Web Application Testing Framework using Agents

Amina Ghufran
PG Scholar
CSE Deptt.
SRMCEM, Lucknow

Raj Gaurang Tiwari, PhD
Assistant Professor
CSE Deptt.
SRMCEM,Lucknow

Pankaj Kumar, PhD
Assistant Professor
CSE Deptt.
SRMCEM,Lucknow

ABSTRACT
Testing is a software development activity, devoted to evaluating quality and improving the end product by identifying defects and problems. It's an important aspect in software application development to insure the application quality which include the application's performance, reliability, speed, security and functionality. Testing can be done by automation tools like Win runner, QTP or manually. Manual testing, is another option, but it takes lot of time and manpower. Automated testing has increased costs and most of the times is not affordable for small or middle level organizations. In this paper we are presenting agent based testing which can fulfill the testing requirements on smaller costs. We present an implementation framework of an online application which can be accessed over the web and on payment of small amounts can provide different kinds of software tests for web based applications. The framework uses Multi Agent Systems to manage, perform, report the testing procedures.

General Terms
Testing Framework, Web Application.

Keywords
Software Testing, Web Based Application, Multi Agent Systems(MAS).

1. INTRODUCTION
Software testing plays an important role to make any software project reliable and successful. Software testing is an exercise to simulate a system. Testing provides the program to get the desired goal. Testing analyze a program with the intent of finding problems and errors that measures system reliability. Testing consists of identification of required requirement and design as well as execution test of code. Software agents represent an interesting paradigm to develop intelligent and distributed systems, because of their autonomy, reactiveness and reactivity; in addition, their sociality enables the distribution of the application logic in different agents that can interact together and with the host environment. Agents have been steadily moving into more and more significant applications [3]. Agents are capable to support more naturally the development of software systems whose components are heterogeneous and autonomous. These properties make agents ideally suited to applications in electronic commerce, virtual enterprises, and other open settings [2].Owing to human effort and deviated results, manual testing is being replaced by automated testing tools. But the automated solution is much costly due to licensing and as it still requires human interaction. Hence, such a testing solution is required, in which on demand testing services based on pay as you go model, can be provided. This framework is intended to highlight the role of software agents or intelligent agents in providing such a multi-agent based testing framework, where different agents collaborate to provide software testing as a service on cloud. Different agents groups are introduced, to support different testing types within an integrated framework. The multi agents system, has been developed using the JADE framework in the proposed system.

2. RELATED WORK
Sabih Jamal et.al [1] have proposed an integrated framework MSTAS which provides software testing as service by utilizing shared pool of resources, over the cloud. It makes use of multi-agents which collaborate intelligently on behalf of testers and interact with cloud providers‟ platforms for allocation of resources, required dynamically based on testing requests. Pay as you use model of testing service, makes it most useful for small and medium businesses, where instead of purchasing licensed based testing tool, users can simply pay as they use.

A similar approach has been followed by Khalid A.M. [5] in which they present MARTS Multi-Agents based Regression Testing Suite framework that makes use of intelligent agents for test cases versioning, report and result analyzing. It provides interface for users to access testing history. It only covers regression testing and lacks in covering other testing types.

Yu Qi et al. [6] highlights an approach, specific test agents are generated from abstract classes. All the test agents test a specific type of web document or object. Web application testing is done by cooperation of a set of agents. These test agents follow the BDI (Belief- Desire- Intention) model agents.

Qingning Huo and Hong Zhu [8] present an agent-based software environment for testing web-based applications. The infrastructure of the system consists of a lightweight agent platform that supports agent communication, an ontology of software testing that enables flexible integration of multiple agents, and a formalism using XML to represent both the basic and compound concepts of the ontology. Relations between testing concepts are defined and their properties are analyzed. A number of agents have been implemented by the authors implemented to perform various tasks in testing web-based applications. Broker agents use the ontology as a means of inferences to manage the knowledge about agents and assign each task to the most appropriate agent.

