Testing GUI Applications

Abstract

- Most clients in client/server systems deliver system functionality using a graphical user interface (GUI).
- GUIs make testing systems more difficult for many reasons:
  - the event-driven nature of GUIs
  - unsolicited events
  - many ways in/many ways out
  - the infinite input domain problems
- The programmer has introduced errors because he could not test every path.

1. Introduction

- GUIs as universal client
  - provide the standard look and feel of a client operating system.
  - so flexible that they can be used in most application areas.
  - provides seamless integration of custom and package applications.
  - The user has a choice of using the keyboard or a mouse device.
  - The user has a more natural interface to applications: multiple windows can be visible simultaneously, so user understanding is improved.
  - The user is in control: screens can be accessed in the sequence the user wants at will

GUIs

- Graphic User Interfaces
- GUI allows multiple windows to be displayed at the same time
- Buttons and keyboard shortcuts enable the user to navigate and access the various functions of their application.
- Tabbing order, free to use the mouse

  → There are no constraints on the order in which a user may enter data on a screen

GUIs

→ GUIs free the user to access system functionality in their preferred way. They have permanent access to all features and may use the mouse, the keyboard or a combination of both to have a more natural dialogue with the system.

2. Some testing difficulties

- Event-driven software
- Unsolicited events
- Object oriented
- Hidden synchronisation and dependencies
- 'Infinite' input domain
- Many ways in, many ways out
- Window management

Event-driven software

- The event-driven nature of GUIs:
  - Many click on any pixel on the screen
  - Many more possible user inputs that can occur
  - At any point in the application, the users may click on any field or object within a window
  - Many events are handled 'behind the scenes'
  - → The '**infinite paths**' problems…

Unsolicited events

- Difficult
  - number of test cases may be high
  - but also special test drivers may be necessary to generate such events within the operating systems

Hidden synchronisation and dependencies

- Behind one click → many rules
- Sync
  - problem: where are these dependencies?
- Other examples ?

Infinite' input domain

- Input data domain (talk later)
- Click, Shortcut Key, other devices
- How many situations in Wordpad, MS Word ?

Many ways in, many ways out

- Many ways in' by which the user reached that point in the application

Window management

- movement, resizing, maximisation, minimisation and closure

3. GUI Test Strategy

- Test Principles Applied to GUIs
- High Level Test Process
- Types of GUI errors
- Four Stages of GUI Testing

Types of GUI errors

- Data validation
- Incorrect field defaults
- Mis-handling of server process failures
- Mandatory fields, not mandatory
- Wrong fields retrieved by queries
- Incorrect search criteria
- Field order
- Multiple database rows returned, single row expected
- Currency of data on screens
- Window object/DB field correspondence

Types of GUI errors

- Correct window modality?
- Window system commands not available/don't work
- Control state alignment with state of data in window?
- Focus on objects needing it?
- Menu options align with state of data or application mode?
- Action of menu commands aligns with state of data in window
- Synchronisation of window object content
- State of controls aligns with state of data in window?

Four Stages of GUI Testing

Stage	*Test Types*
Low Level	Checklist testing Navigation
Application	Equivalence Partitioning Boundary Values Decision Tables State Transition Testing
Integration	Desktop Integration C/S Communications Synchronisation
Non-Functional	Soak testing Compatibility testing Platform/environment

4.Types of GUI Test

- Checklist Testing
- Navigation Testing
- Application Testing
- Desktop Integration Testing
- Client/Server Communication Testing
- Synchronisation Testing
- Non-Functional Testing

5. Test Automation

- Justifying Automation
- Automating GUI Tests

6. Improving the testability of GUI Applications

- The GUI Testing Challenge
  - It is difficult to specify tests
  - It is difficult to prepare tests
  - It is difficult to execute tests
  - It is difficult to analyse tests
- GUI Design for Testability
  - 9 recommendations