MaTeLo

ALL4TEC

MODEL BASED TESTING SOLUTIONS

# APPLICATION NOTE

## HOW TO GENERATE SELENIUM TEST CASES WITH MATELO?

Author René Christian TUYISHIME

Abstract This Application Note explains how to generate selenium test cases with MaTeLo

Keywords MaTeLo, Testor, Test case, Script, Selenium, HTML

Version 1.0 – November 2012

# Table of contents

# Table of figures

# 1. Introduction

This application note aims to present how to generate Selenium test scripts with MaTeLo.

## 1.1 MaTeLo

MaTeLo is a tool chain developed by ALL4TEC (www.all4tec.net ) that implements the Model Based Testing approach. In user friendly environment, MaTeLo offers an integrated set of components and features to enable test engineers in designing their tests based on usage models. Then, the test cases are automatically generated from the model depending on the user selected test strategy.

In general, MaTeLo manages successfully test projects in the fields of automotive, railway or energy, IT by providing professional and valuable features to obtain a better test coverage and a higher engineering productivity.

## 1.2 Selenium

Selenium is an open source tool (http://seleniumhq.org/) which allows performing automated testing of web applications. It provides a test domain-specific language called “Selenese” to write tests in a number of popular programming languages, including C#, Java, Groovy, Perl, PHP, Python and Ruby. The tests can be run on different web browsers.

## 1.3 Why to generate Selenium test cases with MaTeLo?

Actually, the use of websites in different activities like online sales, company communication is essential. That means websites quality and reliability should be high.

Today, It’s important to have nominal use cases tested but also complex. Even, websites evolutions are directly linked to business. Thus, it’s important to reduce time to put in use while ensuring good quality.

MaTeLo as a Model Based Testing tool allows having high quality test plan which correspond to business objectives. In order to reduce time related to the execution and the maintenance of the test plan in a case of evolution, MaTeLo allows the generation of test cases understandable or executable automatically by many test automation tools.

In this application note, the generation of Selenium test cases by MaTeLo is presented.

To illustrate the generation of Selenium test cases, a simple example of MaTeLo model will be used:

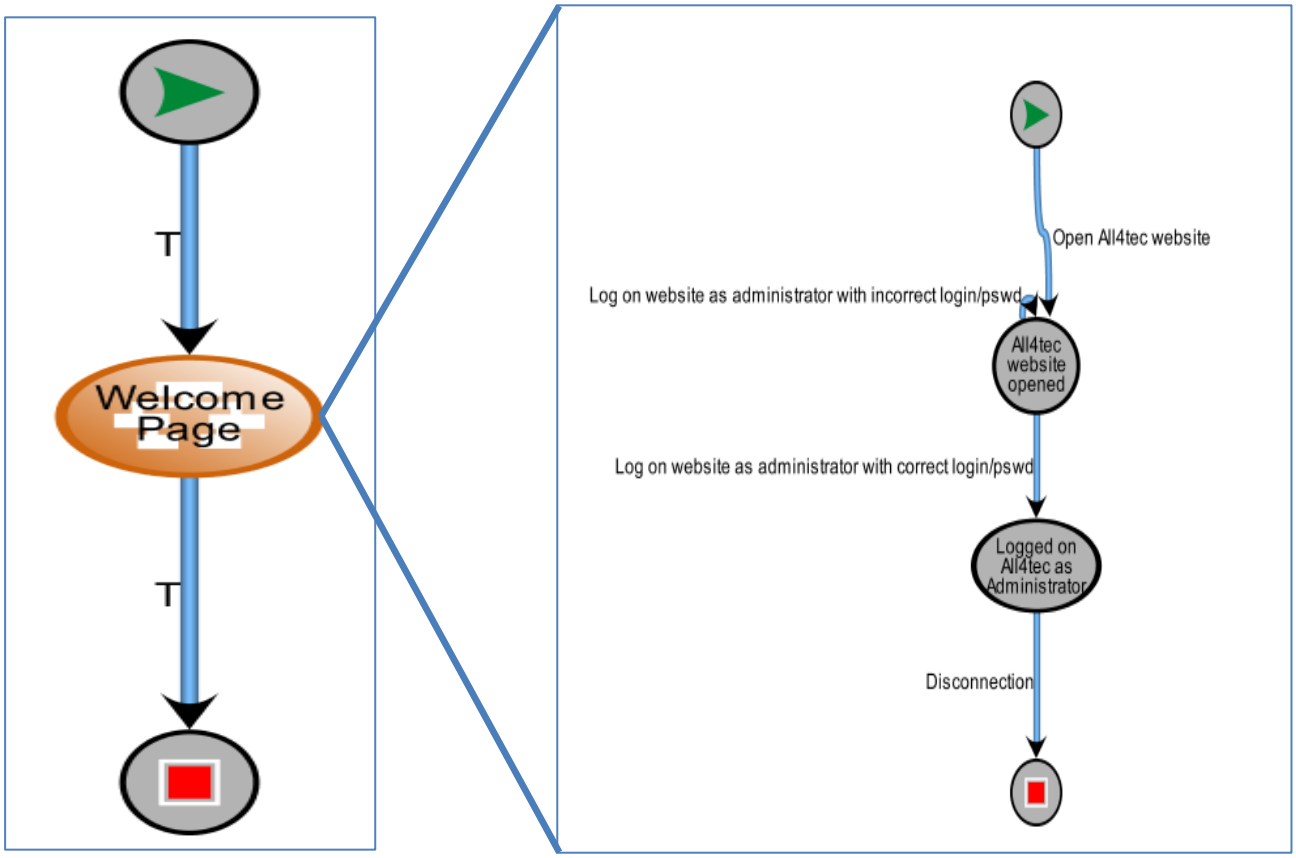



Figure 1: MaTeLo model

This MaTeLo model is composed of three simple actions: Opening of ALL4TEC website, log in as administrator and disconnecting to the website.

# 2 How to model a website with MaTeLo?

## 2.1 Basic architecture

The architecture of the MaTeLo model in website testing could be constructed, based on recommendations below:

- A macro chain shall represent a page of the website because different basic user actions can be performed on it. (see as example figure 1)
  When a user action opens another page, another macro chain in the model is reached.

  In other cases, macro chain can represent activities or functionalities (example: subscription, Modification of an offer or option…)
- A transition will represent basic user action (example: click on a link, open a page …)
- A state will represent a step in the model.