Coordination and communication are more complex features of the distributed testing components. For such general reactions for errors, Time outs, observability, locks, controllability and synchronization problem are build. Azzouzi et al.[7] focuses on the temporal properties that specify the time required for exchanging messages between the various components of the distributed test applications. The study introduces new architecture to avoid the synchronization problem between different testers. The aim of the study is to propose better coordination mechanism between multiple testers and exert more controllability and observability on fault detection. Another objective is to check timing constraints in distributed testing correctly in such a

way that all testers and clocks should be synchronized. This development with the distributed testing framework is a difficult process where the testing system must not only check if the output events have been observed, and also the time whenever most of these events have been happened. The project presented within this research extends result from testing in distributed technique to deal with testing an implementation under test with some testing constraint.

In [10] H. Yamany et. Al propose a new multi-agent framework to test 3-tier distributed systems. This framework includes intelligent agents that work together in parallel in order to decrease testing cost and time. The authors propose the communication mechanism to interact between the agents. The framework is able to perform different testing techniques according to the distributed system behavior. It can generate new test cases by monitoring the users actions and it creates expected outputs from the execution of the various components in a distributed system to compare it with the testing results.

A multi-agent framework has been proposed for testing web-based systems by Samad Paydar et.al [9]. Different agents are designed with specific roles and they collaborate with each other to perform the test. The main design goals as underlined by the authors have been to develop an effective and flexible framework that supports different types of tests and utilize different sources of information about the system under the test to automate the test process. One of the novelties of this work is the use of test code which is based on the idea of mobile code. It provides benefits like increasing the performance, and decreasing the complexity of test executer agents. Another novelty of the work is the modeler agents that use different in-formation sources for automatic test script generation. A prototype of the proposed framework has been implemented and is used to perform some experiments. The results are promising and verify the overall design of the framework.

3. MULTI AGENT SYSTEMS

An agent is a piece of software that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. Agents are autonomous, intelligent, flexible, cooperative and reactive. These properties can be described as follows [10]:

1. Autonomy: Agents automatically monitor a distributed system during its runtime.
2. Intelligence: Agents reveal most of defects that might occur by performing the different testing techniques. As a result, they generate fresh test cases that enhance the performance of testing execution.
3. Flexibility: Agents perform different testing approaches depending on the changes and the development of distributed systems.
4. Cooperation: Agents communicate with each other in the proposed framework and consult with the testers to ensure the valid execution of the different testing techniques.
5. Reactivity: Agents reclaim any found error when misbehavior of a distributed system occurs.

An attractive feature of agent-orientation is that it provides a powerful metaphor for describing, understanding and modeling information systems that contain multiple autonomous active information processing agents and information sources and receivers. A role is intended to enable software engineers to use as a metaphor effectively to develop such cooperative information systems systematically through smooth and ordered transitions from models of the current system and users' requirements to the designs and implementations of new systems in an evolutionary way. The method proposed in this research paper uses this property of the multi agent systems to assign specific roles to the agents based on the different aspects of software testing. The next section gives an overview of the proposed system.

4. THE PROPOSED SYSTEM

The framework proposed by this paper can be described by the following formula: S={U, C, D, V, f}[11] .U represents UML models. Fig1 shows the details of the framework. R=CD represents the property set of the framework, in which C is the conditional properties and D is the result condition. V is the value set of property, V=vr, r ϵ R . f is information function, f : U×R→V. f defines how to select test data generation method in accordance to the specified UML diagrams.

Based on the UML model Up, the framework can choose the test data generation methods agent set D according to conditional property C. The agents which is selected are then to execute method to generate test data[12].According to the process of test data generation, the principles above can be summarized into the following three steps:

1. Extract software test information from UML diagrams;
2. Select agent set Di from rule set Ci;
3. Execute the selected agents to get software test data.

