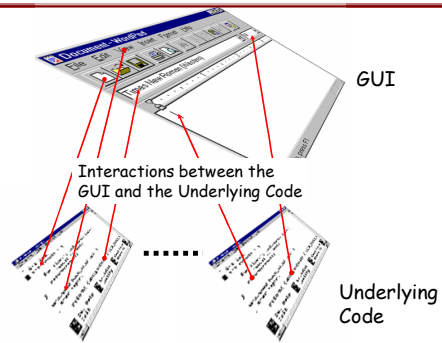


Plan Generation for GUI Testing

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

Research Focus



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

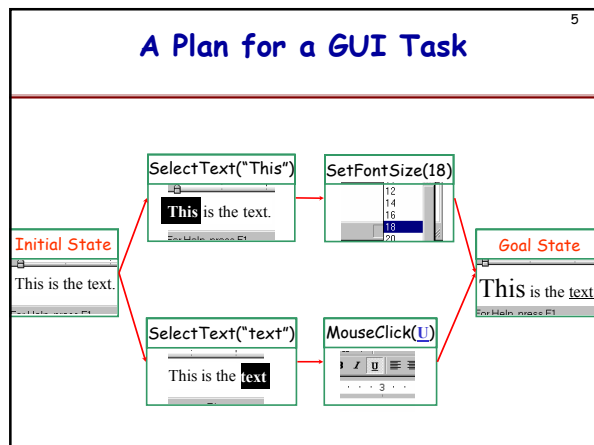
4

Selecting Test Sequences

- Infinitely Many
- Randomly Choose Sequences
- Expert Chooses Sequences
- Automatically Generate Events for
COMMONLY USED TASKS

Initial State

Goal State



6

Outline

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

7

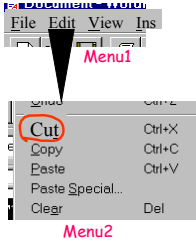
Overview of Test Generation

Phase	Step	Test Designer	Automatic Planning-based System
Setup	1		Derive Planning Operators from GUI
	2	Code Preconditions and Effects of Operators	
Test Case Generation	3	Specify a Task (Initial and Goal States)	
	4		Generate Test Cases

8

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).
          