The usage profiles could be defined based on registered statistics about visits of the website.

# 3 Configuration of MaTeLo to generate SeleniumHQ test scripts

Selenium IDE is a suite of tools for web applications testing. A Selenium Test is a script which can be written in several languages (Java, HTML, C#...). Then, the right commands are sent to the browser.

Before making any automation action in MaTeLo, Test Framework properties have to be defined.

First of all, Selenium test framework must be configured in MaTeLo (Project > Configuration > Test Automation):

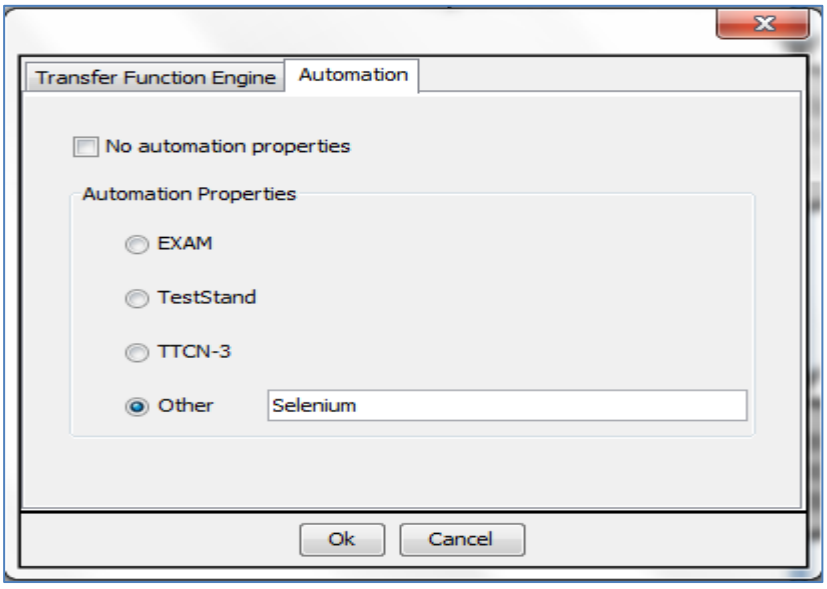



Figure 2: Configuration of Selenium test Framework

After setting up Selenium Test Framework, its properties must be defined:

- the format of the output file which will contain Selenium test scripts. Here, we are choosing html extension.

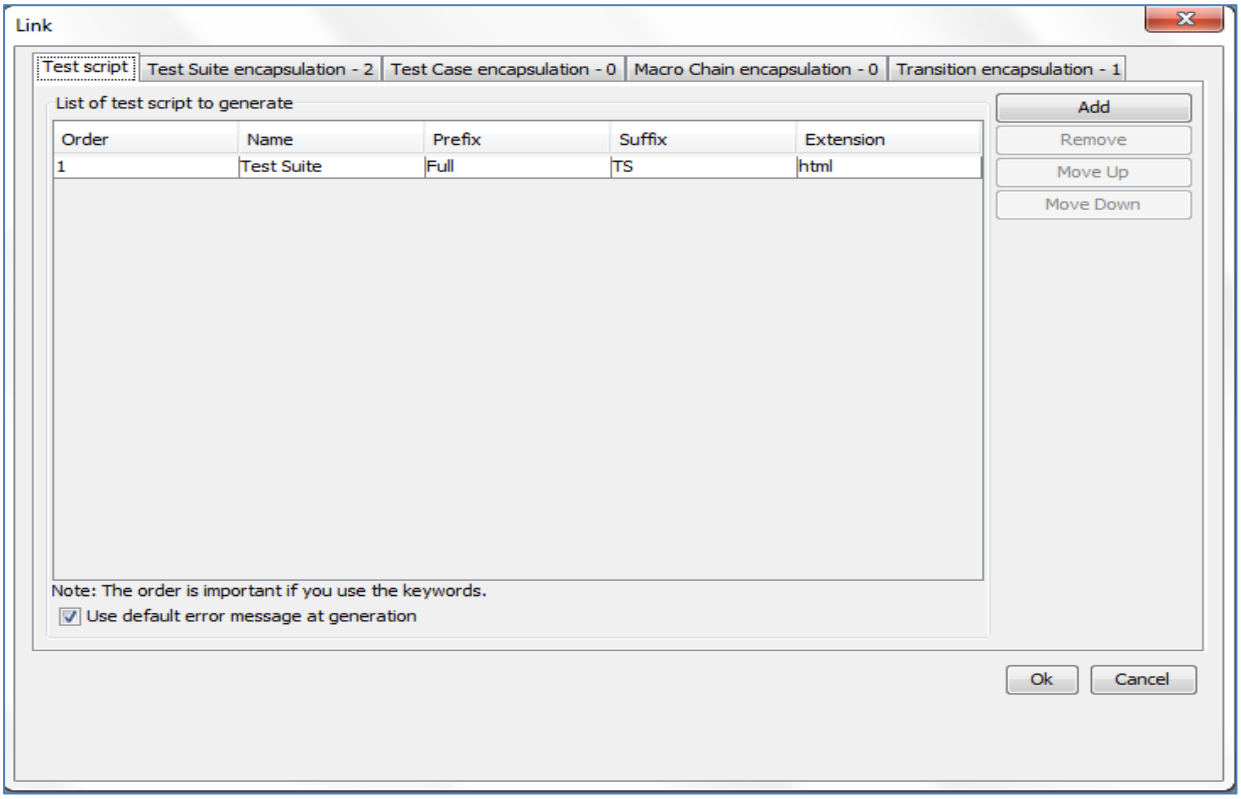



Figure 3: Test script definition

Python, Java or other kind of scripts can be generated by MaTeLo if needed.

MaTeLo allows encapsulation at different level (Test suite, Test case, Macro chain and Transition). For that, different functions which by convention start by “Adm” and used to deal with the construction of the test script structure and its automatic insertion in Selenium test tool are encapsulated at different levels. For example, the test function like” header” and “footer” for the output file can be defined and encapsulated at test suite level.

