IMPLEMENTATION OF AN OPTIMAL APPROACH TO TESTING WEB BASED APPLICATIONS

Amanpreet Singh Brar, Ashish Jalota.

GNDEC Ludhiana, Punjab, India.

ARTICLE INFO

Corresponding Author: Amanpreet Singh Brar
GNDEC Ludhiana, Punjab, India.

Keywords: Architecture of the system, Methodology of the system, Implementation.

ABSTRACT

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Software has become a driving force. It is the engine that drives business decision making and serves as the basis for the modern scientific investigation and the engineering problem solving. It is embedded in systems of all kinds: transportation, telecommunication, medical, military, industrial process, entertainment, office products etc. The failure of software effects economy, data, time, reputation etc all in negative manner. All the disadvantageous consequences with system failure are motivating forces for well planned software through testing.

INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Software has become a driving force. It is the engine that drives business decision making and serves as the basis for the modern scientific investigation and the engineering problem solving. It is embedded in systems of all kinds: transportation, telecommunication, medical, military, industrial process, entertainment, office products etc. The failure of software effects economy, data, time, reputation etc all in negative manner. All the disadvantageous consequences with system failure are motivating forces for well planned software through testing.

During the early days of internet, it was primarily a typical client-server configuration with little flexibility or scalability that supported limited functionality and simple applications. Web applications were typically static, i.e. composed only of web pages stored in some file system, linked together with hyperlinks. Aim of web applications was to provide information across the web in a rather simple and intuitive manner. Quality assurance was therefore a relatively unchallenged task. In the last few years this situation has dramatically changed. An “N-tier” model has been widely adopted to significantly improve quality factors such as scalability, flexibility, functionality and availability. The rise of new technological tools has increased the complexity. This rapid evolving technology provides new challenges to the techniques used to develop software.

1. ARCHITECTURE OF THE SYSTEM.
1) **Model graph intializer**: Model graph intializer the uml use case diagram model and we browse it in the developed software for its workings.

2) **Test Creator**: It creates the test cases for the desired uml use case diagram. It also gives us the path to be followed and covered by our develop software in order to test it properly.

3) **Test case optimizer**: It optimizes the test cases by removing the redundancy and find the shortest possible paths to test the entire project completely. Basic advantage of using this is to speed up the execution of the program. To reduce the number of test sequences we can follow up another coverage criterion i.e. “All Page Coverage”. According to this coverage criteria, remove all the copied pages which are the leaf nodes and links corresponding to these copied pages from the PTT.

4) **Test case output**: This phase generates the actual optimized output.

II. **METHODOLOGY OF THE SYSTEM**

---

**Fig 2 Methodology of the system**

- a) The algorithm to iterate model is:
- b) Define Tracker as Stack, TestList as List
- c) Set Next Available Page = “Home Page”
- d) Push all direct links to Tracker.
- e) Repeat Until Stack Empty
- f) Pop First direct Link
- g) Create test case
- h) Add test to list.
- i) Push all direct links to Tracker.
- j) Repeat
- k) Optimize tests

---

**Fig 3 Tasks**

1) Create list of distinct test case
2) Return test list.

III. **Implementation Tasks**

---

**Fig: 4 News Updates**

Below is the screen shot showing web model of application:

---

**Fig : 5 Web Model**
Here is the screen shot for Application generated test sequences:

Fig: 6 Test Sequences

CONCLUSION
The implementation here is done on small parts of a web application with maximum of eight pages. The proposed approach can be implemented on large and complex web application involving a number of links, frames and other static and dynamic components. We may think of using “Divide & Conquer” technique to analyze and test complex web applications and to model them separately to attain sub UML for each web component. Finally we can generate test sequences and optimize them for each web component.

REFERENCES