

Design and Implementation of Human Space-based Automated Test Cases based on Script

Feng Yang

Department of Aeronautics and Astronautics Engineering, Nanjing University of Aeronautics and Astronautics

Nanjing, 210016, China

Institute of Manned Space System Engineering, China Academy of Space Technology

Beijing, 100094, China

E-mail: yangfeng501@163.com

The spacecraft automation test is to use the computer to complete the control instructions to send and monitor data according to predetermined program braking. The automation test case designer carries out the core work of the spacecraft automation test. The design format is unified, the grammar is simple and the operation is flexible that make the spacecraft testing process no longer the individual and discrete instruction. The test case has the process and the automatic monitoring ability. The automation test case design was introduced in this paper including the syntax of the details and the authentication system in the development and use to verify the reliability of the sub-project of the process control ability, support the test taking, the customizable process monitoring automation test case design. It is to improve the test efficiency, ensure the quality of test security, shorten testing cycle, and simplify test team necessary requirement and effective means.

*ISCC2017
16-17 December 2017
Guangzhou, China*

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).

<http://pos.sissa.it/>

1. Introduction

Manned spacecraft electric logging (also known as the spacecraft e-test) refers to the same power supply conditions through the ascending order to spacecraft, the spacecraft functions, performance, interface, etc. to carry on the comprehensive examination, several research activities in the spacecraft. The space testing work is one of the most critical and hard links. Before electric logging, manual sends commands as dominated, manned spacecraft uplink instruction has hundreds of thousands, manually-sent instructions when the test consumed such considerable human resources and time but with error-prone and low efficiency.

The manned spacecraft automatic test case design is the core of the spacecraft automatic test work, design format unification, grammar test procedure is simple, flexible operation, can provide basic data for spacecraft automatic test resources. Automation test case design makes the testing process no longer an independent individual and discrete instructions. This kind of design has the ability to streamline, automatic program-controlled. It can effectively reduce the manual operation and artificial intervention tester, this kind of design can integrate test resources at the same time, increase the degree of testing electronic.

As the test of manned spacecraft is facing more and more complicated logic and closed-loop dynamic simulation, etc. To respond to more complex test logic requirements, increase the flexibility of the system by using the "script atom" test case design. You can embed a script fragment in a sequence of atoms that can use code to describe complex logic and functionality. The basic framework, based on atomic do testers need to pick up all kinds of atoms from the atomic library of permutation and combination and attribute set, can be formed for a wide variety of logic functions and information flow of the automated test cases. [1-2] this paper introduced the automation test case, designed the syntax of the details and the authentication system in the development and use to verify the reliability of the sub-project of the process control ability, support the test taking, customizable process monitoring automation test case design, this kind of grammatical design will further enhance the automation of manned space flight.

2. Test Script Design Implementation

2.1 Advantages of Script Atoms

The script atom provides an atomic operation based on two different types of automated testing systems based on scripts, giving full play to the advantages of two types of automated testing systems. Users can describe complex logical scripts that can be embedded in an automated test sequence in the form of a test case design. The execution time system will automatically execute the corresponding operation according to the code written in the script. The script editor after no systematic compilation can cooperate with atomic sequence execution, editing software provides the function of the static syntax checking, to ensure that the designer can timely find syntax error code. For scripts that have been edited, replication calls can be made directly, with good reuse.

Because the automated test system is implemented in the Java language, the embedded scripts are written in the Java language. Compared with popular scripting language, Java language provides a powerful standard class library, grammar and more specifications, can facilitate the implementation of all kinds of complicated logic function, does not involve complex stack operation, can give full play to the script the advantages of powerful, flexible design. As one of the most widely used languages, designers can easily get the template code of

all kinds of complicated logic from the Internet. Designers can also add comments to standard syntax in scripts to make code sound readable. Test system in the script encapsulates the test parameters read, sends instructions, and other functions, the user through a simple call standard interface function, can realize send instructions, the basis of the interpretation of remote sensing test logic.

2.2 Generic Script Templates

The comprehensive test system USES test cases to form a more general test script for the standardized design. It makes the script more general and inheritable, and the functional modules are more clearly divided.

The first is to import the Java functionality package, and after the import function, you can directly call the various standard libraries in Java. There are three main packages: lang.*, util.date, text.simpledateformat. Among them, lang.* is the basic language function package, which provides the most common functions such as string processing, mathematical operation and basic data type. Util.date provides the operation interface for time and Date variables, which can be marked with system time in the data record. Text. SimpleDateFormat provides a standard interface to convert the date and time change to a string output.

And then get all sorts of telemetry parameters, mainly using standard interface functions of automation software get (string para_number), can be fuses iconic on remote sensing and ground equipment parameters into a floating point or integer variables, and by calling the Math function of package complete intermediate processing, finally get order parameters.

After get instruction parameters, using the elder sister rich function provided by the automation software send (string inst_number, string inst_type, string para_string) can be sent with or without parameters of instruction, can include send satellites into direct and indirect are discrete, software instructions, also can be used to the ground equipment to send into instruction.