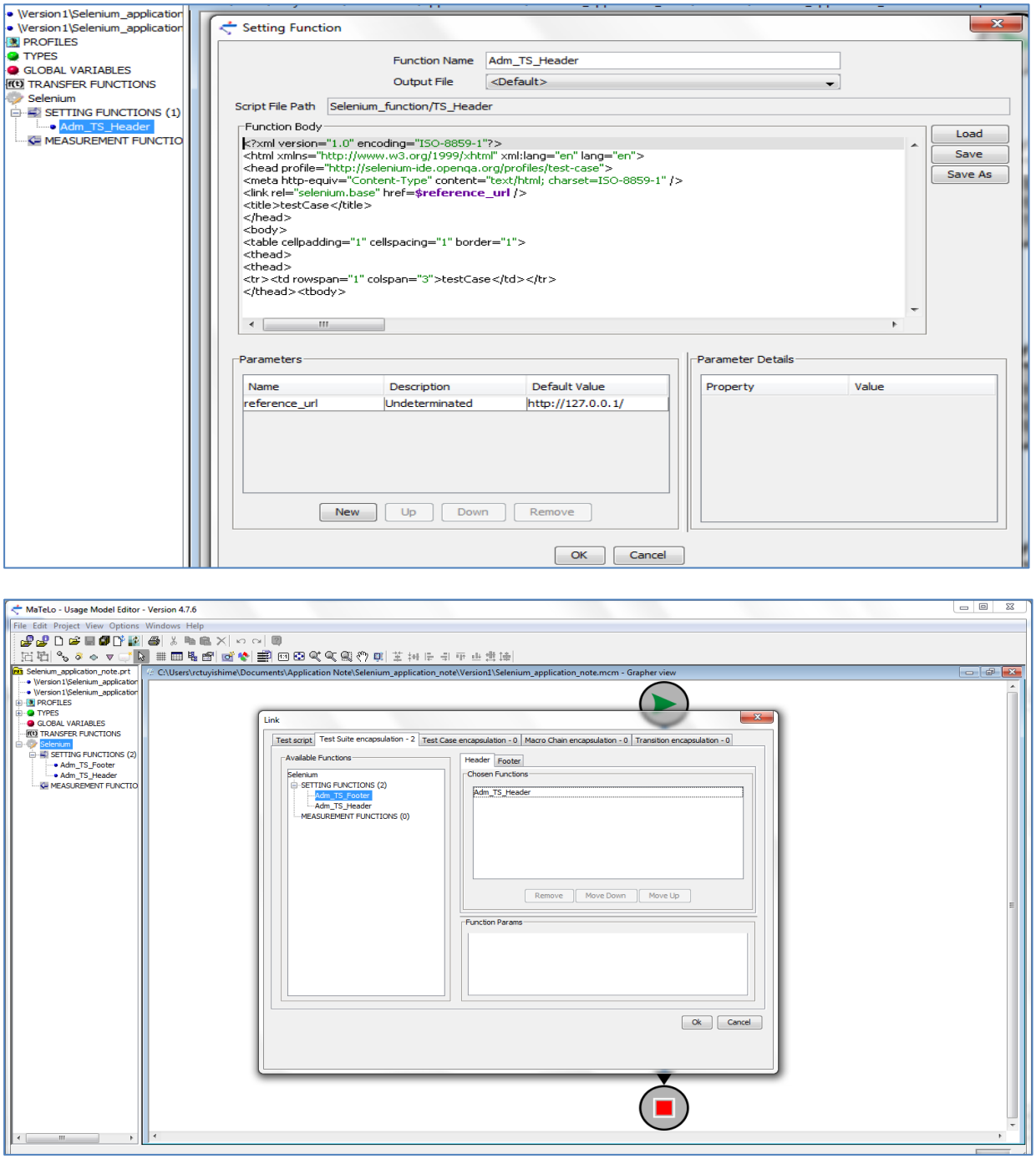



Figure 4: Test script encapsulation

Other administrative function like the function allowing the recognition or the displaying of MaTeLo transition (Step number, Transition name), macro chain names, header and footer of a test case, can be created and encapsulated at transition level, macro chain level and test case level.

To make modeling activity efficient, the creation of a template which contains basic functions and encapsulation done would be recommended to help users begin modeling. ALL4TEC developed an API which

enables the user import the Selenium standard library in MaTeLo. This API can be downloaded on our website.

# 4 Creation of Selenium test functions and its association on transition

## 3.1 Basic use

Basic user actions are performed in every web applications. For websites, basic actions can be a click on an object, opening a website or web page, verifying an object presence or a text presence.

MaTeLo defines two types of test functions:

1. Setting functions related to stimulations of the System Under Test (SUT)
2. Measurement functions related to measurement or checking of the expected results (output)

Selenium test functions must be created or imported before being associated on transition. For illustration, here is the test function for basic action “Open”:

