



Automatic Generation of Graphical Web Model from Analysis of Web Component

Reeta Soni

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

Mrs. Neetu Sharma

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

Abstract-Testing is the process of evaluating system components by manual or automated means to verify that it satisfies specified requirements. Regression testing are used in the web application because in testing modified software to ensure that changes are correct and do not adversely affected other part of the software the changes occur simultaneously and the error produced. Web test sequence represents available navigation from initial node to the ending node by travelling different immediate pages. A method to generate manually test sequence in web component in web application. Our work can be considered as the following step based on coverage criteria. First we construct a model of given web example. Second to generate test tree of the web because the model diagram is cyclic. At the last optimal test sequence are generated manually to reduce the number of test sequence we can follow of coverage criteria "All page coverage" according this coverage criteria. Testoptimal are a model based test tool are used in the web model automatically generate sequence graph, model graph, coverage graph. Testoptimal are used to automatically test the model of the web component and execute without any user help.

Keyword-web application, data flow diagram, data test tree, test sequence graph, testoptimal.

I. INTRODUCTION

Software testing is the process of executing a software system to determine whether its meets its specification and executes in its manner. An incorrect behaviour indicates a software failure. A failure results from a fault, which is usually an incorrect or missing software component. Software testing is performed to display possible failures of the software. The various advantages of testing are as follows increasing accountability and Control, Cost reduction, Time reduction, and Defect reduction A web application is a software application that is access with a client over a network such as internet or an intranet. Web application generates various type of web page like text, image, form etc. Web service systems would have the following properties safe, interoperable, composable, decentralized, end-to-end, and automated..

II. RELATED WORK

Automatic generation of the graphical web model from analysis of web component

Zhongsheng Qian

Web testing model for Web application testing. It started from constructing the PFD (Page Flow Diagram) of the Web application. An algorithm is then designed to derive a PTT (Page Test Tree) from the PFD. From the PTT, a test translator is employed to extract the path expression, in order to generate all the test paths and then translates them into a test specification in XML syntax, which is the input of test engine. Web application Qian has developed to demonstrate our approach the SWLS (simple web login system) starting at the first point. Lei Xu have proposed Web application testing, which includes the requirement analysis, test case generation and selection, testing methods and techniques, testing execution, and testing result metrics.

Gagandeep

An analysis based modelling and regression testing for web application. Model based testing is a test automation approach that generates and maintain more useful and effectively test from explicit description of the application. Model describing web component structure before and after the changes are incorporate. Its helps to identify test case generated for the initial version of the component that can be rerun on the changed component. The regression test suite generated for the component is optimized using "all paths" coverage criteria.

Mr. Rohit

Model based testing is automatic generation of efficient test procedure using model of system requirement and specified facilities. One major issue of using testing method to automatic optimal selection of test cases to test the affected part of the software. Model based regression testing tool has been developed, which is the java based regression tool is used to generation, reduction of test cases and also classify the test cases of obsolete reusability and re-testable test case which result is reduction the time and cost.

Andrew

Testing web application by modelling with FSMs a system-level testing technique that combines test generation based on finite state machines with constraints. Use a hierarchical approach to model potentially large Web applications. The approach builds hierarchies of Finite State Machines (FSMs) that model subsystems of the Web applications, and then generates test requirements as sub sequences of states in the FSMs. This subsequence are then combined and refined to form complete executable tests. The constraints are used to select a reduced set of inputs with the goal of reducing the state space explosion otherwise inherent in using FSMs. This technique with a running example of a Web-based course student information system and introduces a prototype implementation to support the technique

Alshahwan

An approach to web application regression testing based upon repair of user session data. The approach is entirely automated. It consists of a white box examination of the structure of the changed web application to detect changes and a set of techniques to map these detected changes onto repair actions. The results of experiments that explore both the performance and effectiveness of the approach. The effectiveness experiment uses an implementation of the repair algorithm applied to the online bookstore application over a series of 10 releases.

Briand

A TOTEM (Testing Object-oriented systems with the unified Modelling language) functional system test methodology. Briand derived test requirements from early artifacts produced at the end of the analysis development stage, namely use case diagram, use case description, interaction diagram (sequence or collaboration) associated with each use case, and class diagram (composed of application domain classes and their contracts). This early use of analysis artifacts is very important as it helps devising a system test plan, size the system test task, and plan appropriate resources early in the life cycle.

