

An Approach to Generate the Test Cases for GUI Testing

¹Kanchan Gautam , ²Madhuri Sharma

¹Department of Computer Science and Engineering, Bharat Institute of Technology , Uttar Pradesh Technical University
Meerut , Uttar Pradesh ,India

²Department of Computer Science and Engineering , Bharat Institute of Technology ,Uttar Pradesh Technical University
Meerut , Uttar Pradesh ,India

Abstract

Graphical user interfaces are the one of most popular topic in today's software. The correctness of a GUI is required to ensure the, robustness, safety, reliable and usability of an entire software system.

In GUI testing ,the generation of event sequences is the more challenge. In testing phase, testing has been divided into graphical user interface based testing, unit testing, logical testing, integration testing, etc. The testing needs that it provides effectiveness, efficiency, correctness. In this paper there is to study some approaches for GUI testing and techniques used for test case generation.

Keywords- GUI testing, GUI Testing definition ,Test case , Event Flow Graph, Model based testing

1. Introduction

Graphical User Interface testing is the process of testing a product's GUI to ensure it meets its written specification.

A user action, such as clicking a mouse button , keyword trigger an event in the application. An event sequence is an main part of a GUI test case. The goal is to select a set of event sequences in a effective manner. Here, EFG is a graph whose nodes are events and whose paths represent event sequences. An event sequence on a path can be used for a GUI test case.

2. GUI Testing

A GUI is a hierarchical, graphical frontend of a software system that accepts as input from user and system generated events and produces output.

GUI testing is a process to test application's user interface and to detect if application is functionally correct. GUI testing involves carrying set of tasks and comparing the result of same with the expected output and ability to repeat same set of tasks multiple times with different data input and same level of accuracy.

GUI Testing includes

- how the application handles mouse and keyboard events.
- how different GUI components like menubars, toolbars, buttons, edit fields, list controls etc. reacts to user input.
- As it performs or not in the desired manner.

Success of any GUI application depends on how the user actions are performed to access application's features, how it interacts with user through its user interface, and whether application responds in functionally correct manner.

3. Test Case

A test case is a set of variables or conditions. A test case is usually a single step, or a sequence of steps, to test the correct functionality, features of an application.

IEEE Standard 610 (1990) defines test case as follows:

“A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement”.

Objective of Test Cases

- Find the defects in software products.
- Improve software quality.
- Verify that the software meets the end user requirements.
- Minimize the maintenance and software cost.
- Help management to make decisions in software delivery .

3.1 Formal test cases

A formal written test-case is characterized by a known input and by an expected output, which is worked out before the test is executed. The known input is test pre condition and the expected output is test a post condition .

3.2 Informal test cases

For applications without formal requirements, test cases can be written based on the accepted normal operation of programs of a same class. They are usually different from test cases in that test cases are single steps while scenarios cover a number of steps of the key.

4. Approaches for GUI based testing

4.1 Manual based

In this approach, it is done by humans. Graphical are checked manually by testers or developers. In this there are

chances that most test scenarios are left and it is time consuming also.

4.2 Capture and Replay

GUI testing can be done using automation tools. This is done in 2 part . During Record , test steps are captured into the automation tool. During playback the recorded test steps are executed on the Application .The recorded test cases can be replayed by the software using the replay part of the tool.

4.3 Model based testing

There are 2 type of model to use. It is based on the execution of user sessions.

4.3.1 Event-based model

4.3.2 State-based model

A model is a graphical description of system's behavior. It helps us to understand and predict the system behavior . Models help in a generation of efficient test cases using the system requirements. Following requirement to be consider for this model based testing:

- Build the model and determine inputs for the model .
- Calculate expected output for the model .
- Run the tests .
- Compare the actual output with the expected output .
- Decision on action on the model .

Model based testing is an evolving technique for generating the test cases cases from the requirements. Its main advantage compared to above two methods,

is that it can determine undesirable states that the GUI can attain.

4.3.1 Event-based Model

Model the space of GUI event interactions as a graph .

Given a GUI

- create a graph model of all the possible sequences that a user can execute.
- use the model to generate event sequences.

EFG Graph

An Event-flow graph (EFG) is a directed graph modelling all possible event sequences that may be executed on a GUI. An EFG contains nodes representing GUI events (e.g., click on a button, select a check box). Edges representing the "follows" relationships between them. The EFG model provides a "GUI map" for systematically generating test case.

- The main window shows 5 events (3 buttons and 2 checkbox)
- When the *Exit* button is clicked, a confirmation window is opened consisting of 3 other events (2 button and 1 checkbox).

