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

SelectText ("This") ➔ Format ➔ Font ➔ 18 ➔ OK ➔ SelectText ("text")

Format ➔ Font ➔ Underline ➔ OK

This is the text.


What Is Correct Behavior

This is the text.

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

Formal GUI Model

Expected-state Generator

Verifier

Oracle

Expected State

Actual State

Execution Monitor

Run-time information from executing GUI

Verdict
Modeling the GUI

A GUI consists of Objects

- Button
- Form
- Label
- Window State
- Width
- AutoScroll
- Align
- Caption
- Color
- Font
- Caption
- Enabled
- Visible
- Height

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: $S_i$

Event: $e$

Notation: $S_j = [S_i, e]$

State: $S_j$
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 :: *CUT*
Preconditions:
\[ \text{isCurrent}(	ext{Menu2}). \]

Effects:
\[ \text{FORALL Obj in Objects} \]
\[ \text{Selected(Obj)} \Rightarrow \]
\[ \text{ADD inClipboard(Obj)} \]
\[ \text{DEL onScreen(Obj)} \]
\[ \text{DEL Selected(Obj)} \]
\[ \text{ADD isCurrent}(	ext{Menu1}) \]
\[ \text{DEL isCurrent}(	ext{Menu2}). \]

Obtaining next state
Deriving Expected State

- Given $S_0$, the initial state,
- A sequence of events
  $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow \ldots \rightarrow e_n$
- 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 $<\text{Object}, \text{Property}, \text{Value}>$
    $<\text{Button1}, "Caption", "Cancel">$
Automated Execution

Test Executor

GUI Under Test

Execution Monitor

ACTUAL STATE:
(isCurrent ROOT)
(Contains ROOT D)
(Contains ROOT D)

Verifier

Test Cases

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**

<table>
<thead>
<tr>
<th>Test-Case Length</th>
<th>Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
</tr>
<tr>
<td>11</td>
<td>0.2</td>
</tr>
<tr>
<td>16</td>
<td>0.3</td>
</tr>
<tr>
<td>21</td>
<td>0.4</td>
</tr>
<tr>
<td>26</td>
<td>0.5</td>
</tr>
<tr>
<td>31</td>
<td>0.6</td>
</tr>
<tr>
<td>36</td>
<td>0.7</td>
</tr>
<tr>
<td>41</td>
<td>0.8</td>
</tr>
<tr>
<td>46</td>
<td>0.9</td>
</tr>
<tr>
<td>51</td>
<td>1.0</td>
</tr>
<tr>
<td>56</td>
<td>1.1</td>
</tr>
</tbody>
</table>

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

Time (sec.)
Test-Case Length

Relevant-properties verification
Total running time < 10 minutes