



Automatic Generation Test Case Selection for web Application

Reeta Soni

Mtech, Department of Information Technology
Govt. Engineering College Ajmer
Rajasthan ,India

Mrs. Neetu Sharma

Assistance Professor, Department of Information Technology
Govt. Engineering College Ajmer
Rajasthan ,India

Abstract— Web application uses are increased day by day. Web application are used the following field like business, communication, education and entertainment. Testing is the process to help identify the correctness, completeness and quality of developed computer software. Web application designs into source quality attribute security availability, and scalability. Web application changes occurs whenever then the regression testing are used to easily used them and any difficulty not occur during the web uses in this paper present the modelling web application though data flow graph ,data test tree and the optimal test sequence.

Keywords— Regression testing, test case sequence, data flow model, data test tree

I. INTRODUCTION

Web applications are used in many areas as communication, e-commerce, education, government, entertainment sectors, and industry capable of performing business activities. The main feature of the web for developing web product including 1) no installation cost 2) automatic update new feature of all user 3) web are used anywhere from any machine. Web applications are characterize such as distribution, hypermedia, multi-platform, dynamic and interaction. Testing is the process to help identify the correctness, completeness and quality of developed computer software. Web application designs into source quality attribute security availability, and scalability.

The key of software testing is to assess the capabilities of software's ability to adequately meet the applicable standards and customer needs. Software that has been tested on certain inputs may fail to work on those same inputs in the future. Regression testing aims to detect these errors by comparing present behavior with past behavior. Regression testing is used to check the code's integrity and is mainly performed during automated builds tightly to ensure that errors are detected and recovered as soon as possible. Automation testing which is also known as Test Automation is when the tester writes scripts and uses software to test the software. This process involves automation of a manual process. Automation Testing is used to re-run the test scenarios that were performed manually, quickly and repeatedly. Apart from regression testing, Automation testing is also used to test the application from load, performance and stress point of view. It increases the test coverage; improve accuracy, saves time and money in comparison to manual testing

II. RRELATED WORK

Automatic test case Generation for Web Service Testing

Xiaoying Bai

The WSDL -based test case generation is a part of distributed service testing framework that include test case generation, test controller, test agent and test evaluator. The specification of the service is the first phased into a DOM tree representations are based on the WSDL schema. Test cases are generated from four level; test data generated, individual test operation generation, operation flow generation, test specification generation. WSDL which carries the basic information of a service including its interface operation and the data transmitted. WSDL only test case for individual service could be generated either single operation or combination of operation. The WSDL is first parsed and transformed into the structure DOM tree test cases are generated into two perspective

- 1) Test data generation-test data generated by analyzing the message data type according to standard XML schema syntax.
- 2) Test operation generation-operation flows are generated based on the operation dependency analysis. Three type of dependency include input dependency, output dependency and input/output dependency

Zhongsheng Qian

Testing component based web application using component Automata is based on the component interaction Automata. Web applications are dividing into a set of interacting component. For each component, a formal automation is constructed and the interactions among component are modelled by the composition of automata. Component-based Web applications at a high level of abstraction. The concept of generic component, instance component, composite component and component cluster. A generic component is an abstraction of a set of instance components. An instance component is an actual component instantiated in the running environment of a Web application. A composite component is the composition of several interacting components. A component cluster is a collection of related components.

Yang Liu

Automatic generation of test cases is the core content in the model-driven testing. The process of generating test cases is that a platform-independent model is converted into a platform-independent test model through level conversion, and the platform-independent test model is converted into the corresponding test cases through vertical conversion. The basic structures of the system model and the test model are designed using Web-application system. The conversion rules from the system model to the test model are designed using the model to model conversion method. The conversion rules from the test model to the test cases are designed using the model to code conversion method. Ultimately, the relevant test cases are generated.

Gagandeep

A test automaton is the use of software for automatic execution of test cases and their comparison with expected outcome. Test automation is use of the software for automatic execution of test cases and their comparison with expected result. Model based testing is automatic generation of Web Applications need to be analysed and modelled prior to testing. The web model is traversed to generate test sequences. Framework consists of the following steps.1. Domain Analysis and Modelling 2. Model traversal and test case generation 3.Optimizing test cases using coverage criteria 4. Regression test suite generation. Initially we start with detailed analysis of a web application and explore specifications of its different parts. Complete study of the web application need to be done and creation of a manual document is performed for all type of functions and is termed as Specification Document. It is the theoretical explanation of the whole functionality (links, buttons, database interaction etc.) of the web application. In this document, type of page: static or dynamic, title of the page, number and label of links/buttons on the page and the transitions is given. A page id has also been assigned to each page to provide easiness in using the pages implementation. From Analysis, we move to design representation of the web application. We adopt Model-driven representation for designing and implementation. Model-driven approach offers a new and promising approach for the task of designing and automation. With the help of Specification document, a Graphical Web Model of the component is constructed. The model is traversed using All Link Coverage to generate test sequences. By using this criterion we confirm that all the links available in the web application will be tested at least once.