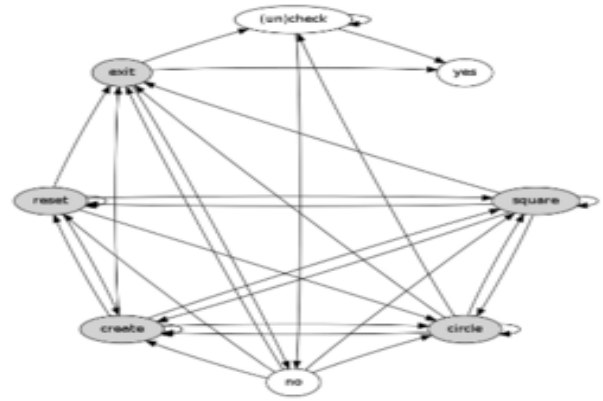


Fig 2 .

In fig 2, the EFG of this GUI is shown , where dark nodes mark the initial events. Moving on this graph, starting from an initial event, produces GUI test cases.

Examples of **GUI test case** generated are: <exit, yes>; <square, create, exit, yes> .

4.3.2 State-based Model

The most popular among them are state-machine models that have been proposed to generate test case for GUIs . The key idea of using these models is that a test designer represents a GUI's behavior as a state machine. A path, i.e. sequence of edges during transitions, in the state machine represents a test case.

Model the space of GUI event interactions as a state model, e.g., by using a finite state machine (FSM):

- States are representation of the GUI.
- Transitions are the GUI events that change the GUI state .
- create a FSM of the possible sequences that a user can execute, consider the GUI state use the FSM to generate event sequences .

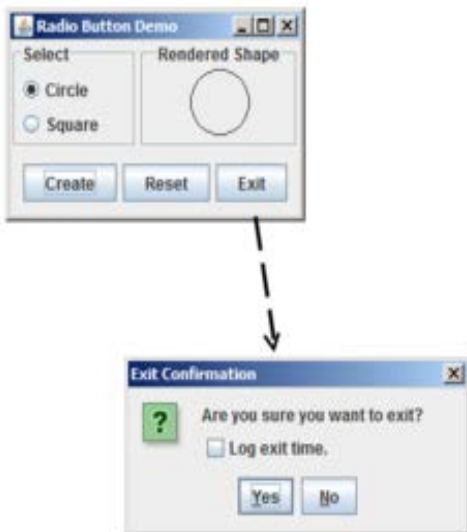


Fig 1.

5. Conclusions

In this paper, there are many different approaches used for GUI testing. Model based testing is better than the other two approach. In this automated tools can used for generating the test cases.

This presented a model (called the event-flow model) of the GUIs .The event-flow model may be used for various aspects of GUI testing. The event-flow model can be used to quickly generate the GUI test cases in a large number that are effective for GUI faults. The model is reusable because once created it can be quickly convert to generate additional test cases for the same or a modify version of the GUI.

References

- [1] Atif M. Memon ,Martha E. Pollack and Mary Lou. Soffa, “Hierarchical GUI Test Case Generation Using Automated Planning”, *IEEE Transactions on Software Engineering*, vol. 27, no. 2, Feb 2001, pp.144-155.
- [2] Xun Yuan, Myra B. Cohen and Atif M. Memon, “GUI Interaction Testing: Incorporating Event Context” , *IEEE Transactions on Software Engineering* ,vol.37 ,no.4, July/August 2011.
- [3] Prabhu. J , Dr. N .Malmurugan,“Automated GUI Test Cases Generation with Optimization Algorithm using a Model Driven Approach” , *Indian Journal of Computer Science and Engineering (IJCSE)*vol.4 ,no.5, Oct-Nov 2013 .
- [4] Atif M .Memon, “An event –flow model of GUI-based applications for testing” ,*Software Testing, Verification and Reliability* , Nov 2006.
- [5] Marlon Vieira, Johanne Leduc, Bill Hasling, Rajesh Subramanyan, Juergen Kazmeier, “Automation of GUI Testing Using a Model-driven Approach” May 2006.
- [6] Atif M. Memon , Martha E. Pollack and Mary Lou Soffa, “Using a goal driven approach to generate test case for guis” *21st International conference on software Engineering* ,1999.
- [7] Om Kumar C.U. , P. Bhargavi & Vinod Kumar. K, “A Single Model for Event -Driven Software”,*International Journal on Advanced Computer Theory and Engineering (IJACTE)*, vol 2, 2013.
- [8] Stephan Arlt, Andreas Podelski, Cristiano Bertolini, Martin Schaf, Ishan Banerjee, Atif M. Memon, “Lightweight Static Analysis for GUI Testing” ,2012.