Research of Mobile Applications Automated Testing Using Uiautomator

Wang Xiu Ming, Huang Xin, Li Gui Zhi, Miao Cui Ping, Shen Ning*

Beijing Software Testing&QA Center, National Application Software Testing Labs, Beijing 100191, China.

wangxm@bsw.net.cn

Keywords: Android, Uiautomator, automatic test.

Abstract. With development of the mobile technology, the quantity of the mobile application is also growing, and how to ensure the quality of the application has become a hot topic. Android is open platform for mobile devices, more and more developers like to develop application based on the Android system, which makes the quantity of the Android application is growing .This paper introduces the automatic testing of Android with Android Uiautomator, and describes the process of automatic test. The test result is analyzed, and the advantages of Android application system are described.

1.Introduction

With the development of intelligent mobile device technology, and which is gradually changing our lives, more and more people are using mobile phones or panel computer and other mobile devices to obtain information. How to ensure the quality of mobile application has become a hot topic. Testing of mobile applications is different from traditional software testing : 1) A wide variety of operating systems such as Android, IOS, Symbian and Windows Phone [1], as well as some operating systems based on the secondary development of the Android system; 2) There are some differences in equipment model, screen size and languages, etc. To solve the above two problems, the most companies have come up with a solution of test automation to the problem, it [2] not only improve the efficiency of the test, but also to some extent can reduce testing costs. And at the same time, the testing engineer can relieved from the heavy repetitive work.

Since Android is open source platform framework, many manufacturers based on Android system for secondary development. Many people like to using android, due to the characteristic of low price. In recent years, according to data statistics in China Android market share of up to 74%, and this paper mainly analyzes the present situation of Android application automation testing. There are several automation tools: 1) Monkey: it is a command line tool, the testing principle is using socket communication way to simulate the user's keystroke system, and send a pseudo random of user events to be measured flow [3], can achieve a stress test on the application being developed; 2) MonkeyRunner: it use of Python language. Support recording function, and provide API, use this API to code programs to control Android devices and simulators. But the main drawback to get UI elements using the coordinate, since can get less UI elements it logic is relatively poor; 3) Appium:it is open source, cross-platform test framework. It can be used to test the native and hybrid mobile terminal application. Appium supports IOS, Android and FirefoxOS platform, and a variety of languages, such as Java, python and ruby, etc; 4) Robotium [4]: repackage the Instrumentation framework, it can only test on single application. And need to test application with the same signature; 5) Calabash: test Android and IOS native and hybrid applications [5]. There are mainly Calabash-Android and Calabash-IOS. Calabash-Android supports Android's UI automated testing framework, PC-end use the cucumber testing framework, test apk install on real machine communicate via http and json with the simulator, and testing apk can call robotium's API to UI automation testing, as the same time it also support webview's operation; 6) TMTS (Taobao Mobile Test System): Alibaba develop Android browser application, it is lightweight open source automation test framework. TMTS draws Robotium and Selenium's Android driver [6]. 7) Uiautomator: Google launched simultaneously in Android4.1 UI testing framework, and it is compatible with JUnit.

Synchronized with the Android version of upgrade, after several iterations has been stable at present. Compared to MonkeyRunner and Robotium, Uiautomator to provide the API easier to use. Bascially supported all Android event actions, event operation does not depend on the control coordinates. It can verify the correctness by assertion and screenshots. At the same time Uiautomator is not subject to the restrictions of the signature and source code, testing engineer can use black box testing. In summary, the test code is written faster, run simple, once compiler the code can run on all Android devices.

2.Introduction Uiautomator framework API

The whole framework of Uiautomator is very clear, this section mainly introduces the API class.

2.1 UiDevice class

The UiDevice class contains two types of API, obtain devices status information and simulate user opertaion. And the coordinate method of the UiDevice class can to complete some simple testcase.When test apk can calling the UiDevice instance of the method to obtain the different properties of device state, such as rotation direction of the screen, screen size,ect. UiDevice also perform device-level operations, such as the direction of the mandatory device screen, press the screen and all kinds of buttons, etc.

UiDevice is Singleton pattern. There are two ways to obtain an instance, UiDevice.getInstance() and getUiDevice ().

Function description:

1) Get the device information: screen resolution, rotation state, screen status;

2) Operation: button, coordinate operation, slide and drap the screen, screenshots, etc.;

3) Monitor function: for processing the script is interrupted.

