

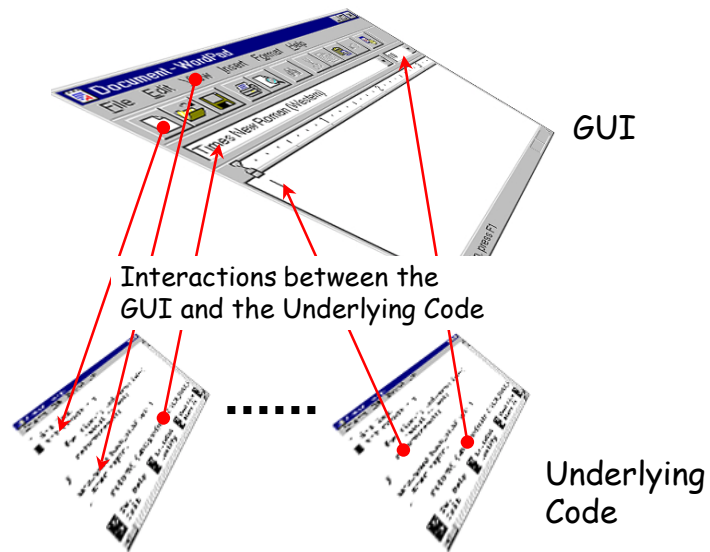
A Planning-based Approach to GUI Testing

Atif M. Memon
Martha E. Pollack
Mary Lou Soffa



Dept. of Computer Science
University of Pittsburgh

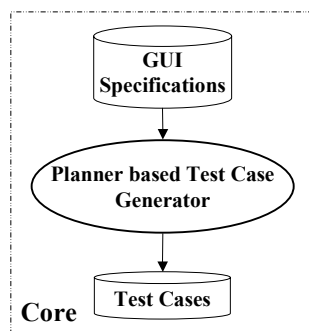
Testing a GUI

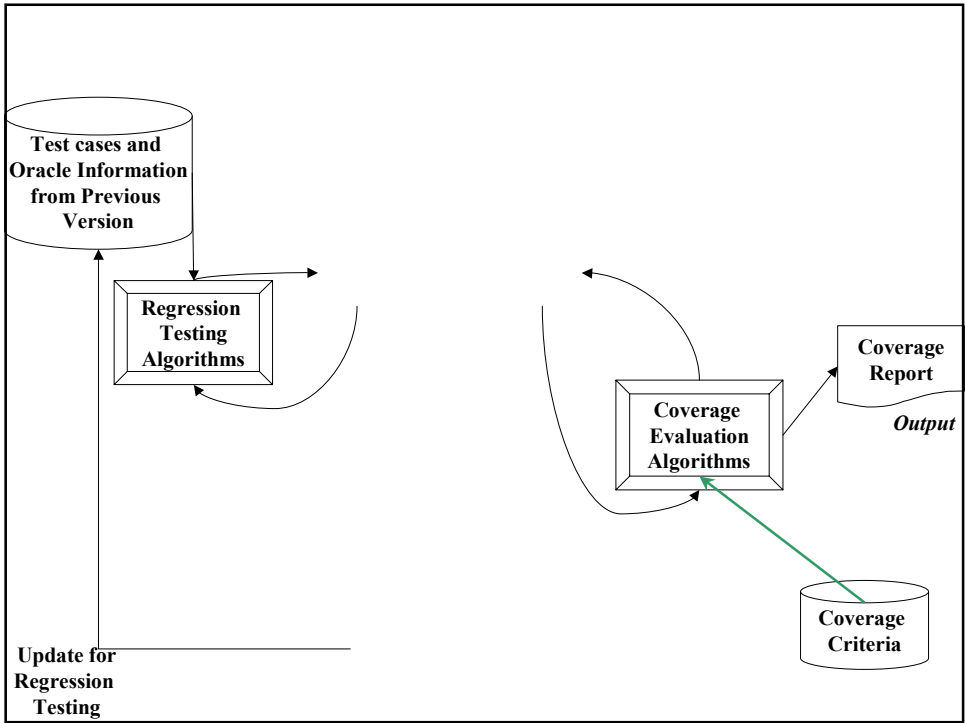
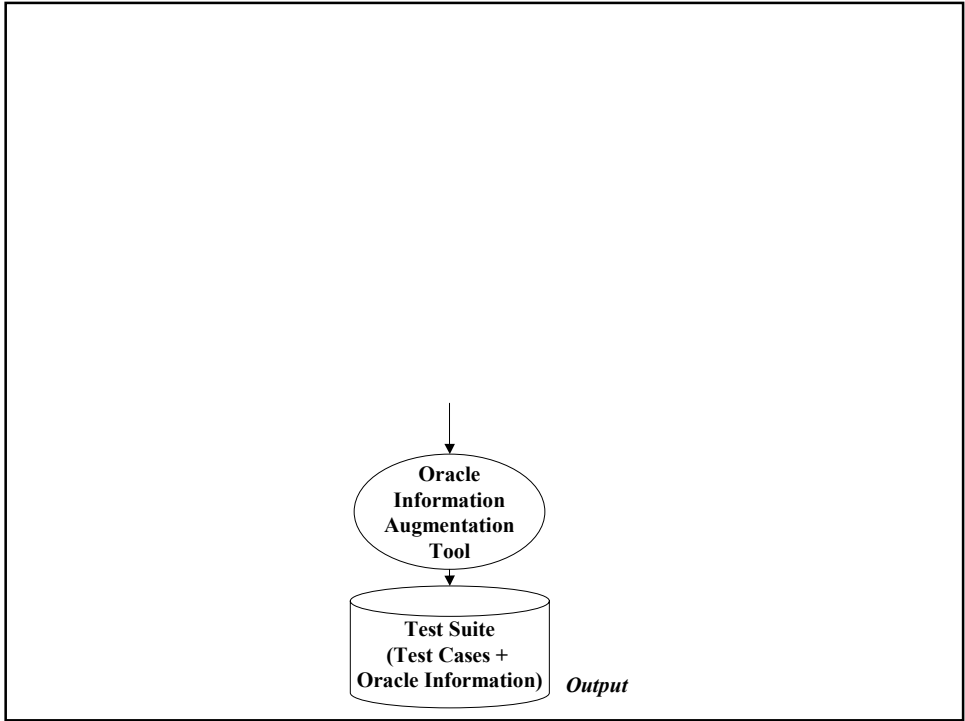


