

A Testing Framework of Adaptive Web Application Based on Multi-agent Collaboration

Yakun Huang^a, Wang Yang^b, Mingxing Wang^c, Jiajing Wu^d

School of Mathematics & Computer Science, Anhui Normal University, Wuhu 241000

^ahyk_it@foxmail.com, ^bwycap@126.com, ^cwujj@foxmail.com, ^dwangmx@foxmail.com

Keywords: Multi-Agent; Agent Collaboration; Adaptive; Web application testing; Selenium.

Abstract. With the increase of the Web applications, efficient and reliable test has become a practical necessary to improve the reliability of Web applications. Due to the dynamic and uncertain characteristics of Web applications, traditional testing methods are hard to make good performance. Agent could be adapted in the distributed environment which are intelligent and initiative. An adaptive Web application testing framework based on Multi-Agent collaboration theory (ATFBMA) is proposed in this paper. ATFBMA satisfies the dynamics and uncertainties of the adaptive Web application testing. Meanwhile, layered and role-divided Multi-Agent architecture is constructed. Agents are divided into different roles depending on the responsibilities and capabilities. The architecture supports the test requirements of function and structure, and improves test efficiency through the division and distribution of test tasks. A user management system is used to verify the practicality and intelligence of ATFBMAC.

Introduction

With the rapid development of Internet and software technology, Web application software develops sharply based on B/S architecture. And it has been widely used in many fields for the characteristics of interactivity and universality [1,2]. Meanwhile, it becomes an important platform for information transmission and interaction. Since the scale and complexity of Web application platform is increasing, user pursues the higher experience while the requirements of the software are getting higher and higher [3,4]. Thus, it is significant to improve the quality and efficiency of Web application.

Currently, the main researches of Web application test are divided into testing methods, testing models and frameworks. The testing methods include testing tools, using case generations, and the researches of specific verification method. The researches of testing model concentrate on establishing the abstract detection model and constructing the concrete testing framework.

In testing methods, McMaster [5] presented a new tool to develop a source code test browser, which searches for defections through feedbacks from the running state of programs. The testing methods based on user's session are proposed by Elbaum [6]. They obtained the concept of case and behavior analysis of conversations without giving a specific and effective case generation algorithm. As for testing model and framework, researchers have modeled all aspects of the system from different angles. F. Ricca [7] uses reverse engineering methods to construct Web application model. The model is not suitable for Agile Development which depends on the document. A new model was put forward by Kung [8] which was combined with dynamic and static interactive action based on the standard modeling language. In the Web application testing, Agents is mainly used to achieve intelligence, they could be cooperated with each other, and realize the consultation and efficient implementation of the task. Meanwhile, some scholars have added Agents to the distributed detection [9]. Dynamic evolution problem of intelligent body analysis was proposed in literature [10]. They solved the dynamic changes in the distributed environment through viewing the software unit as the agent. And these methods did not divide the work processes among agents in detail. Based on the summary and analysis of the above studies, it is significant to design an intelligent testing framework, which could solve the complex problems that are hard for traditional scenarios.

Related Theory and Technology

Adaptive Testing Theory. Web applications have to resubmit the new test cases to the test library, and constantly reorganize the task plan with the dynamic and uncertain natures. Thus, adaptive mechanism is adopted to learn the changes of environment in time and adjust the test strategy dynamically. Fig. 1 shows the adaptive testing framework.

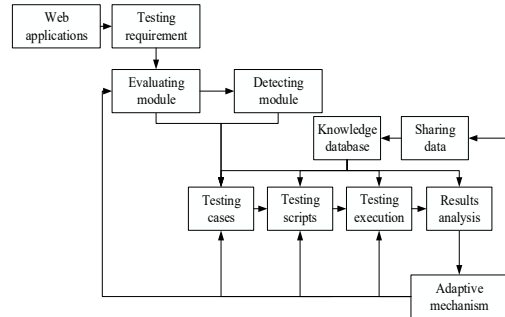


Figure 1. Adaptive Web testing framework.

The main function of this framework is to organize the testing requirements of Web applications. It checks the validation, generates the dynamic test cases and executes test scripts via the evaluation and detection modules. The whole process of adaptive mechanism provides real-time feedback to the various components, which achieves continuous adjustments and learns how to strengthen the test strategies.

Cooperation Mechanism of Mulit-Agent System. Multi-Agent system is widely used in various fields, such as electronic commerce, network technology and language processing, etc [11]. Cooperation is one of the types of Agent. The mechanisms of the cooperation are as follows: (1) Clear the relationships among the Agents. (2) Pursuit the highest interest and efficiency. (3) Enough knowledge to solve the problem. (4) There is no other historical records for the first time. (5) Agents negotiate the exchange proposal. (6) The environment changes after Agent operation.

