IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

**ISSN: 2454-132X**

**Impact factor: 4.295**

***(Volume3, Issue3)***

Available online at www.ijariit.com

# E-commerce Testing Framework

**Ramsha Fatima**
*Babu Banarasi Das University, Lucknow*
*khan03ramsha@gmail.com*

**Upasana Dugal**
*Babu Banarasi Das University, Lucknow*

***Abstract:** E-commerce applications are being built at a rapid pace with the increase in demand of business to be always online. Many of the small shops popularly known as "kirana" shops are going online in our locality to meet the demands of the customer. They basically need an online presence, a product catalogue and payment gateway as the minimum requisites. The high demand of such applications and ease of technology has led to the emergence of several small and medium scale software development companies and freelance developers who can develop such applications at very minimal cost. However, the quality of such applications is compromised in various aspects. Software Testing is one such aspect. With very low costs involved the Testing phase for such applications is altogether skipped by the developers as they couldn't afford the expensive testing tools available for these applications and rely on mostly manual testing. In this research work, this particular problem has been undertaken and a framework has been proposed which gives the developers an automated testing suite which they can use by paying very low cost. A small application with some common testing attribute related to a typical e-commerce application has also been developed to better understand the concept. It is realized that if sufficient research is put in this framework can surely provide a breakthrough in e-commerce application testing.*

***Keywords:** Software Testing, Agents, E-commerce, Framework.*

## I. INTRODUCTION

Software testing is an important phase of the software lifecycle that suffers from an abundance of unsolved problems. Testing has not been ignored by the research community, but common research topics focus on only a small portion of these challenges, many of which may or may not be the most crucial issues for the industry as a whole. The first problem that this research intends to solve is identifying and quantifying hard problems in software testing as defined by testing professionals in order to provide a better research direction, allowing future work to resolve the most frequently occurring and problematic challenges that the industry faces today.

Recently, tool vendors are promoting testing software via web services, known as TaaS that theoretically resolve some of the major issues plaguing the industry. The secondary problem that this thesis examines is determining whether or not Testing as a Service is a potential candidate for solving the major problems identified by the industry by analysing the gap between the existing tools and the needs of the industry.

Software Testing plays a vital role in the lifecycle of any software development. It is that phase where the developers can get assured that software which is being delivered is error-free, reliable, robust, secure and validated against required functionality. If all the efforts have been done in developing, but still somehow lacks in testing, then it may result in software failure, after its deployment. Hence, due to much effort and chances of errors/deviated results, currently, manual testing is being replaced by automated testing tools, which are more reliable and error-free. However automated testing tools are very costly, as tools are licensed based, also manual effort is still required to operate tools, which costs double and most of the tools provide the only single testing type. Hence, purchasing different tools for different testing types is not economical, especially for small and medium organizations. A majority of e-commerce application which is made nowadays generally are intended for a small business like a small kirana store etc, because of huge demand to go online. These businesses go for cost effective solutions. This puts additional pressure on developers to make available a good quality software at minimum cost which makes them skip several phases of the software development lifecycle including testing. They only rely on manual testing, which is prone to errors.

This research work takes an agent-based e-commerce application development and focusses on its testing phase to lay out a framework for future e-commerce application developers to follow. The goal is to chalk out a framework which is low on cost and as such the possibilities of using the application to give a shared space to a multitude of users will be studied and implemented.

The major outlines of this research work can be outlined as below:

*© 2017, www.IJARIIT.com All Rights Reserved*

- Due to the considerable growth in applications of web-based systems, there are increasing demands for methods and tools to assure their quality.
- Testing these systems, due to their inherent complexities and special characteristics, is complex, time-consuming and challenging.
- These complexities increase the cost of testing web-based systems.

A reliable testing framework for web application especially e-commerce applications is indispensable.
The major objectives to be fulfilled by this research work is as mentioned below:

- Provide a framework for E-commerce applications of small and medium scale.
- Possibilities of using online pay per use services for testing.
- Development of multi-agent based e-commerce testing application with various kinds of intelligent agents developed using JADE.
- Testing of the software and the agents at various testing modes.
- Defining the best practices involved in developing an agent-based e-commerce application.

## II. LITERATURE REVIEW

•Research On A Multi-Agent Software Environment for Testing Web based Applications. [1]
This paper is written by Qingning Huo and Hong Zhu the paper explains the agent-based environment for web applications testing. 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 analysed. A number of agents are 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.

