Plug-ins an Added component for automatic testing of Modern Web Applications

S.MD.HAROON,

M.Tech, PG Scholar. ,

D. KHADAR HUSSAIN,

M.Tech (Research Scholar), CSEDept

JNTUA College of Engg, Anantapur-515002, A.P.

ABSTRACT

Current practice in testing JavaScript web applications requires manual structure of test cases, which is hard and tedious. It existing a framework for feedback-directed automated test generation for JavaScript in which execution is monitored to collect information that directs the test generator towards inputs that yield improvedreportage. Present-day web applications are very self-motivated in nature with rich user experience. Such applicationstypically use Web 2.0 and Asynchronous JavaScript and XML (AJAX) technologies. These applications arevery different from conventional web applications as they use stateful C/S communication in an asynchronousfashion. The use agent is able to communicate with web server without explicit form submission. This capabilitymakes the technology to support Rich Internet Applications (RIA). However testing such applications is atedious assignment. This paper suggests a tool that robotically tests AJAX applications. The mechanismmakes use of a crawler to capture the client side fields and generate a state-flow which is basis for thecompletion of automatic analysis. ADOM tree is constructedbuilt on the client side state which is useful to navigate tovarious parts of the tree in order to test the files. Different invariants of DOM tree are constructed for covering all statesof the presentation. This tool also types use plug-in concept for implementation of a tool thatcaters the present and future needs of the application. The real-world results exposed the proposed prototypeis able to capture AJAX faults and report them to development team.

1. INTRODUCTION

When World Wide Web was first developed, the web applications arestill in nature. There were only meantfor sharing information that has been available. Future on many server side technologies and scripting languagescame into existence in order to make the web applications dynamic and interactive. Java Applets also served tomake the web applications popular as they too made them dynamic. The web applications bestow manyadvantages as they can be accessed from any corner of the world without geographical and time restrictions. More over the web applications eliminate the client side installations as they are browser-based applications.

The modern web application development is based on Web 2.0 and AJAX [1] applications. The technologiesmake the applications to support the development of Rich Internet Applications (RIA). AJAX technologyaffected the way web applications are developed. There was aimportantchange in the growthprocess. More usability and more natural web application developers were made possible with

ISSN: 2278 – 1323

AJAX [2]. However, the features come at a price as they are prone to errors due to their tasteful nature the supports event handling. Theuse of DOM at client side also makes it vulnerable [2]. There are many static testing techniques to improveAJAX presentations. But, the static analysis shown to be inefficient as there are many dynamicdependencies when AJAX is used. Its communication with web resources in web servers makes it so complex anddependent on runtime states. In this reason testing is an AJAX application is Challenging task. Recentlydeveloped tools like Selenium1 is capable of capturing state of AJAX application and thus test modern webapplications. However, they also need lot of manual work for testing such applications.

Therefore, this paper aims at developing a tool that can automatically test AJAX applications. The proposedmechanism automatically gets all invariants of states of the application and covers all possible tests. Theproposed mechanism uses a crawler to reach user given values through UI elements presented in browser. Theproposed mechanism uses multiple invariants. Here we develop tool which is plug-in based. This tool has crawlinginfrastructure besides flexibility to add more plug-in in future for incremental functionality building of the tool. The main contribution of this paper is the tool which implements the mechanism based on plug-in and crawler.

The proposed tool is capable testing RIAs for various kinds of inconsistencies. The automated testing of modernapplications is possible with the proposed tool. The tool is influenced by [3] and [4]. The tool also givescomprehensive results that can help development teams to go ahead with fixing bugs if any. The remainder ofthis paper is structured as follows. Section 2 analyses literature on automated testing of modern web applications.Section 3 provides details of the proposed automated testing mechanisms. Section 4 provides details ofexperiments and results while section 5 concludes the paper.

2. PRIOR WORK

Modern web applications make use of sufficiently of client side scripting that interacts with server side technologiessuch as Java Server Pages, Active Server Pages, and PHP. AJAX technologies madethe web applications more interactive and dynamic with rich user knowledge, they also incur errors in the applications. Theworks on modern and modern application testing is presented inthis section.A crawler was built in [5] to test web applications containing many pages.