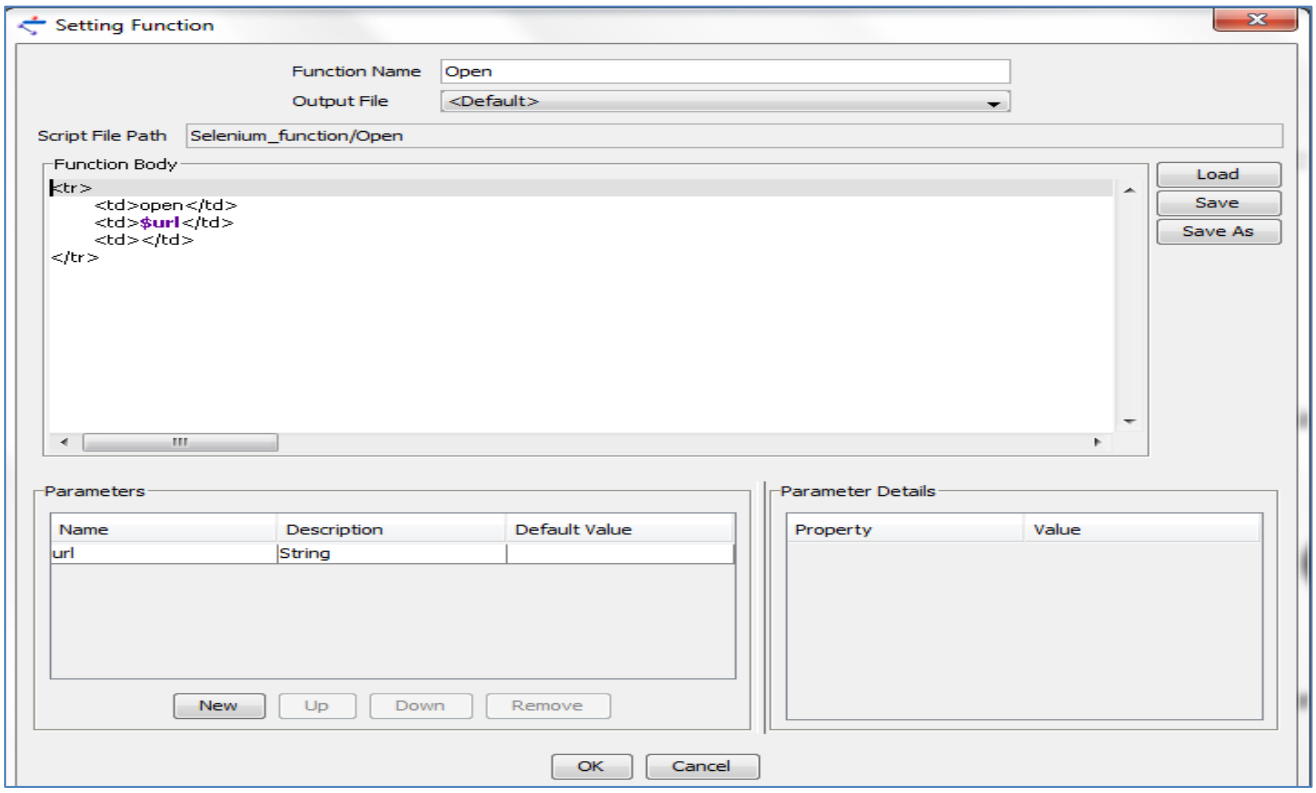



Figure 5: Setting function “Open”

As showed on the figure above, each function calls some parameters. For the function “Open” the parameter is the **url** of the website to open.

When the function “Open” will be associated to the transition, the parameter **url** will take the input’s value. On the figure below, the transition input has the same name as the function parameter.

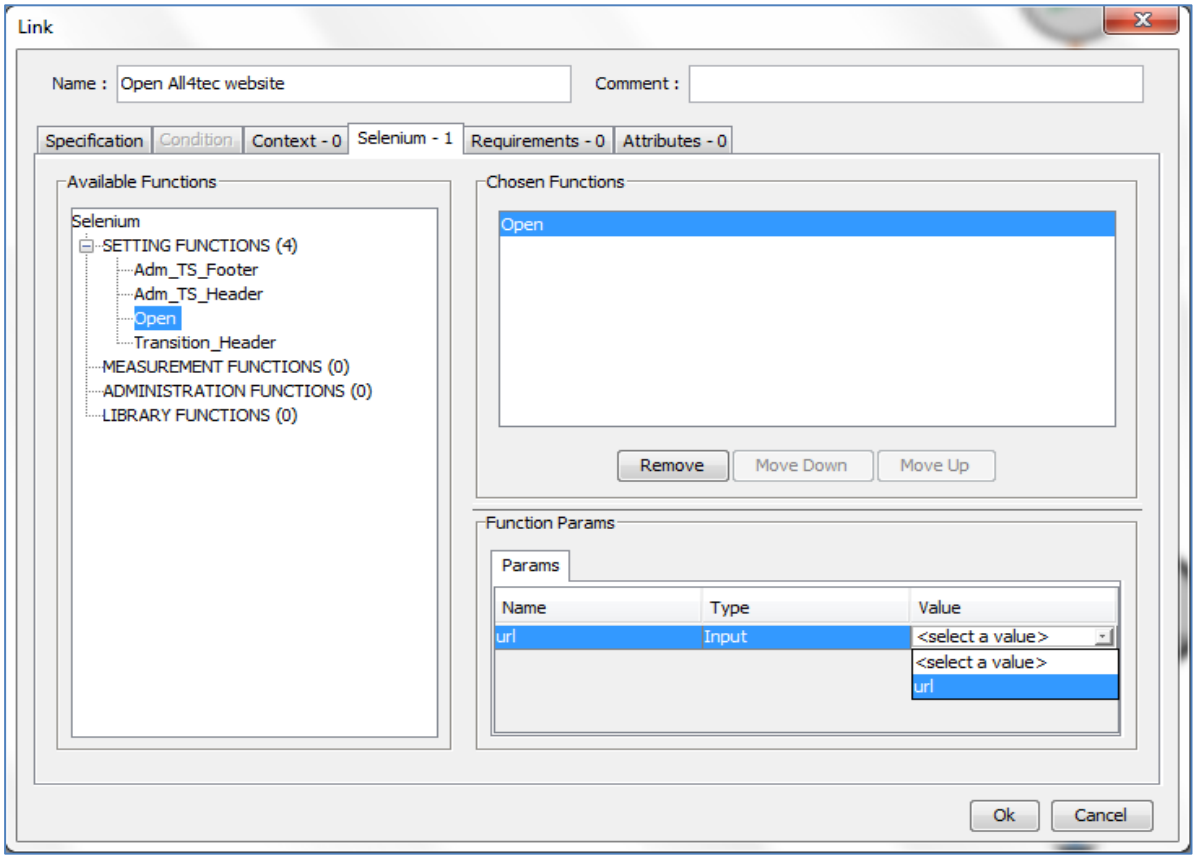



Figure 6: Test function - transition association

The process for creation and association of measurement functions will be the same as for setting functions except the parameters which will take Expected Result Automatic (ERA) value.

## 3.2. Functional use

It's possible to call more than one test functions or same function more than on time on a transition in a case of:

- Operations for which more than one basic actions are needed;

Example: To get connected on a website, we need three setting functions, typing the login, typing the password and click on button connexion or Ok.