•Applying Agent into Web Testing and Evolution [2]
This paper was written by Baowen Xu and Lei Xu. This paper shows the implementation of agents into the area of automated Web Testing. They have explained the necessity and feasibility of automatic web testing and use of agents and also give the general description of the proposed method; then, they apply Agent into the Web performance testing, Web regression testing and Web usability evolution

•An Agent-based Testing Approach for Web Applications [3]
This paper was written by Yu Qi, David Kung, and Eric Wong. This paper highlights an approach for a specific test in which agents are generated from abstract classes. Web application testing is done by cooperation of a set of agents. They propose an agent-based approach for Web application testing. While the agent-based framework greatly reduces the complexity of Web applications, a four-level data flow test approach can be employed to perform structure testing on them. In this approach, data flow analysis will be performed as Function Level Testing, Function Cluster Level Testing, Object Level Testing, and Web Application Level Testing, from low abstract level to high abstract level. Each test agent in the framework takes charge of the testing in an abstract level for a particular type of Web document or object.

•Multi-Agent based Regression Testing Suite [4]

This Paper was written by Khalid A Muhammadi. This paper shows the MARTS Multi-Agents based Regression Testing Suite that makes use of intelligent agents for test cases versioning, report and results analysing. Proposed solution utilizes the benefits of the agent-based systems, along with the proven solutions of distributed computing. In proposed solution, only test scenarios are executed using agents, rather than the entire agent-based system. The main goal was to simulate human testing, along with some level of intelligence and autonomous decisions making capability.

•Agent-based Cloud Computing [5]
This paper was written by Kwang. Mong Sim. This paper provides verification of the system into two classes. Logic gives axiom for every statement and semantic approach checks model for testing. Its uses through a service-oriented interface to offer on-demand services gives an overview of intelligent agents in cloud computing and their role in testing. The significance of this work is introducing an agent-based paradigm for constructing software tools and testbeds for cloud resource management. The novel contributions of this work include:
1) Developing Cloudle: an agent-based search engine for cloud service discovery
2) showing that agent-based negotiation mechanisms can be effectively adopted for bolstering cloud service negotiation and cloud commerce, and
3) Showing that agent-based cooperative problem-solving techniques can be effectively adopted for automating cloud service composition

•Multi-Agents Based Software Testing As a Service on Cloud [6]

This Paper present the MSTAS in which they highlight the role and significance of intelligent agents in providing such testing services. It is intended to conceptualize the multi-agents based framework that emphasise that how different groups of agents can collaborate to provide software testing as a service on the cloud. MSTAS is multi-agents based integrated framework which provides different types of on-demand testing services, just as “pay as you go” model.

## III.PROPOSED WORK

Multi-Agents have been used to develop a framework for testing of e-commerce applications. E-commerce application/sites are web applications or mobile application too. So, they undergo all the typical test types.

- Functional Testing
- Usability Testing
- Security Testing
- Performance Testing
- Database Testing
- Mobile Application Testing
- A/B testing.

In this proposed framework, the various types of testing for e-commerce applications is automatically implemented using the multi-agents. Different agents have been created and put to work to automate the whole testing process of the e-commerce applications.

The flowchart of the system architecture is as shown below in Figure 5.1. The frontend application provides a testing tool to the users, who can either subscribe to the application or purchase as per the use. The working of the application is as follows:

- The user enters the URL of the E-commerce application to be tested.
- The Filter Testing Agent identifies the type of testing to be applied to the e-commerce application.
- The users selects the particular testing and testing is performed
- Results are displayed to the user.

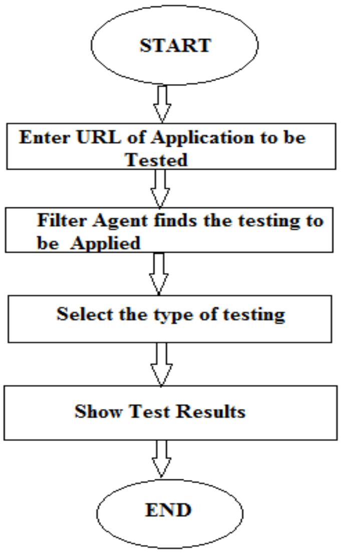
START
Enter URL of Application to be Tested
Filter Agent finds the testing to be Applied
Select the type of testing
Show Test Results
END

**Figure 1: System Architecture**

### Filter Testing Agent

The application Web Testing Tool requires the user to input the URL which he is intended to apply the testing on. The application uses filter agents for this purpose. Filter agents perform URL filtering to find out the kind of testing which is to be applied to the web application, according to the URL.

