Research on Intelligent Cloud Test Platform

Huang Zhonghua, Yang Yuting*, Gao Jing and Wang Kunlong
Beijing Institute of Computer Technology and Applications, Beijing 100854, China
*yyttina@126.com

Abstract. With the increasing complexity of software information systems, software quality assurance faces enormous challenges. The software information system is more and more complex, and the quality requirements are getting higher and higher. The traditional automated test methods and tools can not adapt to the changing test requirements. The research on the independent controllable intelligent test platform based on cloud computing and big data will be carried out. The combination of platform and software testing not only reduces test costs, but also improves test efficiency and effectiveness. This paper analyzes the status quo and gaps of the intelligent cloud test platform at home and abroad, and presents the research content and technical solutions of an autonomous and controllable intelligent cloud test platform. First, we make research on the rapid construction technology of large-scale test environment for heterogeneous independent platforms; that is, mirror virtualization and rapid combined deployment technology of software testing tool environment; secondly, we study complex business scenario model generation technologies, including domain analysis and modeling of information systems, formal definition of model-based complex business scenarios and rapid construction technology; finally, we study the automatic generation technology of test cases for complex business scenarios, aiming to make full use of complex business scenario models and full-link historical test data.

1. Introduction
As a multiplier of system capabilities, information systems have made outstanding achievements in recent decades, such as integrated command platforms, weapon deduction systems, data engineering, and combat mission planning systems, which have all achieved leapfrog development. Also, our country vigorously advocates the construction of national defense information systems to improve the security of software information systems. The construction of localized information systems has entered a period of rapid development. With the integrated and intelligent development of software information systems, the scale of software information systems and the degree of systemization of software systems have continued to increase.

As the intelligent core of software information systems, the quality of software is related to whether the entire information system can successfully complete combat missions. As a key means to ensure software quality and reliability, software testing has always attracted attention[1]. As information systems become more complex and require higher quality, traditional automated testing methods and tools can no longer adopt to the requirements of current information system software testing, exposing many problems that need to be solved. First, the software testing infrastructure is slow to build, time-consuming, and inflexible; secondly, the testing process may also encounter problems such as the difficulty of testing resources to meet the task needs.
The intelligent test platform based on cloud computing and big data can provide a test environment that can be deployed and configured in the cloud for various test verification needs, which can greatly reduce cost investment[2-3]; Cloud test can run test cases concurrently to speed up the test process; In addition, cloud test can also predefine standard mirror files and testing services for various testing environments, providing a unified and standardized testing method. The intelligent test platform based on cloud computing and big data can effectively improve test efficiency and effectiveness, and ensure the quality of complex software information systems[4-5].

2. Related work
Oriented towards new test patterns, major cloud computing vendors in the industry have launched their own cloud computing services and test platforms, and the academic community has also put forward their own theoretical methods and technical results.

2.1 Complex business scenario modeling
There are two mainstream modeling methods: Formal-based modeling method and business process-based modeling method.

①Formal-based modeling method means that an ideal modeling method should have formal semantics to facilitate analysis and verification. At present, modeling methods with a good formal foundation include Petri net-based methods and rule-based methods.

Petri net is a modeling tool with strong mathematical theory support and suitable for modeling discrete systems. Using Petri nets, the workflow can be clearly and strictly defined. Most of the properties and theorems of Petri nets can be used for workflow feature description and performance analysis, such as reachability, boundedness, security, liveness, strong connectivity, etc. Advanced variants of Petri nets, such as colored nets, time nets, hierarchical nets, and object nets, can describe and analyze complex workflows.

The most researched rule-based method is the event-condition-action ECA (Event/Condition/Action) rule. ECA rules are derived from active database technology and are currently adopted by many workflow projects to describe the control flow relationship between activities. In addition, the rule-based method can support the dynamic modification of the process, so that the model is flexible. However, as a low-level implementation mechanism, ECA rules lack graphical means and structured construction methods. In addition, when the rules increase sharply, the processing of the rules becomes a very tedious task.

