2010 3rd International Symposium on Knowledge Acquisition and Modeling

Research on Automated Testing Framework Based on Ontology and Multi-Agent

Chongchong Zhao[1], Guoxin Ai[1], Xiao Yu[2], Xiaofeng Wang[3]

[1]*School of Information Engineering*

University of Science and Technology Beijing, Beijing, China

[2]*China Academy of Space Technology*

[3]*China National Aero-Technology Import & Export Corporation*

aiguoxin2008@gmail.com

***Abstract* - To solve the problems of software testing automation,such as poor scalability, compatibility and other problems, this paper puts forward a automated performance testing framework based on ontology and multi-Agent. An extensible BDI Agent model is the core part of this framework. The dynamic loading in BDI Agent improves the inclusiveness and expansibility of automatic test. On this basis, a software performance testing interaction model based on multi-Agent is proposed. At last, the performance and feasibility of this model by using ontology technology based on JADE is verified through a prototype system.**

Index Terms* - *ontology, agent, BDI model, JADE, testing framework.

I. INTRODUCTION

With the software expanding, the scale and complexity of the dramatic increase of software, more and more software producers and users start to attend to software testing. The current manual software testing can't meet the growing demand, so software producers prompted development of automated testing tools. In order to improve the quality of software, we usually concern about the software quality and accurate evaluation. Currently, there are some testing tools mature such as: Load Runner, QALoad, etc., but there are very expensive, and it can only meet the demand for certain types of applications [1].

Automated testing framework includes test configuration, test execution and test management. Testing framework in the whole, testing process plays an important role and affects the success of automated testing. Currently, the testing framework of the study, functional testing framework generally two components, and one is the script generator; the other is the script execution and management components. Script execution and management components include the definition of test tasks, issued, test results collection and analysis. The current framework for automated testing studies have focused on the generation and reuse the script, script execution and results collection and analysis, but the lack of distribution of the test tasks, arrangements[2].

The current testing framework assignments are carried out manually. First of all, testers make testing task deco posited; the numbers of interdependent tasks are classified into a group, the distribution of tasks as a minimum unit. This task of testing is made up of a series of task modules. Then, the tester through the test tool manually defined task allocation model, and generate the relevant documents. Finally, the test tasks by the form of the document are sent to the test execution nodes [3]. Distribution of this manual, is issued by the task the way, on the one hand is inefficient; the other hand, it may lead to unreasonable allocation of resources. This paper presents a basic ontology and multi-Agent framework to automate testing and validation through prototype system, each function of Agent are based on the BDI model to control the life cycle, and implementation tasks issued are based on the capacity of the node. Technical coordination between the nodes is through the body of work, according to system resources, etc. to make the appropriate adjustments. The needs of the task by testing the ability and test node, the node automatically is selected a reasonable and allocation of tasks. This paper is organized as follows: Section presents an overview the entire testing framework; Section III puts emphases on the application of ontology and test rules; Section IV introduces a prototype system based on JADE; Section V rounds up the conclusions of this research effort.

II. TESTING FRAMEWORK OVERVIEW

The testing framework uses hierarchical architecture, that is, different layers and Knowledge Agent layer.

With the Agents in accordance with their knowledge base to complete the dynamic coordination of the user's

978-1-4244-8005-0/10/$26.00 ©2010 IEEE

designated task. The chart is shown in Fig.1.

Fig. 1, testing framework graph

Throughout the testing framework for each test is different from Fig.1. Agents are under the different roles to accomplish the task. Agents are divided into five main roles, namely Controller Agent (center machine), Recorder Agent (Script Builder), Replayer Agent (load generator), Monitor Agent (monitor) and the Analysis Agent (Data Analyzer).

Agents are through the different roles between the communication and collaboration, to complete the testing process. The Agent-based BDI (Belief, Desire, Intension) model, are through the communication and collaboration model to address the test group of users within and among the various tasks to improve test automation.

