Research and Implementation of Automated Testing Framework-PATF Framework

Liuren Wang[1, a]

[1]Room 32-2-902, E3 Zone, Quanshui Renjia, Ganjingzi District, Dalian, Liaoning Province, P. R. China

[a]20954323@qq.com

Keywords: Automated Testing, PATF Framework, Test Efficiency

Abstract. Through the research on automated testing framework, the efficiency of automated testing can be improved, and the problems in the testing process be solved. The PATF framework provides a good solution for automated testing.

The Origin and Evolution Process of PATF Framework

The Development of Automated Testing. After a long-term development and evolution, software testing develops from a pure manual manner to today's combination way of automation and manual, professional requirements for software testing have become more sophisticated, the quality and technical level request for the testing engineers are also increased.

In the non-automated testing phase, all the test jobs were done manually by the non-professionals. At the initial phase of automated testing, they were semi-automated testing driven by the script file, tested by non computer software professionals, no testing tools and no laboratory. At the test automation development phase, some test cases realized automation, developing test code by computer software professionals with Perl scripts, using tools with professional lab testing environment [1]. At the mature stage of automated testing, a large number of test cases realized automation; the test case automatic operation system is combined with product construction system, test reports and statistical analysis service are provided, and ultimately evolving to the advanced stage of automated testing. There are high level test developers, test architectures, highly automated laboratory and professional management team, and the perfect automated testing process, and external business services of automated testing can be provided [2].

The Selection of Automated Testing Software. At present, most testing tools can only be used for the software testing of windows and web pages. If you want to test UNIX system software or command line (no graphical interface, only the instruction display), you may have to look for others test software or use complete programming. The black box test of the command line software is mainly to input various kinds of instruction, and then whether the information returned from the server is correct and if the instruction is executed correctly can be observed.

In order to fit for the command line software test, SHELL, EXPECT, TCL, DEJAGNU, PERL, etc. can be used; through constant trial and testing, the final choice of PERL becomes the basic language to achieve the type of command line software.

The Advantages of PERL Language. Perl started very early in automated testing, in the earliest version of Perl1.0, there is comprehensive test set of authors. Perl does not need the compiler and linker to run the code, who with rich expressive abilities, and only a little power released can do a lot of things. Perl can simplify the design work without losing the language ability to process difficulties. Moreover, with the power and flexibility of advanced language (such as C), actually many features are borrowed from the C language. And it’s easy for PERL to operate digit, text, files and directories, computer and network, especially the programming language [3].

The Overview of Research Process of PATF Automated Testing Framework. Through the previous attempts, we found that the real realization of automated testing can not be reached just by mastering certain automated testing tools or the scripting technology. Facing the complex communication system, the simple record/playback can not achieve the requirements of automated testing. By the method of complete writing the script, the workload is enormous, the maintainability is poor, and can not be reused. Automation is to improve the test efficiency, that who do not have the maintainability and with poor reusability will become the most fatal factor leading to the failure of automated testing, the huge price is paid but little effect is achieved.

Based on above factors and combined with the development of the industry, before the formal implementation of automation, a set of appropriate automated testing framework must be set up, so that the script can be effectively organized and used to provide the maintainability and readability of the test script.

The PATF automated testing framework enables the testing personnel to accomplish the automation system design and implementation independently through the familiarity with the framework case writing format and business logic and business process of function points, which can provide the testing personnel with automated testing framework constructed rapidly, so that the future automated testing can be conducted orderly and efficiently.

Research Process of PATF Automated Testing Framework. a. QTP、LR；

b. Open source testing framework staf/stax;
c. Regression testing framework Tcl/expect/Dejagnu;
d. 1st version of regression testing framework Perl/jsp/oracle;
e. 1st version of PATF automated testing framework Perl/Flex/XML;
f. 2nd version of PATF automated testing framework Perl/Flex/XML, 200 test cases of industry gateway kernel module are completed, and 12 basic use case of Monternet gateway are completed.