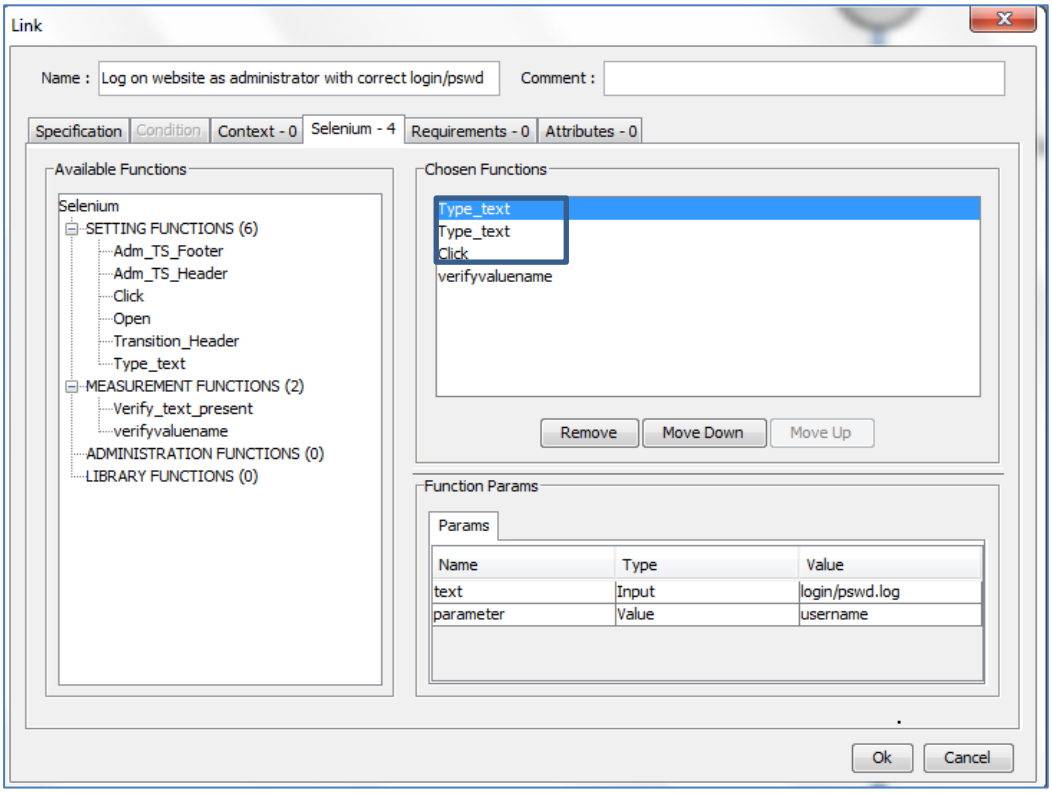



Figure 7: Several setting functions on a transition

- Many verifications on the same transition.

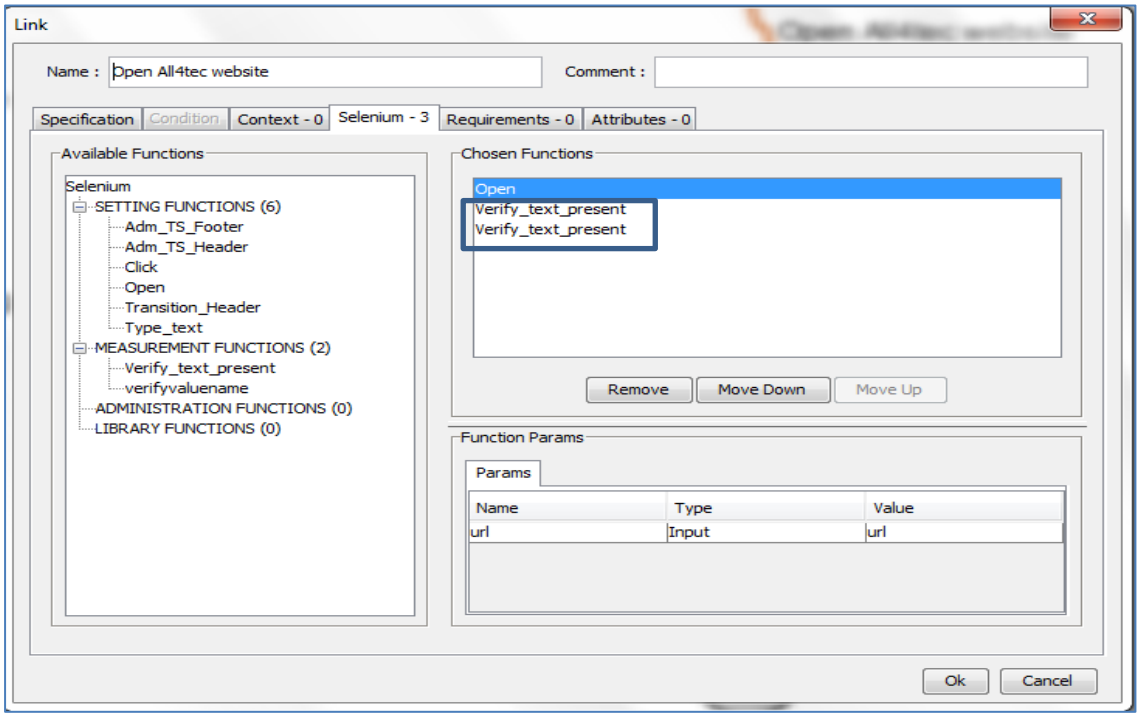



Figure 8: Several measurement functions on a transition

# 5 Generation of the test suite by MaTeLo Testor

## 5.1 Test strategies and correspondent algorithms

After the construction of the usage model and the setting up of Selenium test function on the usage model; with MaTeLo Testor you can generate the test suite in accordance with the test strategy defined beforehand.

For website testing, three test generation algorithm of MaTeLo Testor can be used for specific strategies:

- **Most probable**: to test nominal paths of the model. For that profile has to be defined for each nominal path. (profile management : http://www.all4tec.net/wiki/index.php?title=Profiles_Management)
- **Arc coverage**: to cover the whole model. With this algorithm, MaTeLo Testor will pass minimum one time on each transition of the model. All scenarios or functionalities represented in the usage model will be covered.
- **User oriented**: to perform functional load testing by generating randomly many functional test cases.

## 5.2 Test cases generation

After choosing the algorithm for the test cases generation, the generation can be performed. For each generation, MaTeLo Testor will generate:

- an XML file (see figure 7) which contains the description of the test suite (input and expected results for each test step, Selenium test functions associated to each step or transition, requirements covered …);
- file related to requirements coverage (requirements coverage rate);
- an html file which contains the test script.