Rohit N. Devikar

Model-Based Testing is the automatic generation of efficient test procedures using models of system requirements and specified functionality. Model-based regression testing is an important activity that ensures the reliability of evolving software. One of the major issues in this type of testing is the optimal selection of test-cases to test the affected portion of the software. Model-based selective regression testing promises reduction in cost and labour by selecting a subset of the test suite corresponding to the modifications after system evolution. Identification of such changes in modification and selection of test cases is very difficult task. MBRT (Model Based Regression Testing) tool has been developed, which is a java base testing tool used for generation, reduction of test cases and also classify the test cases into obsolete, reusable and re-testable test cases which results in reduction in time and cost. In this paper use of class diagram and state machine diagram has been made for presenting the idea of regression testing and also use of flow graph to generate the test cases.

III. MODELLING WEB APPLICATION

Web applications are need to analyse and modelling prior to testing. Web model are traversed to generate test sequence. A Web Application involves information and navigation mechanisms so that user may find and use that information effectively. The purpose of this workflow detail is to organize the Website’s content and features into a logical structure of the whole Website. For testing Web-based application, Gmail System is considered which helps to user for mailing to one another. Users first go to Gmail page then login through new or already registered transition. Test sequence are generated though regression testing. Modelling of web application in the regression testing used the following step

- 1) Data flow diagram- Data flow graph of web application show the relationship between various page. Data flow graph represent in fig. 1 entry the first node A (browser).
- 2) Data test tree- The advantageous of using data flow graph is that we easily find path but disadvantageous is that the path can be cyclic with no termination. Starting node but not present ending node. Data test tree are used to remove the disadvantageous of the data flow graph and generated shorter path .