BDI model is composed of beliefs (Belief), desire (Desire), intention (Intention). The model is mainly used to solve the problem that is how to determine the subject (Agent) goals and how to achieve this goal.

Definition 1: Agent of the BDI model is a five-group.

$$M = < T, ACT, BEL, DES, INT >$$

Where T is the set point in time, ACT is a set of atomic actions, BEL is the belief, DES is the desire, and INT is the intention. According to a particular point in time the set of atomic propositions established to determine the time point of belief, desire and intention, and then act accordingly.

BDI model using multi-Agent test ,First of all, according to testing requirements through use case knowledge base, describe the different roles of each Agent to achieve different objectives; Then, the Agent with the provisions of the definition of an approach to the design of each Agent, Agent state according to their own belief, desire and intention of the state, Implementation of a series of operations, without affecting the target to complete the same to communicate with other Agent and collaboration; Ultimately test automation.

In the testing process, the testing framework used five main roles and functions of Agent. First use of BDI model defines a common base class Agent, the role of Agent inherited from the base class, and realized their corresponding functions. Such as: Controller Agent that central machine is a core part of the test platform, the demand for access to user testing, coordinate the operation of the test system. According to the definition of a design is as follows: where T contains each playback Agent ready to playback point in time (mainly for the collection point), etc. ACT includes a playback device issued the script commands to monitor control orders issued; Agent BEL including playback information, monitor Agent information. Agent DES include all normal operation, coordination of work between Agent; INT Agent, including the coordination task.

Compared with the previous test framework, all the test tasks are different from the ability of Agent done. Each Agent is relatively simple, but it is overall really strong and extensible. It is because of the testing framework for a more detailed division Agent role, which is also more diversified portfolio. For different testing requirements, they can build a suitable combination of tasks to complete the corresponding tests. The same role as Agent has similar functionality, but different Agents can use different programming language and can also be on different computers and system platforms. This makes the system more flexible integration of different functional components of Agent to improve the system scalability and flexibility.

III. Ontology-based Multi-Agent Interaction Model

Agents' interaction between the signal layer and symbols, includes the interaction layer. Signal level that is the message interaction, and symbolic level refers to the word semantics and domain knowledge. As each Agent may be deployed in different systems, ontology is as semantic understanding to solve the problem, so they need a public ontology [9] to represent knowledge in the field test. Each agent needs to achieve local knowledge base and testing in the field of semantic consistency between the shared ontology conversions to get knowledge interaction, ultimately agents can effective communicates and collaborates with each other.

Xml format is used by the local knowledge base [4] to maintain, to facilitate data exchange between the various agents and communications. Test domain ontology defines a set of concepts within the field, words and hidden knowledge. According to the test demand, the paper used in agent communication environment for formal ontology is defined as follows [5]:

Definition 2: The ontology O is defined as a triple, in which the concept of class C that the set of concepts expressed by Ci to describe things in an abstract; A collection of classes that action, action Ai that the concept class C, concept of operation of one or more of the collection; R is said that the concept of class C and class A relationship between the action. According to the ontology definition, structure-related body and based on the characteristics of tests constructed ontology-based multi-Agent interaction model diagram shown in Fig.2.