In the section below, some screenshots on those MaTeLo Testor deliverables are exposed:

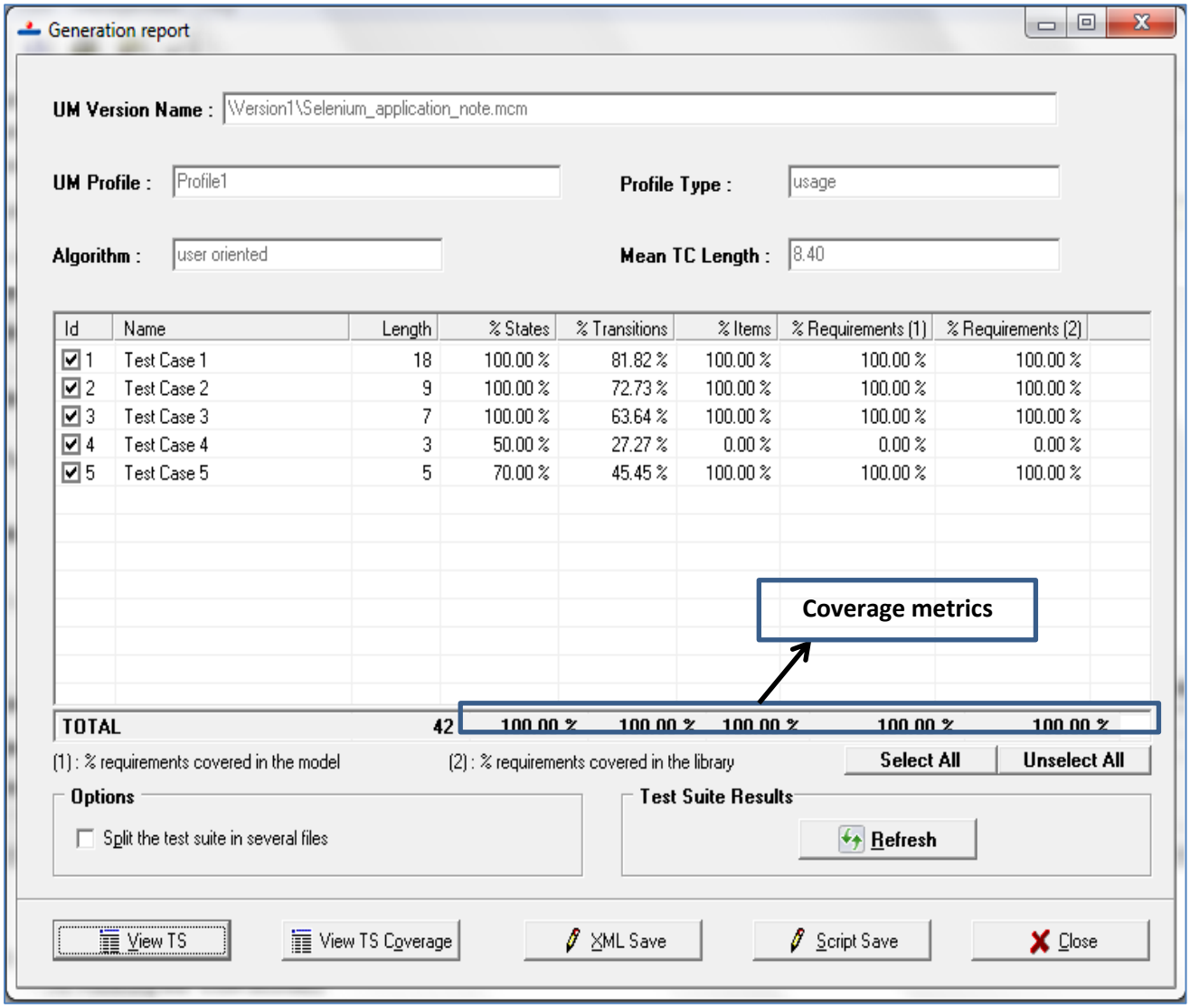



Figure 9: MaTeLo Testor generation tab

For the example, user_oriented algorithm has been chosen and 5 as number of test cases to be generated, has been specified.

