

## DEVELOPING A TESTING TOOL FOR TESTING BOTH GUI AND WEB APPLICATIONS TOGETHER

Shashank Joshi, Shital Pawar

Deptt. of Computer Engineering, Bharati Vidyapeeth College of Engineering, Pune, India.

### ABSTRACT

*An event driven software is the class of the software which changes its state with respect to occurrence of the events. The most common examples of this Event Driven Software are GUI and web applications. Therefore it's challenging to test such Event Driven Software. The main goal of the software testing is to uncover the errors. There are various tools are available for automated GUI testing and web application testing. Here our specific contribution is to develop a single testing tool for testing both GUI and Web Applications together. The web application is built using asp, jsp, php, servlet and GUI is built through the java technology. So we are developing a testing tool which is applicable to web applications of asp and jsp and GUI which is built through the java technology.*

**KEYWORDS:** *ambiguity problem, checking delay, GUI components, GUI testing, name of hyperlinks, swing based GUI, test cases, testing tool, web applications testing.*

### I. INTRODUCTION

In this paper we have presented two approaches to test GUI. One is for swing based GUI and another is for web application GUI such as html, jsp, asp pages. There are lots of tools available for GUI testing such as FEST, ABBOT etc. To test the GUI using these tools we have to write test cases for each page and these tools use test scripts to check against the results of GUI components [1] [2]. For example in fest tool we have to write test case class for each swing based frame and then it checks the GUI behavior against each test case. So each tool requires additional work to do. In our testing tool we are overcoming this drawback. In our tool we are checking the developers mistakes which he has done while coding but which are not the programming errors at all. Hence once the developer does the coding and program complies correctly then our tool takes this page as input and finding out the bugs in that page dynamically without writing any test cases. If developer added some components on the frame then we are checking whether the component has action listener or not and also we are checking that whether the components have same name or not.

#### 1.1 Existing Tools:

There is no any tool available which dynamically test the web based GUI. The different tools use different approaches to test the GUI. Mostly used tools are FEST, ABBOT etc. Each tool gets help of other frameworks such as UISpec4j java library which provides the different inbuilt classes and an interface which helps to test the swing based GUI [3] - [7]. This component works on static pages, means developer has to develop the test suite class and test cases depending upon the functional requirements. So it's not possible to test any swing based window. If you want to test the window then you have to create the test cases and have to develop the class according to test cases. In our testing tool we are not using any inbuilt API library or tools or even we are not generating classes for each window. So our tool works dynamically that is it can check any swing based application

restricting some GUI components. So there is no need to generate separate classes for each window [8].

## **II. TESTING OF WEB APPLICATION**

In web Application user can perform following types of events

- 1) Clicks on hyperlinks
- 2) Submit the data

In our testing tool user first select the web page which he wants to test from web application folder. After selecting the page our testing tool separates out the html, java and script code from the page. The user tests the code by clicking the test next button. After that our testing tool checks the all hyperlinks on that page, name of hyperlinks, if there is any form tag in the field then we checks its action attribute values. We store all available web pages in web application folder. If all hyperlinks which the developer provides on the web pages are available in the web application folder then there is no fault. That means there are no any fake hyperlinks or pages. After that we are checking the name given to the hyperlink. If there are same name for two different hyperlinks this will create ambiguity problem. So we are checking this problem also. If there is any such mistake then we are finding out and displaying this fault [9].

Next we are checking whether developer has given same hyperlink for different names. That will also create problem. For example for login and home if developer has given the same hyperlink i.e. login.jsp then after clicking on home and login the page will redirect to the login.jsp Now we are checking the form attribute value. If the value specified by the developer in action Attributes is not present in the web application folder then this will create problem. So we are also checking for this problem.

In the web application we write action method in the page to denote where the request will be forwarded after submitting the form. So when developer forgot to write the action tag attribute we are checking the code and finding out if code contains action or not. We are also checking that if GUI contains any similar component names. If GUI contains two buttons with the same name, then it will create ambiguity for user to perform further action.

### **2.1 Checking delay:**

After opening the page if user doesn't perform any action then it means he is in confusion of which action to perform [10]. So we are checking the time after page is opened. If user doesn't perform any action within some period that means the user has some confusion then our system detects there is fault in GUI. In desktop application pages we are getting all components from that window and checking whether each component behaving like its properties. For example we are checking whether any event is generating or not after clicking on button. If any event is not generating after clicking the button then there is fault in the page. This procedure is same for all components on that window [11].

## **III. SWING GUI TESTING (DESKTOP APPLICATION TESTING)**

In this we are testing swing based application. Each swing Application has window and each window contains the number of components and each component generates the events. So in our system we are checking whether each component generates specified event or not [12]. For example on clicking on submit button the new window will be opened, this is the requirement. Then we are checking that after clicking on the submit button new window is opening or not if opened then there is no fault in the system, if not opened then there is fault in the system. We are checking the event handler of each component. In our system first user select the page which he wants to test. Then after selecting the page our system checks the code. Then the page will get displayed. The user performs action on that page. In swing application there are number of components and each component has event handler. So we are restricting our application to test only some component and event handlers [13]. Such as label buttons, frame, checkboxes, radio buttons etc. Developer develops swing application simply on swing window. Each swing window consists of frame, number of components and event handlers. So just only by looking the component we can't recognize that components are working properly. So we are providing this developed window as input to our tool that parse the code and starts the testing [14].