III. TEST SEQUENCECE GENRATION OF WEB COMPONENT

Web store is like online shopping considered which helps to user for shopping anywhere anytime without going any shopping mall. We select the Using web store shopping management system, customers can shop using without internet. Customer when enter the site after entering a shop, customer can browse through the products available in the shop, can select some of them and put into the shopping cart. Finalize product list of items he finally wish to buy and make the final payment.

1. Web store flow diagram

Data diagram of the web store is like online shopping considered which helps to user for shopping anywhere anytime without going any shopping mall. We select the Using online shopping management system, customers can shop using internet.

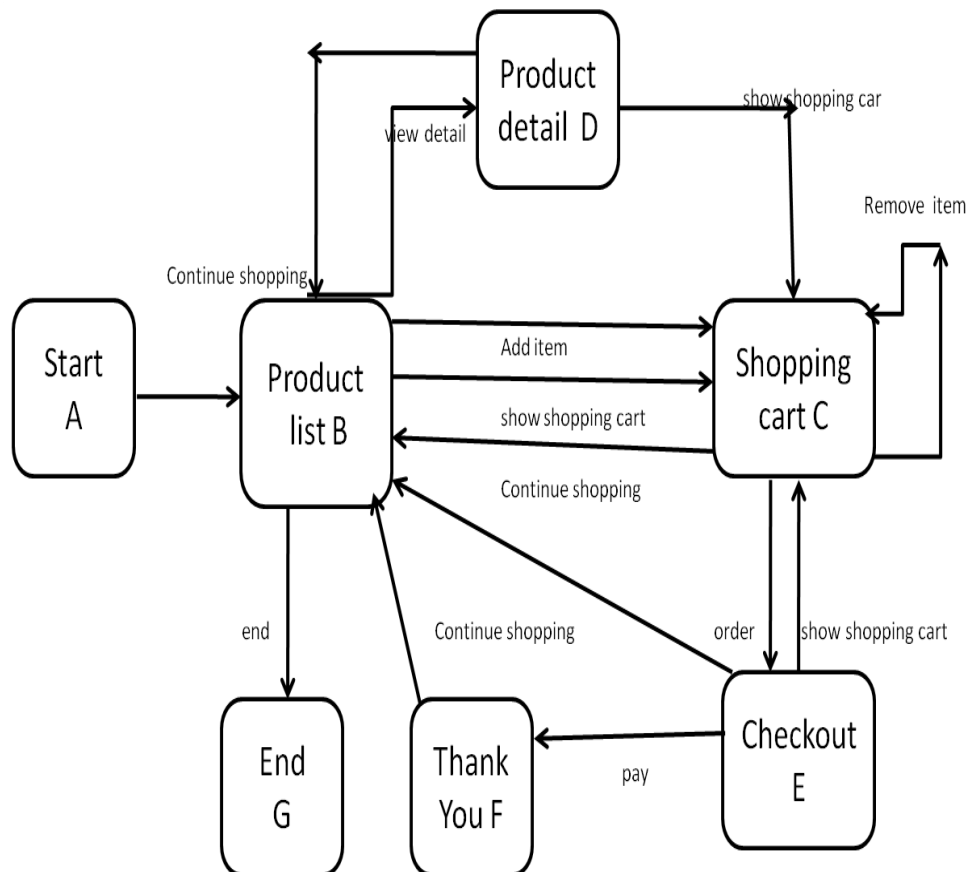


Fig.1 Web store flow diagram

2. Web store test tree

One of the important characteristic for a tree structure is branching. The advantageous of using data flow diagram is that we can easily find path but disadvantageous is that the path is cyclic with no termination there are starting node but not an ending node specified. Data tree diagram are used to generate shortest path are generated.

