

The Robot Framework and a Related Testing Workflow

Kinga Andrea Gémes

3 December 2020

Table of Contents

- 1 Robot Framework
 - Building blocks
 - Conclusion

- 2 Using Robot Framework at NNG in 2017
 - Testing Workflow with Robot Framework at NNG
 - Conclusion

Robot Framework

- Has been introduced in 2011 in the paper *Usage of Robot Framework in Automation of Functional Test Regression* [SCH11]

Robot Framework

- Has been introduced in 2011 in the paper *Usage of Robot Framework in Automation of Functional Test Regression* [SCH11]
- **Goal:** Automate manual test processes, especially regression testing

Robot Framework

- Has been introduced in 2011 in the paper *Usage of Robot Framework in Automation of Functional Test Regression* [SCH11]
- **Goal:** Automate manual test processes, especially regression testing
- Python based tool with RIDE GUI

Robot Framework

- Has been introduced in 2011 in the paper *Usage of Robot Framework in Automation of Functional Test Regression* [SCH11]
- **Goal:** Automate manual test processes, especially regression testing
- Python based tool with RIDE GUI
- Easy, keywords based framework for quick test definition

Robot Framework

- Has been introduced in 2011 in the paper *Usage of Robot Framework in Automation of Functional Test Regression* [SCH11]
- **Goal:** Automate manual test processes, especially regression testing
- Python based tool with RIDE GUI
- Easy, keywords based framework for quick test definition
- Test cases built from keywords

Robot Framework

- Has been introduced in 2011 in the paper *Usage of Robot Framework in Automation of Functional Test Regression* [SCH11]
- **Goal:** Automate manual test processes, especially regression testing
- Python based tool with RIDE GUI
- Easy, keywords based framework for quick test definition
- Test cases built from keywords
- Test suites built from test cases

Robot Framework

- Has been introduced in 2011 in the paper *Usage of Robot Framework in Automation of Functional Test Regression* [SCH11]
- **Goal:** Automate manual test processes, especially regression testing
- Python based tool with RIDE GUI
- Easy, keywords based framework for quick test definition
- Test cases built from keywords
- Test suites built from test cases

RIDE

TCS2F185

Documentation

		Edit	Clear
--	--	------	-------

Setup

Clean Batch Data	Edit	Clear
----------------------------------	------	-------

Teardown

Clean Batch Log	Edit	Clear
---------------------------------	------	-------

Tags

	Edit	Clear
--	------	-------

Timeout

5 minutes	Edit	Clear
-----------	------	-------

Template

	Edit	Clear
--	------	-------

1	Transfer Batch	`\${SERVER_IP}`	`\${SERVER_USER}`	`\${SERVER_PW}`	
2	Check Batch				
3	Generate Include File				
4	Compile	SIP	SIP	SCSCF	
5	Compile	CAP	CAP_V4	test	
6	Run Protocol Simulator	CAP	test		
7	Run Protocol Simulator	SIP	SCSCF		
8	`\${OUT}`	Run TC	runme.cmd		
9	Should Contain	`\${OUT}`	TC run finished		
10	`\${OUT}`	Decode	SIP	SIP	SCSCF
11	Should Contain	`\${OUT}`	Sucesfully		
12	`\${OUT}`	Decode	CAP	CAP_V4	cap
13	Should Contain	`\${OUT}`	Sucesfully		
14					

Keyword definition

- New keyword written in python (or java)

Keyword definition

- New keyword written in python (or java)
- Already existing keywords

Keyword definition

- New keyword written in python (or java)
- Already existing keywords
- Defining keywords based on already existing keywords in the Robot Framework syntax

```
def FTP_Delete(self, host, user, pwd, file_remote):
    ftp = ftplib.FTP()
    ftp.connect(host, 21)
    try:
        try:
            ftp.login(user, pwd)
            ftp.delete(file_remote);
            return True
        finally:
            ftp.quit()
    except:
        traceback.print_exc()
        return False
```

Keyword definition in Robot

```
*** Test Cases ***
Withdraw From Account
    Withdraw From Account    $50
    Withdraw Should Have Succeeded

*** Keywords ***
Withdraw From Account
    [Arguments]    ${amount}
    ${STATUS} =    Withdraw From User Account    ${USER}    ${amount}
    Set Test Variable    ${STATUS}

Withdraw Should Have Succeeded
    Should Be Equal    ${STATUS}    SUCCESS
```

Building a test case

- Optional setup and teardown

Building a test case

- Optional setup and teardown
- Optional timeout time

Building a test case

- Optional setup and teardown
- Optional timeout time
- At least one passing criteria

