Plan Generation for GUI Testing

- The 21st International Conference on Software Engineering
- The Fifth International Conference on Artificial Intelligence Planning and Scheduling
- IEEE Transactions on Software Engineering

Research Focus

Interactions between the GUI and the Underlying Code
Why Planning for GUI Testing

- GUIs are Event Driven
- Individual User Events
  - NOT ENOUGH!
  - Sequences of User Events lead to Different States
- Test Case: Sequence of User Events
- How to Generate Test Cases?
- Use Planning to Select Likely Test Cases

Selecting Test Sequences

- Infinitely Many
- Randomly Choose Sequences
- Expert Chooses Sequences
- Automatically Generate Events for COMMONLY USED TASKS
A Plan for a GUI Task

Outline

• Using Planning for Test Case Generation
  - Overall Approach
  - Exploiting GUI Structure
  - Generating Alternative Test Cases
• Experimental Results
• Related Research
• Concluding Remarks
Overview of Test Generation

<table>
<thead>
<tr>
<th>Phase</th>
<th>Step</th>
<th>Test Designer</th>
<th>Automatic Planning-based System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Setup</td>
<td>1</td>
<td>Derive Planning Operators from GUI</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Code Preconditions and Effects of Operators</td>
<td></td>
</tr>
<tr>
<td>Test Case Generation</td>
<td>3</td>
<td>Specify a Task (Initial and Goal States)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Generate Test Cases</td>
<td></td>
</tr>
</tbody>
</table>

Straightforward Approach

- Define One Operator for each User Action

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).
Exploit the GUI’s Structure

- Reduce the Number of Operators
  - System more Efficient
  - Easier for the Test Designer

Opening Modal Windows
Opening Menus

- **File**
  - New... Ctrl+N
  - Open... Ctrl+O
  - Close

- **Save**
  - Save As...
  - Save as HTML...
  - Versions...

- **Page Setup**
  - Print Preview
  - Print...
  - Ctrl+P

- **Send To**
  - Mail Recipient
  - Routing Recipient...
  - Exchange Folder...
  - Fax Recipient...
  - Microsoft PowerPoint

Interacting with the Underlying Software
Create Hierarchical Operators

Two Types of Abstractions
- Combine Buttons? Create System-Interaction Operators
- Decompose GUI Hierarchically? Create Abstract Operators

Create System-Interaction Operators

Sys-Interaction Operator:
File_SendTo_MailRecipient
= <File + SendTo + MailRecipient>
Create Abstract Operators

Straightforward Approach
Main GUI's Operator Set

... Set Language
SelectFromList()
Default
OK
Cancel
...

Using Abstraction
Window's Operator Set

... SelectFromList()
Default
OK
Cancel

Language

Set Language

Define Abstraction

SetLanguage()

Abstract Operator

High Level Plan

SetLanguage() → ...

Planner

Sub Plan

SelectFromList
("English(US)"")

OK
Effects of Exploiting the GUI's Structure

• Reduction in Planning Operators
  - 325 operators ? 32 operators
  - Ratio 10:1 for MS WordPad
  - 20:1 for MS Word

• System Automatically Determines the System-interaction and Abstract Operators
Test Case

INITIAL

Primitive Operator
SelectText ("This")

Abstract Operator
FormatFont ("This", 18pt)

Primitive Operator
SelectText ("text")

Abstract Operator
FormatFont ("text", Underline)

GOAL
Alternative Test Case

```
Primitive Operator SelectText ("This")
Abstract Operator FormatFont ("This", 18pt)

Primitive Operator SelectText ("text")
Abstract Operator FormatFont ("text", Underline)
```

```
Primitive Operator SelectText ("This")
Abstract Operator SelectFromList (18)

Primitive Operator SelectText ("text")
Abstract Operator SelectText ("text")
Abstract Operator FormatFont ("text", Underline)
```
Methods to Generate Alternative Test Cases

- Different Results from Planner
- Abstract Operator Decompositions
- Linearizations of the Partial-order Plan

Feasibility Study

- **Purpose**
  - To Determine whether Planning is a Feasible Approach for GUI Test Case Generation
  - Execution Time
  - Human Effort

- **Experimental Design**
  - **GUI**: MS WordPad
  - **Planner**: IPP [Koehler et al. ‘97]
  - **Hardware Platform**: 300 MHz Pentium based Machine, 200 MB RAM, Linux OS
  - **8 Tasks, Multiple Test Cases for each Task**
Experimental Results

<table>
<thead>
<tr>
<th>(Task) Plan No.</th>
<th>Plan Time (sec.)</th>
<th>Sub Plan Time (sec.)</th>
<th>Total Time (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.16</td>
<td>0</td>
<td>3.16</td>
</tr>
<tr>
<td>2</td>
<td>3.17</td>
<td>0</td>
<td>3.17</td>
</tr>
<tr>
<td>3</td>
<td>3.2</td>
<td>0.01</td>
<td>3.21</td>
</tr>
<tr>
<td>4</td>
<td>3.38</td>
<td>0.01</td>
<td>3.39</td>
</tr>
<tr>
<td>5</td>
<td>3.44</td>
<td>0.02</td>
<td>3.46</td>
</tr>
<tr>
<td>6</td>
<td>4.09</td>
<td>0.04</td>
<td>4.13</td>
</tr>
<tr>
<td>7</td>
<td>8.88</td>
<td>0.02</td>
<td>8.9</td>
</tr>
<tr>
<td>8</td>
<td>40.47</td>
<td>0.04</td>
<td>40.51</td>
</tr>
</tbody>
</table>

Concluding Remarks

- Automatic Planning is a Feasible Approach for GUI Test Case Generation
- Automatic Generation of Preconditions and Effects from GUI Specifications
- Generate Expected Output (Automated Verification)