

Research on Application Testing Method Based on Test Script

Junhao Zhai, Rui Wang

University of Science and Technology Liaoning, China

Abstract: This project provides a research on application testing method based on test script, especially relating to the technical field of big data. The method includes: according to a preset interface document, obtaining the environment information, field information and field standard information corresponding to the field information of the tested application; Filling the environmental information, field information and field standard information corresponding to the field information into a preset test script template to obtain a test script; Running the test script to perform the application test. This project can automatically generate test scripts based on the current interface documents, thus improving the speed and accuracy of generating test scripts, and further improving the efficiency of the overall application test.

Keywords: Test script, Application testing, Big data.

1. Introduction

In the process of application testing, it is necessary to write and generate the corresponding test script and carry out the corresponding test by executing the test script. However, in the prior art, testers often manually write codes based on predetermined test interface documents to generate test scripts. However, the speed of writing test scripts manually is slow, and it is easy to make mistakes in the writing process, which leads to the low accuracy of the generated test scripts, resulting in the low efficiency of the overall application test.

2. Project Content

One purpose of this project is to provide an application testing method based on test scripts, so as to solve the problem that the speed and accuracy of generating test scripts are low, which leads to the low efficiency of overall application testing. Another purpose of this project is to provide an application testing device based on test scripts. Another object of this project is to provide a computer device. Another object of this project is to provide a readable medium. Another object of this project is to provide a computer program product.

In order to achieve the above purpose, one aspect of this project discloses an application testing method based on test scripts, which includes: according to the preset interface document, obtaining the environment information, field information and field standard information corresponding to the field information of the tested application; Filling the environmental information, field information and field standard information corresponding to the field information into a preset test script template to obtain a test script; Running the test script to perform the application test.

Before obtaining the environmental information, field information and field standard information of the tested application according to the preset interface document, packaging the test script template according to the preset environmental information type, the environmental access logic corresponding to the environmental information type, the field standard type and the verification logic corresponding to the field standard type.

Before obtaining the environmental information, field

information and field standard information of the tested application according to the preset interface document, packaging the environmental information type and field standard type to obtain an interface document template; Respectively filling preset environmental information corresponding to the environmental information type, preset field standard information corresponding to the field standard type and preset field information into the interface document template to obtain the interface document.

According to the preset environmental information type, the environmental access logic corresponding to the environmental information type, the field standard type and the verification logic corresponding to the field standard type, packaging to obtain the test script template, including:

Environment type, and obtain address information type and interface information type; Field type, get format standard type and content standard type; Address type, interface information type, format standard type, content standard type, environment access logic and verification logic, and packaging to obtain the test script template.

Address type, interface information type, format standard type, content standard type, environment access logic and verification logic, and packaging to obtain a test script template, which comprises address information type and the environment access logic, and packaging to obtain an application address access code segment; The information type, the application address access code segment and the environment access logic are encapsulated to obtain the application function interface access code segment; The format type and the check logic are packaged to obtain a format check code segment corresponding to the format standard type; A content standard type and the verification logic are encapsulated to obtain a content verification code segment corresponding to the content standard type; And package that access code segment, the format check code segment and the content check code segment based on the application function interface to obtain a test script template.

A format standard type and the verification logic are encapsulated to obtain a format verification code segment corresponding to the format standard type, which comprises a format standard type, a format standard receiving logic, a format test data generating logic and a format test application

calling logic for encapsulating and receiving the format standard corresponding to the format standard type; Wherein, the format test application calling logic is used for inputting the format test data generated by the format test data generating logic to the corresponding application function based on the application function interface access code segment, so as to make the application function run and return the first result; The logic is checked with the format test data generation logic to obtain the expected format return result; Checking the logic and the expected format return result to obtain a format assertion corresponding to the format standard type; Wherein the format assertion is used for receiving the first result and judging whether the first result is consistent with the expected format return result, if so, returning format check success information corresponding to the format standard type, and if not, returning format check failure information corresponding to the format standard type; Based on the format standard receiving logic, the format test data generating logic, the format test application calling logic and the format assertion, packaging to obtain a format check code segment.

