An Implementation Method of Automatic Switch and Operations between Applications

Kong Liangliang[a,*], Chen Lin, and Wang Zhenxing

College of Engineering, Shanghai Polytechnic University, Jinhai 2360, Shanghai 201209, China

[a]llkong@sspu.edu.cn

Keywords: automatic switch, simulation, automatic operation, message mechanism, API

Abstract: In the field of electronic and information, computers and applications are commonly applied to various commercial and industrial businesses. Different companies of different businesses develop various types of hardware and software. Since software is easy to be copied and hard to be developed, many companies will not publish their source code, they will ask for intellectual property rights protection of government. The applications people operate are based on keyboards, mouse, display, etc. If people want to alter the applications to solve some special problems, APIs will be called to implement the alteration. However, in some cases calling APIs may not be enough to implement the alteration. In this paper, a new kind of methods is proposed to solve the problems that calling APIs cannot solve. It uses program handle to switch different applications, simulate click events that may be inspired by keyboards, mouse to provide inputs for these applications, and then switch between these application can be implemented automatically.

1. Introduction

In the field of electronics and information technology, various applications are developed by different companies to solve various problem. Sometimes, a problem may be solved under the collaboration of different applications. In such cases, message passing and collaboration by APIs between different applications is necessary. However, many applications published by companies are not free and they are not open source. For example, a hardware testing environment usually consists of several applications to implement a test case. These different application are provided by different companies, they share little APIs even nothing, so message passing and collaboration by APIs between them is impossible though they share a common computer they are installed. People must to operate on them manually in a short time to implement the test case. Another example is editing documents that consist of many pictures. These picture may be produced by different graphical applications. Operating on different graphical applications manually may be error prone. So, if there is a method that implement automatic operations between different graphical applications, the document editing will be easy to be implemented. In this paper, we propose an implementation method of automatic switch and operations between different applications. It takes Windows applications as an example to exam the method and we implement the method by VB.NET[1][2][3].

2. Background

Windows OS has a complete mechanism of message sending and receiving. Key press, single click, double click, release of a key of keyboards or mouse will be viewed as events which will produce messages in confirmed form. These messages will be grouped into a queue with other messages. Applications will look for these messages in the message queue and assign them to appropriate windows. There will be a loop for the process to continue until the message queue is empty. Each window has a corresponding process to receive messages and respond to them. The return value of each response will be send to Windows OS and will be called by Windows OS or Modules outside Windows OS[1].

PeekMessage() or GetMessage() is usually called to get messages from message queues. Message queues are grouped by threads. Each thread has its own message queue. The messages in the queue is send to the application which the thread correspond to[1].

SendMessage() and PostMessage() are usually called to send message to windows. Message are translated and processed by SendMessage(), and the results will be returned once the process is completed rather than be added into the message queue. PostMessage() responds to add messages into the message queue and returns “Ture” if it succeeds, otherwise returns “False” if it fails.

3. Application Automatic Switch and Operation

In order to implement automatic switch between different applications and automatic operation of application, we implement a method, which is, programming for background applications based on Windows OS. The programs call handlers of different applications to record the position coordinates of all control windows and simulate users to switch automatically between different applications and operate on applications. And it calls HOOK() to set up special instructions to execute functions such as pause and end. In Fig.1, each application complete its own function, which executes different functions such as setting up or operating on hardware. To make these background programs which make the switch automatically and simulate various operations of mouse or key press of keyboards[4][5].

