操作方法各不相同。当 UI 元素的类型是文本框时, 使用文本框的 SetValue 模式更新文本框中的文字; 当 UI 元素的类型是按钮时, 使用 Invoke 模式实现按钮点击; 当 UI 元素的类型是下拉框时, 使用 Select 模式实现下拉框选值, 如果 Select 模式无效, 则使用 Invoke 模式实现下拉框选值。

实验结果显示, 自动化操作模块可实现将字符串自动赋值给文本框, 实现自动选择下拉框以及按钮的自动点击等操作, 通过这些操作实现了图 5(a)所示窗口的流程自动化。

4.3 对比实验

针对以上测试案例, 对本文提出的自动化场景进行了实验, 并与模拟鼠标键盘方法进行了比较, 实验结果如表 1 所列。

表 1 两种方法在图 5 上的测试结果

Table 1 Test results of two methods on Fig. 5

Method	UIA			Win32		
	Start	End	Spend	Start	End	Spend
Edit	0.638	0.926	0.288	0.417	1.311	0.894
ComboBox	0.685	1.231	0.546	0.723	1.656	0.933
Button	0.027	0.274	0.247	0.211	0.795	0.584

在下文的实验结果中, 分别对输入框(Edit)、下拉框(Combo Box)和按钮(Button)进行了对比实验, 其中“UIA”表示使用 UIA 接口进行自动化, “Win32”表示使用 Win32 的模拟鼠标键盘方法进行自动化, “start”和“end”分别表示开始时间和结束时间, “spend”表示算法消耗的总时间, 时间的单位均为 ms。所有实验均在 Windows10 PC Intel(R)Core(TM) i5-8250 CPU @1.60 GHz 1.80GHz, 8.00GB RAM 环境中实现, 采用 Python3.8 实现本文算法。

实验结果表明, 在图 5(a)所示的窗口中, 对于不同的 UI 元素, 本文提出的基于 UIA 的自动化系统性能均比 Win32 方法有明显提高。其中输入框 UI 元素的执行效率提高了 67.8%; 下拉框 UI 元素的执行速度提高了 41.5%; 按钮 UI 元素的执行速度提高了 57.7%。

实验结果进一步表明, UIA 事件不基于广播机制, UIA 客户端注册特定的事件通知, 并且将事件与引发该事件的 UI 元素进行绑定。与 Win32 事件相比, 不仅没有广播寻找的时间开销, 而且事件的执行是在操作系统内核中进行, 没有用户层的执行时间开销, 执行效率得到大大提高。

结束语 本文首先介绍了机器人流程自动化技术的发展状况; 其次详细介绍了微软的 UIA 技术的相关知识, 以及利用 UIA 技术实现 UI 自动化的优势; 然后阐述了基于微软的 UIA 技术构建 RPA 流程自动化应用的详细步骤; 最后使用该自动化应用, 完成了对两个应用程序窗口的自动化测试, 并对测试结果进行了分析, 结果表明其在 Windows 操作系统上具有广泛的适用性。将本文方法与模拟鼠标键盘方法进行了对比实验, 实验结果显示, 本文提出的流程自动化应用的执行时间平均可缩短 55.67%, 效率得到显著提高。

下一步的研究工作将从两个方面展开,一方面是完善应用的自动化操作模块,提供更多类型 UI 元素的自动化操作;另一方面是针对少量 UIA 技术无法识别的窗体,使用目标检测技术来识别这类窗体。