After the instruction to send, you need to determine the corresponding parameter changes, while break infinite loop structure was adopted to realize, with the method of increasing each cycle of the timer, if again by timer timeout time judgment logic, it returns the right conclusion and adopt the way of break or return to launch cycle, if more than time limit judge logic is still failed, it returns error or failure, and return the entire script function directly.

2.3 Implementation of Key Functions

Similar to general script template, comprehensive testing system of testing requirements is mainly composed of some commonly used functions or logical unit, including the transformation of the realization of continuous judgment, float, time delay and other operations, the conclusion of atoms output, the output of time information, etc., the functional units can also be directly applied to the realization of the other subsystem test requirements, detailed as follows:

- The realization of consecutive interpretation

Consecutive interpretation while + break infinite loop structure was adopted to realize, and through the timer to avoid into dead circulation, if the timer again over time within the time limit for the logic of judgment, it returns the right conclusion and adopt the way of break or return launch cycle, if more than time limit interpretation logic is still failed, it returns error or failure of the conclusion, and put the entire script function returned directly. The basic code is as follows:

```

Int overtime = 0;           //timer initialization
While(true)                // circular judgment
{
Tread.sleep(5000);        //the timer step is set in ms
Overtime = overtime+1;    // the timer counter is increasing
If(result == true)        //read through
{
    setResult(true,result_string); // return the correct conclusion
    return;                       //Exit the script, and then use break
}                                 //to exit the loop if you need to run
subsequent code
If(overtime>240)           //timeout threshold setting
{
    setResult(false,result_string); //the failure result is returned
    return;                       //Exit the script, and if you
                                   //need to run subsequent code,
                                   //break out of the following code
}
}

```

- Implementation of floating-point conversion

Convert floating point Numbers to the IEEE standard instruction code is frequently in spaceborne software test case design to use a function, the method of using into Java class library of basic data types can facilitate the transition of system. It is important to note that the Java floating-point number is already in line with the big-endian standard when it comes to the IEEE code, so the hexadecimal code is not required to adjust the byte order, but it should be noted that the high level of the complementary operation is required. The basic code is as follows:

```

Integer bitValue = Float.floatToIntBits(floatValue);
//Converts floating-point Numbers
// to the number of integers
//corresponding to the IEEE standard code
String hexString = Inter.toHexString(bitValue);
//Gets the hexadecimal code string
hexString = ("00000000"+hexString).toUpperCase();
//Make up the high level and format the
//string into uppercase characters
String result_String = hexString(hexString.length()-8,
hexString.length()-4);
//Intercept the code for the corresponding
// width as needed

```

- Implementation of delay

Delay function plays an important role in steady-state judgment and telemetry continuous judgment. In the Java script, you can use the thread.sleep(sillisecong) function to make it easy to wait and wait.

- Information output

Automated software provides an output interface function for test results. The function is setResult (bool result, string result_String), and only when the function of set Result (true) is invoked, the script atom returns the successful value and keeps the automated sequence running. If you call set Result (false) or not call the setResult function, the script atoms return back to the failed conclusion and abort the operation of the automated sequence. The use case designer can use the second parameter to print the intermediate variable of the script to the execution result, which will be reflected in the system generated test results report.

- The implementation of time markers

In order to make the test results more comprehensive in the test state, the use case designer can record the read pass or fail time in the script execution result. This function is implemented primarily by calling the Date class library in Java. The basic code is as follows:

```

SimpleDateFormat df = new
    SimpleDateFormat (yyyy-MM-dd HH:mm:ss);
    //Set time format,
    //output according to time format
String succ_point = df.format(new Date());
    //Gets the system time of the read pass
    // or failure moment ,and formats it

```

3. Research on Automated Test Case Technology

3.1 Test Drive Automation Technology

From the test script perspective, include the following features:

- Universal test platform provides test script recording, storage, editing functions. The tester can record all of the operating records in the test for subsequent models or other relevant models. When used, the tester can reedit the script, modify the parameters, and execute the time.
- Through test platform support structured design function of the test scripts. This function is similar to structured programming, with structured scripts with various logical structures (sequence, branch, loop), and function call functionality. Select control script has the judgment function as a result, by judging the authenticity of a certain conditions, the script can perform a particular operation: support the script calls at the same time, a script to control point of transfer to another is the beginning of the script, when the child after the script execution, control points back to the first script to start the location of the call.
- Support scripting sharing, that is, a script (test case) can be used by multiple test cases, when a test project is a test script after implementation, other test cases to repeated calls to this function.