The specific organizational structure is divided into non-core and core management. Specifically, the structure of non-core management is that the main members of the cooperative group spontaneously produce cooperative intention and form a cooperative group. As for core management, it is the core manager of the whole process, and the responsibility is determining the cooperation method and cooperation strategy.

Adaptive Web Application Testing Framework Based on Mulit-Agent (ATFBMA)

Logical Model of ATFBMA. In Fig. 2, an adaptive logic model based on Multi-Agent is proposed for Multi-Agent collaboration.

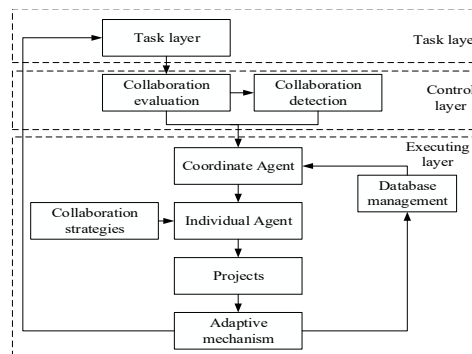


Figure 2. Adaptive Web testing framework.

The main steps of the model are as follows: (1) Organize task plan via the demands of task. (2) Judge whether the work should be coordinated through the evaluation of the task. (3) The same

behaviors could be achieved among Agents with the cooperation strategy. (4) Break the tasks up, and allocating the breaking task for executing. (5) Feedback the results to the task level and adjust the task plan continuously by the ranks and records of the results.

The Role Introduction of Mulit-Agent. In order to design more intelligent testing framework, we assign the role of Agent into Test Manager Agent, Test Interface Agent, Test Assign Agent, Test Monitor Agent, Test Analysis Agent, Test Run Agent, Test Object Agent and Test Atom Agent. The responsibility of these eight Agents are different, and each Agent could cooperate with each other to achieve the goals.

Adaptive Testing Framework Based on Mulit-Agent.

An adaptive Web application testing framework based on Multi-Agent cooperation is designed in Fig. 3 which uses Multi-Agent's intelligence to perceive the external environment. The whole testing framework is divided into three layers, the task layer, the control layer and the implementation layer. The test framework separates the test business and the object at different levels, so as to reduce the correlation among different modules.

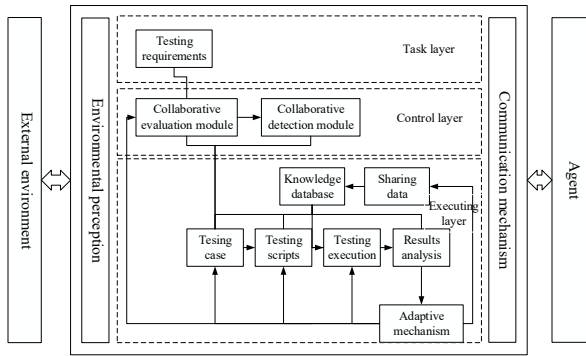


Figure 3. Adaptive Web testing framework.

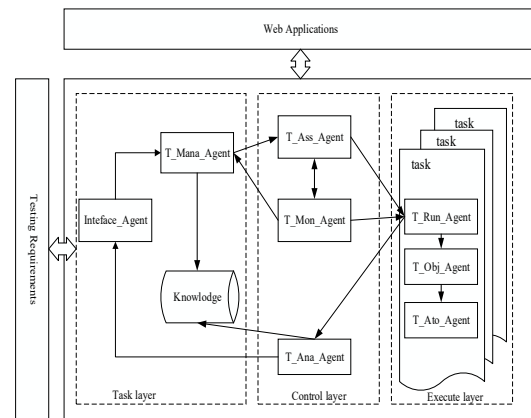


Figure 4. The hierarchical adaptive testing architecture based on Multi-Agent.

The key structure details of ATFBMA model are as follows: (1) Environment perception. It is responsible for the interaction to obtain external environment information, which is to adjust itself and adapt to the external environment. For example, when the Web application changes, the corresponding information will be passed to the environment perception module Agent, and Agent develops the next step in the work plan. (2) Communication mechanism. The database exchange of test data and information among agents are the main communications. In order to improve the efficiency of task execution, the knowledge base is divided into four layers which are the planning area, the task area, the data area, and the result area.