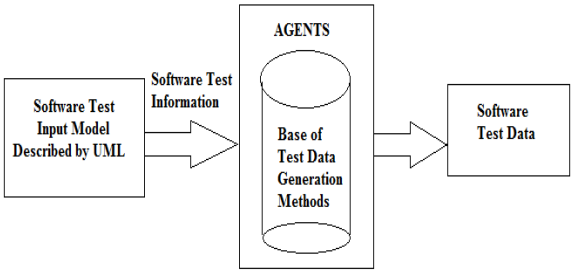

Figure 1: The framework of agent based software test data generation

The proposed Web based Application Testing Framework, consists of testing agents, which are independently configured to perform different kinds of software testing e.g. such as SSL testing agent, broken links testing agent, security testing agent, load testing agent, performance testing agent, smoke testing agent etc. The framework gives the option to the user to select the type of testing, which he wants to perform and the corresponding testing agents are employed to test the web application accordingly.

The working of the agents can be described by the following use case diagram:

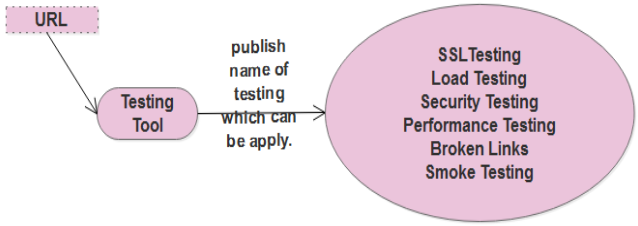

Figure 2: Use case Functionality for the Entire Framework

4.1 Use Cases

Description:

- The application provides a common interface for testing .
- This framework is only for web based application.
- The user can simply copy URL and paste it on text box of application and apply the testing.
- Application define that how many types of testing may be applied on the given URL.
- The system doesn't require the user to register first, to perform the testing.

Mission Statement

To support users for the testing of web based application on the internet.

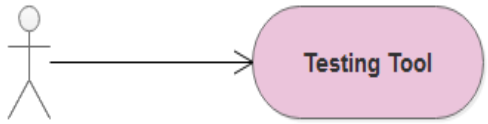

Figure 3: Audience Classification

4.2 Audience Characterization

- User can paste the URL on text box after that testing tool define that how many testing may be apply according to URL.
- User select the testing one by one and application process the testing and give the output as a response.

4.3 System Scope

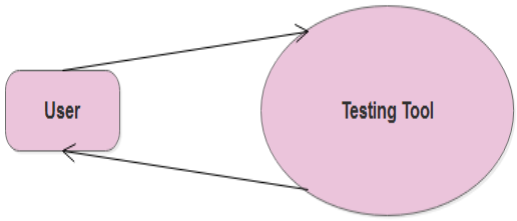

Figure 4: System Scope

4.4 User Type Description

It represent all the user that can access the public part of the web application

Profile data: No profile required
Object access in read mode: Testing
Relevant Usage : Applying different testing

4.5 Functional Use Case Diagram

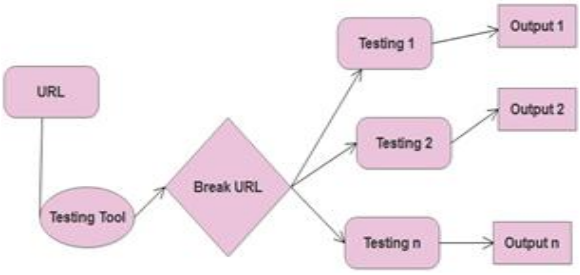

Figure 5 : Functional Use Case Diagram

This framework is intended to highlight the role of software agents or intelligent agents in providing such a multi-agent based testing framework, where different agents collaborate to provide software testing as a service on demand. Different agents groups are introduced, to support different testing types within an integrated framework. The multi agents system, has been developed using the JADE framework in the proposed system.