Building a test case

- Optional setup and teardown
- Optional timeout time
- At least one passing criteria
 - ➔ Should Be True

Building a test case

- Optional setup and teardown
- Optional timeout time
- At least one passing criteria
 - ➔ Should Be True
 - ➔ Should Be Equal

Building a test case

- Optional setup and teardown
- Optional timeout time
- At least one passing criteria
 - ➔ Should Be True
 - ➔ Should Be Equal
 - ➔ Should Not Be Empty

Building a test case

- Optional setup and teardown
- Optional timeout time
- At least one passing criteria
 - ➔ Should Be True
 - ➔ Should Be Equal
 - ➔ Should Not Be Empty
 - ➔ [More on the Robot Framework website](#)

Building a test case

Test case	Action	Argument	Argument
Test_name	[Tags]	Test_tag1	Test_tag1
	[Setup]	Setup keyword	
	[Timeout]	10 minutes	
	Keyword1	$\${Global_Var1}$	$\${Global_Var2}$
	$\${New_Var1}$	Keyword2	
	Should Be True	$\${New_Var1}$	
	[Teardown]	Teardown keyword	

Building a test suite

- Optional test suite setup and teardown

Building a test suite

- Optional test suite setup and teardown
- Contains the test cases and specific keywords

Building a test suite

- Optional test suite setup and teardown
- Contains the test cases and specific keywords
- Set global variables

Building a test suite

- Optional test suite setup and teardown
- Contains the test cases and specific keywords
- Set global variables

```
*** Settings ***
Suite Setup      Set Active User

*** Variables ***
# Default system address. Override when tested against other instances.
${SERVER URL}   http://sre-12.example.com/
${USER}         Actual value set dynamically at suite setup

*** Keywords ***
Set Active User
    ${USER} =    Get Current User    ${SERVER URL}
    Set Suite Variable    ${USER}
```

Running tests

- From RIDE

Running tests

- From RIDE
 - ➔ Run individual test cases

Running tests

- From RIDE
 - ➔ Run individual test cases
 - ➔ Run the whole test suite

Running tests

- From RIDE
 - ➔ Run individual test cases
 - ➔ Run the whole test suite
- From command line

Running tests

- From RIDE
 - ➔ Run individual test cases
 - ➔ Run the whole test suite
- From command line
 - ➔ *robot* command with command line options

Running tests

- From RIDE
 - ➔ Run individual test cases
 - ➔ Run the whole test suite
- From command line
 - ➔ *robot* command with command line options
- Automatically generated test reports

Running tests - RIDE

Acc Test Report

Generated: 20110224 22:30:20 GMT +02:00
1 minute 24 seconds ago

Summary information

Status: **2 critical tests failed**

Start Time: 20110224 22:30:17.023
End Time: 20110224 22:30:20.032
Elapsed Time: 00:00:03.009

Test Statistics

Total Statistics				
	Total	Pass	Fail	Graph
Critical Tests	136	134	2	
All Tests	136	134	2	

Statistics by Tag				
	Total	Pass	Fail	Graph
No Tags				

Statistics by Suite				
	Total	Pass	Fail	Graph
Acc	136	134	2	

Test Details by Suite

Name	Documentation	Metadata / Tags	Crit.	Status	Message	Start / Elapsed
Acc			N/A	FAIL	136 critical tests, 134 passed, 2 failed 136 tests total, 134 passed, 2 failed	20110224 22:30:17 00:00:03
TCHD010			yes	PASS		20110224 22:30:17 00:00:00
TCHD015			yes	PASS		20110224 22:30:17 00:00:00
TCHD020			yes	PASS		20110224 22:30:17 00:00:00
TCHD025			yes	PASS		20110224 22:30:17 00:00:00
TCHD030			yes	PASS		20110224 22:30:17 00:00:00
TCHD035			yes	PASS		20110224 22:30:17 00:00:00
TCHD110			yes	PASS		20110224 22:30:17 00:00:00
TCHD111			yes	PASS		20110224 22:30:17 00:00:00

Running tests - Command line

```
=====  
Example test suite  
=====
```

```
First test :: Possible test documentation | PASS |  
-----
```

```
Second test | FAIL |
```

```
Error message is displayed here  
=====
```

```
Example test suite | FAIL |
```

```
2 critical tests, 1 passed, 1 failed
```

```
2 tests total, 1 passed, 1 failed  
=====
```

```
Output: /path/to/output.xml
```

```
Report: /path/to/report.html
```

```
Log: /path/to/log.html
```