Test scripts have preconditions that test atoms to execute when the current conditions are satisfied. At the same time, due to the real-time influence, each test script has time constraint, indicating the validity of the test action in a certain time range. The test task is composed of a variety of complex operations that are performed by the test script. In the test model, the test script coupling compound with real-time operation, these complex called constraint parallel operation, the choice of time constraints and time constraints and cycle time constraints, to test script, the meaning of these composite operation time constraints as below:

```

//Test script definition:
Atom ::= Precond Aname (Runtime)[TestP]
//The test task short test task T syntax is defined as:
T ::= skip | AtomL ||i T | AtomL;iT
AtomL ::= skip | Atom;iAtomL ||i AtomL
        | whilei Bexp do AtomL od
        | ift Bexp AtomL else AtomL

```

```
// Time constraint serial " ;t " : only when a test script executes to meet the time constraint,
the result is true; Execute, whether to abort , return to false ;
//Time constraint parallel "||t" : when and only if two test scripts execute to meet the time
constraint, the result is true; Execute, whether to abort , return to false ;
// Time constraint selects "ift" : when the conditional Boolean expression is true , the result of
its operation is the first test script execution result, otherwise the result of the second
execution;
// Time constraints to choose "whilet" : when after entering cycle , only when the test script
execution meet time constraints in the loop and the result is true , will return cycle entrance ,
continue to cycle, or the end of the cycle.
```

Anyhow, test-driven automation can ensure the correctness of the logical relationship between test incentive, ensure the uniqueness of test inputs, avoid tests for test data into chaos lead to changes in the test results or testing error, and can realize automatic test to cover all the purpose of the test area as much as possible.

3.1 Test Data Automatically Generated and Data Mining

A large number of different types of test data are filled in the database to achieve the test data entry and playback and reanalysis purposes. Capturing is recording every process of the test. Including test operation and corresponding generated test data, data processing is deposited into the database. In the database, the relationship between the two is based on the time relationship, and the subsequent testing is ready to be used.

The specific automation process is shown in the figure:

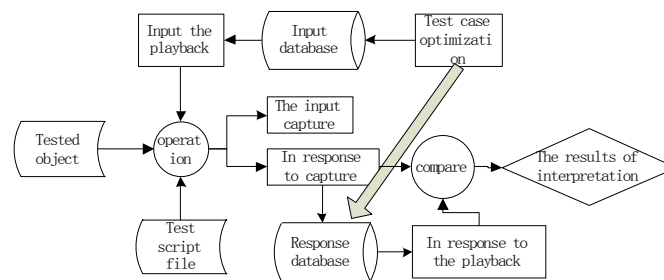


Figure 1: Data capture automation process

The research of test data mining includes the following three aspects:

- First, analysis and statistics of test case data. For each test case execution do after the analysis, the test cases associated with the tested software function, performance, classification management, implement test cases for subsequent software testing script automatically generated provide basic information database and the software reliability testing.
- Second, statistics and analysis of the fault report of the tested object and the data involved. According to the cause of failure, the frequency of occurrence of failure and the degree of failure, the fault event database is established. A fault - data - test correlation matrix, a Boolean matrix that responds to all kinds of physical checkups, is a fault - data - test correlation matrix. Therefore, a clear fault set and test set can be obtained to further complete the analysis of fault characteristics and test parameters. The test platform can be used to test the test. According to the correlation matrix, the tested objects are sampled. From the test point of view, we can confirm whether the various faults can be avoided effectively.
- Third, for complex object under test, when test or a few specific point of failure, according to the historical data in the database, determine the applause of the scene, as well as the transitive relation between them. Test platform can provide feedback analysis function, help testers quickly query a test pass in the model, and the

corresponding relation of fault test, to enhance the visibility and operability of the model. After the module is completed, a variety of analysis can be carried out to help understand the reliability index of the measured module.

4. Conclusion

The test automation software based on script atom has been successfully applied to several models, which has played an important role in the design and implementation of key use cases in the comprehensive test subsystem test. Demand provides extremely flexible implementation framework, implements the runtime parameters of dynamic translation, ensure the comprehensive test system of test cases "one-click" performance, greatly reduced the pressure of the tester. The continuity of dynamics model in test engineering is ensured, and the coverage of different dynamic scenes is enhanced. The most important implementation of the comprehensive test case is that the configuration is not independent, which enhances the maintainability of the test case. From the perspective of application, the script atoms can be further studied in the implementation process with the user's dynamic information interaction, basic function encapsulation and invocation flexibility. Predictably, the script combined with test atomic architecture will be increasingly complex test requirements to achieve in the future play a more and more important role, is the function of the test system and efficiency are improved.

References

- [1] Q.C. Wang, *Spacecraft Electric Measurement Technology*[M], China Science and Technology Press, 2007:183
- [2] Y.C. He, H.F. Li, F. Yang, et al. *Design and Application of Electrical Test Automation Program for Manned Spacecraft*[J], Computer Measurement and Control, 24 (12) : 53-55, 2016.
- [3] M. Ni, P. Huang, *The Script Based Component Test Automation Framework*[J], Computer Engineering, 6:94-99, 2010.
- [4] J.X. Zhang, Y. Zhang, C.L. Li, *Automated Software Testing Framework for Scripts*[J], Terahertz Science and Electronic Information Journal, 5:797-801, 2013
- [5] J.M. Zhao, L. Ren, F. Yang, *Integrated Design Method of Spacecraft Power Supply and Distribution Test Software Based on Configuration Files*[J], Computer Measurement and Control, 22(7): 2316-2317,2014
- [6] J. Tong, Y.W Xing, *Application of Next Generation Automatic Test System in Aerospace Test Architecture in China* [J], Aerospace Control, 29(2): 75-80, 2011