②There is a key concept in the business process-based modeling method: Artifact, which is the carrier of data in the business process. Artifacts are created, updated and archived in the business process to achieve business goals. The description of Artifact includes two aspects: information model and life cycle model. The information model describes the data structure of Artifact, which is similar to the database model; The life cycle model describes the operational steps of Artifact to accomplish business goals. Artifact is the key data in the business process. The establishment method of the business Artifact information model is similar to the data modeling method in the database, but the focus is different. When establishing the database model, the main purpose is to manage the data, using the E-R (Entity-Relationship) model to give the entity types and their connections, and further divide the data through the standardized theory. When building an Artifact information model, the main purpose is to complete the business process. Artifact contains all the data needed to complete a business process. Each stage of the process corresponds to part of the data in the Artifact.

When traditional workflow modeling methods model business processes, data is in an auxiliary position in the modeling process. However, the actual business process is often based on a document containing key business data as the center to complete various business operations, so the data-centric business process modeling method is more conforming with the actual business process operation mode. The Artifact-centric modeling method combines data and processes, emphasizes the importance of key business data in the modeling process, also models business processes based on changes of key data in the business process.
2.2 Cloud test framework

The construction of cloud computing test platform is to construct a large-scale platform architecture dedicated to software testing through cloud computing technology.

Many researchers in our country have explored cloud test platform architecture and testing service (TaaS) mode and gradually developed a new type of testing mode centered on the testing in cloud computing, related research on "cloud" testing, and migration testing to "cloud". Chen Tienan proposed the large-scale performance testing service platform Bench4Q in response to the 1:20 miniature simulation simulation method in the traditional performance test, which may cause many serious problems. The use of cloud computing technology enhances the control and management of the load injection agent pool, reduces the cost of performance testing, and can obtain the performance of the platform under normal and peak test conditions.

The representative cloud test service providers are shown in the table 1.

| Name                    | Cloud Test Service                                                                 |
|-------------------------|------------------------------------------------------------------------------------|
| AlldayTest              | Released AlldayTest V3.0, providing users with functional testing and performance testing services. |
| YiTAO                   | Based on KVM and sheepdog distributed storage clusters, apply for and use virtual machines for testing according to user needs. |
| Testin                  | Released RealTestTM, RealAutoTM and RealMonitorTM to provide users with terminal cloud test services based on the real terminal device environment. |
| MTC                     | Provide cloud test, cloud debugging, cloud audit and other testing services to facilitate real-time development and testing of mobile applications. |
| YICEYUN                 | Provide real-machine automated cloud test services such as adaptation testing, performance testing, functional testing, and traversal testing for mobile App products. |
| ALM                     | Shanghai Zezhong ALM cloud test, including automated testing, performance testing, mobile APP testing and test management tools and other service platforms. |
| China Aerospace Recognition cloud test platform | China Aerospace Recognition cloud test platform, including test tool rental, software online development and testing, test task supervision, etc. |
| Cloud Test              | Provide multiple browser platforms, any web application, provide function, performance, load, regression test. |
| ThoughtWorks            | Provide a real and reliable working environment for mingle project management and collaboration; Provide cruise management and continuous integration tools; Provide twist automated testing collaboration platform;Provide Selenium open source testing framework. |
| Keynote                 | Release kite tool to provide users with performance test standards throughout the life cycle of Web applications. |
| **SOASTA** | Publish CloudTest to test any web application. |
|------------|-----------------------------------------------|
| **Skytap** | Provide users and software providers with services to develop, test, migrate and evaluate applications in the cloud. |
| **Intel**  | Proposed iLab, which not only supports software platforms and services, but also supports the verification design of physical products. |
| **Spirent**| Propose a PASS test method to verify the quality of service (QoS) of cloud infrastructure and the quality of experience (QoE) of cloud services. |

As shown in the table 1, domestic cloud test platform such as the mobile cloud test center of Baidu MTC provides fully automated testing services for mobile developers, covering more than 160 mobile devices from mainstream manufacturers, facilitating developers to perform real-time mobile application testing, developers can also choose different testing modes for different needs. However, the various cloud test platforms have serious homogeneity in their business scope. More than 90% of the test platforms only focus on two aspects: website stress testing (Such as the testing platform operated by the National Information Center Software Evaluation Center, China Test Cloud and UserReplay) and mobile phone application testing (Such as testin cloud test platform, Banmo cloud test platform, easy test cloud, JD mobile cloud test). However, this type of online test platform cannot satisfy software testing for multiple test types and multiple tested objects. China Aerospace Science and Technology has built a cloud test platform with online rental of test tools and online distribution/acceptance of test tasks.