Key Technologies of Automated Testing Framework-Test Script Technology

Automated Test Script Design Standards. a. *Maintainability*: refers to whether the test update is easy to keep up with the pace of software upgrade. In the highly competitive IT field, the software will continue to upgrade or change in order to adapt to demand change. It is important to prevent excessive maintenance workload of automated testing caused by software upgrade.

b. *High efficiency*: means the efficiency is closely related to the cost. An important goal of automated testing is to run the test more economically, so there should be a comprehensive survey of various factors in the design phase of testing and automated testing. On the basis of scientific and reasonable testing process, the use of excellent automation strategy will be the combination of automated testing and manual testing, so as to achieve the maximum efficiency.

c. *Accessibility*: refers to if the creation or change of test case is easy, and if it is easy for test personnel or user to grasp and understand the use of the method.

d. *Reliability*: refers to whether the test system can give accurate and reproducible results. If the questionnaire submitted has a lot of problems caused by errors due to the automated test script itself, then the senior managers and software developers may distrust automate planning, so that the implementation of automated testing becomes more difficult.

e. *Compatibility*: refers to whether the test cases are allowed to be mixed in different ways for different test objectives.

f. *Portability*: refers to the ability to run tests on different platforms.

g. Robustness: refers to handle unforeseen circumstances without unnecessary exit or termination, and try to give the correct useful information.

Design and Analysis of Automated Testing Script. What we need to note is these 7 standards are constrained by each other; one standard is satisfied may weaken another. We should consider these seven standards comprehensively when designing automated scripting architecture based on software features to improve the test standard, and to implement automated testing system that meets the actual needs.

If the software runs on the different platform, then the portability and accessibility are very important, other standards not be met well may not matter, because the time saved can make the test execution more faster, wining the time competitive advantage for the software; if the software complexity increases, but very stable with little changes, then the design should consider how to improve the compatibility and reliability, than consider much of maintainability and portability.

The Design of Automated Testing Script. Although automated testing can bring many advantages, but it must be noted that with the gradual accumulation of automated scripts, especially the continuously updated software, automated script maintenance work will increase dramatically to burn off a lot of energy of test engineer. The initial creation cost and late maintenance cost of automated testing scripts are big, which is why a lot of companies do not want to implement an automated testing. Thus, in the early implementation of automated testing, deliberate design should be made about the architecture of automation scripts according to the characteristics of the company's software to reduce the automated testing script creation and maintenance costs.

The Implementation of Automated Testing Script. The script exists in the form of file. In order to ensure the flexibility of the entire script, we should not only consider the independence of each script, so that each test case can be run separately when designing the script, we should also take into account the actual needs. We can execute test case by customizing the execution conditions.

Now the development of the script design is at the stage of collaborative development of multi designer. In the automated testing script development process, the workload of script coding is quite large. Each designer might has his own experience and habit to write Perl script code, and different style of Perl scripting code makes post-maintenance and upgrade work more complex and difficult. To enhance the readability of Perl script code and the stability of system, reduce maintenance and upgrade costs, and improve designer's efficiency to write Perl scripts, unified standard is required for Perl script coding work.

For example, we can standardize from the file name format, the encapsulation of the process function, the variable initialization and so on.

Perl script omitted form is as follows:

```
my $sp20_test = &sp20_default;
$sp20_test->{sim_args} = "-d";
my $smsc20_case = &create_case_info("local_20mo");
&case_add_sim_check($smsc20_case, “default_sp20” ,[ ‘receive one MO']);
&case_process($smsc20_case);
```

Key Technologies of Automated Testing Framework-Core Support Module

Design prerequisite of log module are:

a. Time synchronization between automated testing framework and testing system;
b. The server of the tested system supports ftp;
c. Each log file must contain log print time, and time format should be yy/mm/dd and hour:min:sec;
d. Log analysis uses regular expressions.