The crawler is accomplished ofdetecting inconsistencies such pertaining to page errors and navigation. Itcan support testing web applications with client side scripting. Means it can't cope with AJAX webapplications. Web applications might have security liabilities. Such web applications there are sometools such as SecuBat [6] and WAVES [7]. The approach in these tools is to have a crawler to obtain data anddetect defects. They also detect meandesignsrelating to XSS and SQL with respect to variousinjection attacks. Model checking is used in [8] using a tool named [9] for testing applications built on web 1.0.In [10] a testing approach is followed which is prototypical based. The tool implemented for creating webapplication using UML and a variety of test scenarios. Andrews et al [11] existing another way that depends onstate machine with constraints. All model based web application testing tools have a common drawback that istheir inability to deal with AJAX applications [3]. For generating automated test cases in [12] and [13] thesession data of logged in users is used. This kind of method needs enough communication with users of the webapplication. The session – based testing applications also focused on only synchronous requests that aresynchronous in nature. The messages that come from server, commonly known as delta-server messages, aredifficult to analyze. However, updates on such messages are more meaningful once they are processed at clientside and injected into DOM. Static analysis concept is used to analyze server content. Apollo is a tool implemented by Artzi et al [14] which is to find errors in web applications madeup of PHP. The tool is meant for detecting runtime errors that make HTML output inconsistent. In [15] and [16]presented method that examines server side Java code statically through request parameters to test theirvalues. Symbolic execution of server side code is used in [17] for identifying possible interface of given webapplication. However, all such techniques have disadvantages. They can't

ISSN: 2278 – 1323

cope with difficult client side behavior.Testing desktop applications was attempted by Memon in [18]. However, AJAX applications can be envisagedto be the combination of web applications and desktop applications [2]. To test such applications new tools areto be made that can specific structures of AJAX applications such as asynchronous c/s, dynamic etc.

They are different from traditional interface applications. They make use of DOM – based interface that is notsimilar to desktop applications with Graphical User Interface (GUI) [19]. Any traditional testing tool can beused to test AJAX applications. JSUnit3 is one of the unit testing tools that can be used to test web applicationsthat includeclient side JavaScript.

The world widely used AJAX web application testing tools include Sahi, WebKing, and Selenium IDE. Thesetools make use of APIs in order to control browser. However, itneeds lot of physicalwork for testing webapplications. These techniques work fine with web1.0 application. This paper proposes a tool that is plug-in-basedto test modern web applications that make use of AJAX technology. Here mainfocuses on crawler which isimportant for automated testing of present web applications.

GUI-Driven Software Testing

Most of today's software applications feature a graphical user interface (GUI) front-end. System testing of these presentations requires that test cases, modeled as structures of GUI measures, be created and executed on thesoftware. We term GUI testing as the process of testing a software application concluded its GUI. Investigators and consultants agree that one must employ a variety of techniques (e.g., model-based, capture/replay, manually scripted) for ejective GUI testing.

3. IMPLEMENTATION OF PROPOSED TOOL

The future tool is applied to test modern web applications that make usage of Web 2.0 version and AJAXtechnologies. There are many challenges in testing such presentations. They include discoveryprocedures thatsimulate user functionality in web presentation, achievement access to all states of DOM with respect to client;developing a method that can be used to find the correctness of the invariant situations. In modern webapplications, the states are explicit in nature and can be tested easily [21].

However, testing the applications thatuse AJAX is not easy. This is because the event driven nature of AJAX is exposed. It may cause manyaddition attacks including SQL. In order to assert performance of program, the proposed system makes use ofinvariants. In fact any dynamic application needs various DOM invariants and the testing has to consider allinvariants of the DOM. testing such invariants without human intervention are stimulating. Previous workstowards this end include Daikon [22] and DoDom [23]. The former takes invariants from runtime execution suggestions while the concluding is also capable of obtaining DOM invariants in the client.

Obtaining AJAX States

Here we built a crawler in Java programming language that can derive many states of an AJAX webapplication. This crawler was influenced the work in [3] and [24]. The crawling process has been implementedusing 2 algorithms which are presented in fig 1 and 2.

```
1. Procedure START(url, Set Tags)
2. Browser initEmbeddedBrowser(url)
3. Robot initRobot
4. sm initStatemachine()
5. precrawlingPlugin(browser)
6. crawl(null)
7. postcrawlingPlugin(browser)
8. end Procedure
9. procedure CRAWL(STATEps)
```

```
1. Procedure GenerateEvent(States,Clikable c)
2. Robot.enterFormValue(c)
3. Robot.fireEvent(c)
4. dom browser.getDom()
5.ifstatechanged(cs.getDom(),Dom)then
6. xe getXpathExpr(c)
7. ns sm.addState(Dom)
8. sm.addEdge(cs,ns,Event(c,xe)
9. sm.changeToStaePlugins(ns)
```


