

Review Paper on Multi-Agent Based Software Testing

Amina Ghufran
P.G Scholar,

Dept. Of Computer Science and Engineering
Shri Ramswaroop Memorial College Of Engineering and
Management
Lucknow, India

Dr. Raj Gaurang Tiwari
Assistant Professor

Dept. Of Computer Science and Engineering
Shri Ramswaroop Memorial College Of Engineering and
Management
Lucknow, India

Abstract— Software testing plays vital role in software development life cycle to give out an accurate, sensible and a secure product. In recent times manual testing replaced by automated tools. But automated testing solution to is not economically efficient as it requires licensing and human involvement so a framework is required to overcome the above shortcoming. Agents have been introduced in this field to work on behalf of manual testing; also do they compensate the above complexities as they cognitively test application in place of testers The paper examines the aspects of software testing as well as the role of multi agent in software testing in addition to the contribution of cloud computing in this region

Keywords— *Software testing, Agent, Multi-agent System*

I. INTRODUCTION

Software testing is used to spot errors and generate an error free program with a better attribute. Software testing requires ample of time and is a toilsome process as development of an error free software needs a number of test cases. Users now days incline towards a secure, sensible and time efficient system. To develop cost effective and efficient testing process, new approaches are required for test automation. The most important prospect of test automation is the automation of test case generation. Although automation yields better result, but it is not profitable as the tools required licensing also to operate tools manual efforts are required which again is a costly process besides the tools are mostly single testing type. Therefore purchasing different tools for different testing types is not prudent especially for small and medium scale industries.

To compensate for all these problems an entity is needed which can smartly test application on behalf of testers and is economically apt. Hence intelligent agent came into play to meet the demand of testing services with some splendid features as autonomy, pro-activity, re-activity and with learning abilities.

A single agent can look up to a given task in an adaptive manner. An agent exemplar or a model was set up as a distributed system, where a group of agents interact one another by sharing information and performing tasks in co-operation with intelligence. If we consider agent paradigm as a system where agents act like entities working in co-ordination with other agents, it would be a manipulation of the agent exemplar.

II. RELATED WORK

Qingning Huo and Hong huo [11] propose an environment based on agent for web application testing. The framework has a light weight agent platform to provide agent communication. It put forth a theory of software testing which facilitates flexible integration of multiple agents and a protocol which takes XML in use as to exhibit the basic and compound concept of ontology. To lessen the communication traffic an agent can be distributed among many systems.

A tool call ACL analyzer [16] process on the JADE [18] platform, It obstructs the transfer of messages among agents and stores them in a relational database. This method take advantage of clustering techniques to prepare agent interaction graph that footholds the identification of missed communication among agents that are supposed to interact data overhead transfer between agents.

Padgham [6] developed an agent interaction protocol to spot the errors automatically from the source code at run time. There is a central debugging agent that examines the communication between agent. It receive a message copy of communication held between agent

D.Lam and k.Barber [7] put forth a structural framework to apprehend the behavior of software agents. The system mimics the actions of human user in software perception. The work did not include the problem like, to generate and execute test cases.

Nunez [10] proposed framework which designate the conduct of sovereign e-commerce agents. The required behavior of the agents under speculation are laid out using new formalism, called utility state machines that suggest user's choices in its states. Two testing techniques were put forth to examine the proper implementation of the specified agent. In the process of active testing, each agent under test was placed under a further test that indicates some formal specifications into the agent and facilitates it to reach a specific state. The operational trace of the agent is then compared to the specifications as to look out for errors. On contrary some authors proposed the use of passive testing in which only the agents under test were keenly observed without any incision special case.

Yin, Jin Zeng [19] focuses on a protocol where specific test agents are developed from abstract classes. All the test agents are assigned for testing a specific type of web document or object. Web application testing is carried out with the help of a set of agents. These set of agents follow the Belief-Desire-Intention model agents

Z.Zhang and J.Thangarajah [5] directed his work towards model based testing using the design models of the Prometheus agent development methodology. Unit in agent system are executed and triggered in a more complicated way unlike the traditional software system. The system aims on testing agent plans and mechanism for developing more suitable test cases and for deciding the orders of the units to be tested

Khalid A.M [9] proposes a Regression testing suit for multi agent based system using intelligent agent. Multi-agent based Regression testing suit grasps the advantages of proving distributed computing solution and capability of making decisions by intelligent software agents.

Zina Houhamdi and B.Atamena [4] proposed a novel approach for goal oriented software integration testing. This technique establishes an integration testing process that counterparts the goal oriented methodology Tropos and Brace, the relationship between goal analysis and testing. The derived test series from the system goals can be implied to examine the properties which result from the agent interactions and assure that the group of agents and contextual resources work in co-ordination. This approach describe a structured and discursive integration test suit derivation process for engineering software agent by implementing analytic way of test cases derivation from goal analysis.