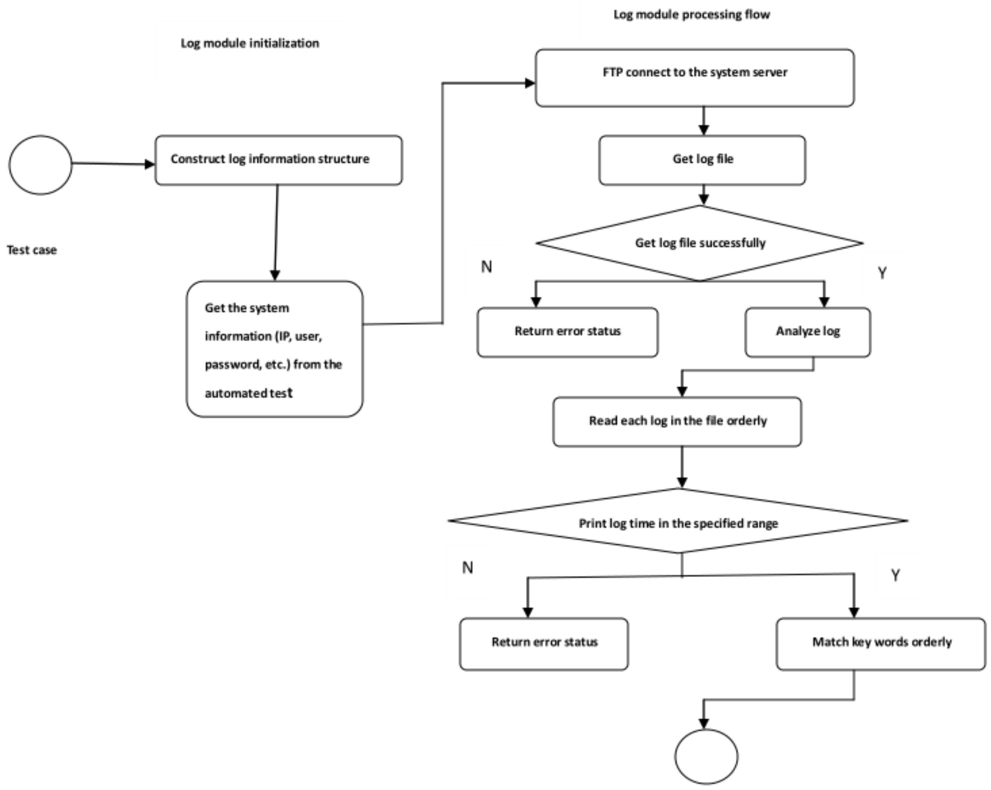

Log module initialization
Log module processing flow
Test case
Construct log information structure
Get the system information (IP, user, password, etc.) from the automated test
FTP connect to the system server
Get log file
Get log file successfully
N
Y
Return error status
Analyze log
Read each log in the file orderly
Print log time in the specified range
N
Y
Return error status
Match key words orderly

Fig.1 Log module flow chart

The Use of Automated Testing Framework

a. The display mode of FLEX interface.

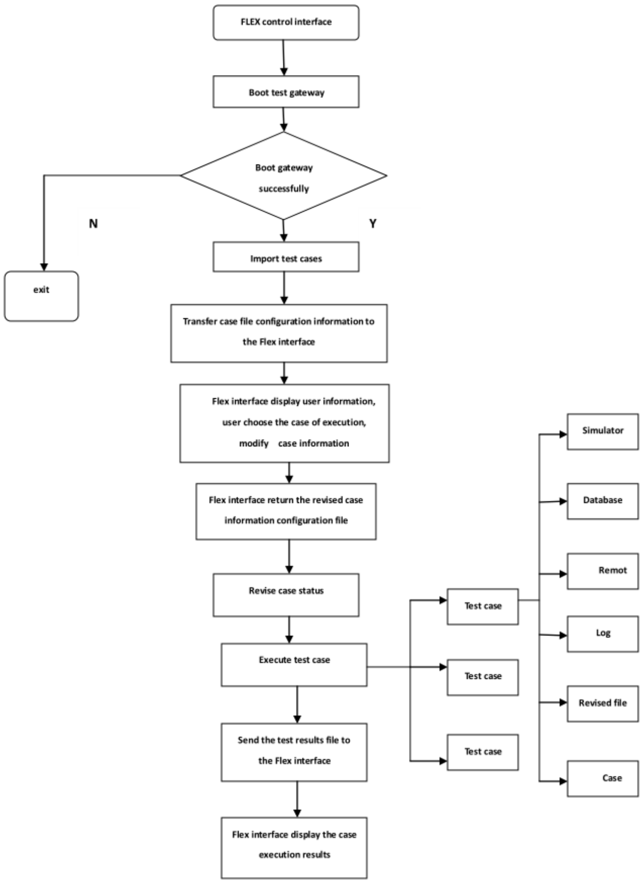

FLEX control interface
Boot test gateway
Boot gateway successfully
N
Y
exit
Import test cases
Transfer case file configuration information to the Flex interface
Flex interface display user information, user choose the case of execution, modify case information
Flex interface return the revised case information configuration file
Revise case status
Execute test case
Send the test results file to the Flex interface
Flex interface display the case execution results
Test case
Test case
Test case
Simulator
Database
Remot
Log
Revised file
Case