This testing is done as follows:

- 1) First it checks how many components are on the window.
- 2) Then it checks whether they are initialized or not.
- 3) Then checks whether they are added on frame or not.
- 4) Then checks each component have action listener or not.
- 5) Then checks label of each component to check different component having same name or not
- 6) After finding out the faults, we are displaying all those faults with simplified messages.
- 7) Whether developer forgot to set frame visible true or not. We are checking whether he /she set the frame visibility mode true or not.

#### **IV. CONCLUSION**

In web based application we checking GUI component such as images, hyperlinks and forms. If the developer has provided same hyperlink for different names then our system automatically checks it and displays the information about the developer's mistake. This mistake is not the syntax error all hyperlinks work correctly but it crates the ambiguity problem. If there are different images for same hyperlink or same image for different hyperlink such a types of mistakes can be done by the developer which are GUI faults. So our system checks is there any such a kind of fault present in web application or not. Just we have to provide testing page as an input to the system which will checks all these faults and informs the developer about the faults present on the page which he has developed.

#### **ACKNOWLEDGEMENT**

We thank the anonymous reviewers whose suggestions and comments helped to extend and improve the flow of the work.

#### **REFERENCES**

- [1] Bryce RC, Sampath S, Memon A (2011) Developing a single model and test prioritization strategies for event-driven software. *IEEE Transactions on Software Engineering* 37(1):48–64.
- [2] Andrea Adamoli , Dmitrijs Zapanuks , Milan Jovic , Matthias Hauswirth , “Automated GUI performance testing”, *Software Quality Journal*, Apr 2011.
- [3] Xun Yuan, Atif M. Memon, Generating Event Sequence-Based Test Cases Using GUI Runtime State Feedback , VOL. 36, NO. 1, Feb 2010.
- [4] A. Marchetto and P. Tonella, “Search-based testing of Ajax web applications, “in 1<sup>st</sup> International Symposium on Search Based Software Engineering, May 2009, pp. 3–12.
- [5] S. Sampath, R. Bryce, G. Viswanath, V. Kandimalla, and A. G. Koru, “Prioritizing user-session-based test cases for web application testing,” in the International Conference on Software Testing, Verification and Validation. Lillehammer, Norway: IEEE Computer Society, Apr. 2008, pp. 141–150.
- [6] W.Wang, S. Sampath, Y. Lei, and R. Kacker, “An interaction-based test sequence generation approach for testing web applications,” in IEEE International Conference on High Assurance Systems Engineering. Nanjing, China: IEEE Computer Society, 2008, pp. 209– 218.
- [7] S. Sampath, S. Sprenkle, E. Gibson, L. Pollock, and A. S. Greenwald, “Applying concept analysis to user-session-based testing of web applications,” *IEEE Trans. on Software Engineering*, vol. 33, no. 10, pp. 643–658, Oct. 2007.
- [8]. S. Sprenkle, L. Pollock, H. Esquivel, B. Hazelwood, and S. Ecott, “Automated Oracle Comparators for Testing Web Applications,” *Proc. Int<sup>l</sup> Symp. Software Reliability Eng.*, pp. 253-262, Nov. 2007
- [9] D. Jeffrey and N. Gupta, “Test case prioritization using relevant slices,” in the International Computer Software and Applications Conference. IEEE Computer Society, Sep. 2006, pp. 411–418.
- [10] A. M. Memon and Q. Xie, “Studying the fault-detection effectiveness of GUI test cases for rapidly evolving software,” *IEEE Transactions on Software Engineering*, vol. 31, no. 10, pp. 884–896, 2005.
- [11] Srikanth H, Williams L, Osborne J (2005) System test case prioritization of new and regression test cases. In: International Symposium on empirical software engg. pp 64–73.
- [12] A.M. Memon and Q. Xie, “Studying the Fault-Detection Effectiveness of GUI Test Cases for Evolving Software,” *IEEE Trans. Software Eng.*, vol. 31, no. 10, pp. 884-896, Oct. 2005.
- [13]F. Ricca and P. Tonella, “Analysis and Testing of Web Applications,” *Proc. Int<sup>l</sup> Conf. Software Eng.*, pp. 25-34, May. 2001.

[14] D. C. Kung, C.-H. Liu, and P. Hsia, "An object-oriented web test model for testing web applications," in The First Asia-Pacific Conf. on Quality Software. Singapore: IEEE Computer Society, Oct. 2000, pp. 111–120.

### **Authors Biographies**

**Shashank Joshi** received his B.E. degree in Electronics and Telecommunication from Govt. College of Engineering, Pune in 1988, the M.E. and Ph. D. Degree in Computer Engineering from Bharati Vidyapeeth Deemed University Pune. He is currently working as the Professor in Computer Engineering Department Bharati Vidyapeeth Deemed University College of Engineering, Pune. His research interests include software engineering. Presently he is engaged in SDLC and secure software development methodologies. He is innovative teacher devoted to Education and Learning for the last 23 yrs.



**Shital Pawar** received her B.E degree in Computer Engineering from Bharati Vidyapeeth College of Engineering, Pune in 2010, pursuing M.tech degree in computer Engineering, Pune.