2.2 UiSelector class

Represents a query for one or more target UI elements on a device screen. UiSelector class represents a search condition, can get the page to find a specific element of the handle in the current, if the result found contains more than one matching element, the layout of the hierarchy of the first element is returned as target UiObject. When constructing a UiSelector class usually using multiple properties to reduce the search range. Positioning components through various attributes and nodes.

UiSelector automatic operation steps is to locate the target object, and then operate the object, as shown in figure 1.

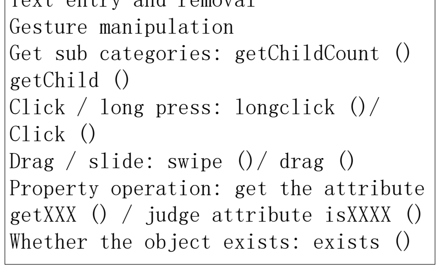
2.3 UiObject class

UiObject class represents a UI element, through the UiSelector class to find conditions to create an instance of UiObject, after the UI instance is found, and then to carry out a variety of simulation operations.

In the UiObject class, there are many methods and properties for simulating the actual operation of a mobile phone, as shown in figure 2.

Locate the target object(UI)
Operate the object(UI）

Fig. 1 UiSelector automation operation



```
Text entry and removal
Gesture manipulation
Get sub categories: getChildCount ()
getChild ()
Click / long press: longclick ()/
Click ()
Drag / slide: swipe ()/ drag ()
Property operation: get the attribute
getXXX () / judge attribute isXXXX ()
Whether the object exists: exists ()
```

Fig. 2 attribute of the UiObject class

2.4 UiCollection class

The UiCollection class is a subclass of UiObject, which represents a collection of elements.

Function description:

1) according to certain conditions enumerated container class interface all eligible sub-elements;

2) collection from qualifying element, again through a certain search criteria to locate the target final components.

UiColletcion usage scenarios:

1) Generally used as a parent class container;
2) Generally used in the need to find a subclass and subclass difficult to locate;
3) Get the number of a certain type of collection.

2.5 UiScrollable class

Provides support for searching for items in a scrollable UI container. UiScrollabe is a subclass of UiCollection. Figure 3 shows the inheritance relationship UiScrollabe, UiCollection and UiObject three.

Main method: forward and backward rolling, rapid rolling, rolling to a certain object, set the rolling direction, set the number of rolling, etc.

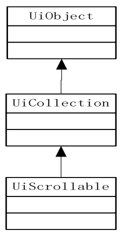


Fig. 3 inheritance relationship

2.6 UiWatcher class

The UiWatcher class is used to process the exception (not expected) in the execution of the script.

Application scenarios:

a) The telephone rang during the test.
b) Testing process receives the message.
c) Alarm clock in the process of testing.
d) Other interrupt during the test.

As that happens in the implementation process, the need to use UiWatcher class to handle exceptions, and return to the scene of the original test script.

Use the UiDevice.registerWatcher () method to register the UiWatcher class in UiAutomator.

3. Testcase analysis

The author takes the Yinxiang notes APP (abbreviation note APP) as the test target, studies the application of UiAutomator automation testing framework. The step of testcase are as follows:

a) Location notes APP
b) Add a new record
c) In the new recording process of telephone interruption test
d) New record back to the home page

3.1 Prepare the test environment

1. Install JDK Development Tools
2. Install the Android SDK development tools
3. Install the simulator Genymotion
4. Configure environment variables: JDK, adb (Android Debug Bridge)
5. Install notes APP test package

ADB is a command line tool for the C/S architecture, which is easy for Android developers to use ADB commands to operate a mobile phone or emulator [7]. Figure 4 using adb install command to install test APP.

Composition of adb:

- ➢ Client: run on the PC-end, can install, uninstall and debugging of Android application.
- ➢ Service: run on the PC-end, the connection management client to the Android device on the ADB background process.
- ➢ ADB daemon: ADB daemon running on Android devices.

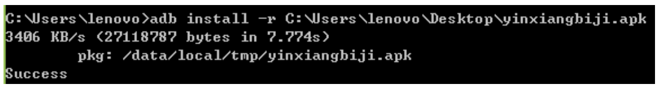
Genymotion simulator used in this case instead of real mobile devices to run the demo script result, Genymotion is a complete tool that provides the Android virtual environment to support most of the simulator function.