Fig. 4 gives a hierarchical adaptive testing architecture based on Multi-Agent. The responsibilities and capabilities of each Agent are defined in the system. The specific test could be completed by the mutual consultation and cooperation of Multi-Agent. Task layer is composed of Test Interface Agent, Test Manager Agent and knowledge database. And it completes the analysis of the requirements, test plan, data record and update. The control layer is responsibility to complete the work assignment, the work monitoring and the result analysis. As for the executive layer, it is composed of Test Run Agent, Test Object Agent and Test Atom Agent. Agent could be run independently and work with each other to complete the task. A single Agent is similar to the running thread of program. It could run independently or concurrently, which improves the efficiency of testing by realization of a variety of ways.

Adaptive Testing Process of ATFBMA.

The specific test procedure is as shown in Fig. 5 which introduces the detail test procedures.

STEP 1. Test Manage Agent arranges the testing plan via the testing requirements. Meanwhile, more comprehensive using cases are designed through the specific requirements, modules and business logic.

STEP 2. We should test script development, record the new testing script with the testing tool, and package the testing script in the form of the object library. Test Object Agent and Test Atom Agent package the test pages and elements, and then integrate them into a complete testing project.

STEP 3. Test Run Agent executes the test, and runs the defined testing scrip. Next, they test the pages of Test Agent Object and elements of Test Atom Agent in the object library. At last, it outputs the testing reports and logs.

STEP 4. Test Assign Agent allocates the testing tasks through the transitive testing plan of Test Manage Agent, and assigns to Test Run Agent for executing intelligently.

STEP 5. Test Monitor Agent is responsible for monitoring and real-time feedback via the implementation of Test Run Agent. It also returns the progress and results to Test Manage Agent and Test Assign Agent in real time.

STEP 6. Test Analysis Agent analyzes the testing reports and logs. Then, it analyzes and gives the real-time feedbacks via the testing results, which records the results in the knowledge database to generate a new plan.

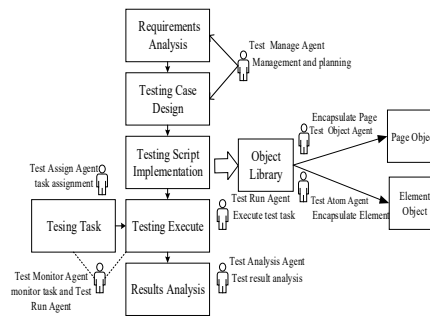


Figure 5. Adaptive testing process of ATFBMA.

Actual Case of ATFBMA

Multi-Agent cooperation structure is designed in view of the actual Web application. Each task is performed by a Test Run Agent, and the specific task could be combined by Test Object and Test Atom Agent. Test results are demonstrated through a visual graphical interface and detailed test reports. The testing process of the whole user management system verifies the intelligence and practicability of the adaptive framework.

Testing Project Introduction. In this part, the multiple broadcast controlled by center application software is used as the testing object. It mainly includes nine function modules, the project flow chart is shown in Fig. 6.

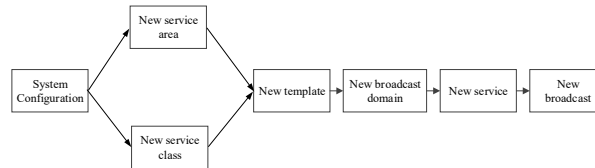


Figure 6. Project flow chart.

Testing Implementation Process.

(1) Testing Case Design.

The functions of Web application are tested and verified, and the quality of the system is proved by verifying the functions and operations of Web application. Testing contents include the Web interface loading time, form, request response and etc. Test case design is mainly in accordance with the functional modules and business logic. TestNG is to achieve the Keyword Driven which can flexibly set test cases through “include” and “exclude”. Complex test cases could be used to implement grouping and concurrent testing with Groups.

(2)Language Selection and Agent Object Structure.

In user management system, Test Run Agent is divided via the characteristics of pages. Pages and Test Object Agent have one to one corresponding relations. Each page is encapsulated into an object with the Object Agent. And Test Atom Agent, elements or objects should be relative.

(3)Testing Environment and Configuration File Settings.

Before developing testing scripts, you need to configure the testing project development environment. The tools we used are Eclipse, Maven, Git and Log4j. Firstly, you should determine the browser type and version information according to the demands. Meanwhile, the address of Web application and server address should be determined. And if the server address is in remote, you may need to set up a proxy. Next, you need to install the JDK and Maven operating environment when you build the local development environment. Finally, the testing case management information should be set up.

(4)Testing Script Implementation.

Testing items are run primarily by scripts. The main tool is Selenium which records the script program. The specific process is exhibited in the Fig. 7.

