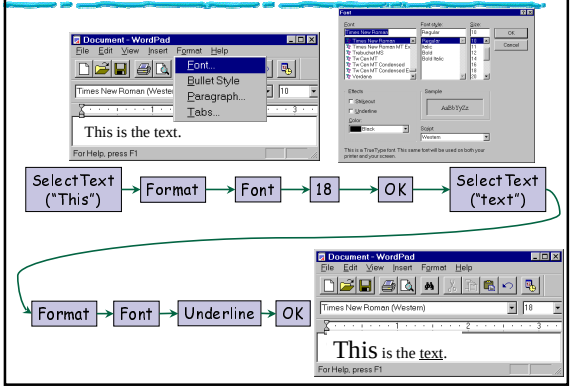
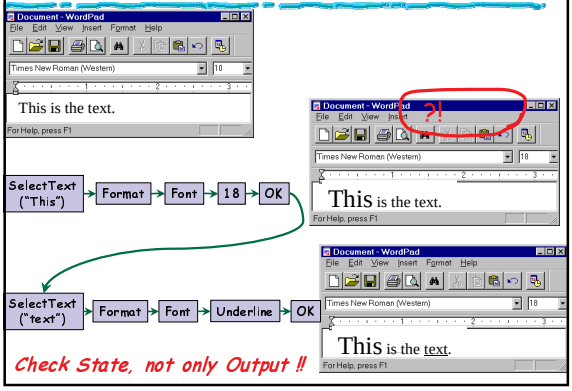
Automated Test Oracles for GUIs

Eighth International Symposium on the Foundations of Software Engineering, San Diego, CA, Nov. 6-10, 2000.

A Test Case for WordPad

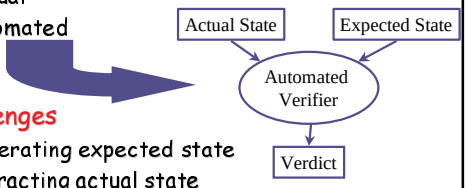


What Is Correct Behavior



Research Focus

- Goal
  - To check the GUI's state after each event
- Approaches
  - Manual
  - Automated

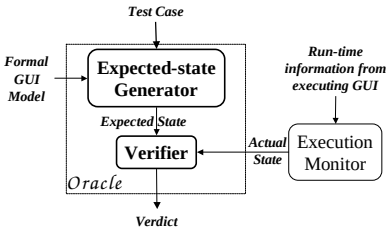


- Challenges
  - Generating expected state
  - Extracting actual state
  - Comparing expected & actual states

Outline

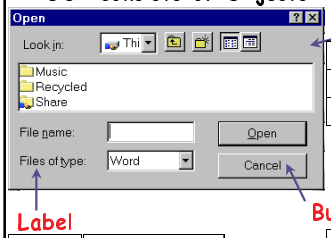
- Overview of GUI Oracle
- Generating Expected State
  - Modeling the GUI's State
    - Objects
    - Properties
  - Modeling the Events
- Obtaining Actual GUI's State
- Comparing Actual & Expected States
- Case Study: MS WordPad
- Concluding Remarks

Overview of GUI Oracle



Modeling the GUI

A GUI consists of Objects



Form

Window State	wsNormal
Width	1088
AutoScroll	TRUE

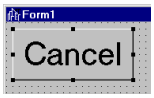
Label

Align	alNone
Caption	Files of type:
Color	clBtnFace
Font	(tFont)

Button

Caption	Cancel
Enabled	TRUE
Visible	TRUE
Height	65

All Properties of Cancel



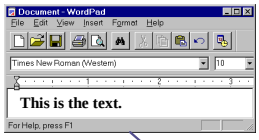
Determining Properties

- Manual Examination of GUI
- Specifications (Reduced Set)
  - GUI being tested
- Toolkit/Language (Complete Set)
  - All available properties