High-level Outline

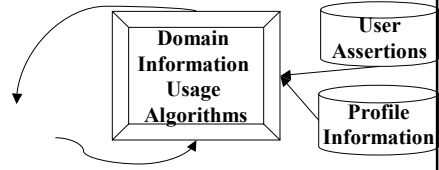
- Design of PATHS (Planning Assisted Tester for graphical user interface Systems)
- Implementation Status
- Overview of Planning & Test Case Generation

The GUI Testing Framework



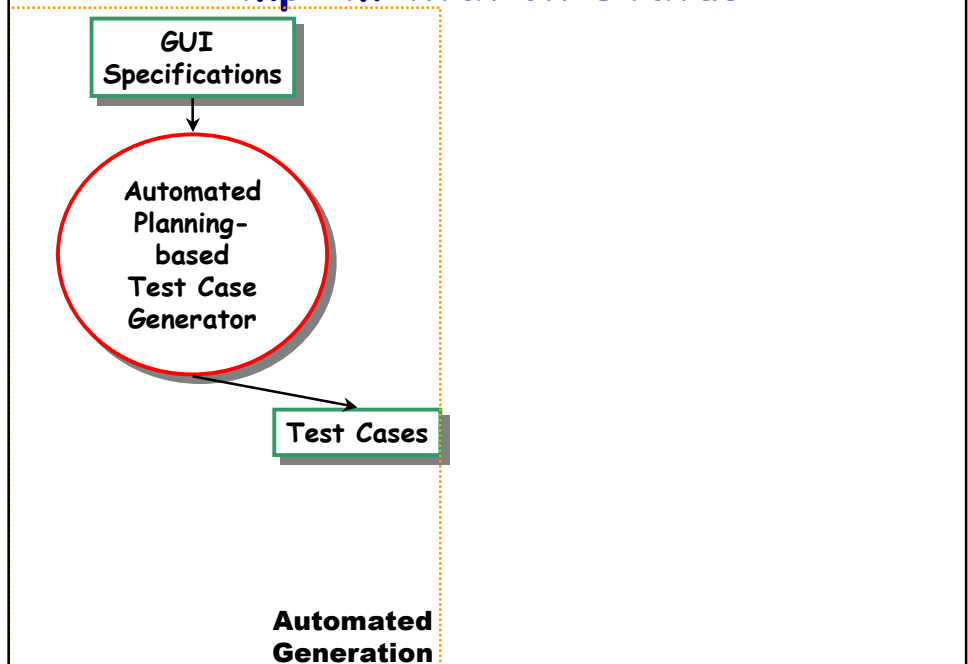


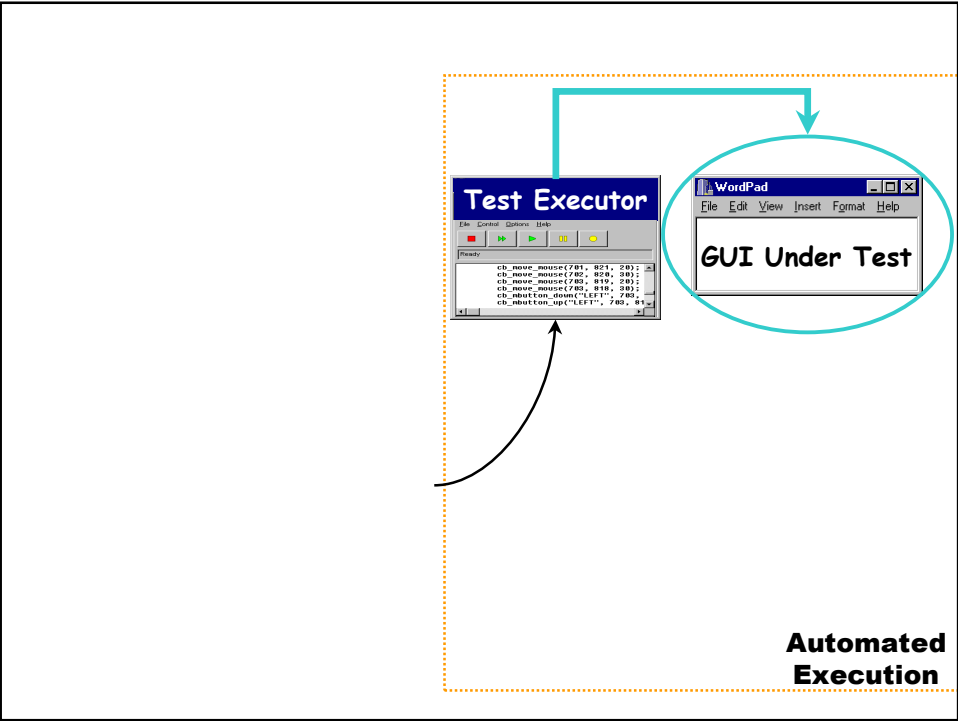
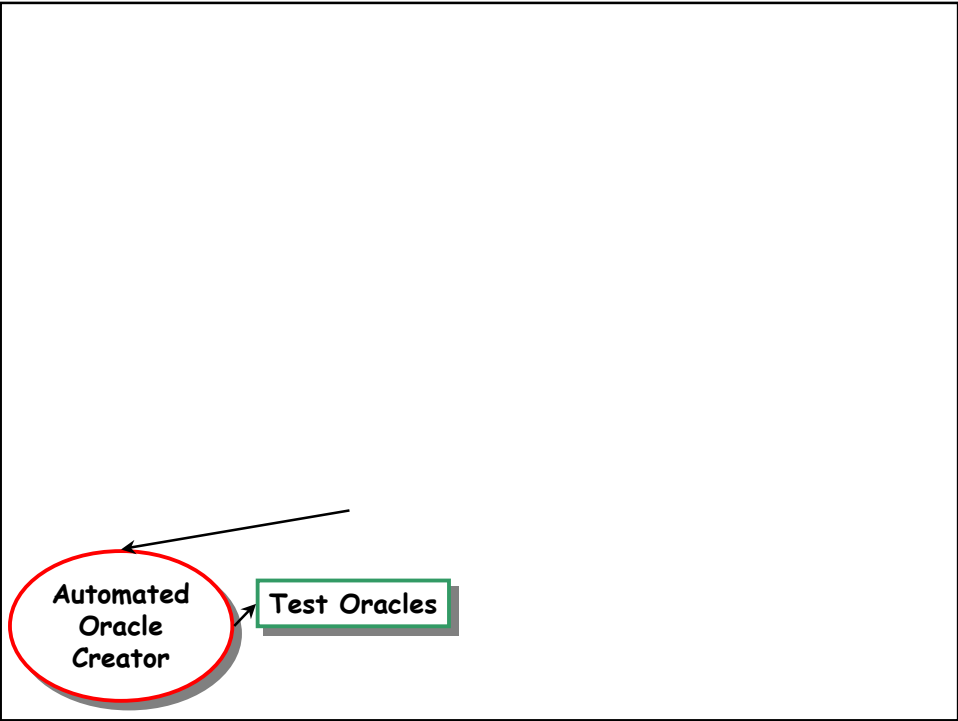
Entire