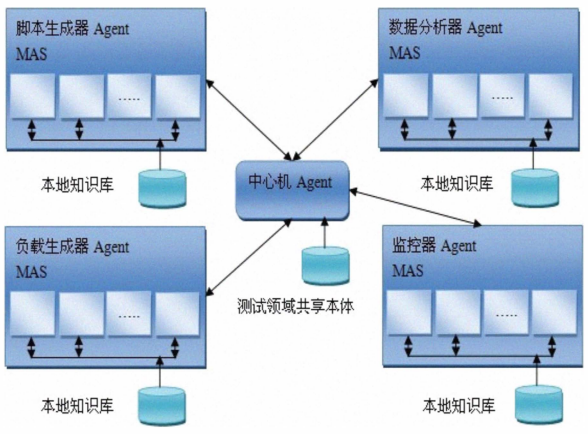

Fig. 2, ontology-based multi-agent Interaction model graph

Fig.2 Agent as the basic unit, according to different building MAS (Multi-Agent System), each MAS Agent roles is composed by the same kind of composition. MAS have a different function in each of the corresponding local knowledge base to provide the same functionality of the agent role of collaboration between MAS in different functional areas and through the center plane agent field testing different functional areas of shared ontology to achieve semantic consistency between the conversion, to achieve the field of communication and collaboration between the agents. According to the above definition of the ontology, Multi-Agent Model and interaction between the characteristics of the test, the following constraints and rules [6]:

Process of implementation of the constraint, interactive model is an abstraction in the implementation of Africa when the host computer playback agent resources can't be occupied by more than a certain proportion. Or they may test environment. The process of implementation of the constraint can guarantee that the relative accuracy of test results.

Portfolio construction process of the rules: $\forall p \in (C \cup A), O = composeOf(p).$ Used for testing body such as: Playback body (Replay Ontology), in the process of building the ontology, ontology composition subject to the requirements of the definition of 2.

If-Then-Else construct rules: $\forall a \in Ai, a \neq null.$ In the If-Then-Else construction on then the corresponding control constructor can't be empty. For testing, such as when the host computer playback agent shares resources exceed a certain percentage, the notification center machine, the center machine rules on the use of the building to take corresponding measures, can't be NULL, or not respond.

IV. Prototype SystemBased On JADE

Based on the above, a prototype system based on JADE was designed. In order to verify the feasibility of the testing framework, we designed and implemented a prototype system to test the software. Agent systems are distributed in different roles in more than one computer.

The prototype system uses the java language to develop, agent based on the JADE platform [7], agents' communication between the using of FIPA-ACL language, content with the SL (semantic language) described and encapsulated using XML. Communication's process of ontology construction is based on the definition of 2. To playback the following body (Replay Ontology) implementation of an example, Replay Ontology body mainly by the concept of Script and VUGroup and action ReplayAction, the relationship between actions and concepts that are replay playback. The chart about Replay Ontology is shown in Fig. 3:

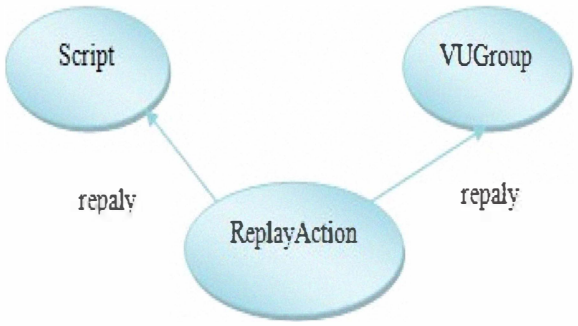

Fig. 3, replay-ontology graph

To verify the prototype system, this choice of a Web system login module for the object to be tested. During testing, implementation constraints and the *If-Then-Else* construct rules playback agent deployment machine is well reflected, Agent assumes that when the playback on the unit found in over 50% of CPU, they would send a notification center machine, center the next step is to machine the decision to perform testing or decreasing the amount of the Agent on the concurrent, or the longer period of observation, to see an incident or sustained long occupation of CPU. In addition, set the script set-point strategy, the use of *If-Then-Else* construct rules is when all Vuser(visual user) collection of X% to reach point, the center machine to release all of the Vuser. Of course, there are other logic systems are based on the realization of the above constraints and rules, not list them one by one. Management systems using the JADE platform can see different agents' deployment shown in Fig.4.