3.2 UiAutomator development kit

Google provided uiautomator.jar and android.jar located in <Android SDK installation path> \ sdk \ platforms \ android-19.

3.3 Control identifier: uiautomatorviewer

Before you start writing test cases, use the Android SDK comes uiautomatorviewer tool (located in <Android SDK installation path> \ sdk \ tools), it helps tester to be familiar with the various components(including the views and controls) of layout level [8]. Uiautomatorviewer tool can display any UI screen snapshot that is connected to the Android device. Figure 5 shows the UI in the uiautomatorviewer tool APP screen snapshot.



```
C:\Users\lenovo>adb install -r C:\Users\lenovo\Desktop\yinxiangbiji.apk
3406 KB/s (27118787 bytes in 7.774s)
        pkg: /data/local/tmp/yinxiangbiji.apk
Success
```

Fig. 4 command to install the APK

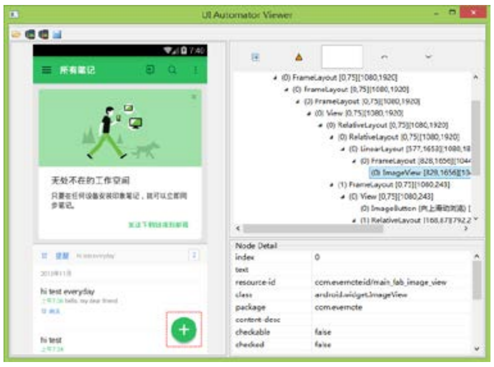


Fig. 5 uiautomatorviewer tool

3.4 Test script

When create a new test scripting, need to create a java Project, and add uiautomator.jar and android.jar to referenced libraries of the project. Part of the test script as follows:

Register Listener:

```
UiDevice.getInstance().registerWatcher("phone", new UiWatcher() {  @Override
 public boolean checkForCondition() {
 UiObject call = new UiObject(new UiSelector().text("call"));
 UiObject view = new UiObject(new UiSelector().className(""));
                if (call.exists()) {
      System.out.println("####call#### ");
      try { view.swipeLeft(20);
                 return true; }
     catch (UiObjectNotFoundException e) {
                 e.printStackTrace();}
     }
    return false;}});
```

Location notes APP:

```
UiObject app = new UiObject(new UiSelector().descriptionMatches("应用"));
  app.click();
  sleep(2000);
  UiObject note= new UiObject(new UiSelector().text("Yinxiang"));
  note.click();//select note APP
    sleep(3000);
```

Add a new record:

```
UiObject add = new UiObject(new UiSelector().className("android.widget.ImageView").index(0));
    add.click();//
    sleep(3000);
    UiObject textnote=new UiObject(new UiSelector().text("text notes"));
    textnote.click();//text notes
    sleep(3000);
    UiObject addtitle=new UiObject(new UiSelector().text("notes title"));
```

```
        UiObject addcont=new UiObject(new UiSelector().text("write your notes"));
        addtitle.setText("hello this test");
        sleep(1000);
        addcont.setText("hi,the first context");
        sleep(1000);
        UiObject addbutton = new UiObject(new UiSelector().className("android.widget.ImageView").index(0));
        addbutton.click();

   }
```

3.5 Execution script

To execute the script using the ADB command line, the main steps are as follows:

1.create create uitest-project -n <jars> -t -p 1 <project workspace> build:android

2. modify the build file: project node under the build as "default"

3. start compilation: -buildfile <build.xml ant file path >

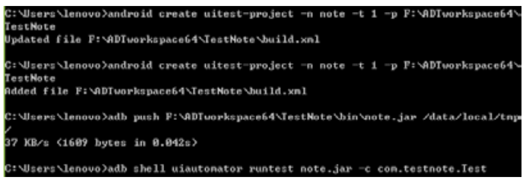
4.push file: push <path_to_out_jar> /data/local/tmp/ ADB

5. Run script: adb shell uiautomator runtest <jars> -c <package name> <class name> [# testname].

Figure 6 shows uses the ADB command to execute the com.testnote.Test.

3.6 Analysis of test results

Figure7 show the execution of the script: the number of uescase(numtests), the flow of information and error(stream), operational framework (id), execution time, package and class name. The state information INSTRUMENTATION_STATUS represents the state of operation, before running, ru-
nning and running end, and state codes 1,0 and -1. In practical applications, various state informati-
on can be customized according to requirements.