4.6 Working with the Jade Platform:

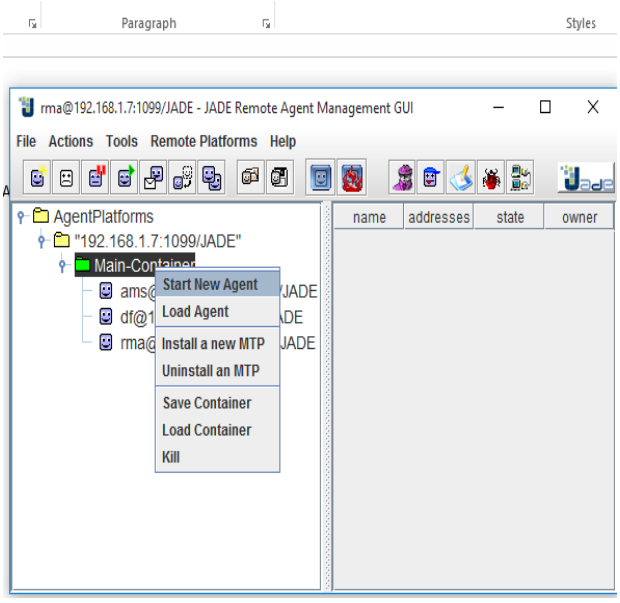

Figure 6: Creating an agent

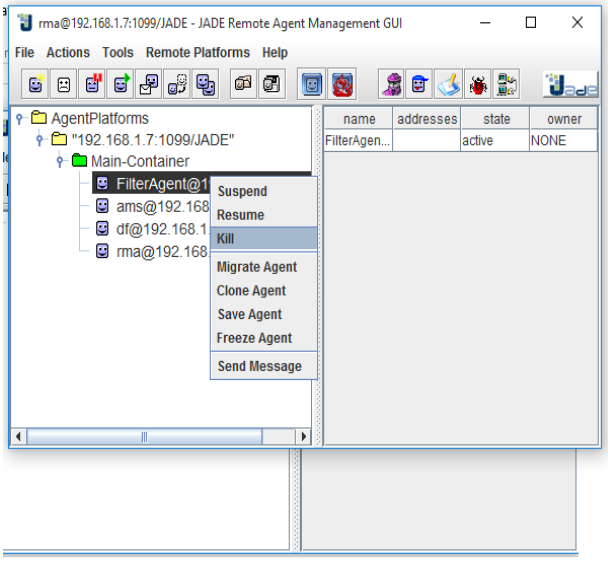

Figure 7: Kill Agent(Remove)

When starting a Jade container, we can specify many options. In particular, the "-gui" option causes Jade to create a special "Remote Management Agent" with a graphical interface which shows all participating agents and containers. It can also be used to kill agents, start new ones or to trace messages sent between agents (with the "sniffer" tool).