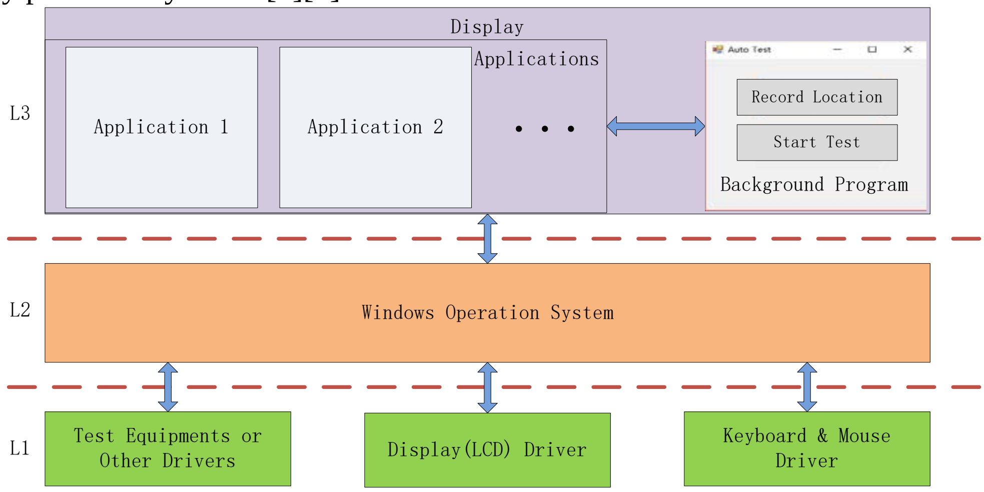

Figure 1 Keyboard and mouse monitor software diagram

3.1. Application Switch

To switch applications, at first, handlers of applications and their window’s positions are recorded, then handlers are switched and operations of mouse are simulated. In the process of switching, some APIs of Windows OS are called. If an application is selected by mouse, GetCurrentThreadID () will be called to obtain the current thread’s ID, and then move the cursor to the windows or buttons which need to be recorded and record the position of the cursor by calling GetCursorPos (), finally, the current application is switch to another application and the process of switching is recursive. In detail, SetForegroundWindow() is called to set the selected application as the front windows, SetCursorPos () is called to set the position of cursor in the target region, AttachThreadInput() is called to manage the input queue, and background programs simulate various operations of mouse or key press of keyboards.

3.2. Background Programs Which Simulate Operations of Mouse or Key Press of Keyboards

Once the target application is switched to, background program must simulate the operation of key press. In detail, set the target window of the target application as the accept window for the press of keyboard or mouse button at first, and then send various output of keyboard press or mouse operation to the accept window, finally, if there is a response of the target application to these outputs, the application will be switched to another application and the process will be looped until all simulated operations of the mouse and keyboards are implemented.

In the process of sending operations that are simulated to the target accept window, key presses to be send are encapsulated into SendMessage() as the parameter wParam, while the operations of key presses are encapsulated into PostMessage() as the parameter lParam. So, the key presses and operations of them to be send can be send to the target accept window waiting for the following acceptance and response of the window. As illustrated in Section 2, if SendMessage() is called, real-time operations of user will not be respond until all simulated operations are respond. While if PostMessage() is called, the response of simulated operations will be interrupted by real-time operations.

Once the background program works, if PostMessage() is called to send the message of simulated operations of key press, real-time operations of user could be responded by the background program. Some hot keys are configured to represent special instructions which implement pause or end some operations. HOOK() is used to catch real-time operations of user.

4. Experimental Analysis

In the proposed method, it is easy to implement automatic switch between different applications and automatic operations of key press instead of manual ones. This method is suit to simulate operations of different applications and key press of keyboard and mouse. Thus, if a rule of operations of applications and key press of keyboard and mouse can be concluded, this method can be used to implement these operations automatically instead of manual operations. And this method can be applied to many kinds of applications as long as we update the code of the background program.

Take hardware testing in Section 1 as an example, at first, the two applications of Matlab and Labview are setup in Windows OS. Then, edit the background program by VB.NET. To make it is easy for user to execute the hardware testing, we designed a window for the running of background program, which is consist of “Record Location” button, “Start Test” button and so on. The background program implements catching handlers of applications and position of window by the “Record Location” button, and makes the applications run and automatically switch between them and simulates operations of keyboards and mouse by the “Start Test” button[6].

The program flow of the background program we implemented can be shown as Fig.2. Once Matlab and Labview begin to run, the background program will be invoked at once also. Users need to click the “record position” button, and a dialog reminding users to move the cursor onto the icon button of Labview will be pop up. The background program will record the handler of current application and the coordinate of the button. And then the cursor will be switched to the window of Matlab, and the background program will record the similar data for the application of Matlab. After done, users need to click the “begin testing” button, the background program will switch the handler of current application to Labview by simulating operations of keyboards and mouse. After test data are obtained by running Labview, it will switch the handler of current application to Matlab again by simulating operations of keyboards and mouse. And Matlab will calculate and analy the test data by automatic operations inducted by the background program. Finally, the analysis of the test data and data of test environment will be generated and saved in an Excel file, a test case is completed by the background program. The process is recursive until all test cases are executed automatically. However, there is point that is it usually spends must time to exeute automatic testing as we inferred. If an exception occurs during testing, it is necessary to pause or end the testing by some special hot key. So it’s better to invoke HOOK() in the background program.