A content standard type and the verification logic are encapsulated to obtain a content verification code segment corresponding to the content standard type, which comprises a content standard type, a content standard receiving logic, a content test data generating logic and a content test application calling logic for encapsulating and receiving the content standard corresponding to the content standard type; Wherein, the content test application calling logic is used for inputting the content test data generated by the content test data generating logic to a corresponding application function based on the application function interface access code segment, so as to make the application function run and return a second result; Check the logic and the content test data generation logic to obtain the expected content return result; Checking the logic and the expected content return result to obtain the content assertion corresponding to the content standard type; Wherein the content assertion is used for receiving the second result and judging whether the second result is consistent with the expected content return result, if so, returning the content verification success information corresponding to the content standard type, and if not, returning the content verification failure information corresponding to the content standard type; Based on the content standard receiving logic, the content test data generating logic, the content test application calling logic and the content assertion, packaging to obtain a content check code segment.

Based on the application function interface access code segment, format check code segment and content check code segment, packaging to obtain a test script template, which includes: docking the application function interface access code segment, format check code segment and content check code segment according to a preset logical relationship and time sequence to obtain a test script template. According to the preset interface document, the environment information, field information and field standard information corresponding to the field information of the tested application are obtained, including: interface document to obtain environment information identification and field information identification; The environmental information identifier determines the address of the environmental information in the interface document, and reads the environmental information based on the address of the

environmental information; Field identification: determining the field information address in the interface document, and reading the field information and the field standard information corresponding to the field information based on the field information address.

Filling the environmental information, the field information and the field standard information corresponding to the field information into a preset test script template to obtain a test script, which comprises: environmental information to obtain address information and interface information; Field information to obtain format standard information and content standard information corresponding to the field information; Filling the address information corresponding to the address information type into the application address access code segment, filling the interface information corresponding to the interface information type into the application function interface access code segment, filling the format standard information corresponding to the format standard type into the format check code segment, filling the content standard information corresponding to the content standard type into the content check code segment, And correspondingly filling the field information into the format check code segment and the content check code segment, so as to complete the step of filling the environment information, the field information and the field standard information corresponding to the field information into a preset test script template to obtain a test script.

In order to achieve the above purpose, another aspect of this project discloses an application testing device based on a test script, which includes: an interface document parsing module, which is used to obtain the environment information, field information and field standard information corresponding to the field information of the tested application according to a preset interface document; The test script generation module is used for filling the environmental information, the field information and the field standard information corresponding to the field information into a preset test script template to obtain a test script; A test script running module, configured to run the test script to perform the application test.

This project also discloses a computer device, which includes a memory, a processor and a computer program stored in the memory and executable on the processor, and the processor implements the above method when executing the program.

This project also discloses a computer readable medium on which a computer program is stored, which when executed by a processor realizes the above-mentioned method.

This project also discloses a computer program product, including a computer program, which when executed by a processor realizes the above method.

3. Specific Design Methods

The implementation example of this project discloses an application testing method based on test script, which specifically includes the following steps:

S101: according to the preset interface document, the environment information, field information and field standard information corresponding to the field information are obtained.

S102: the environment information, field information and field standard information corresponding to the field information are filled into the preset test script template to obtain the test script.

S103: running the test script to carry out the application test.

The code language of test script can be but not limited to Perl, python, PHP, TCL, guile, ruby, C, C++, Java or Pascal, and Python is preferred.

The preferred format is JMX, but not limited to test files.

The test script can be run through but not limited to Gatling, apibox, badboy or JMeter test tools, preferably JMeter test tool.

This project provides the application testing method research based on test script. According to the preset interface documents, the environment information, field information and field standard information corresponding to the field information of the application under test can be obtained, and the test characteristic information required for generating test script can be obtained accurately and quickly. It provides the necessary foundation for the automatic generation of test scripts in the following steps, and improves the speed and accuracy of generating test scripts in subsequent steps; By filling the environment information, field information and field standard information corresponding to the field information into the preset test script template, the test script can be obtained. The test script can be generated quickly and accurately based on the test characteristic information obtained from the interface document in an automatic way and simple steps. It does not need the tester to write the test script in person, and reduces the possibility of errors in the test script, so as to improve the speed and accuracy of generating the test script. When the interface document is replaced or the content of the interface document changes, the corresponding new test script can be automatically generated based on the new interface document. No need for testers to modify or rewrite the test script, which further improves the efficiency of the overall application testing. By running the test script to carry out the application test, the application test can be carried out conveniently and quickly, and the efficiency of the application test can be improved. To sum up,

the research on application testing method based on test script provided by this project can realize automatic generation of test script based on current interface document, so as to improve the speed and accuracy of test script generation, and then improve the efficiency of overall application testing.