参 考 文 献

- [1] GRABOWSKI R, SIEGEL B, JOHNSON C, et al. Finance in a Digital World: Technology as a partner, not a threat[J]. The Journal of Government Financial Management, 2019, 67(4): 18-23.
- [2] MCCANN D. Special report: The new digital workforce[R]. CFO. Com. Ireland, 2018, 51(2): 24.
- [3] SUTIPITAKWONG S, JAMSRI P. The Effectiveness of RPA in Fine-tuning Tedious Tasks[C]// 2020 6th International Conference on Engineering, Applied Sciences and Technology (ICEAST). 2020:1-4.
- [4] YOKOYAMA H, FUKUI M, MATSUMOTO T, et al. Use Gameplay RPA as Programming Materials[C]// 2020 Nicograph International (NicoInt). 2020:92-92.
- [5] CHEN D, XIAO D, LI Y, et al. Automatic approval method for financial reimbursement of electric power enterprises based on RPA robot[C]// 2021 International Conference on Computer Technology and Media Convergence Design (CTMCD). 2021: 231-234.
- [6] WEWERKA J, MICUS C, REICHERT M. Seven Guidelines for Designing the User Interface in Robotic Process Automation [C]// 2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW). Gold Coast, Australia, 2021:157-165.
- [7] WEWERKA J, DAX S, REICHERT M. A User Acceptance Model for Robotic Process Automation[C]// 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC). Eindhoven, Netherlands, 2020:97-106.
- [8] BAIDYA A. Document Analysis and Classification: A Robotic Process Automation (RPA) and Machine Learning Approach [C]// 2021 4th International Conference on Information and Computer Technologies (ICICT). HI, USA, 2021:33-37.
- [9] LE V C, PHAN D H, VU T D. Robotic Process Automation and Opportunities for Vietnamese Market[C]// Proceedings of the 2019 7th International Conference on Computer and Communications Management (ICCCM 2019). 2019:86-90.
- [10] MAALLA A. Development Prospect and Application Feasibility Analysis of Robotic Process Automation[C]// 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). 2019:2714-2717.
- [11] KETKAR Y, GAWADE S. Effectiveness of Robotic Process Automation for data mining using UiPath[C]// 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). 2021:864-867.
- [12] ISSAC R, MUNI R, DESAI K. Delineated Analysis of Robotic Process Automation Tools[C]// 2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAEECC). 2018:1-5.
- [13] MARTINS P, SÁ F, MORGADO F, et al. Using machine learning for cognitive Robotic Process Automation (RPA) [C]// 2020 15th Iberian Conference on Information Systems and Technologies (CISTD). 2020:1-6.
- [14] YATSKIV N, YATSKIV S, VASYLYK A. Method of Robotic Process Automation in Software Testing Using Artificial Intelligence[C]// 2020 10th International Conference on Advanced Computer Information Technologies (ACIT). 2020:501-504.
- [15] GUPTA S, RANI S, DIXIT A. Recent Trends in Automation—A study of RPA Development Tools[C]// 2019 3rd International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE). 2019:159-163.
- [16] YATSKIV S, VOYTYUK I, YATSKIV N, et al. Improved Method of Software Automation Testing Based on the Robotic Process Automation Technology[C]// 2019 9th International Conference on Advanced Computer Information Technologies (ACIT). 2019:293-296.
- [17] FENG K, GAO J H. GUI Event Interaction Graph Generation Method Based on Model[J]. Computer Science, 2013, 40(Z6): 184-187, 203.
- [18] JASON G, MASAAHIKO K. Engineering Software for Accessibility[M]// Microsoft Press. 2009:1-7.
- [19] MORGADOI C, PAIVAAC R. The iMPAcT Tool: Testing UI Patterns on Mobile Applications[C]// 2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE). 2015:876-881.
- [20] COSTA P, PAIVAAC R, NABUCO M. Pattern Based GUI Testing for Mobile Applications[C]// 2014 9th International Conference on the Quality of Information and Communications Technology (QUATIC 2014). 2014:66-74.
- [21] HUANG S Q, LIU Y B, HUANG X S. Research on Automatic Testing Technology of Model Driven Development Tools[J]. Computer Science, 2021, 48(6A): 568-571.



WANG Yan-song, born in 1997, post-graduate. His main research interests include artificial intelligence and big data processing.



QIN Yun-chuan, born in 1983, Ph.D, is a member of China Computer Federation. His main research interests include autonomous unmanned systems and high-performance embedded computing.