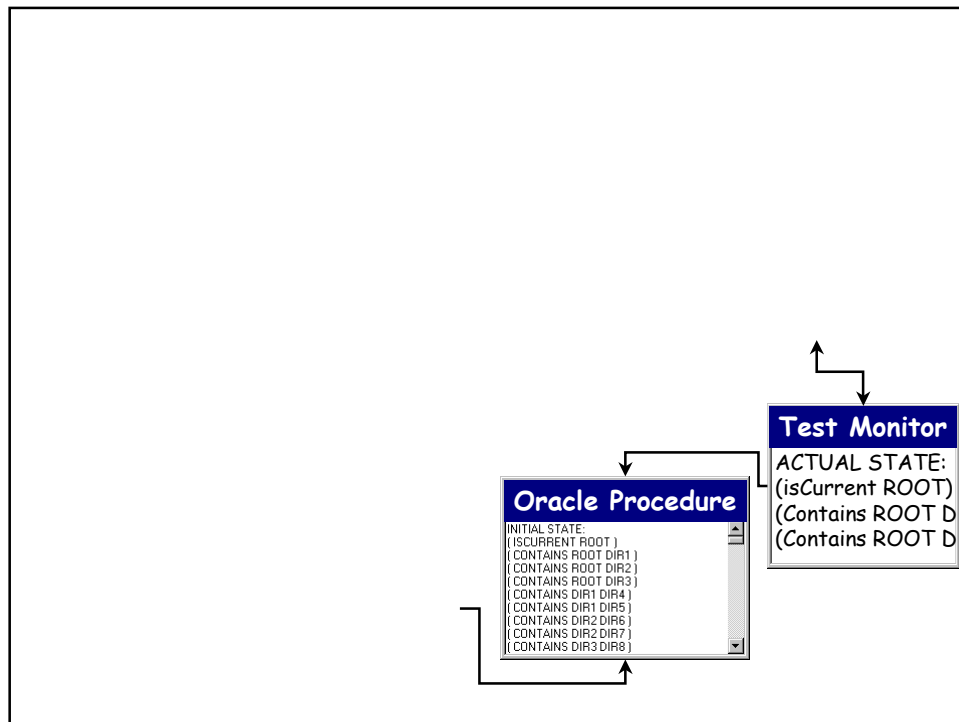


Implementation Status

8





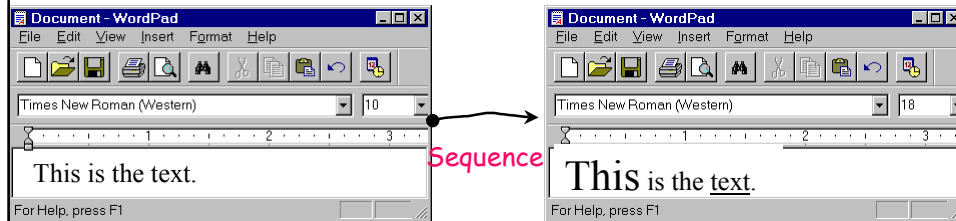


What is a GUI Test Case ?

