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

Test Case

Expected-state Generator

Run-time information from executing GUI

Actual State

Oracle

Verifier

Actual State

Execution Monitor

Verdict
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.

Representing Events

- We define an event as:
  \[ \text{State}_j = [\text{State}_i, \text{event}] \]
- For example:
  \[ \text{State}_j = [\text{State}_i, \text{cut}] \]
- Need a compact representation

Operators

Operator :: \( \text{cut} \)

Preconditions:
\[ \text{isCurrent}(\text{Menu1}). \]

Effects:
\[
\begin{align*}
\text{FORALL } & \text{Obj in Objects} \quad \text{Selected(Obj)} \\
& \text{ADD incClipboard(Obj)} \quad \text{DEL onScreen(Obj)} \quad \text{DEL Selected(Obj)} \quad \text{ADD isCurrent(\text{Menu1})} \quad \text{DEL isCurrent(\text{Menu2})}
\end{align*}
\]

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

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

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

Executing Test Cases, Verifier and Execution Monitor

Test Case
Verifier + Execution Monitor
Test Case + Verifier + Execution Monitor

Relevant-properties verification
Total running time < 10 minutes