Now we know how to represent the GUI's state

Modeling Events

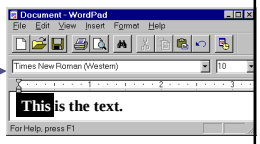
- Events are State Transducers



State: S_i

SelectText ("This")

Event: e



State: S_j

Notation: $S_j = [S_i, e]$

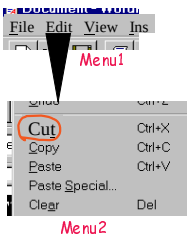
Representing Events

- We define an event as:

  $State_j = [State_i, event]$
- For example:

  $State_j = [State_i, cut]$
- Need a compact representation

Operators



Operator :: *CUT*
Preconditions:
isCurrent(Menu2).

Effects:
FORALL Obj in Objects
Selected(Obj) ⇒
ADD inClipboard(Obj)
DEL onScreen(Obj)
DEL Selected(Obj)
ADD isCurrent(Menu1)
DEL isCurrent(Menu2).

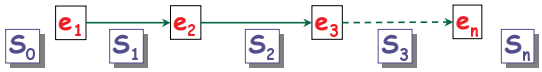
Obtaining next state

Deriving Expected State

13

- Given S_0, the initial state,
- A sequence of events

- Obtain $S_1 = [S_0, e_1]$
- And $S_i = [S_{i-1}, e_i]$



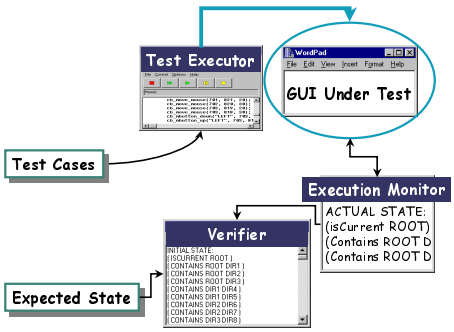
Obtaining Actual GUI's State

14

- Execution Monitor
  - Screen Scraping
  - Queries
  - Compatible with Expected State
  - Returns <Object, Property, Value>
    <Button1, "Caption", "Cancel">

Automated Execution

15



Comparing Actual and Expected States

16

- Verifier
- Three Levels of Testing
  - Changed Property Set *(Operators)*
  - GUI Relevant Property Set *(Specifications)*
  - Complete Property Set *(Toolkit/Language)*
- Hybrid Approach
  - Use all 3

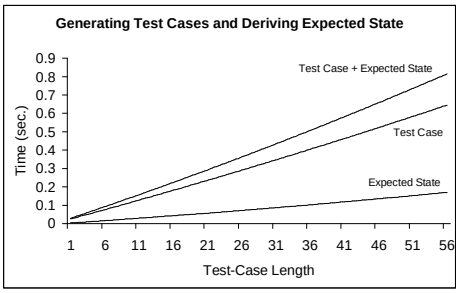
Case Study

17

- Purpose: Determine
  - Time to Derive Expected State
  - Time to Execute Monitor and Verifier
- Experimental Design
  - GUI: Our Version of MS WordPad (36 Modal Windows, 362 events)
  - Test Cases: Generated 290 Test Cases (6-56 events) using an AI Planner
  - Hardware Platform: 350 MHz Pentium based Machine, 256 MB RAM
  - Properties: Reduced Set
  - Level of Testing: GUI Relevant Property Set

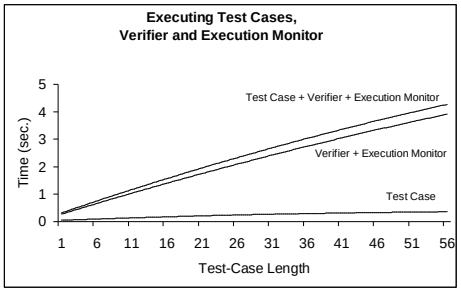
Deriving Expected State

18



Total CPU time (test case and expected state)
75.84 sec.

Execution



Relevant-properties verification
Total running time < 10 minutes