Automated Test Oracles for GUIs


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

<table>
<thead>
<tr>
<th>Label</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Align</td>
<td>Caption</td>
</tr>
<tr>
<td>Color</td>
<td>Font</td>
</tr>
<tr>
<td>Button</td>
<td>File Name</td>
</tr>
<tr>
<td>Window State</td>
<td>Width</td>
</tr>
<tr>
<td>AutoScroll</td>
<td>Height</td>
</tr>
</tbody>
</table>

All Properties of Cancel

<table>
<thead>
<tr>
<th>Button</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cancel</td>
<td>Window State</td>
</tr>
<tr>
<td>Width</td>
<td>AutoScroll</td>
</tr>
</tbody>
</table>

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(Menu1)

Obtaining next state
Deriving Expected State

Given $S_0$, the initial state,
A sequence of events $e_1, e_2, e_3, \ldots, e_n$
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 $<$Object, Property, Value$>$$<$Button1, “Caption”, “Cancel”$>

Automated Execution

Test Cases
Execution Monitor
Verifier
Expected State

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

Generating Test Cases and Deriving Expected State

Test Case + Expected State
Total CPU time (test case and expected state)
$75.84$ sec.
Executing Test Cases,
Verifier and Execution Monitor

Test Case Length

Time (sec.)

Relevant-properties verification
Total running time < 10 minutes