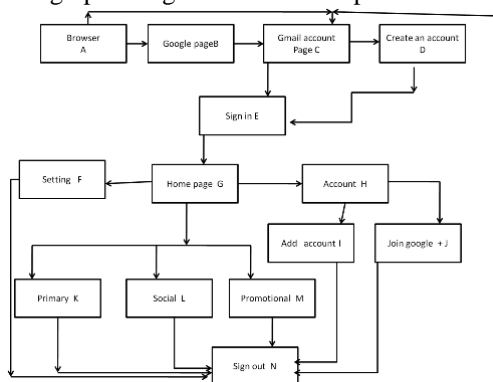


Fig 1. Data flow diagram for Gmail account

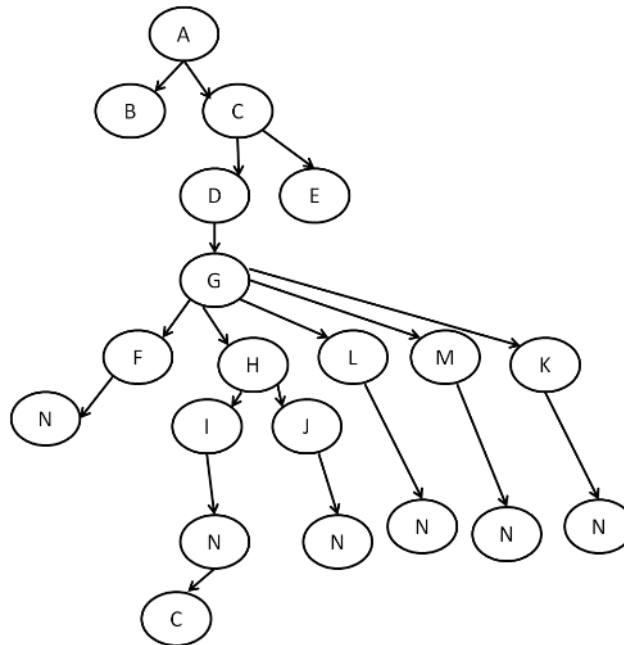


Fig 2. Data tree diagram for Gmail account

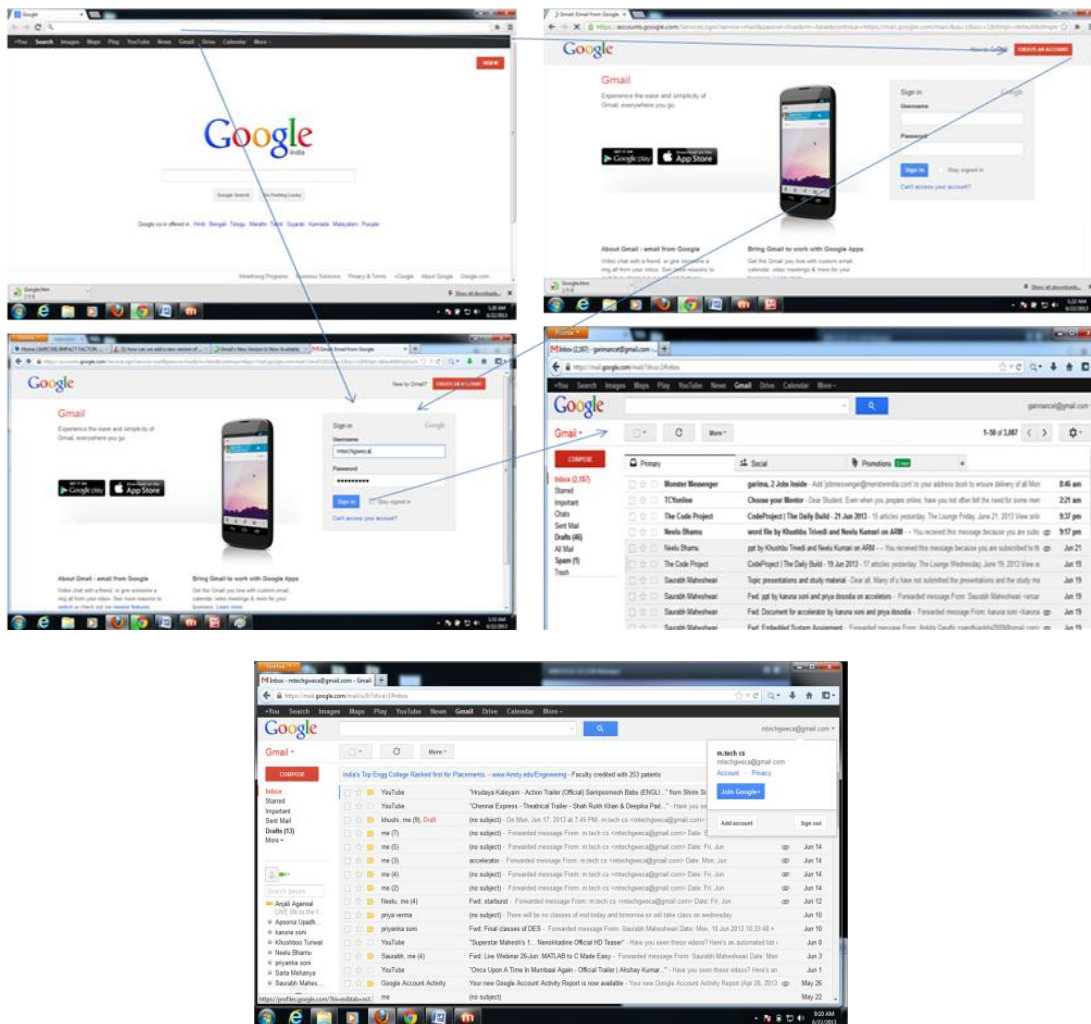


Fig 3 Process of Gmail system

- 3) Generating optimal test sequence- web application test sequence are starting from root node (page) to the same leaf node that can traversed M_0 is the initial page and n represent total number of test sequence where (n>0). Web test sequence represents available navigation from initial node to the ending node by travelling different immediate pages.

Table 1. Optical test sequence of Gmail system

Test sequence		Optimal test sequence	
S1:A->B->C->D		S1:A->B->C->D->E	
S2:A->C->D->E		S2:A->B->C->D->E->C->E->A->C	
S3:A->B->C->E->G		S3:A->B->C->E->G->F->N->C	
S4:A->B->C->E->G->F->N		S4:A->B->C->E->G->H->L->N->H->M->N	
S5:A->B->C->E->G->H->L->N		S5:A->B->C->E->G->I->N->G->J->N->G->K->N	
S6:A->B->C->E->G->H->M->N			
S7:A->B->C->E->G->K->N			
S8:A->B->C->E->G->I->N			
S9:A->B->C->E->G->J->N			
S10:A->B->C->E->G->J->N->C			
Total no of test sequence	10	Total no of optimal test sequence	5
Total no of page traversed	65	Total no of page covered	46
Total no of link traversed	55	Total no of link covered	21

IV. CONCLUSION

Automatic generations of test sequence are based on the testing. The regression testing is used whenever changes occur in the web in the help of regression testing solve the problem. In this paper main focus on the Gmail system process in which they are show in data flow model, data test tree and the optimal test sequence generated are used to minimize the number of test sequence used in the model.

Reference

- [1] Andrews, A., J. Offutt, R. Alexander, 2004. Testing web applications by modelling with FSMs. Software Syst. Modelling, 4: 326-345.
- [2] Xiaoping Bai, 2005, WSDL-based Automatic Test Case Generation for Web Service Testing, proceeding of the IEEE international Workshop on service –oriented system engineering (SOSE'05), pp: 1-6.
- [3] Zhongsheng Qian, 2009. Testing Component-Based Web Applications Using Component Automata. International Conference on Information Engineering. ICIE '09. Vol. 1, pp. 455 – 458, DOI 10.1109/ICIE.2009.64.
- [4] Yang Liu. Yafen Li, Pu Wang (2010). Design and Implementation of Automatic Generation of Test Case Based on Model Driven Architecture. Proceedings of the Second International Conference on Information Technology and Computer Science, pp. 344-347
- [5] Gagandeep, "Automatic generation of regression test cases for web component using Domain analysis and modelling", International Journal of computer application), Volume 11-NO-12, December 2010