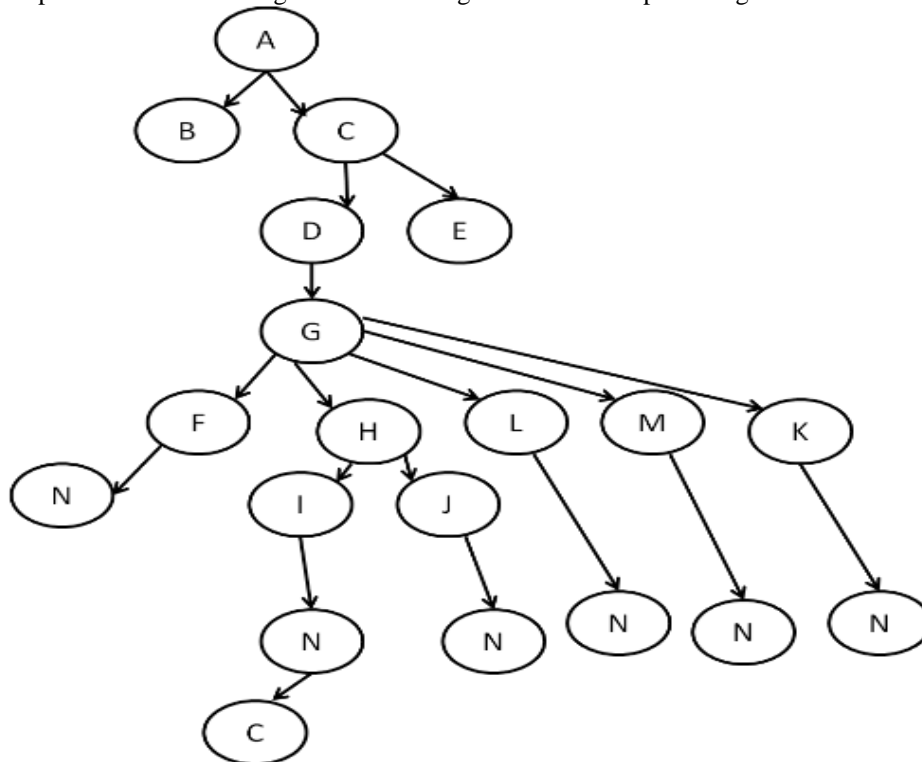


Fig.2 Web store flow diagram

Web store data tree, each form starting node to end node corresponding to a simple test path (Sequence) therefore there are as following

1. A-> B->C->B
2. A->B->C->E->C
3. A->B->D->B
4. A->B->D->C
5. A->B->C->E->F->B
6. A->B->C->E->B
7. A->B->C->E->C

In simple test path

Total number of test sequence=7

Total number of page travel=33

Total number of links travel=26

3. Optimization test case using coverage criteria:

According to the coverage criteria all state transaction have been coverage at least one time, it satisfy sets transition coverage. Consequently, we construct minimum test set of a web store system test tree figure 2 derived from figure 1 whose element are as follow

Reduce the no of test sequence we can follow of coverage criteria "All page coverage" according to this coverage criteria ,remove all copied page which are the leaf node and link corresponding to these copied page from the web model. By applied this method test sequence will automatically be reduced as it ensures that all the page is tested at least one.

