

**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

Check State, not only Output !!

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	<i>wsNormal</i>
Width	<i>1088</i>
AutoScroll	<i>TRUE</i>

Label

Align	<i>alNone</i>
Caption	<i>Files of type:</i>
Color	<i>clBtnFace</i>
Font	<i>(tFont)</i>

Button

Caption	<i>Cancel</i>
Enabled	<i>TRUE</i>
Visible	<i>TRUE</i>
Height	<i>65</i>

All Properties of Cancel

Properties	Events
Cancel	true
Caption	<i>Cancel</i>
Cursor	crDefault
Default	false
DragCursor	crDrag
DragMode	dmlManual
Enabled	true
Font	(TFont)
Height	65
HelpContext	0
Hint	
Left	8
ModalResult	mrNone
Name	Button1
ParentFont	false
ParentShowHint	true
PopupMenu	
ShowHint	false
TabOrder	0
TabStop	true
Tag	0
Top	8
Visible	true
Width	153

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

Event: e

Notation: $S_j = [S_i, e]$

State: S_j

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:
 $isCurrent(Menu2)$.

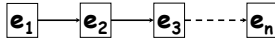
Effects:
FORALL Obj in Objects
 $Selected(Obj) \Rightarrow$
 ADD inClipboard(Obj)
 DEL onScreen(Obj)
 DEL Selected(Obj)

ADD isCurrent(Menu1)
DEL isCurrent(Menu2).

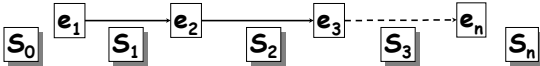
Obtaining next state

Deriving Expected State

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



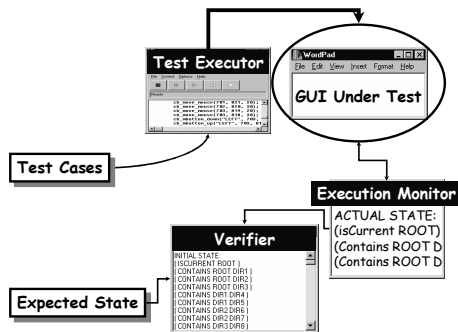
- Obtain $S_1 = [S_0, e_1]$
- And $S_i = [S_{i-1}, e_i]$



Obtaining Actual GUI's State

- Execution Monitor
 - Screen Scraping
 - Queries
 - Compatible with Expected State
 - Returns $\langle \text{Object, Property, Value} \rangle$
 $\langle \text{Button1, "Caption", "Cancel"} \rangle$

Automated Execution



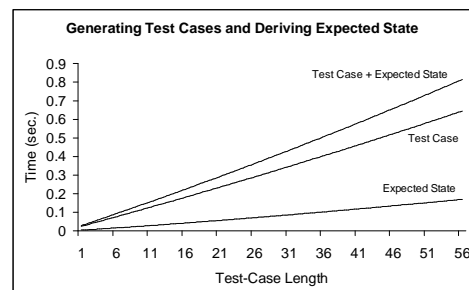
Comparing Actual and Expected States

- 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

- 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

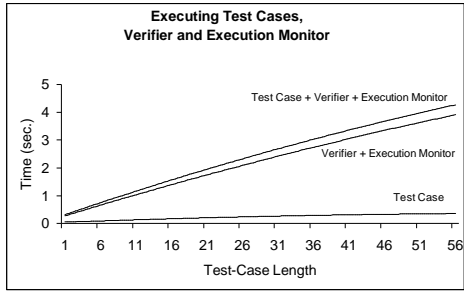
Deriving Expected State



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

Execution

19



*Relevant-properties verification
Total running time < 10 minutes*