Foreign cloud test frameworks such as SOASTA-CloudTest provided by SOASTA are products for rapid test creation and real-time analysis, and a lightweight new generation of java unit test framework. Use its convenient reusability to easily build application software testing; Sauce Labs provides enterprises with automatic and manual network application testing services. Sauce's cloud testing uses Selenium, an open source testing solution and browser automation framework, which can perform automated acceptance testing on web browsers. Using Sauce's expanded Selenium-based technology, developers can test all mainstream browsers under Windows and Linux platforms, including IE, Firefox, Chrome, Opera, Safari, etc., and use them to take screenshots and playback of errors during testing; The smart business test cloud in fiiM's Smart Business Cloud solution can help companies establish a secure, dynamically scalable virtual test environment, which is located in the corporate firewall and is part of the corporate private cloud.

2.3 The gap between domestic and foreign
In general, modeling for complex business scenarios is difficult and formal modeling methods are not perfect, its credibility still needs to be improved. The research of software testing based on cloud computing belongs to the frontier research field in the world, and it is in its infancy in China so it has not yet formed a perfect theoretical system. For information system basic software and integrated application testing, build an intelligent test system based on cloud services and big data, provide fine-grained inspection, standardized testing, and intelligent test data analysis mechanisms for typical basic software products, effectively improving software quality and reliability, will be the future development direction of information system software testing.

3. Intelligent cloud test platform architecture
Carrying out research on intelligent test platforms based on cloud computing and big data to combine cloud platforms and software testing can not only reduce test costs, but also improve test efficiency and effectiveness.

The overall technical implementation scheme is shown in Figure 1.
Intelligent cloud test platform technology

Rapid construction technology of large-scale test environment for heterogeneous platforms

Image virtualization and rapid combined deployment technology of software test tool environment

Complex business scenario model generation technology

Domain Analysis and Modeling Technology of Military Information System

Provide test environment

Formal definition and rapid construction technology of model-based complex business scenarios

Provide the tested object

Automatic test case generation technology for complex business scenarios

Software testing knowledge graph construction technology integrating multi-dimensional testing scenarios

Automatic test case generation technology based on knowledge graph

Test case heuristic generation technology based on black box testing

Figure 1. Smart cloud test platform overall plan

As shown in Figure 1, the first is to study the rapid construction technology of large-scale test environment for heterogeneous independent platforms, that is, mirror virtualization and rapid combined deployment technology of software testing tool environment, they aim to study various testing requirements of basic software and information systems, the research includes the use of virtual mirroring to realize the rapid deployment of test tools, test environment and test requirements-oriented flexible combination, so as to break through the test environment and quickly build related technologies, providing a technical foundation for cloud test.

The steps of mirror virtualization technology for software testing tool environments are as follows:

① Define a detailed hierarchical classification directory and naming rules for the mirror according to the type of virtual machine (KVM or Docker), the sharing of the mirror, the type of operating system supported by the mirror, the type of software, the test attribute, etc. Then realize the life of the mirror based on Jenkins Automated management of cycles and the test suites are classified and managed by test types or test requirements.

② Use the Docker tool to write the Dockerfile of the mirror, and use the XML syntax to define a markup language to meet the reuse of single-component mirror code. Then use the single-component mirror code to define the combined mirror. After that, the Dockerfile generates the mirror on the Docker host. The user can use the combined mirror as a test suite and conduct tests.

The rapid deployment technology is divided into two situations:
For the testing of office applications, since the office applications are installed on the KVM virtual machine, the virtual machine can be deployed through OpenStack. At the same time, in order to achieve optimized virtual machine scheduling, the following algorithm can be used:

First, we determine the number of virtual machines to be created according to the current load situation, task time complexity and user task priority. Then, sort the physical machines according to the current physical machine load, and select the physical machine whose load does not exceed and is closest to the load for virtual machine generation. Among them, the physical machine in the off state is calculated according to the load.

