

Displacement Detection-Comparison Dual Engine for Dynamic Web Pages

Upasna Suman¹, Raunak Kasera²

¹Student, CSE, RVCE, Bangalore, India

²Student, CSE, RVCE, Bangalore, India

Abstract— This paper involves developing a displacement – based compare test module which compares corresponding elements in the test port and production website and generates a failure report in case the elements are misplaced by more than a centimeter. This module can be integrated into existing quality assurance platforms as an additional filter and can also be used as a plug in for location – specific tests by testing teams. Such a feature helps us move closer to the goal of automating the testing process and at the same time, make it more meticulous and rigorous.

Keywords — automation frameworks, quality assurance, software quality, Selenium, WebDriver, Appium, cross browser testing, quality metrics, XPath.

I. INTRODUCTION

Software quality is an important component in software development cycle. In the current scenario, frameworks do not provide a standardized way to assess the software quality. This leads to inconsistency in the tests, affects readability, affects reuse of code, impacts time to add tests. This new framework aims to mitigate this shortcoming and give a quantifiable measure of quality to the code developed. This platform will assess the location of the features developed and quantify how good the code is performing at any given point in time, by comparing displacements in real-time.

II. OBJECTIVE

The purpose of this research is to augment an automation framework, which will be used for functional testing and quality assessment of the backend service APIs in an easy, maintainable and configurable fashion. One use-case for displacements calculated can be to ensure that critical features do not overlap each other.

III. METHODOLOGY

The automation platform aims to extract data from the tests, measuring and accumulating usage statistics to indicate the behavior of the code in quantifiable terms. This dual engine will be developed as an additional filter to existing automation platforms making it a one stop shop for automation initiation and report analysis. This framework project will extract data using jQuery from the databases stored over the cloud [1][2][3]. For any new test, the test case author can choose to organize the components in any fashion. The existing framework just notifies the developers about their code using Aristotelian logic – either pass or fail. This is not very helpful. A framework is required which actually specifies how good the code is.

Running JavaScripts and using servlet programming will help in such extraction and eventual estimation. Quality of code can be regularly tracked with quality metrics [4].

IV. CROSS-BROWSER TESTING

Any test automation job initiated from an automation platform which needs actual browser to finish the test can be linked to cross browser testing framework [5][6]. This is because the cross browser testing framework provides the WebDriver of the respective browser which the user requests for the test.

- Works on actual browser session and does not use user agents for PC browsers
- Capable to running on user agents for tablet and mobile
- Uses WebDriver to open a selenium based sessions of all the browsers
- Selenium based drivers on all test machines support the launching of all browsers
- The framework is pluggable with automation jars easily
- It creates a driver session for the browsers which you have specified and passes the session to your automation to use it the way you want

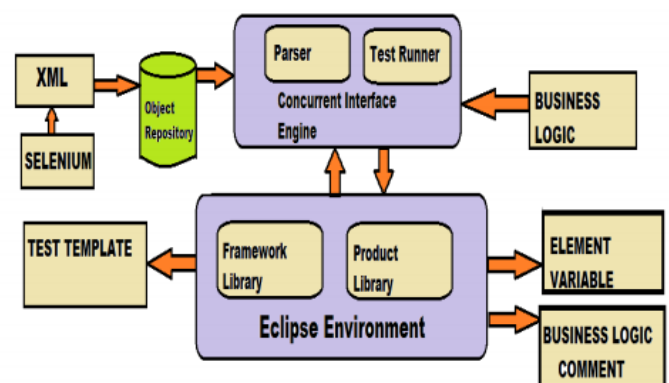


Fig. 1 A test automation interface model

V. READJUSTMENTS OF COORDINATES FOR DYNAMIC WEB ELEMENTS

A simple 3-step process to take into account the dynamic elements on web pages to calculate and compare displacements in real-time is as follows:

- Find the dynamic elements on the treatment page and production page using the XPath from a given .xml file.