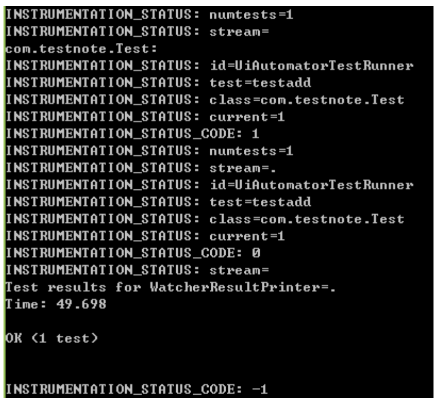
```
C:\Users\lenovo>android create uitest-project -n note -t 1 -p F:\ADTworkspace64\
TestNote
Updated file F:\ADTworkspace64\TestNote\build.xml

C:\Users\lenovo>android create uitest-project -n note -t 1 -p F:\ADTworkspace64\
TestNote
Added file F:\ADTworkspace64\TestNote\build.xml

C:\Users\lenovo>adb push F:\ADTworkspace64\TestNote\bin\note.jar /data/local/tmp
/
37 KB/s (1609 bytes in 0.042s)

C:\Users\lenovo>adb shell uiautomator runtest note.jar -c com.testnote.Test
```

Fig. 6 execute the com.testnote.Test



```
INSTRUMENTATION_STATUS: numtests=1
INSTRUMENTATION_STATUS: stream=
com.testnote.Test:
INSTRUMENTATION_STATUS: id=UiAutomatorTestRunner
INSTRUMENTATION_STATUS: test=testadd
INSTRUMENTATION_STATUS: class=com.testnote.Test
INSTRUMENTATION_STATUS: current=1
INSTRUMENTATION_STATUS_CODE: 1
INSTRUMENTATION_STATUS: numtests=1
INSTRUMENTATION_STATUS: stream=.
INSTRUMENTATION_STATUS: id=UiAutomatorTestRunner
INSTRUMENTATION_STATUS: test=testadd
INSTRUMENTATION_STATUS: class=com.testnote.Test
INSTRUMENTATION_STATUS: current=1
INSTRUMENTATION_STATUS_CODE: 0
INSTRUMENTATION_STATUS: stream=
Test results for WatcherResultPrinter=.
Time: 49.698

OK (1 test)


INSTRUMENTATION_STATUS_CODE: -1
```

Fig. 7 test result

4. Summary

In this paper, through an example introduce Uiautomator testing framework. Although Uiautomator have many advantages such as the creating UI testcases efficiently, no signature, but also have disadvantages, access control, more similar to the highly privileged operations. To solve these problems, Uiautomator constantly perfected. There is no tool based on the Uiautomator framework, and then does not have the script recording function, this is also the future Uiautomator needs to enhance the space.

Acknowledgments

Wang Xiu Ming (1982-) , Intermediate project manager,main research area: software testing.

Huang Xin(1991-) Test engineer.

Li Gui Zhi(1973-) MA.Eng, Senior project manager.

Miao Cui Ping(1981-)Test engineer.

Shen Ning(1986-) MA.Eng, Test engineer.

References

[1] Song Chunyu, The research and practice of automation test on Android platform, D. Beijing Jiaotong University. 2012

[2] Wang Wentao, Design and Implementation of Automated Software Testing Based on Android Phone, D. Beijing Jiaotong University.2015

[3] WANG Yan, Continuous Integration Based Study on Automated Testing for Android Platform, J. Computer Systems & Applications. 2015, 24(5): 261-266

[4] ZHU Yang-yang, HOU Yong-hong, WANG Bao-liang, Application of automatic test tool Robotium for Android, J. Information Technology.2015(10): 198-205

[5] Lin Ling, Research on the design of mobile phone software testing framework .Fujian Computer, J. 2013,3: 123-125

[6] Xu Fang, Research on automatic test technology of mobile application software, J. Electronic Technology & Software Engineering. 2015,18:63-65

[7] Lin Xiaojie. Research and Implementation of Automated Testing Platform Based on Android , D. South China University of Technology.2013

[8] Zhang Enhai, Wang Duo, Yu JinXuan, Design and implementation of automatic test system for mobile terminal application layer software , J .Science & Technology Information.2015,3:12-14