An alternative embodiment includes:

Before the environment information, field information and field standard information corresponding to the field information of the application under test are obtained according to the preset interface document,

According to the preset environment information type, the environment access logic corresponding to the environment information type, the field standard type and the verification logic corresponding to the field standard type, the test script template is encapsulated.

Through the above steps, the test script template can be packaged accurately and quickly based on the actual target test environment and target test field, so as to improve the accuracy and applicability of the test script template, and indirectly improve the accuracy and speed of generating test script based on the test script template in the subsequent steps.

The systems, devices, modules or units described in the above embodiments can be implemented by computer chips or entities, or by products with certain functions. A typical implementation device is a computer device. Specifically, the computer device can be, for example, a personal computer, a laptop computer, a cellular phone, a camera phone, a smart phone, a personal digital assistant, a media player, a navigation device, an email device, a game console, a tablet computer, a wearable device, or a combination of any of these devices.

In a typical example, a computer device specifically includes a memory, a processor, and a computer program stored in the memory and executable on the processor, and the processor implements the above-mentioned method when executing the program.

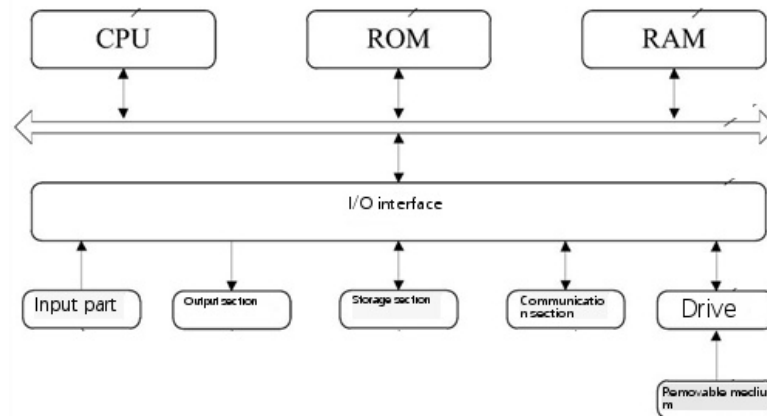


Figure 1. Structure diagram of computer equipment

Referring to fig. 1 below, it shows a schematic structural diagram of a computer device 700 suitable for implementing the embodiment of the present application.

As shown in fig. 1, a computer device 700 includes a central processing unit (CPU)701, which can perform various appropriate jobs and processes according to a program stored in a read only memory (ROM)702 or a program loaded into a random access memory (RAM)703 from a storage section 708. The RAM703 also stores various programs and data

required for the operation of the system 700. A CPU701, a ROM702, and a RAM703 are connected to each other through a bus 704. Input/output (I/O) interface 705 is also connected to bus 704.

The following components are connected to the I/O interface 705: an input part 706 including a keyboard, a mouse, and the like; Includes an output section 707 such as a cathode ray tube (CRT), a liquid crystal feedback (LCD), a speaker, etc. A storage portion 708 including a hard disk or

the like; And a communication part 709 including a network interface card such as a LAN card, a modem, etc. The communication section 709 performs communication processing via a network such as the Internet. A drive 710 is also connected to the I/O interface 705 as required. Removable media 711, such as magnetic disk, optical disk, magneto-optical disk, semiconductor memory, etc., is installed on the drive 710 as needed, so that the computer program read from it can be installed as the storage part 708 as needed.

Particularly, according to the embodiment of this project, the process described above with reference to the flowchart can be implemented as a computer software program. For example, an embodiment of this project includes a computer program product, which includes a computer program tangibly embodied on a machine-readable medium, the computer program including program code for executing the method shown in the flowchart. In such an embodiment, the computer program can be downloaded and installed from the network through the communication part 709, and/or installed from the removable medium 711.

Computer readable media, including permanent and non-permanent, removable and non-removable media, can realize information storage by any method or technology. Information can be computer readable instructions, data structures, modules of programs or other data. Examples of computer storage media include, But not limited to, phase change memory (PRAM), static random access memory (SRAM), dynamic random access memory (DRAM), other types of random access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technologies, CD-ROM, DVD or other optical storage, magnetic According to the definition in this paper, computer-readable media does not include temporary computer-readable media, such as modulated data signals and carrier waves.

