

Android Robotic Process Automation

Denis Leskovar

Program specification

Robotic process automation (RPA) is a term for means of performing complex actions automatically. Tools for RPA are popular on desktops, allowing users to automate routine tasks. However, in the Android setting, here is no such tool.

The Android RPA tool should be standardized in terms of its input. As such, it will support the WAW format described in detail [here](#).

The scope of the project can become very large if not kept in check. In order to avoid any complications, the project will be split into two parts — an RPA library and a WAW-to-Java transpiler.

WAW format

In short, the WAW format is a declarative way of specifying *what* action should be performed in which state (*where*). The format specifies many additional web specific keywords, such as selectors and cookies. These specifics will be ignored when working on the Android RPA.

The documentation of the project will present the user with all possible ways of specifying a state (*where*). Actions (*what*) should work the same as in the described format.

UI testing

Automating tasks on Android, either in an emulator or on a real device, requires leveraging the accessibility features of the operating system. Using these, we can read text fields on the screen, extract information present in the UI controls and interact with them.

Luckily, there are wrappers that allow for interacting and extracting data from the UI. For this project, we will be using Google's UI Automator, which is a tool distributed with the Android Studio IDE. The UI Automator is described [here](#) and allows developers to write UI tests for their applications. In practice, it can be used to access and control any applications with sufficient accessibility features.

Using the tool, we want to build a library — an automation framework. The framework should allow developers to perform common actions easily, instead of using lower-level functions presented by the UI Automator. Moreover, the functionality offered by the library will be used by the WAW-to-Java transpiler in order to generate UI tests, which automate the workflow declared in the provided WAW files.

Functional requirements

When referring to *all UI controls*, we mean those controls that have accessibility fields, such as name, defined.

1. The RPA library must allow the user to search for an UI control (button, text field, checkbox, ...) based on the control's display name.
2. The RPA library must allow the user to search for an UI control based on the its `AutomationId`.
3. The RPA library must allow the user to search for an UI control based on the its absolute position on the screen.
4. The RPA library must allow the user to search for an UI control based on the its relative position on the screen.
5. The RPA library must allow the user to search for an UI control based on the its ordering on the current screen.
6. The RPA library must allow the user to extract the text value of a text field.
7. The RPA library must allow the user to extract the logical value of a checkbox.
8. The RPA library must allow the user to extract the selected value in a combo box.
9. The RPA library must allow the user to extract an image from an image view.
10. The RPA library must allow the user to extract display names of all visible UI controls.
11. The RPA library must allow the user to alter all editable UI content.
12. The RPA library must allow the user to press found controls.
13. The RPA library must allow the user to perform a press action on a given absolute position.
14. The RPA library must allow the user to perform a press action on the given relative position.
15. The RPA library must allow the user to swipe vertically and horizontally in both directions, starting from an absolute position.

16. The RPA library must allow the user to swipe vertically and horizontally in both directions, starting from a relative position.
17. The RPA library must allow the user to press any system button (menu, back, volume control, ...).
18. The WAW interpreter must parse a given WAW file and generate an imperative UI test performing the declared workflow.
19. Should the format be faulty, the WAW interpreter must report any encountered errors.
20. Should the RPA library not support a given action, the WAW interpreter must report this incompatibility.
21. The UI tests generated by the WAW interpreter must work in the emulator provided by the Android Studio IDE, running on version Android 11 (API level 30).