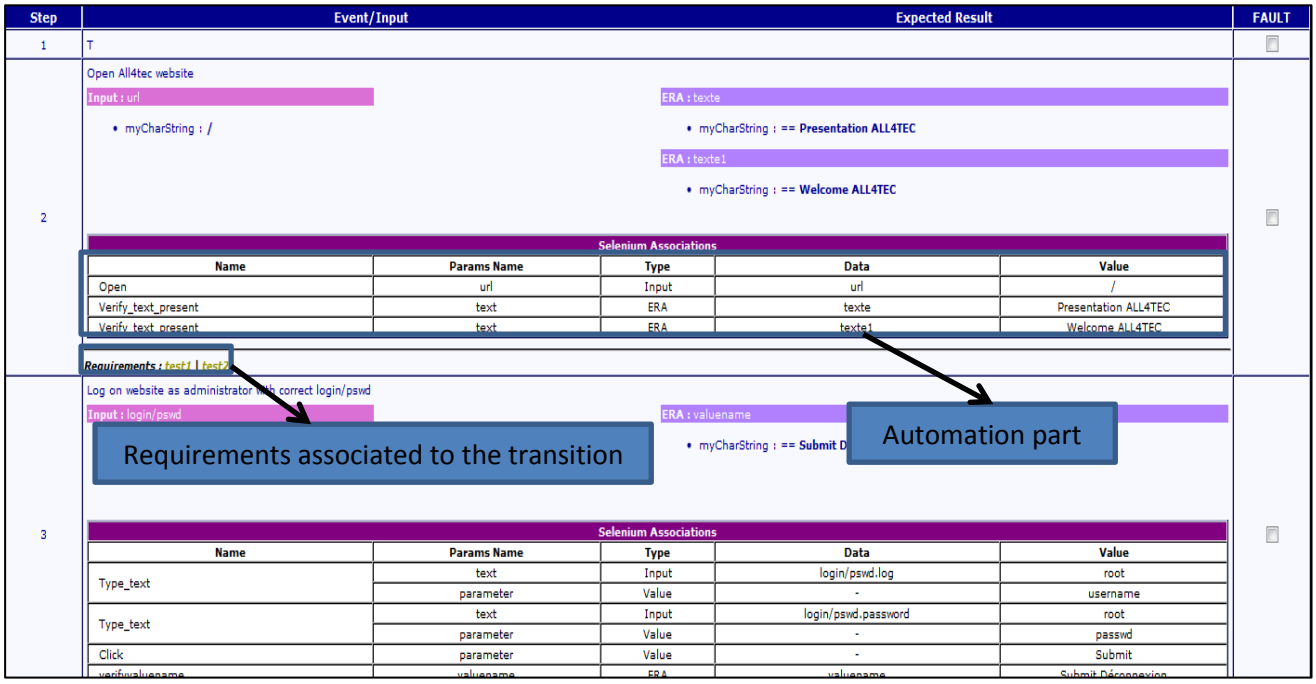
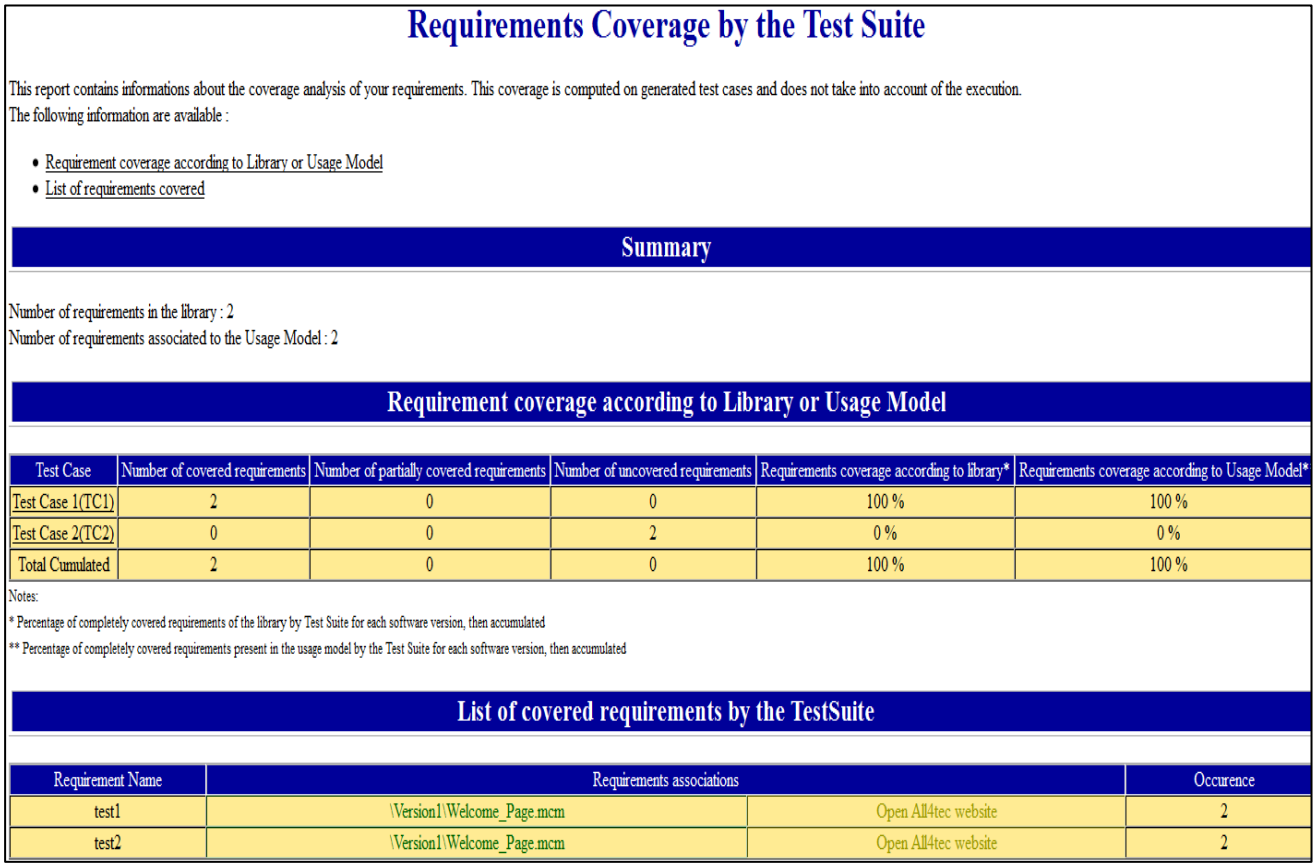



Figure 10: XML file for the test suite description



# Requirements Coverage by the Test Suite

This report contains informations about the coverage analysis of your requirements. This coverage is computed on generated test cases and does not take into account of the execution.
The following information are available :

- Requirement coverage according to Library or Usage Model
- List of requirements covered

## Summary

Number of requirements in the library : 2
Number of requirements associated to the Usage Model : 2

## Requirement coverage according to Library or Usage Model

| Test Case | Number of covered requirements | Number of partially covered requirements | Number of uncovered requirements | Requirements coverage according to library* | Requirements coverage according to Usage Model* |
|---|---|---|---|---|---|
| Test Case 1(TC1) | 2 | 0 | 0 | 100 % | 100 % |
| Test Case 2(TC2) | 0 | 0 | 2 | 0 % | 0 % |
| Total Cumulated | 2 | 0 | 0 | 100 % | 100 % |

Notes:
* Percentage of completely covered requirements of the library by Test Suite for each software version, then accumulated
** Percentage of completely covered requirements present in the usage model by the Test Suite for each software version, then accumulated

## List of covered requirements by the TestSuite

| Requirement Name | Requirements associations | | Occurence |
|---|---|---|---|
| test1 | \Version1\Welcome_Page.mcm | Open All4tec website | 2 |
| test2 | \Version1\Welcome_Page.mcm | Open All4tec website | 2 |

Figure 11: Test suite requirement coverage report

**Note:** Even test suite requirement coverage report, a requirement coverage report for each test case of the test suite is generated.