Nicholas [10] verified the system using two classes; logic provides principle for each statement and semantic methodology verify model for testing

The research in succession of test data for structural testing was done by Miller and Spooner in order to enhance the objective function with the use of numerical maximization technique. The first one to imply hill climbing was korel.

Xanthakis [14] was the first who tested the generation of data for path coverage using evolutionary computation. This work has been carried forward by many other authors for branch coverage.

Memetic algorithm [14] for search based testing and evolutionary testing with Hill climbing were put forth by Wang and Jeng.

Harman and Minn [13] compared evolutionary testing, Hill climbing and Random testing. Mansour and salame [14] proposed to compare evolutionary testing, Hill climbing and simulated Annealing for test data generation of path coverage. But Hill climbing is only used to examine object with input as integer. Also Simulated Annealing performs better than evolutionary testing and Hill climbing. The

suggested system applies alternating variable method for structural test case generation.

III. SOFTWARE TESTING

The main objective of software testing revolves around responding to the question like Is the system being developed in an sustainable manner? Is the system efficient enough for the users? Thus testing stands for software directed activity, which is used to have a check on product quality and correcting the defects in order to improve the overall quality.

The aim of software testing is to locate errors in software products that require lots of resources which in turn improves the quality of the product and makes it more sustainable. In testing process decision are made in order to locate errors and reduce the vulnerability of the product. The objective is to achieve the balance between the expected value of errors present and the number of tests to be carried out. It is not cost efficient to perform testing within a certified limit, due to the risk of high expenditures.

Researches reveal that 30%-70% of the resources that are associated with software development are involved in software testing. Due to this extent use of resources result are not satisfactory.

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. TEST LEVELS

Since previous few years, the testing scenario has been changed and is no longer a step which comes into an action only after the implementation phase. Software testing now includes the development maintenance activities. The work that has been done in testing software agents And Multi-agent system can be divided into different testing levels: units testing, integration testing, system testing, acceptance testing.

- Unit testing is used to test all units that are building blocks of an agent like blocks of code, agent implementation unit like target, work plan, knowledge base, analysis, rules specification and ensures their proper functioning
- Integration testing is used to keep a check on the communication and interactions among agents sand their structure; their interaction with the environment, shared resources; integration of agents with regulations enforcement; to have a count of emergent properties, collective behavior and to ensure that a group of agents and the resources of environment perform together..

- System testing test the multi agent system as a whole system running at a desired environment. It test the anticipated developing properties of the system; test the properties regarding the expected quality of the system like accommodation and receptiveness, fault-tolerance and functioning
- Acceptance testing test the multi agent system in user execution platform and verifies that if fulfill with customer expectation..

IV. AGENT

An agent is a software program or a unit that acts for a user or other program in a relationship of agency that is acts on behalf of another unit to achieve a particular objective. Agent systems are sovereign software programs engulfing the domain knowledge and having an aptness to be an independent entity to a certain degree, to achieve specified objectives and goals. They effectively respond to the environmental alteration.

Agents possess some particular set of features. The main features are as follows;-

Autonomy- agents have the ability to act autonomously and to make decision about their actions without the involvement of other system, on behalf of the users or other program in a way to execute best result.

Pro-activity- agents are capable of prosecuting their individual set of objectives by carrying out actions based on their internal decisions.

Re-activity- Agent have the ability to react impetuously to the external events by adapting the change and making decision in order to execute the task effectively.

Communication and Co-operation- Agents have the capability to communicate and co-operate effectively to other agents in the systems by responding, sharing information and receiving instructions to achieve their own objectives.

V. MULTI-AGENT SYSTEM

A Multi agent system is an interconnection of loosely coupled computational entities having sovereignty to perform various activities, possessing resources, intellect abilities and skills. They are located in a common atmosphere and they can communicate through a set of rules, namely an interaction protocol.

The software is designed using the structure of multi-agents based upon the exemplar, which induces the original objects and their behavior. Developers of multi-agent model often opt for the issues that are of more relevance to the complex systems from the real world, and they exhibit simplified versions of the system, like other intricate processes, in software testing too, organization of the process is of immense importance. So that resources which evolve from the internal abilities and awareness of the system can be used effectively hence it can become compatible to the surrounding atmosphere. Accordingly, it is offered that more

focus to be on organizational tenets that are necessary to ordain multi-agent models, In order to make management of testing procedure efficient. This would help to take such issues in observation which we cease to take notice of in daily routine scenario.

A prolonged research has been done to final out the main problems in complex systems and to comprehend the most important principles on which the system operates, various theories and principles have been developed on the basis of the result of the researches done to construct the model of the complex system. This is an incompatible model of real life, possessing only the important features required to replicate the original system. This implicates conversion from the original system to an exemplar of the system which can be adopted for several purposes like to design software.