For the test deployment of application software, we use Kubernetes for container management. Kubernetes is a distributed solution of Docker. Docker mirrors can be used on Kubernetes to generate a specified number of Pods. Pods contain containers that can be used to build clusters. Since the application software is deployed in Docker, after the combined mirror is made, a cluster can be built based on Kubernetes and the mirror.

Secondly, we study complex business scenario model generation technologies, including domain analysis and modeling of information systems, formal definition of model-based complex business scenarios and rapid construction technology. First, modeling the sub-scenario, which is divided into service layer, functional layer and behavior layer. In the service layer, domain analysts, domain experts, users, technical personnel, etc. are required to communicate and consult existing literature to define application software fields and business requirements, then use these information as service layer characteristics. Analyze the functional characteristics of services in the application software field at the functional layer stage, and define an unambiguous name and description in the field for each function; Behavior layer describe the behavioral characteristics of the function (such as the early behavioral characteristics of the function execution, the execution condition, the post-condition of the function execution, etc.), establish an unambiguous name and description in the field for each behavior characteristic, then identify what the function contains and put the behavior characteristics into characteristics layer.

Then perform meta-modeling, the steps are as follows:

a) Analyze the similarity of sub-scene models in the field of office application software. Take each service in the service layer of the sub-scene model as a unit, and perform similarity analysis in two aspects: first, analyze whether there is similarity between each unit, and aggregate similar service features and lower-level subtrees into a feature tree; Secondly, within each unit, analyze whether the subtrees contained in the feature tree have similarities with the subtrees of other feature trees, and record these similar subtrees.

b) Establish a meta model for each merged feature tree. That is, according to the function and behavior characteristic information in each feature tree, a meta-model is established in the meta-modeling tool to describe the modeling language that meets the characteristics of this part.

Then, model-based formal definition and rapid construction technology of complex business scenarios realize the formal construction method of top-level scenarios and technical scenarios of complex information systems based on model checking technology on the basis of meta-models obtained by domain analysis.

Next, we study the automatic generation technology of test cases for complex business scenarios, aiming to make full use of complex business scenario models and full-link historical test data. We based on the extraction of test knowledge to build a software testing knowledge graph, we also associate test cases and business Scenarios, software systems, problem sheets, etc. Then heuristic test case generation technology is integrated to automatically generate test cases. On the basis of the complex business scenario model generation technology, information system’s demand characteristics and full-link test data, we make research on the information system business model extraction, construct business scenario model data. Then study the combination of business scenario modules to integrate the entire chain Carry out data analysis on road test data and extract test knowledge data to enrich the software testing knowledge graph. The knowledge graph covers requirements, test objects, test requirements and other information.
Further develop test case automated generation technology based on knowledge graphs[6-7] and test case heuristic generation technology based on black box testing.

The automatic generation technology of test cases based on knowledge graphs need to generate the test case template, and first adopt the FODA method to disassemble the business latitude by establishing a scenario model. Then combines the deep semantic model to identify the correlation between different requirements, obtains candidate test cases, and then realizes test case generation based on test case repair and amplification technology.

Test case heuristic generation technology based on black box testing is to selecting test cases based on the principle of comprehensive coverage of each interface parameter of the system. First, a test case table is generated, and a set of test cases that maximizes coverage of various combinations of various interface parameters is realized with the least test cases. Then add test cases according to the following algorithm mentioned in literature[8]:

1. Find a missing item from the missing item table, generate a use case, fill in the value number of the corresponding parameter in the corresponding parameter position, mark other parameter positions as vacant, and remove the missing item from the missing item table;
2. Respectively center on the determined parameter value in the use case generated above, look up the missing table, find the relevant missing item, and fill in the parameter number in the corresponding vacant position. At the same time, delete the corresponding missing item in the table;
3. Centering on the newly filled parameter value, perform the same search as in step 2 until there is no vacancy mark in each parameter position of this use case or no relevant missing items can be found in the table;
4. Output the test case;
5. If there are still missing items in the table, go to step 1.