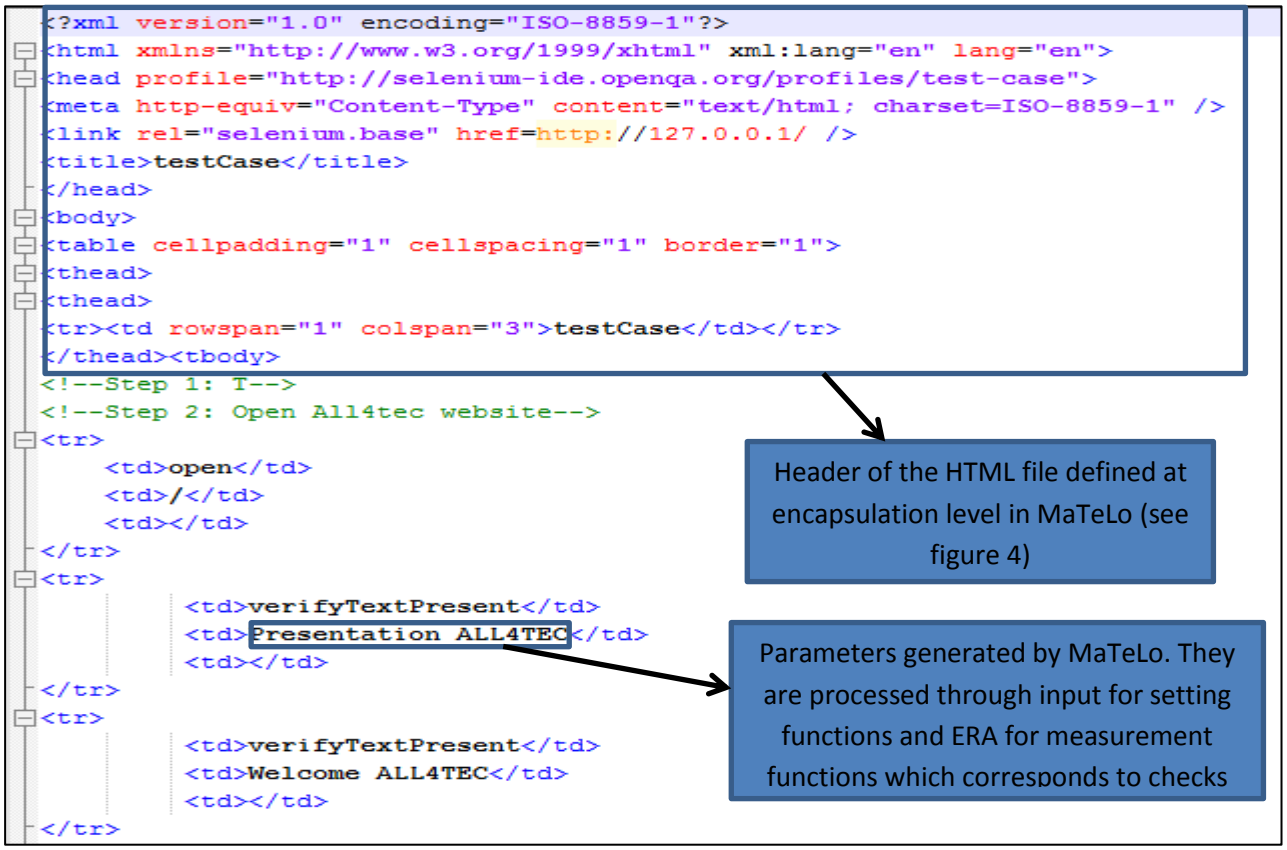



Figure 12: HTML file

The execution of the script is done thanks to Selenium.

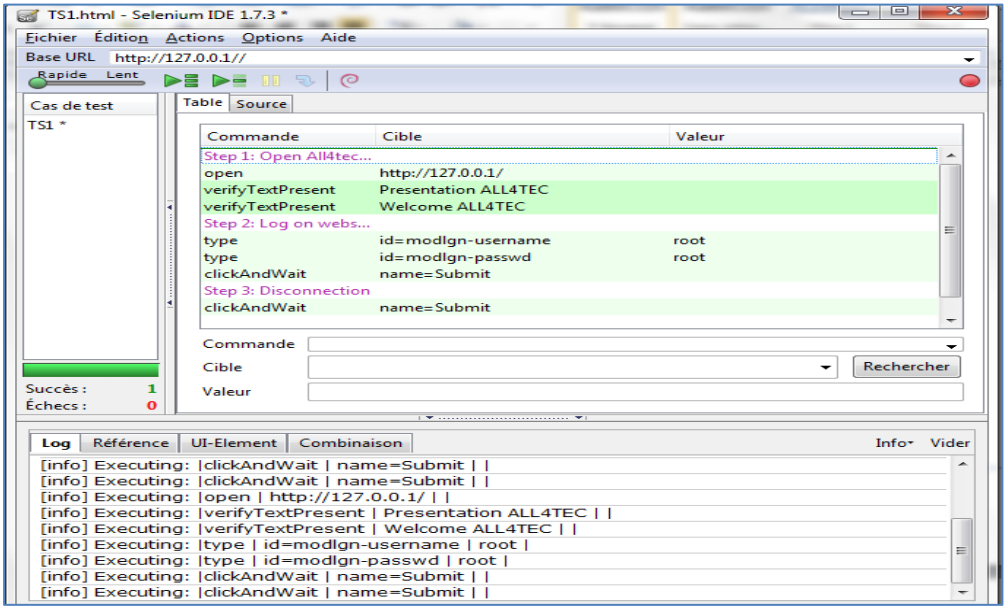



Figure 13: Script execution on Selenium

# 6 Conclusion

As presented in this application note, the generation of Selenium test script is possible in MaTeLo, easy to handle and can be adapted or customized for every specific need of the project.

The encapsulation feature is very helpful for administrative functions. This avoids setting administrative function on each transition of the model. It enables to build MaTeLo project template that will be used to generate “turn key ready” test script for Selenium

The MaTeLo usage model, help to represent the “website” use cases, in an integrated and graphical manner. This model can be review/verify by the “stakeholder”. The maintenance is eased, by simply just reorganizing states and transition to respect the new website workflow.

The different test strategies allow generating different test cases depending on the development maturity, and enable to generate test cases that represent the usage of the web site. It means that it will reduce bugs that appear to the most of the visitors.

To end, MaTeLo as a Model Based Testing tool offers the possibility to automate the test cases patrimony in order to reduce the test cost.

# 7 References

**Websites:**

- MaTeLo: http://www.all4tec.net/
- Selenium: http://seleniumhq.org/

**Wiki MaTeLo :**

- MaTeLo profiles : http://www.all4tec.net/wiki/index.php?title=Profiles_Management
- Test cases generation : http://www.all4tec.net/wiki/index.php?title=%E2%80%9CTest_Suite_Generation%E2%80%9D_Tab

**MaTeLo demo projects references:**

- ProprietaryTestBench_Selenium_HTML
- ProprietaryTestBench_Selenium_JUNIT

If you already have MaTeLo license, those demo projects can be found at this directory: C:\Users\Public\Documents\All4tec\MaTeLo\Demos\Demo MaTeLo V4.7.7