Fig.2 Flow chart of FLEX interface display mode

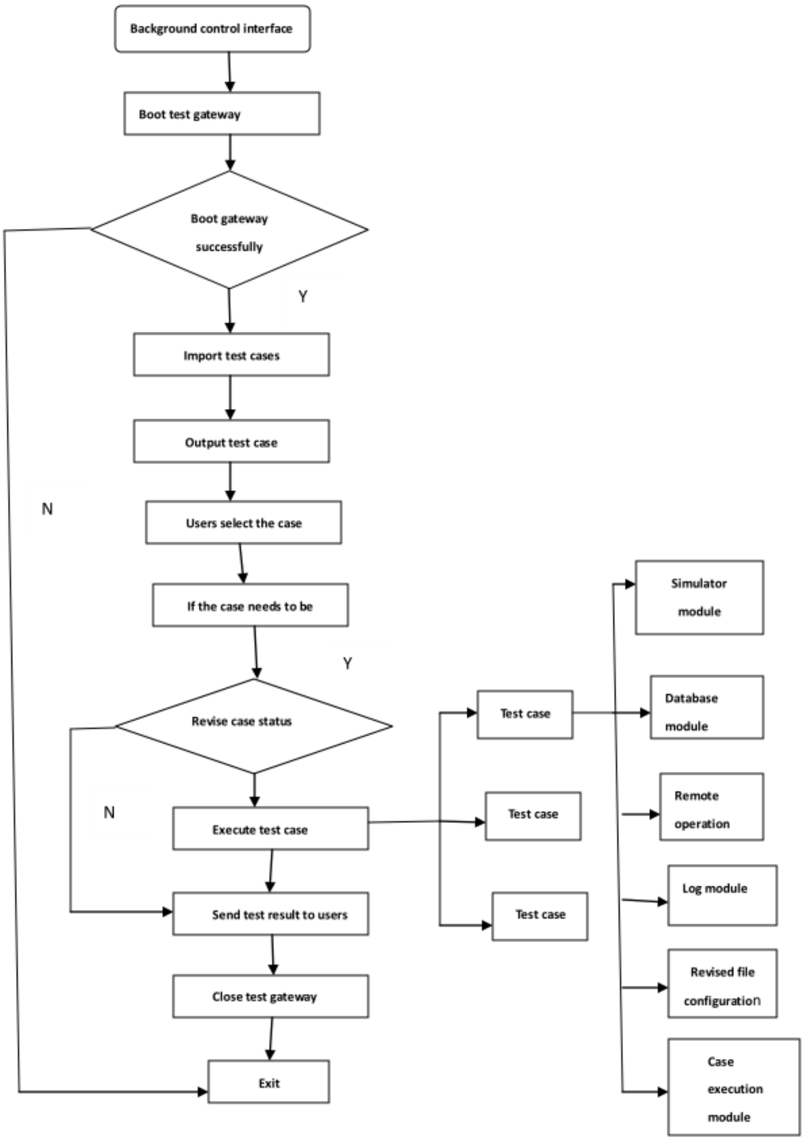
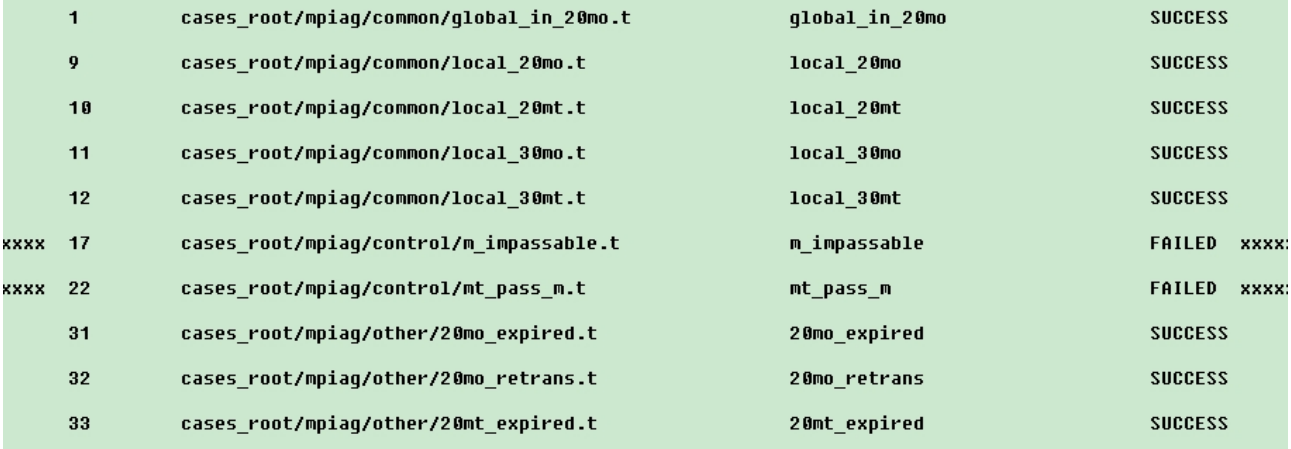


