Automation Test Framework

Proof of Concept Proposal

Authors
Carlos Alberto Souto Junior
Daniel Barboza Franco
Daniel Drigo Pastore
Marcel Augusto Gorri

October/2010

Author	Version	Description
Carlos Alberto Souto Junior Daniel Barboza Franco Daniel Drigo Pastore Marcel Augusto Gorri	1.0	Initial Version

1 - Motivation

In order to nurture the further development and improvements to the ATF and GEL specifications, it would be interesting if a initial implementation, comprising a reduced set of commands and a well constrained scenario, to be performed as a proof of concept of the Automation Test Framework design.

2 - Proposed Commands to be implemented:

`Type(String input)`

This command enters the text described by the input argument as it was typed on a standard keyboard

`Click(int x,int y)`

This command performs a click event on the designed x,y coordinates, given in pixels

`Assert(String filename)`

In this command, the presence of the file given by filename is checked. Filename can be the full path, represented in UNIX path format.

3 - Extension Point Schema and Handler

Every command should extend an specific extension point, the TestCommand, which in turn has the following fields that must be filled:

`commandId`: The internal command's Id

`alias`: Describes the command name to be used on GEL

`handler`: the command handler in charge of providing the actual implementation of the command.

The handler class should implement the Interface *ITestCommand*, which uses two methods:

`run()`:

- this method start the execution of the command, performing all necessary internal steps required by it.

`Report()`:

- this method returns the command execution status report, giving information pertaining to the command procedure(e.g. Image matching confidence level)

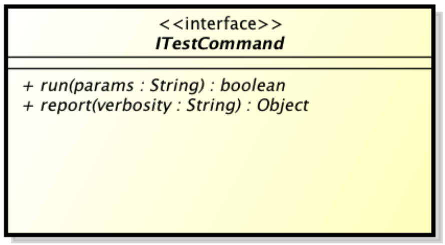
<<interface>>
ITestCommand
+ run(params : String) : boolean
+ report(verbosity : String) : Object

Figure 1: Class Diagram for the ItestCommand Interface

4 - Sample Test Script

In this script, a single UI test is performed. For the sake of this test, some assumptions must be taken regarding the scenario:

- An Android Emulator instance must be open and running
- One application that creates a file must have been already run and a file with the name mytest.txt must have been created by the application.

By fulfilling these conditions, the following script can be ran:

```
click(50,30)
type("http://www.eclipse.org/sequoyah")
assert("/data/mytest.txt")
```

The expected result is the text given by the type command being searched using the google service provided in the main screen of the emulator, and then the presence of the described file being verified.