```

9

Exploit the GUI's Structure

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

Opening Modal Windows

Set Language
 Thesaurus... Shift+F7
 Hyphenation...

Language
 Mark selected text as:
 English (Ireland)
 English (Jamaica)
 English (New Zealand)
 English (South Africa)
 English (Trinidad)
 English (United Kingdom)
English (United States)
The spellier and other proofing tools automatically use dictionaries of the selected language, if
 OK Cancel Default...

Opening Menus

File Edit View Insert Format Tools Table Window Help
 New... Ctrl+N
 Open... Ctrl+O
 Close
 Save Ctrl+S
 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

Underlying Software

Document - WordPad
 Edit Insert Format Help
 Cut Ctrl+X
Copy Ctrl+C
 Paste Ctrl+V
 Paste Special...
 Clear Ctrl+Z
 Select All Ctrl+A
 End... Ctrl+F
 Find... Ctrl+F
 Replace... Ctrl+H
 Format
 Document Properties...
 Help

13

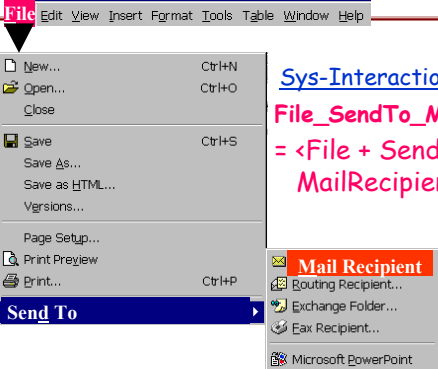
Create Hierarchical Operators

Two Types of Abstractions

- Combine Buttons \Rightarrow Create **System-Interaction** Operators
- Decompose GUI Hierarchically \Rightarrow Create **Abstract** Operators

14

Create System-Interaction Operators



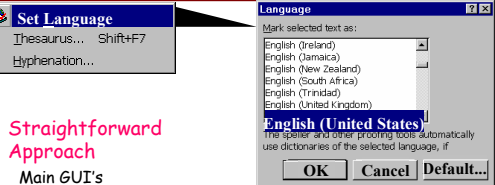
Sys-Interaction Operator:

File_SendTo_MailRecipient

= <File + SendTo + MailRecipient >

15

Create Abstract Operators



Straightforward Approach

Main GUI's Operator Set

...

Set Language

SelectFromList()

Default

OK

Cancel

...

Using Abstraction

Main GUI's Operator Set

...

Set Language

...

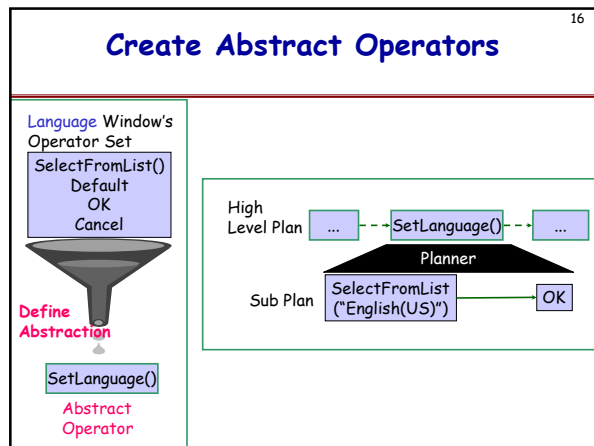
Language Window's Operator Set

SelectFromList()

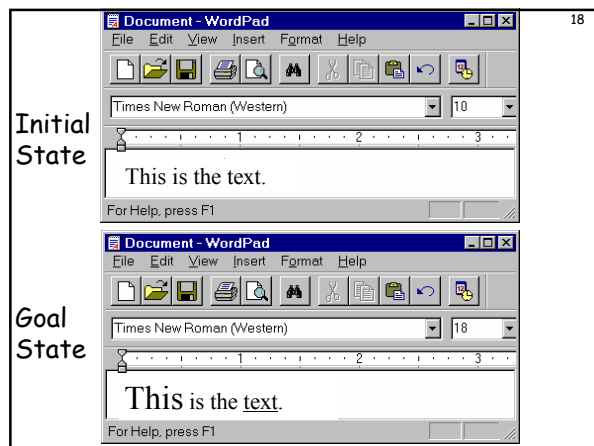
Default

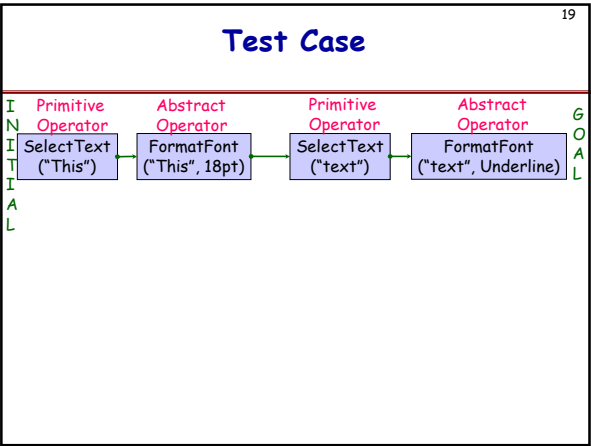
OK

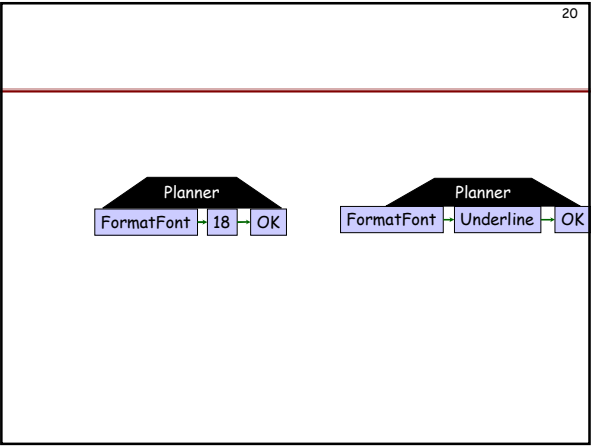
Cancel

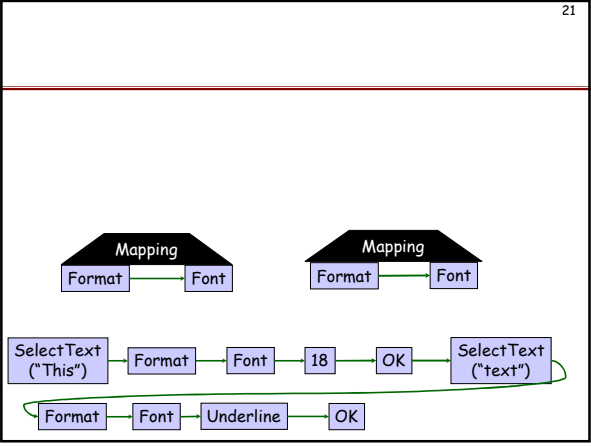


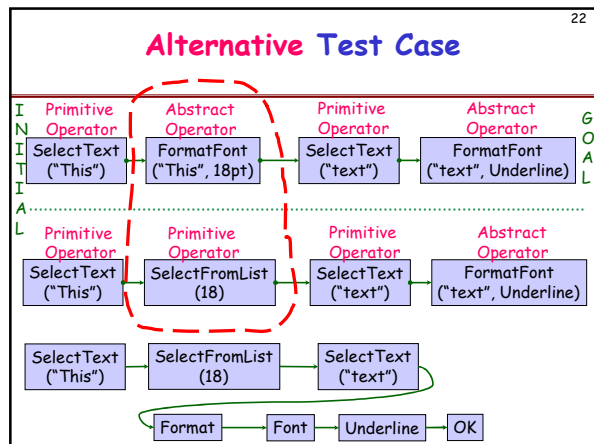
- ## Effects of Exploiting the GUI's Structure
- 17
- Reduction in Planning Operators
 - 325 operators \Rightarrow 32 operators
 - Ratio 10:1 for MS WordPad
 - 20:1 for MS Word
 - System Automatically Determines the System-interaction and Abstract Operators











23

Methods to Generate Alternative Test Cases

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

24

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

25

Experimental Results			
(Task) Plan No.	Plan Time (sec.)	Sub Plan Time (sec.)	Total Time (sec.)
1	3.16	0	3.16
2	3.17	0	3.17
3	3.2	0.01	3.21
4	3.38	0.01	3.39
5	3.44	0.02	3.46
6	4.09	0.04	4.13
7	8.88	0.02	8.9
8	40.47	0.04	40.51

26

Concluding Remarks
<ul style="list-style-type: none"> 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)