- Some dynamic elements may have no effect on a static element. We map the influencing dynamic elements to each static element using the Java HashMap class [9].
- Negate the effect of the unpredictable nature of the relevant dynamic elements by subtracting the displacement from the x and y coordinates of each element.

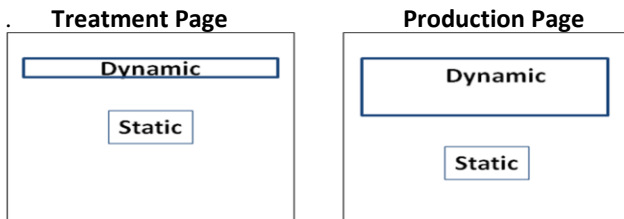


Fig. 2 An illustration showing the necessity to include the use case of dynamic web pages in the displacement detection-comparison dual engine

VI. EXPERIMENTAL RESULTS AND ANALYSIS

The typical quality assurance dashboard operations including displacement calculations were carried out on a cloud service as part of an experiment to evaluate its working. As the dashboard is a part of the development process, it was experimented for successful deployment [7]. This section lists the result of the experiment conducted and the inferences that were made from the testing. The evaluation metrics have been listed and the results have been accordingly quantified. The results that were got from the experiment describe the general performance trend of the automation platform, when it includes the dual engine.

A. Evaluation Metric

The displacement detection-comparison dual engine was designed and developed with the main intention to support various automation frameworks under a single roof. Hence the performance evaluation was done based on criteria of supportability of the implemented methods with different platforms and browsers. This metric evaluated the platform to be performing well if the method call returned success status to the client and corresponding updates were rendered in the Graphical User Interface of the automation portal [8].

B. Experimental Dataset

The dataset on which the experiment was conducted consisted of Apparel and Accessories product category in various e-commerce websites. Selenium-server-standalone-2.44.0 jar for browser automation and the Appium client library for Java were used [10]. These engines had access to the initiator ports, target ports present in the test environments. The input parameter list for the operation was given through the dual engine dashboard interface that was executing at the client machine.

C. Performance Analysis

The website content method which is responsible for creating snapshots on the engine was run on two different browsers through cross-browser testing framework. The

execution of the location-based test engine was verified on user interface in the Mozilla Firefox browser [11][12]. Figure 1 gives a week-by-week analysis of the bugs caught manually as compared to those caught by this automation platform for the time period February - April, 2015. 78 and 38 bugs were caught by the automation platform including the dual engine and the existing framework setup respectively. This gives a 105.26% improvement over the existing frameworks as shown by the following equation:

$$\frac{(78 - 38)}{38} \times 100 = 105.263 \%$$

VII. CONCLUSION

Having a cross-platform automation dashboard is an efficient way of code management as it supports one-stop checkpoint for automation at the framework level which has less time and memory constraints. The automation platform is developed offering operational support to various storage vendors, thus providing multiple vendor supportability. The Quality Assurance platform also provides modularity, code reusability and loosely coupled independent design which are required traits of a generic automation platform. The platform performed commendably well when experimented with location-based automation, accomplishing the requested backup job.

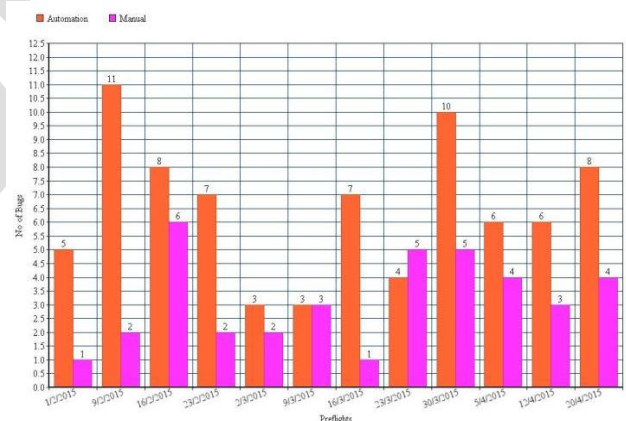


Fig. 3 A week-by-week analysis of the automation platform including the displacement detection-comparison dual engine

A. Future Enhancements

- Having UI initiation point from the automation dashboard itself with customizable options like other automations.
- Having a function to integrate refreshed data from dynamic web pages to all automations.
- This portal was tested on Mobile under Appium Framework. Thus, this module can also be implemented more extensively on other platforms with further deep-diving.

ACKNOWLEDGMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals,

elders and friends. A number of personalities, in their own capacities, have helped us in carrying out this research. We would like to take this opportunity to thank them all. We thank the faculty of Department of Computer Science & Engineering at RVCE for their constant support and encouragement. Last, but not the least, we would like to thank our peers who provided us with valuable suggestions to channel our research in the right direction.

REFERENCES

- [1] Manoranjitham, R.; Jagdale, B.N., "Designing an automation framework to improve the performance of SIP based test suites in PBX feature testing," *Wireless Computing and Networking (GCWCN), 2014 IEEE Global Conference on* , vol., no., pp.258,262, 22-24 Dec. 2014
- [2] Anbunathan, R.; Basu, A., "An event based test automation framework for Android mobiles," *Contemporary Computing and Informatics (IC3I), 2014 International Conference on* , vol., no., pp.76,79, 27-29 Nov. 2014
- [3] Shenoy, S.; Abu Bakar, N.A.; Swamy, R., "An Adaptive Framework for Web Services Testing Automation Using JMeter," *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on* , vol., no., pp.314,318, 17-19 Nov. 2014
- [4] Angmo, R.; Sharma, M., "Performance evaluation of web based automation testing tools," *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference -* , vol., no., pp.731,735, 25-26 Sept. 2014
- [5] Sivanandan, Sandeep; Yogeesh C.B., "Agile development cycle: Approach to design an effective Model Based Testing with Behaviour driven automation framework," *Advanced Computing and Communications (ADCOM), 2014 20th Annual International Conference on* , vol., no., pp.22,25, 19-22 Sept. 2014
- [6] Binsen Qian; Cheng, H.H., "A bio-inspired mobile agent-based coalition formation system for multiple modular-robot systems," *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on* , vol., no., pp.1,6, 10-12 Sept. 2014
- [7] Jamro, M., "SysML modeling of POU-oriented unit tests for IEC 61131-3 control software," *Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference On* , vol., no., pp.82,87, 2-5 Sept. 2014
- [8] Jamro, M., "Automatic generation of implementation in SysML-based model-driven development for IEC 61131-3 control software," *Methods and Models in Automation and Robotics (MMAR), 2014 19th International Conference On* , vol., no., pp.468,473, 2-5 Sept. 2014
- [9] Bose, L.; Thakur, S., "GRAFT: Generic & Reusable Automation Framework for agile testing," *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference -* , vol., no., pp.761,766, 25-26 Sept. 2014
- [10] Pasqualetti, F.; Zanella, F.; Peters, J.R.; Spindler, M.; Carli, R.; Bullo, F., "Camera Network Coordination for Intruder Detection," *Control Systems Technology, IEEE Transactions on* , vol.22, no.5, pp.1669,1683, Sept. 2014
- [11] Wiklund, K.; Sundmark, D.; Eldh, S.; Lundvist, K., "Impediments for Automated Testing -- An Empirical Analysis of a User Support Discussion Board," *Software Testing, Verification and Validation (ICST), 2014 IEEE Seventh International Conference on* , vol., no., pp.113,122, March 31 2014-April 4 2014
- [12] Eldh, S.; Andersson, K.; Ermedahl, A.; Wiklund, K., "Towards a Test Automation Improvement Model (TAIM)," *Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference on* , vol., no., pp.337,342, March 31 2014-April 4 2014