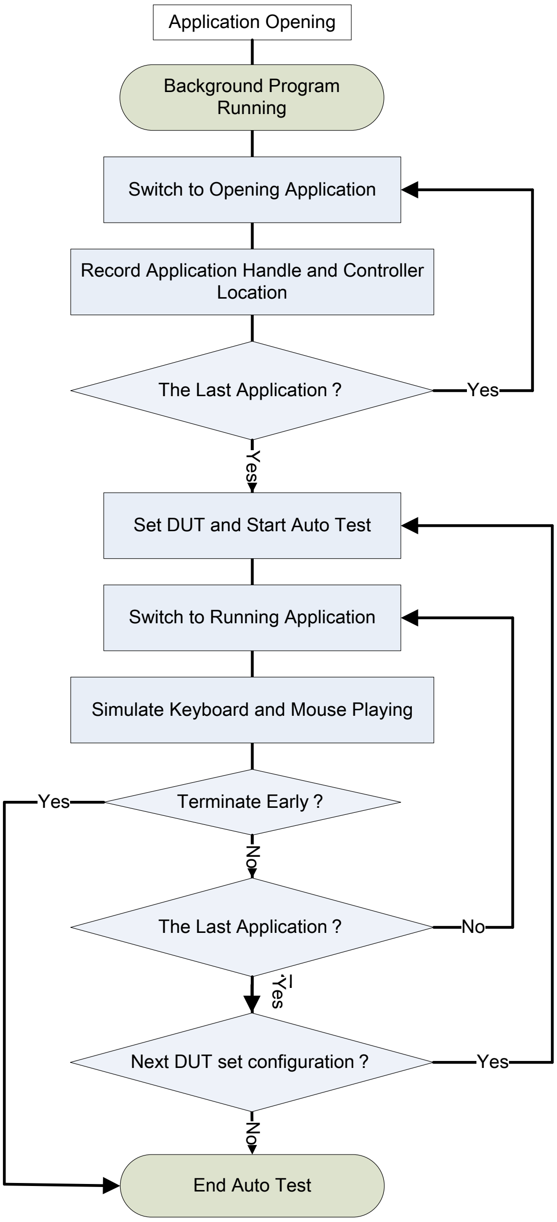
Application Opening
Background Program Running
Switch to Opening Application
Record Application Handle and Controller Location
The Last Application ?
Yes
Yes
Set DUT and Start Auto Test
Switch to Running Application
Simulate Keyboard and Mouse Playing
Terminate Early ?
Yes
No
The Last Application ?
No
Yes
Next DUT set configuration ?
Yes
No
End Auto Test

Figure 2 Background program flow

5. Conclusion

Since the method we implement could simulate operations of users, it is unnecessary to obtain the source code of applications which are objects of operations. In addition, APIs of Window OS is not necessary for the simulation either as long as the keyboard and mouse is available. Thus, it is easy for the simulation method proposed in this paper to implement automatic testing, documents editing, etc. which need many manual operations. And we believe it is a way to improve work effectiveness for running such applications.

Acknowledgements

This work has been supported by the Discipline of Computer Science and Technology of Shanghai Polytechnic University (Grant No.XXKZD1604).

References

[1] Pavel Yosifovich, etc. (2018) Windows Internals. POSTS&TELECOM PRESS.

[2] Jeff Martin. (2014) Visual Studio 2013 Cookbook. Packt Publishing.

[3] Wang, Z.X. (2016) Study of an Innovative Indoor Robotic Navigation Approach Based on Beacons and PSD. MATEC Web of Conferences 59 05003, DOI: 10.1051/matecconf/20165905003.

[4] Wang, L., Zhou, X.S., Jiang, Z.J., etc. (2012) A Real-Time Process Scheduling Policy in Windows. International Conference on Computer Science & Service System, DOI: 10.1109/CSSS.2012.14.

[5] Gupta, R.G., Reddy, R.S., Parmar, N.N. (2016) Virtual Keyboard Using Image Processing. International Research Journal of Engineering and Technology, 3(4), 2009-2013.

[6] XU F.X. (2011) Adopting Windows API to Draw Waveform Based on the LabVIEW Platform. MICROCOMPUTER INFORMATION, 27(12), 123-125.