The flow chart is as figure 2:

![Flow Chart](image)

**Figure 2.** Algorithm for adding test cases
4. Case verification
This chapter takes a student management system as an example to verify the effectiveness of the intelligent cloud test platform in this article. The steps are as figure 3:

1. Define a detailed hierarchical classification catalog and naming rules. According to the module, the student management system can be divided into new students registration management, school student daily management, organization and club management, graduate information management, student performance query and other mirror mirrors.

2. Based on Jenkins to realize the automated management of the life cycle of the mirror in step 1. Through the scripts under the Jenkins continuous integration tool, service mechanisms such as update, recovery, and removal of the mirror are realized.

3. The student management system is a web application with B/S architecture. Its test suite covers system stability test, user concurrency test, file read and write time test, page response time test, browser compatibility test, network transmission speed test, Test cases and case libraries, etc. A standardized test suite management system can be formed by comprehensively sorting out relevant standards and technical specifications, test methods, test cases and cases for office application testing.

4. The above steps group the heterogeneous mirror and test suites, and then they are combined and managed. First define a markup language and define the combined mirror. After the combined mirror is filled, the integrated platform will generate the corresponding Dockerfile and optimize the syntax.
according to it, and then generate the combined mirror on the Docker host. The generated combined mirror can be used as a test suite Carry out tests.

5. Carry out sub-scenario modeling, first communicate with teachers, students, and schools, also consult literature, define the business requirements of the student management system as the service layer characteristics; then analyze the functional characteristics of the student management system, such as freshman registration, daily Attendance, club information management, examination arrangements, etc., and define an unambiguous name and description for each function; then describe the behavior characteristics of each function, and establish an unambiguous name and description for the behavior characteristics.

6. Analyze the similarity of the service layer in the sub-scenario, generate a feature tree, build a meta-model through the feature tree, and quickly build a test scenario through the original model.

7. Based on the extraction of test knowledge, construct a software testing knowledge graph, and then extract business models according to the needs of the student management system (such as high concurrency in class selection, high requirements for real-time performance), build business scenario model data, and pass scenario models Extracting tests is just data to enrich and perfect the knowledge graph, which contains information such as test cases, business scenarios, software systems, and problem lists.

8. Automatically generate test cases based on the knowledge graph or generate test cases based on the heuristic method of black box testing.

It can be seen that the platform can realize complete mirroring and test suite management, scenario modeling, knowledge graph construction, and test case generation and management processes.

5. Summary
The intelligent cloud test system architecture studied in this paper which is combined with cloud computing, big data and other technologies can fundamentally solve the problems of traditional software testing architecture. The system is of great significance to the advancement of intelligent software testing, it can effectively use the latest achievements in the current knowledge engineering, big data analysis, artificial intelligence, natural language processing and other fields to provide a new and practical method for the intelligent analysis and efficient use of information system test results. It also can effectively improve test quality and testing and will further promote the development of related fields.

References
[1] Xiao Mingqing, Yang Zhao, Zhao Xin. The concept of cloud testing and its application exploration [J]. Computer Measurement and Control, 2016 24 (1): 1-3.
[2] Zhao Tao. Exploration and Research of Software testing Technology based on Cloud Computing [D]. Xi'an University.
[3] Zhang Yaping. Research on key technologies of software testing based on cloud computing[D],2016.
[4] Liu Jihua, Ma Yuechuan. Research on software testing technology based on cloud computing[J]. Journal of Luliang University, 2014, 4(2): 24-26.
[5] Wang Xiaohong. Research on computer software testing technology based on cloud computing[J]. Journal of Anyang Normal University, 2018, No.112(02):75-77.
[6] Sun Hui,Zhang Shumei. Automatic generation technology of software test cases based on UML[J].Journal of Changchun University of Science and Technology,2006,29(3):58-60.
[7] Tan Xin,Peng Yaopeng,Yang Shuai,Zheng Wei. Research on Automatic Generation Technology of Test Cases Based on SPEA2+SDE Algorithm[J].Computer Science,2015,42(11):450-453.
[8] Nie Changhai,Xu Baowen. Automatic generation algorithm of black box test cases based on interface parameters[J].CHINESE JOURNAL OF COMPUTERS,2004,(27(3):382-388.