ISSN: 2278 – 1323

```
10. cs
sm.getCurrentState
()
11. update
diff(ps.cs)
12. f
analyseForms(upd
ate)
13. set C
  getCandidateClic
  kables(update,t
  ags,f)
14. for c= do
15. Generate
Event(cs,c)
16. End for
17. End procedure
```

```
10.
runOnNewState(ns)
11. test Invariant(ns)
12.
ifstateAllowedToBeCr
awled(ns) then
13. crawl(cs)
14. end if
15.
sm.changeToState(cs)
16.
ifbrowser.history.can
GoBack then
17.
browser.history.goBac
k()
18. else
19. {We have to back-
track by going to
initial state}
20. Browser. reload()
21. List E sm
.getPathTo(cs)
22. For e=E do
23. re resolve
Element(re)
24. robot .enter Form
Values(re)
25. robot .fire
Event(re)
26. end for
27. end if
28. end if


29. end procedure
```

Figure.1-Crawling Process

Fig. 2 –Result MotivatedAnalysis of AJAX States

As can be seen in fig. 1, the algorithm has functionalities pertaining to crawling web pages including precrawlingand post – crawling functionalities. While crawling the algorithm makesmany events. Theprocess of generating events is in figure. 2.

As can be seen in figure. 2, the generation of events is done robotically. A software robot moves into HTMLforms and fields and generate appropriate events that are used to test the modern AJAX web applicationsautomatically.

Plug-in Development As part of the tool two plug-ins was implemented. The first plug-in is meant for findingincreasefaults. Such faults are due to wrong understanding of requirements and design. The second plug-inis meant for discovering security vulnerabilities. The improvementfaults are pertaining to AJAX coding andalso the responses from web resources that run in theweb server. Such faults are identified based on theexpected behavior of the web application. Navigation and other faults are identified and reported. With respectto security vulnerabilities, the future tool examines dynamic AJAX functionality by inspecting the codeclosely to know whether the code is intact or tampered to inject SQL and other injections. Even JavaScript codecan be injected into a web application to make it compromise towards certain functionalities. This will helphackers or intruders to gain control above web pages. The plug-in checks the exposures and information toincrease team. More plug-ins can be developed in future to make the application more robust and covervarious kinds of errors.

4. EMPERICAL RESULTS

The proposed tool is tested with Web 2.0 requests that type use of AJAX. The tool is tested to know itsvarious functionalities. First of all its ability to obtain meaningful invariants is verified. It has been observed that text invariants needs good accepting of design; Xpath expressions can be used to represent invariants; invariants can be used to discover faults; with invariants, manual effort is reallydecreased. In anotherexperiment fault revealing capabilities of the tools are verified. It has been observed that the tool is capable of writingerrors; provides acceptable performance with respect to crawling and also automatic testing of webapplications. In the third experiment focus is on AJAX enabled applications. The results exposed that the tool is accomplished of finding faults with the application.

5. CONCLUSION

ISSN: 2278 – 1323

Here we implemented a tool in Java which can test modern web applications that make use ofAJAX technology. Towards this we have implemented a crawler which captures data given by the user in webbrowser. Then the tool makes use of DOM tree and invariants of the state flows in order to test the webApplication for discovering inconsistencies. Many errorprototypes have been used in order to uncover hidden bugsin the AJAX applications. An algorithm is built for obtaining all state flows while crawling the web application. The application is modular and scalable in nature.

The plug-ins concept makes it to increment its functionality infuture with simplicity. In addition to basic level testing, its (tool) supports discovery of development faults and also identifies potential security vulnerabilities in the web application and architectural faults. It performs a black-box testing. Unlike source code programscanners, web application scanners don't have access to the source code and therefore detect vulnerabilities by actually performing attacksSecurity vulnerabilities. The experimental results revealed that the tool is very useful in testing real time web applications (uses) provide rich user involvement.

References

[1]J.Garrett,"Ajax: A New Approach to Web Applications,"adaptivepath,http://www.adaptivepath.com/publications/
essays/archives/000385.php, Feb. 2005.

[2] A. Mesbah and A. van Deursen, "A Component- and Push-Based Architectural Style for Ajax Applications," J. Systems andSoftware,vol. 81, no. 12, pp. 2194-2209, 2008.

[3] A. Mesbah, E. Bozdag, and A. van Deursen, "Crawling Ajax by Inferring User Interface State Changes," Proc. Eighth Int'l Conf.WebEng., pp. 122-134, 2008.