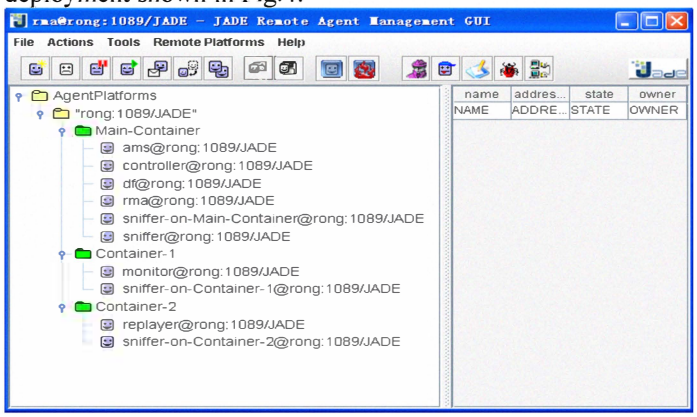

Fig. 4, agent deployment graph

In the testing process, only the system resource consumption was monitored from monitoring data were show in Fig. 5:

序号	CPU利用率	内存利用率	交换区利用率	传输总量	接收总量
1	0.027316294610500336	0.1622626930475235	0.4545297622680664	778457.0	1.1982747E7
2	0.260869562625885	0.162335306406021 12	0.4545767903327942	784110.0	1.1983375E7
3	0.19138756394386292	0.162335306406021 12	0.45467081665992737	789759.0	1.1984155E7
4	0.4114832580089569	0.162330552935600 28	0.45487427711486816	795410.0	1.1985401E7
5	0.1055045872926712	0.162325784564018 25	0.45440441370010376	801063.0	1.1985937E7
6	0.150234743952751 16	0.162325784564018 25	0.4545924663543701	806714.0	1.1986501E7

共[29]条[5]页第[1]页 [首页] [上一页] [下一页] [末页]

Fig. 5, system resource data graph

The prototype system is based on Linux platform [8], it has a good tolerance for script recording and playback tools are used XNEE, if you select other tools such as LDTP, etc. To test the scalability of the prototype system, on the need to monitor indicators added such as: system response time monitoring. To test the scalability of the prototype system, on the need to monitor indicators added such as: system response time monitoring. By BDI model, Agent interaction between the better communication, collaboration, and realized a test automation, a significant reduction in manual testing.

V. CONCLUSION

This article introduces the concept of Agent-test platform, the platform through the use of Agent technology, all agents were designed by BDI Model agent, and the use of XML and ontology technology to solve communications collaboration between agents. And then proposing a multi-Agent based automated testing framework. Follow this framework, the test platform designed with cross-platform, the deployment of flexible, open, and easy to expand and so on. This is a good solution to the current cross-platform testing tool for poor, weak extension, not very good in all areas of software testing to meet the needs of the shortcomings.

REFERENCES

[1] Yiwen Zhu, Chao Qian, Yong Lin. Software Performance Testing Tool Review [J].Financial Computer Of China.2009,7

[2] Xubao Zhao, Xiangyi Geng,Xue Zhang. XML-based message communication mechanism of multi-Agent Research and Implementation [J]. Computer Engineering and Applications.2008,44(31):108~111

[3] Lei Xu,Baowen Xu, ”A framework for Web applications testing.”, Proceedings of the International Conference on Cyberworlds .2004,pp.300-305

[4] Xiaoming Zhang, Changjun Hu,Huayu Li, Chongchong Zhao. Relational database to ontology mapping from the Research. COMPUTER SYSTEMS.2009,7;(7):1367~1372

[5] Aihua Bao, Weiming Zhang, Jinping Yuan, Li Yao. Rule-based OWL-S ontology syntax consistency maintenance methods. Journal of Applied Sciences, 2009,31(3):97~103

[6] Fabio Bellifemine, Giovanni Caire,Dominic Greenwood. Developing Multi-Agent Systems with JADE.2007.

[7] http://jade.tilab.com

[8] Junzhao Wang, Chongchong Zhao, Changjun Hu. Linux desktop operating system vulnerability testing of Research. Computer.2007,23(12-3)：71~74

[9] Changjun Hu, Chunping Ouyang.NON-structured materials science data sharing based on semantic annotation.Data Science Journal,vol 8,P52-61,2009.