Using Robot framework - Conclusion

Test step	Manual time	Automatic time
Preparation of one test case	8 hours	8 hours
Execution of one test case	2 minutes	2 minutes
Check of one test case	5 minutes	1 minute
Automation of one test case	-	2 hours
Report for one test case	3 minutes	-
Total time used for one test case	8 hours 10 minutes	10 hours 3 minutes

Using Robot framework - Conclusion

Test step	Manual time	Automatic time
One test run cycle	10 minutes	3 minutes
For 100 test cases - one suite run	16 hours 40 minutes	5 hours
20 suite runs	333 hours 20 minutes	100 hours
20 suite runs with automation time in- cluded	333 hours 20 minutes	300 hours

Using Robot framework - Conclusion

- Simple, easy to read syntax, non-programmers can learn to use it quickly

Using Robot framework - Conclusion

- Simple, easy to read syntax, non-programmers can learn to use it quickly
- Clean UI for quick test definition and execution

Using Robot framework - Conclusion

- Simple, easy to read syntax, non-programmers can learn to use it quickly
- Clean UI for quick test definition and execution
- Running a test suite all together or one test case at a time

Using Robot framework - Conclusion

- Simple, easy to read syntax, non-programmers can learn to use it quickly
- Clean UI for quick test definition and execution
- Running a test suite all together or one test case at a time
- Automatic test report generation

Using Robot framework - Conclusion

- Simple, easy to read syntax, non-programmers can learn to use it quickly
- Clean UI for quick test definition and execution
- Running a test suite all together or one test case at a time
- Automatic test report generation
- Well documented

Using Robot framework - Conclusion



Figure: Robot Framework's development is nowadays sponsored by Robot Framework Foundation

Using Robot Framework at NNG in 2017



Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make
 - ➔ Start up iGO

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make
 - ➔ Start up iGO
 - ➔ Configure settings

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make
 - ➔ Start up iGO
 - ➔ Configure settings
 - ➔ Step-by-step reproduction of the manual test

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make
 - ➔ Start up iGO
 - ➔ Configure settings
 - ➔ Step-by-step reproduction of the manual test
 - ➔ Evaluate the results

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make
 - ➔ Start up iGO
 - ➔ Configure settings
 - ➔ Step-by-step reproduction of the manual test
 - ➔ Evaluate the results
 - ➔ Stop iGO

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make
 - ➔ Start up iGO
 - ➔ Configure settings
 - ➔ Step-by-step reproduction of the manual test
 - ➔ Evaluate the results
 - ➔ Stop iGO
- Determine if the test case is a part of an existing test suite or if it should be in a new test suite

Testing Workflow with Robot Framework at NNG - Test case

- Define a manual test case
 - ➔ Goal of the test case
 - ➔ Pass and fail criteria
 - ➔ iGO version, map, language, and other settings
- Define the necessary arbitrary steps that a manual tester would make
 - ➔ Start up iGO
 - ➔ Configure settings
 - ➔ Step-by-step reproduction of the manual test
 - ➔ Evaluate the results
 - ➔ Stop iGO
- Determine if the test case is a part of an existing test suite or if it should be in a new test suite
- Determine the necessary keywords for the test case

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➡ Plan a route from X to Y

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z
- Automated description:

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z
- Automated description:
 - ➔ Set the current coordinates to X

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z
- Automated description:
 - ➔ Set the current coordinates to X
 - ➔ Add destination Y

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z
- Automated description:
 - ➔ Set the current coordinates to X
 - ➔ Add destination Y
 - ➔ Add waypoint in Z

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z
- Automated description:
 - ➔ Set the current coordinates to X
 - ➔ Add destination Y
 - ➔ Add waypoint in Z
 - ➔ Handle popup that may or may not appear

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z
- Automated description:
 - ➔ Set the current coordinates to X
 - ➔ Add destination Y
 - ➔ Add waypoint in Z
 - ➔ Handle popup that may or may not appear
 - ➔ Write new keywords when necessary

Testing Workflow with Robot Framework at NNG - Manual to Automated

How to transform a manual test to an automated test - case study on routing:

- Manual description:
 - ➔ Plan a route from X to Y
 - ➔ Add waypoint in Z
 - ➔ Passing criteria: the route from X to Y includes Z
- Automated description:
 - ➔ Set the current coordinates to X
 - ➔ Add destination Y
 - ➔ Add waypoint in Z
 - ➔ Handle popup that may or may not appear
 - ➔ Write new keywords when necessary
 - ➔ Check if the route description includes Z, fail if it does not.