For the convenience of description, when describing the above devices, the functions are divided into various units and described separately. Of course, in the implementation of this application, the functions of each unit can be implemented in one or more software and/or hardware.

This project is described with reference to flowcharts and/or block diagrams of methods, devices (systems), and computer program products according to embodiments of this project. It should be understood that each flow and/or block in flowchart and/or block diagram, and combinations of flows and/or blocks in flowchart and/or block diagram can be realized by computer program instructions. These computer program instructions may be provided to the processor of a general-purpose computer, a special-purpose computer, an embedded processor or other programmable data processing equipment to produce a machine, so that the instructions executed by the processor of the computer or other programmable data processing equipment produce means for implementing the functions specified in one or more flow charts and/or one or more blocks of the block diagram.

These computer program instructions can also be stored in a computer-readable memory that can direct a computer or other programmable data processing equipment to work in a specific way, so that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means that implement the functions specified in one or more flow charts and/or one or more blocks of the

block diagrams.

These computer program instructions can also be loaded on a computer or other programmable data processing equipment, so that a series of operation steps are executed on the computer or other programmable equipment to produce a computer-implemented process, so that the instructions executed on the computer or other programmable equipment provide steps for realizing the functions specified in one or more flows of the flowchart and/or one or more blocks of the block diagram.

4. Concluding Remarks

This project provides the application testing method research based on test script. According to the preset interface documents, the environment information, field information and field standard information corresponding to the field information of the application under test can be obtained, and the test characteristic information required for generating test script can be obtained accurately and quickly. It provides the necessary foundation for the automatic generation of test scripts in the following steps, and improves the speed and accuracy of generating test scripts in subsequent steps; By filling the environment information, field information and field standard information corresponding to the field information into the preset test script template, the test script can be obtained. The test script can be generated quickly and accurately based on the test characteristic information obtained from the interface document in an automatic way and simple steps. It does not need the tester to write the test script in person, and reduces the possibility of errors in the test script, so as to improve the speed and accuracy of generating the test script. When the interface document is replaced or the content of the interface document changes, the corresponding new test script can be automatically generated based on the new interface document. No need for testers to modify or rewrite the test script, which further improves the efficiency of the overall application testing. By running the test script to carry out the application test, the application test can be carried out conveniently and quickly, and the efficiency of the application test can be improved. To sum up, the research on application testing method based on test script provided by this project can realize automatic generation of test script based on current interface document, so as to improve the speed and accuracy of test script generation, and then improve the efficiency of overall application testing.

References

- [1] scenario-event-driven research on script language for simulation test of security demanding software system [J]. Yu Gang, Xu Zhongwei, Du Junwei. Computer Application .2010 (02).
- [2] Research on automatic test script generation technology [J]. Jiang Yun, Zhao Jiabao. Computer Technology and Development .2007 (07)
- [3] Design and implementation of embedded software simulation test script language [J]. Yin Yongfeng, Wang Yichen, Liu Bin. Computer Engineering and Design .2006 (12).
- [4] Research on automatic test script technology of engine control software [J]. Yang Xuemin, Jia Shangjie, Jiang Zhichun. Measurement and Control Technology .2017 (12).
- [5] Research and implementation of model-driven test script generation method for embedded system [J]. Huang Song,

- Chen Mingyu, Yan Yunqiang, Bai Yunxiang. Measurement and Control Technology .2020 (06).
- [6] A hierarchical test script technology [J]. Wan Lin, Liao Feixiong. Computer System Application. 2011(07).
- [7] Research on real-time embedded software test script technology [J]. Yin Yongfeng, Liu Bin, Lu Minyan. Computer Engineering .2003 (01).
- [8] Research on automated test script development technology based on SAP project [J]. Huang Sheng. Computer Engineering 2004(S1)
- [9] A test script reuse method for unit test framework [J]. Qi Linying, Hong Mei, Feng Liyun, Zhou Ning, Wen Tingting. Computer Application Research .2013 (06).
- [10] Problems and solutions of software testing tools [J]. Tan Liqun, Cao Wenjing, Liu Yu. Journal of Armored Forces Engineering College .2004 (02).