Optimal test sequence in figure 2

1. A->B->C->E->C->B
2. A->B->D->B->C
3. A->B->C->E->E->F->B
4. A->B->C->E->B->C

In optimal test sequence

Total number of test sequence=4

Total number of page travel=23

Total number of links travel=19

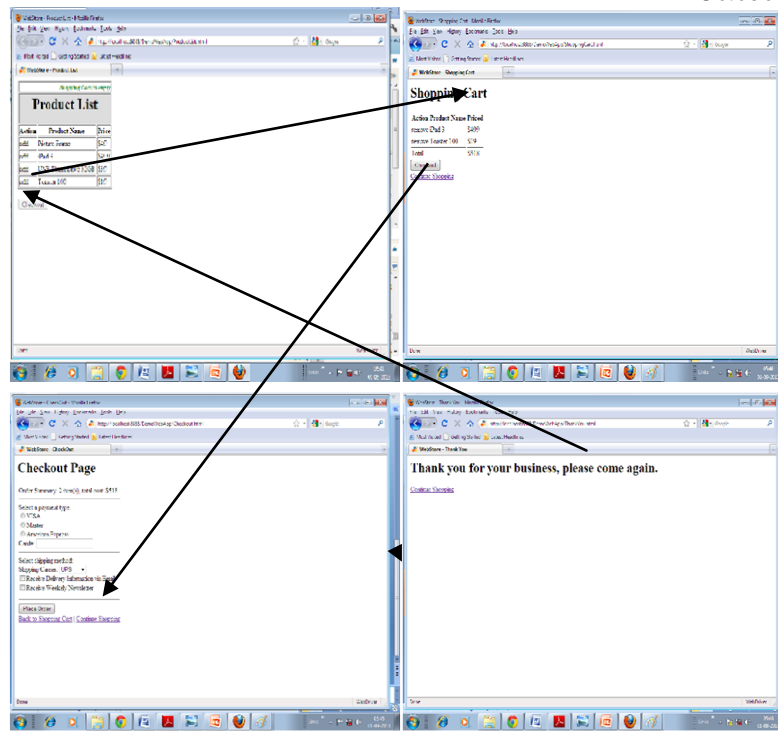


Fig 3.Processing Web store system

IV. GRAPHICAL WEB SEQUENCE GENERATED THOUGH TESTOPTIMAL

TestOptimal is a model-based test development, deployment and execution tool. This tool is created as a result of strong collaboration between highly skilled and experienced software development engineers and world class test engineers. Comprising the best attributes of both communities has given rise to a highly capable, adaptable and easy to apply model-based test environment. Traditionally, TestOptimal is installed and hosted via a web browser, although hosting on Eclipse is possible. Outside of using TestOptimal to develop models for live automatic testing of web applications any of the popular web browsers may be used to host TestOptimal. Model based Testing is black box testing technique for derivation of test cases from a model that describe functional aspects of the system under test and executing those test cases. "Model Based testing is the automation of test design of black box testing". Several types of test cases can be designed based on the design model. The use of a model to describe the behavior of a system is a proven and major advantage to test development teams. Models are used to understand, specify and develop systems. Models give an abstract view of the product hiding all the low level details.

1. Modeling
2. Test generation
3. Automation
4. Execution
5. Analysis

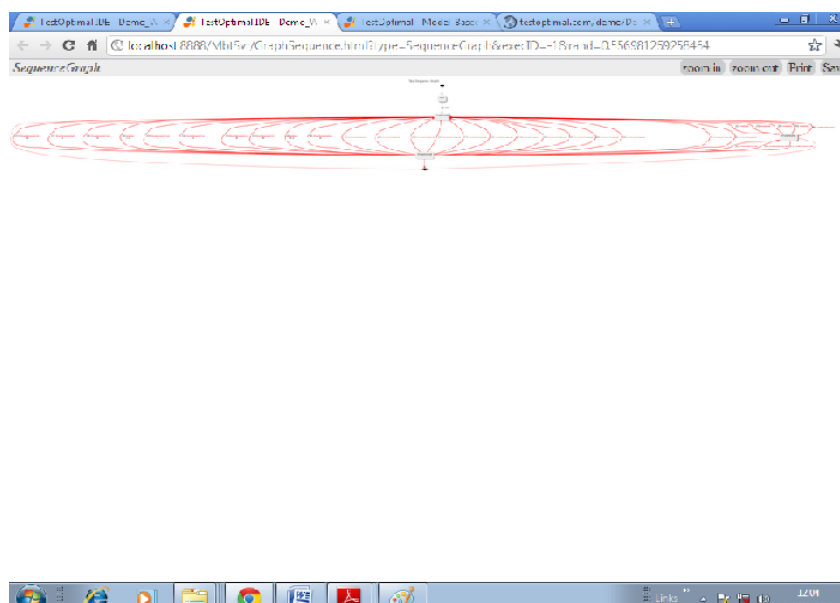


Fig 4 web store sequence graph

V. CONCLUSION

The primary aim of our work was to develop automatic generation of graphical web model of web component based on regression testing we have propose a method to manually and automatically generate test sequence graph from web store. In this thesis main focus on the web store system process in which they are show in data flow diagram, data test tree and the optimal test sequence generated to minimize the number of test sequence used in the model.

First web store model are converted into web store test tree to remove cyclic property of the web store model simple test sequence and optimal test sequence are obtain. Web store website are automatically execute through testoptimal model based testing tool. Testoptimal through automatically generate sequence graph, coverage graph and web store execute coverage graph.

We conclude this thesis and highlight some further extension, which are discussed below:

1. We work primarily on test sequence generation. Next, we want to optimize the test sequence for getting an appropriate amount of test sequence by eliminating redundant test sequence.
2. In the present work, we have implemented generate test sequence automatically using our approach. In future, we will implement a prototype tool.

REFERENCES

- [1] 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
- [2] 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.
- [3] Mr.Rohit N.devikar, ”automation of model based regression testing”, International Journal and scientific and research publication, Volume 2, Issue12, December, 2012.
- [4] Andrews, A., J. Offutt, R. Alexander, 2004. Testing web applications by modelling with FSMs. Software Syst. Modeling, 4: 326-345.
- [5] Alshahwan, N. and M. Harman, 2008. Automated session data repair for web application regression testing. Proceeding of the International Conference on Software Testing, Verification, and Validation, April 9-11, Lillehammer, pp: 298-307. 10.1109/ICST.2008.56
- [6] Briand, L. and Y. Labiche, 2002. A UML-based approach to system testing. Software Syst. Modelling, 1: 10-42, DOI: 10.1007/s10270-002-0004-8