A. Challenges in Multi-agent system

- Intricacy is increased because of several distributed systems running simultaneously and heteronomously.
- Regeneration effect, since it cannot be ensured that two processes would load to same conclusion even if the same input is applied. Consequently, spotting errors is not feasible if it is impossible to reproduce every time.
- They are non-deterministic as it is not possible to determine actions prior the execution.
- Agent communicate through sending messages instead of summoning fellow agents, making existing object oriented testing approaches inapplicable.
- Agents act autonomously and work in co-operation with other agents, running correctly in isolation but incorrectly in community.

CONCLUSION

The existing research work focuses only agents and their alliance to make software testing efficient. They develop a multi-agent system which integrate intelligently and perform testing on behalf of the testers.

It is under consideration for developers how to achieve a confidence at acceptance level as agent's behavior is autonomous. Agents with their own individual objective are to be evaluated from human tester's point of view. Eliminating side effects in test execution and monitoring due to introduction of new entities like mock agent, tester agent and monitoring agent as in many approaches, can affect the behavior of the agents under test and the performance of the system altogether.

Thus multi-agent system introduces a framework that performs a different kind of testing but some new activities can also be considered like test plan development. In present scenario multi-agent system is implemented on Java agent development framework but we can consider some new platform for implementation.

REFERENCES

- [1] B. Beizer, "Software Testing Techniques", 2nd edition, Van Nostrand Reinhold Co., New York, NY, USA, 1990.
- [2] Z. Houhamdi, "Test Suite Generation Process for Agent Testing", Indian Journal of Computer Science and Engineering, Vol. 2, 2, 2011.
- [3] Z. Houhamdi, and B. Athamena, "Structured Integration Test suite Generation Process for Multi-Agent System", Journal of Computer Science, Vol. 7, 5, 2011.
- [4] Z. Houhamdi, and B. Athamena, "Structured System Test Suite Generation Process for Multi-Agent System", International Journal on Computer Science and Engineering, Vol.3, 4, pp.1681-1688, 2011.
- [5] Z. Zhang, J. Thangarajah, and L. Padgham, "Automated unit testing for agent systems", 2nd International Working Conference on Evaluation of Novel Approaches to Software Engineering, ENASE'07, Spain, pp. 10-18, 2007.
- [6] L. Padgham and M. Winikoff, "Developing Intelligent Agent Systems: A Practical Guide", John Wiley and Sons, 2004.
- [7] D. Lam, and K. Barber, "Debugging Agent Behavior in an Implemented Agent System", 2nd International Workshop, ProMAS, Springer, Berlin, pp. 104-125, 2005.
- [8] M. Nunez, I. Rodriguez, and F. Rubio, "Specification and testing of autonomous agents in e-commerce systems", Software Testing, Verification and Reliability, Vol. 15, 4, pp. 211-233, 2005.
- [9] Khalid A. Muhammadi., "Multi-Agent based Regression Testing Suite" in Athabasca University Thesis, April 2009.
- [10] Nicholas R. Jennings and M. Wooldridge, "Agent-Oriented Software Engineering", in 13th International Workshop on Agent Oriented Conference, June 2012.
- [11] Qingning Huo and Hong Zhu, "A Multi-Agent Software Environment for Testing Web based Applications", in Proceedings of the 27th Annual International Conference on Computer Software and Applications, pp 210-215, 2003
- [12] Miller W and Spooone D. Automatic Generation pf Floating Point Test Data, IEEE Transaction on software engineering. 1976;2(3):223-226
- [13] Harman M. and McMinn P.A. Theoretical and Empirical study of Search-Based Testing: Local, Global, Hybrid Search, IEEE Transactions on Software Engineering. 2010; 36(2):226-247
- [14] Manosure N. and Salame M. data Generation for path testing. Software Quality Journal. 2004; 12(2): 121-134
- [15] M S Geetha Devasena , M L Valarmathi Multi Agent Based Framework For Structural and Model based Test Case Generation: Internationa Conference on Modelling, Optimization and Computing(ICMOC 2012)
- [16] J. Botia, A. Lopez-Acosta, and G. Skarmeta, "ACLAnalyser: A tool for debugging multi-agent systems", Proceeding of the 16th European Conference on Artificial Intelligence, pp. 967-968, IOS Press 2004.
- [17] J. Botia, A. Lopez-Acosta, and G. Skarmeta, "ACLAnalyser: A tool for debugging multi-agent systems", Proceeding of the 16th European Conference on Artificial Intelligence, pp. 967-968, IOS Press 2004.
- [18] TILAB. Java agent development framework. <http://jade.tilab.com/>.
- [19] Yu Qi, David Kung and Eric Wong, "An Agent based Testing Approach for Web Applications", In Proceedings of the 29th Annual international Computer Software and