[4] A. Mesbah and A. van Deursen, "Invariant-Based Automatic Testing of Ajax User Interfaces," Proc. IEEE 31st Int'l Conf.SoftwareEng., pp. 210-220, 2009.

[5] M. Benedikt, J. Freire, and P. Godefroid, "VeriWeb: Automatically Testing Dynamic Web Sites," Proc. 11th Int'l Conf. World WideWeb, app. 654-668, 2002.

[6] S. Kals, E. Kirda, C. Kruegel, and N. Jovanovich, "Secubat: A Web Vulnerability Scanner," Proc. 15th Int'l Conf. World WideWeb, pp. 247-256, 2006.

[7] Y.W. Huang, C.H. Tsai, T.P. Lin, S.K. Huang, D.T. Lee, and S.Y.Kuo, "A Testing Framework for Web Application SecurityAssessment," J. Computer Networks, vol. 48, no. 5, pp. 739-761,2005.

[8] L. de Alfaro, "Model Checking the World Wide Web," Proc. 13thInt'l Conf. Computer Aided Verification, pp. 337-349, 2001.

[9] L. de Alfaro, T.A. Hen zinger, and F.Y.C. Mang, "MCWEB: A Model-Checking Tool for Web Site Debugging," Proc. World WideWeb Conf., posters, 2001.

[10] F. Ricca and P. Tonella, "Analysis and Testing of Web Applications,"Proc. 23rd Int'l Conf. Software Eng., pp. 25-34, 2001.

[11] A. Andrews, J. Offutt, and R. Alexander, "Testing Web Applications by Modeling with FSMs," Software and Systems Modeling, vol.4, no. 3, pp. 326-345, July 2005.

[12] S. Erlbaum, G. Rothermel, S. Karre, and M. Fisher II, "Leveraging User-Session Data to Support Web Application Testing,"IEEETrans. Software Eng., vol. 31, no. 3, pp. 187-202, Mar. 2005.

[13] S. Sprenkle, E. Gibson, S. Sampath, and L. Pollock, "Automated Replay and Failure Detection for Web Applications," Proc.IEEE/ACM 20th Int'l Conf. Automated Software Eng., pp. 253-262, 2005.

[14] S. Artzi, A. Kie_zun, J. Dolby, F. Tip, D. Dig, A. Partaker, and M.D. Ernst, "Finding Bugs in Dynamic Web Applications, "Proc. Int'lSymp. Software Testing and Analysis, pp. 261-272, 2008.

[15] W. Halfond and A. Orso, "Improving Test Case Generation for Web Applications Using Automated Interface Discovery," Proc.SixthJoint Meeting of the

ISSN: 2278 – 1323

European Software Eng. Conf. and the ACMSIGSOFT Symp. The Foundations of Software Eng., pp. 145-154, 2007.

[16] W. Halfond and A. Orso, "Automated Identification of Parameter Mismatches in Web Applications," Proc. 16th Verification andValidation, pp. 128-136, 2010.

[17] W. Halfond, S. Anand, and A. Orso, "Precise Interface Identification to Improve Testing and Analysis of Web Applications,"Proc.18th Int'l Symp. Software Testing and Analysis, pp. 285-296, 2009.

[18] A. Memon, "An Event-Flow Model of GUI-Based Applications for Testing: Research Articles," Software Testing, Verification andReliability, vol. 17, no. 3, pp. 137-157, 2007.

[19] A. Marchetto, P. Tonella, and F. Ricca, "State-Based Testing of Ajax Web Applications," Proc. IEEE First Int'l Conf. SoftwareTesting Verification and Validation, pp. 121-130, 2008.

[20] A. Marchetto, F. Ricca, and P. Tonella, "A Case Study-Based Comparison of Web Testing Techniques Applied to Ajax WebApplications," Int'l J. Software Tools for Technology Transfer, vol. 10,no. 6, pp. 477-492, 2008.

[21] A. Mesbah and A. van Deursen, "Migrating Multi-Page Web Applications to Single-Page Ajax Interfaces," Proc. 11th Europea n Conf.Software Maintenance and Reeng. pp. 181-190, 2007.

[22] M.D. Ernst, J. Cockrell, W.G. Griswold, and D. Notkin, "Dynamically Discovering Likely Program Invariants to Support ProgramEvolution," IEEE Trans. Software Eng., vol. 27, no. 2, pp. 99-123,Feb. 2001.

ISSN: 2278 – 1323