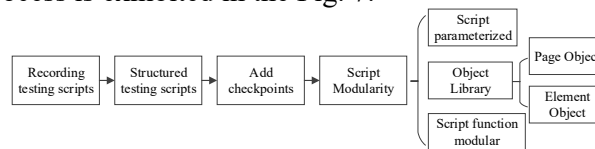


Figure 7. The process of testing script implementation.

(5)Testing execution and results.

There are two methods to run the testing cases. One is directly executing the testing xml in eclipse, and testing xml stores the testing cases. Another one is commanding which runs directly in the window for the setup configurations. You could directly set in the configurations through the different demands. For instance, it is easy to include keyword in the test file in the process of testing the Web application on the two server. If the page is too slow to load, we can set the waiting time for a few seconds. If loading is failed, the excepted codes will be generated in the detail testing report.

ReportNG tool is used to generate the testing reports. It is an extension of TestNG report. It is more beautiful and legible compared with TestNG report. Several testing cases are designed which aim to the testing requirements of the user manage system. And the running result is shown in Fig. 8. The left side is the testing component which mainly includes the five groups. The right is the corresponding testing results. You could see the details of the operation of the usage of the process by clicking on the specific using cases. Meanwhile, the log records could be seen through the LogOutPut as shown in the Fig. 9.



Figure 8. Testing reports.

The testing framework achieves more efficiently by combining different testing cases. The concurrent testing can be performed on different machines according to the specific testing demands. It is necessary to adjust the testing strategies and testing tasks dynamically based on the testing results.

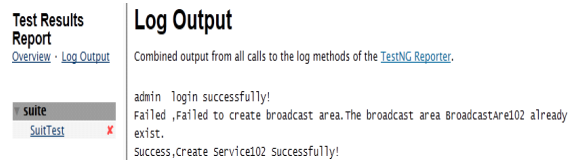


Figure 9. Log records.

Summary

A testing framework of adaptive Web application based on Multi-Agent collaboration is put forward. And we apply it to the practical system to validate the correctness and intelligence. However, it has the shortages in the detail functions. The next work could concentrate on the following aspects. Firstly, for each frame module, the researches and analysis should be carried out and the more reasonable and intelligent techniques should be put forward for improving the various modules and achieving the more intelligent testing framework. Next, a more intelligent Multi-Agent system is built. The system includes the realization of the mechanism of reward and punishment of the task division, efficient and convenient communication mechanism and mutually beneficial cooperation among Agents, which could achieve more efficient completion of the task.

References

- [1] Habibi E., Mirian-Hosseiniabadi S.H. Event-driven web application testing based on model-based mutation testing [J]. Information & Software Technology, 2015, 67(C): 159-179.
- [2] Serdar Doğan, Betin-Can A., Garousi V. Web application testing: A systematic literature review [J]. Journal of Systems & Software, 2014, 91(1): 174-201.
- [3] R. Girgis M, M. Mahmoud T., A. Abdullatif B., et al. An Automated Web Application Testing System [J]. International Journal of Computer Applications, 2014, 99(7): 37-44.
- [4] Serdar Doğan, Betin-Can A., Garousi V. Web application testing: A systematic literature review [J]. Journal of Systems & Software, 2014, 91(1): 174-201.
- [5] McMaster S., Yuan X. Developing a feedback-driven automated testing tool for Web applications[C]. International Conference on. IEEE, 2012: 210-213.
- [6] Elbaum S., Rothermel G., Karre S., et al. Leveraging user-session data to support web application testing [J]. Software Engineering, 2005, 31(3): 187-202.
- [7] Ricca F., Tonella P. Construction of the System Dependence Graph for Web Application Slicing [C]. IEEE Computer Society, 2002: 123-123.
- [8] Kung D.C., Liu C.H., Hsia P. An object-oriented web test model for testing web applications [C]. Proceedings. First Asia-Pacific Conference on. IEEE, 2000: 111-120.
- [9] Zhang H., Feng T., Yang G.H., et al. Distributed Cooperative Optimal Control for Multi-agent Systems on Directed Graphs: An Inverse Optimal Approach. [J]. Cybernetics IEEE Transactions on, 2015, 45(7): 1315-1326.
- [10] Li Q.S., Wang L., Chu H., Zhang M. Agent-Based software adaptive dynamic evolution mechanism [J]. Journal of Software. 2015, 26(4): 760-777.
- [11] Li, Jinsha, and J. Li. "Coordination control of multi-agent systems with second-order nonlinear dynamics using fully distributed adaptive iterative learning [J]. Journal of the Franklin Institute 2015, 352(6): 2441-2463.