Testing Workflow with Robot Framework at NNG - Test suite

- What should be in a test suite?

Testing Workflow with Robot Framework at NNG - Test suite

- What should be in a test suite?
 - ➔ Are the settings the same through the test suite?

Testing Workflow with Robot Framework at NNG - Test suite

- What should be in a test suite?
 - ➔ Are the settings the same through the test suite?
 - ➔ Does one of the test cases add or remove something that can influence the test cases after that (and do we want it)?

Testing Workflow with Robot Framework at NNG - Test suite

- What should be in a test suite?
 - ➔ Are the settings the same through the test suite?
 - ➔ Does one of the test cases add or remove something that can influence the test cases after that (and do we want it)?
- Why should we use test suites?

Testing Workflow with Robot Framework at NNG - Test suite

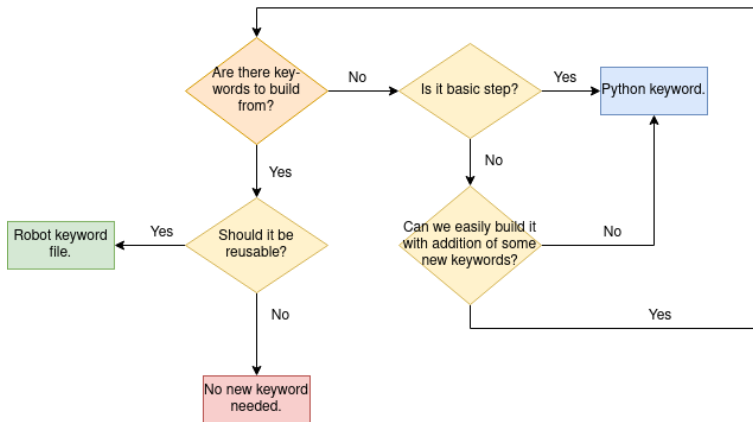
- What should be in a test suite?
 - ➔ Are the settings the same through the test suite?
 - ➔ Does one of the test cases add or remove something that can influence the test cases after that (and do we want it)?
- Why should we use test suites?
 - ➔ The test cases do not exist in a vacuum. A test suite can represent a flow of iGO usage in its natural, intended way.

Testing Workflow with Robot Framework at NNG - Test suite

- What should be in a test suite?
 - ➔ Are the settings the same through the test suite?
 - ➔ Does one of the test cases add or remove something that can influence the test cases after that (and do we want it)?
- Why should we use test suites?
 - ➔ The test cases do not exist in a vacuum. A test suite can represent a flow of iGO usage in its natural, intended way.
 - ➔ Time: starting up the iGO under test, configuring the parameters, stopping it all take up significant amount of time.

Testing Workflow with Robot Framework at NNG - Keywords

How to decide if you need keywords defined in python or in robot framework?



Test automation at NNG - final thoughts

- The positives

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python
 - ➔ Large amount of automated tests that rarely need update

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python
 - ➔ Large amount of automated tests that rarely need update
 - ➔ Thematic test suites

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python
 - ➔ Large amount of automated tests that rarely need update
 - ➔ Thematic test suites
- The negatives

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python
 - ➔ Large amount of automated tests that rarely need update
 - ➔ Thematic test suites
- The negatives
 - ➔ Very specialized keywords needed - builtin keywords rarely used

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python
 - ➔ Large amount of automated tests that rarely need update
 - ➔ Thematic test suites
- The negatives
 - ➔ Very specialized keywords needed - builtin keywords rarely used
 - ➔ Keywords organized by areas, lots of overlaps

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python
 - ➔ Large amount of automated tests that rarely need update
 - ➔ Thematic test suites
- The negatives
 - ➔ Very specialized keywords needed - builtin keywords rarely used
 - ➔ Keywords organized by areas, lots of overlaps
 - ➔ Robot keywords rarely reused

Test automation at NNG - final thoughts

- The positives
 - ➔ Massive keyword base in python
 - ➔ Large amount of automated tests that rarely need update
 - ➔ Thematic test suites
- The negatives
 - ➔ Very specialized keywords needed - builtin keywords rarely used
 - ➔ Keywords organized by areas, lots of overlaps
 - ➔ Robot keywords rarely reused
 - ➔ No actual company-wide rules for keyword definition

Thank you for your attention. 😊



Stanislav Stresnjak, Siemens Cmt, and Z. Hocenski, *Usage of robot framework in automation of functional test regression*, ICSEA 2011, 2011.