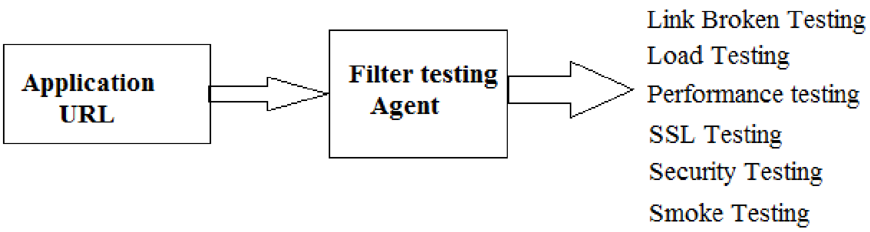

**Figure 2: Filter Testing Agent Working**

© 2017, www.IJARIIT.com All Rights Reserved

The Link Broken Agent checks for the broken links, appearing in the Application URL. A broken link or dead link is a link on a web page that no longer works because the website is encountering one or more of the reasons below.

- An improper URL entered for the link by the website owner.
- The destination website removed the linked web page causing what is known as a 505 error.

Load testing is a kind of non-functional testing. A load test is the type of software testing which is conducted to understand the behavior of the application under a specific expected load.
The Load Agent defined in the Web Testing Tool performs the load testing and measures the response time of the system.

Performance Testing Agent checks the performance of the URL being tested in terms of response time generated.
The SSL testing Agent checks for the SSL encryption of the application.
The Security Test agent testes the application for data security of users visiting the application. The Build Verification test is managed by the Smoke Testing Agent.

Table 1: Performance and Load testing Results of Three major Ecommerce Platforms

| Ecommerce Application | Load Test Result (sec) | Performance Test Result(sec) |
|---|---|---|
| Ebay | .55621 | 2.541262 |
| Amazon | 1.241658 | 3.985752 |
| Alibaba | 0.304788 | 0.751361 |

Table 1 gives the load testing and performance testing results for three major e-commerce giants in terms of their load testing and performance testing. The load testing checks for the response time of the e-commerce application in processing a single request, while a request bundle is sent to the application in performance testing. As can be seen from the table above, in both load testing and performance testing Alibaba fares extremely well as compared to Amazon and Ebay.

## CONCLUSION

Software testing will never be entirely free of challenges; however, many of the current and, frequently, timeless issues can be mitigated or their impacts reduced by focusing research on the needs of the industry and applying new methodologies where applicable. Ecommerce Application Testing is an important issue because of the emergence of several low-cost e-commerce applications. The suggested framework based on multi-agents promises to provide a low cost easy to use software which can be used for better testing by resolving many of these existing problems while promoting dynamic solutions easily capable of adapting to future needs. With more support from other researchers and more effort put into classify and implement test cases relevant to an e-commerce application, this framework can be extended to a solution that can truly revolutionize the future of testing, simplifying the creation and runtime of diverse and manageable test suites with little cost and effort on the consumers. To maintain this path, though, software testing problems must be re-evaluated, redirecting the tool requirements as necessary to develop and maintain strong products that consider overcoming testing challenges as a priority.

## 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] Luciano Baresi and Mauro Puzzle. An introduction to software testing. Electr. Notes Theor. Comput. Sci., 148(1):89–111, 2006.

[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] K. Beck and E. Gamma. Test-infected: Programmers love writing tests. Java Report, 3(7):37–50, 1998.

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

[14] S. Russel, P. Norvig, "Artificial intelligence – A modern approach", Prentice Hall, 1995.

[15] P. Maess, “Artificial life meets entertainment: Life like autonomous agents”, Communications of the ACM, vol. 38, no. 11, pp. 108-114, 1995.

[15] P. Stone, M. Veloso, “Multiagent systems: A survey from a machine learning perspective”, Autonomous Robots, vol. 8, no. 3, pp. 345–383, 2000.

[16] J. Doran, S. Franklin, N. R. Jenkins, T. J. Norman, “On cooperation in multi-agent systems”, In UK Workshop on Foundations of Multi-agent Systems, Warwick, 1996.

[17] 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.

[18] E. Jee, S. Kim, S. Cha, and I. Lee. Automated Test Coverage Measurement for Reactor Protection System Software implemented in Function Block Diagram. Computer Safety, Reliability, and Security pages 223– 236, 2010.

[19] E. Jee, J. Yoo, S. Cha, and D. Bae. A Data Flow-based Structural Testing Technique for FBD Programs. Information and Software Technology, 51(7):1131–1139, 2009.

[20] S. Khurshid, C. S. P˘as˘areanu, and W. Visser. Generalized symbolic execution for model checking and testing. In Tools and Algorithms for the Construction and Analysis of Systems, pages 553–568. Springer, 2003.