Fig.3 Flow chart of the display mode of command line text interface


```
     1    cases_root/mpiag/common/global_in_20mo.t     global_in_20mo    SUCCESS
     9    cases_root/mpiag/common/local_20mo.t         local_20mo        SUCCESS
     10   cases_root/mpiag/common/local_20mt.t         local_20mt        SUCCESS
     11   cases_root/mpiag/common/local_30mo.t         local_30mo        SUCCESS
     12   cases_root/mpiag/common/local_30mt.t         local_30mt        SUCCESS
xxxx 17   cases_root/mpiag/control/m_impassable.t      m_impassable      FAILED  xxxx
xxxx 22   cases_root/mpiag/control/mt_pass_m.t         mt_pass_m         FAILED  xxxx
     31   cases_root/mpiag/other/20mo_expired.t        20mo_expired      SUCCESS
     32   cases_root/mpiag/other/20mo_retrans.t        20mo_retrans      SUCCESS
     33   cases_root/mpiag/other/20mt_expired.t        20mt_expired      SUCCESS
```

Summary of Automated Testing Framework

The Effect. System design effect is very clear, greatly simplifies module needed to be designed in the test script development, to ensure the high quality completion of test cases, achieve the expected result and greatly improve the work efficiency.

Experience Summary.

a. Establish professional team of automated testing framework;

b. Experienced developers engage in the development of automated testing framework code;

c. Development discussion forms the automated testing specification (log, database table, etc.);

d. The 100% automation goal does not be pursued; communicate with the developer and the manual test personnel in time to avoid the bug repeat submittion;

e. Select the appropriate automation test interface people from the project;

f. Function test of the first round is not recommended to be introduced into automated testing.

Existing problems and Improvement. This framework has been operated into the practical work, which simplifies the complexity of the development of test cases in a certain extent, and improves the work efficiency of the test engineer, but there still exists some problems, therefore, the following questions should be studied from the beginning:

a. Due to the limitation of the simulator, the frame can only be run on the solarie platform, C++ or Perl should be considered at the next step to implement the simulator development;

b. If an error case occurred during the execution of the test case script, the related information of the error case script can be saved;

c. Existing task distributor of the automated testing framework only supports single user, multiple users supporting should be considered at the next step;

d. Keyword is separated from test case script; keyword shared by all test case scripts should be implemented;

e. Provide design that is more close to the test case, and simplify the automated testing scripts written and designed by the test engineer;

f. Continue to abstract automated testing framework, decoupling with the network management software as far as possible to apply for more products and projects.

References

[1] Ruijing Jia, Research and Implementation of Software Automation Testing Framework (in Chinese), Master Thesis of East China Normal University, 2011.

[2] Forster Gray (UK), translated by Shaomin Zhu, Qiuhua Zhang and Yanan Zhao, Best Practice of Automated Testing, Machinery Industry Press (in Chinese), April 2013.

[3] Ju Zhu, Zhijian Wang and Xue Yang, Software Automated Testing Framework Based on Data Driven (in Chinese), Computer Technology and Development, 2006 VOL.5.