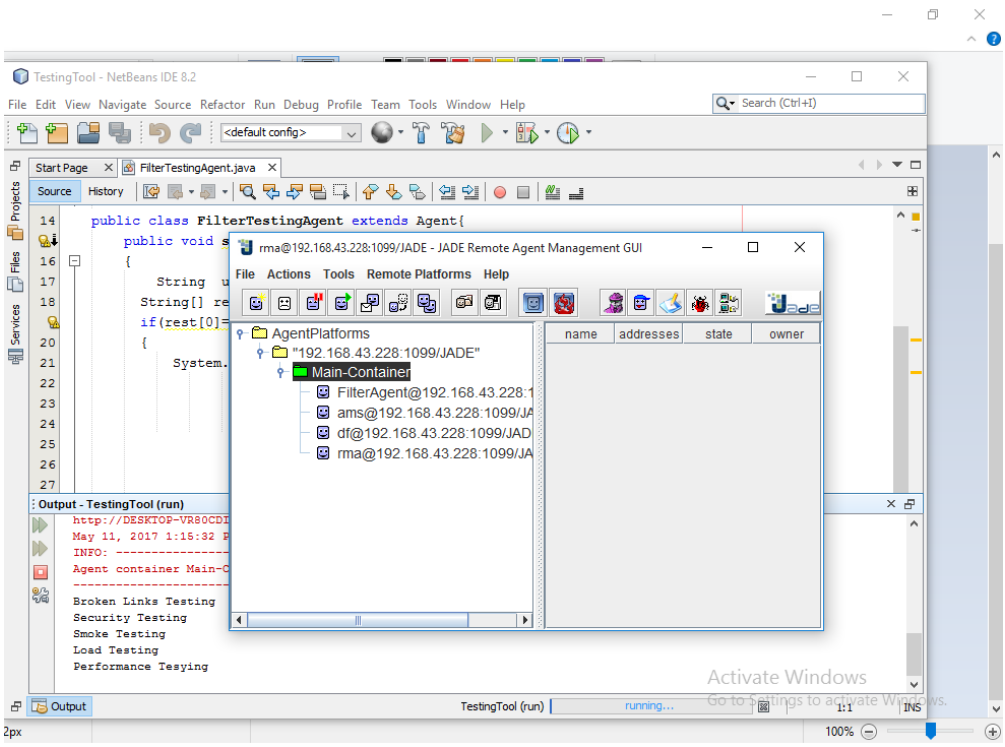

Figure 8: Filter Agents in Main Container

5. CONCLUSION

A new framework for testing web based applications has been proposed and implemented in this paper. The Java Agent Environment Development Environment was used to develop the framework. The tests was performed with a number of popular web applications and was found to be giving quick automatic test results for a number of issues. The framework provides a blueprint for development of easy to use testing solutions for small and middle level software organizations, which can't afford the highly priced testing software, which has been the main aim behind this research work.

In future we can add more functionality through which a software developer could apply his/her code for functional testing and unit testing, depending on the type of application he is working on. The attempt will be made to make it an open source development repository so as to include contributions from software developers and testers working on varied applications around the worlds to make the application more exclusive and extensive.

6. REFERENCES

[1] Sabih Jamal, Muhammad Aslam, Tauqir Ahmad," Multi-Agents Based Software Testing As A Service On Cloud", Sci.Int.(Lahore),27(3),2209-2215,2015.

[2] Kwang. Mong Sim, "Agent based Cloud Computing", IEEE Transactions on Services Computing, Vol. 5.pp 564-577, 2012.

[3] Baowen Xu and Lei Xu, "Applying Agent into Web Testing and Evolution" in National Natural Science Foundation of China (NSFC), pp794-798, 2004

[4] Nicholas R. Jennings and M. Wooldridge, "Agent-Oriented Software Engineering", in 13th International Workshop on Agent Oriented Conference, June 2012.

[5] Khalid A.Muhammadi, "Multi-Agent based Regression Testing Suite" in Athabasca University Thesis, April 2009.

[6] 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 Applications Software, 2005.

[7] Gouiouez Mounir, N. Rais,M. Azzouzi Idrissi," The Multi-Agent System and Stochastic Process in the Road Traffic", International Journal of Computer Applications (0975 –8887) Volume 62–No.12, January 2013

[8] 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

[9] Samad Paydar, Mohsen Kahani,"An Agent-Based Framework for Automated Testing of Web-Based Systems", Journal of Software Engineering and Applications, 2011, 4, 86-94

[10] H.Yemany, Miriam A, "A Multi-Agent Framework for Testing Distributed Systems", Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06)

[11] Wang Linzhang,Yuan Jiesong,Yu Xiaofeng, "Generating test cases from UML activity diagram based on Gray-box method[J]", Software Engineering Conference, 2004. 11th Asia-Pacific on 30 Nov.-3 Dec. 2004, Page:284 – 290.

[12] Siwen Yu and Jun Ai," Software Test Data Generation Based On Multi-Agent", International Journal of Software Engineering and Its Applications Vol. 4, No. 1, January 2010.

[13] Zina Houhamdi," Multi-Agent System Testing: A Survey", (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 2, No. 6, 2011.