12

- Individual User Events
 - NOT ENOUGH!
 - Sequences of User Events lead to Different States
- Test Case: Sequence of User Events

Generating Test Case Sequences



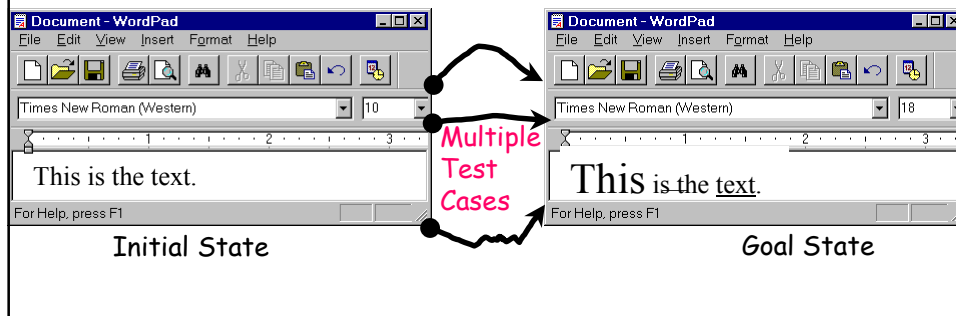
Each Sequence Transforms the GUI State

- Randomly Choose Sequences
- Expert Chooses Sequences
- Generate Sequences from a Description of the GUI



Novel Idea

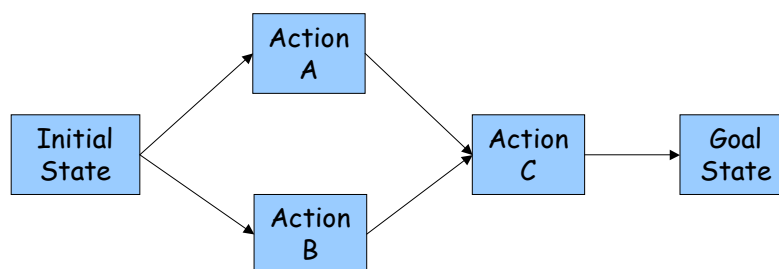
- Automatically Generate Test Cases for EXPECTED TASKS

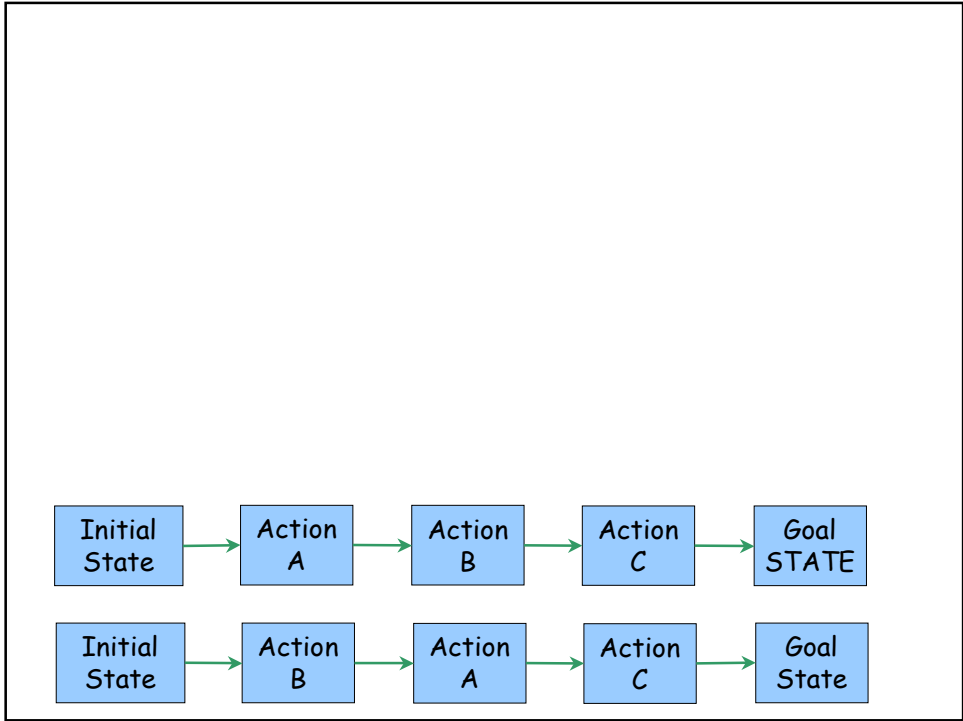


AI Planning and GUI Testing

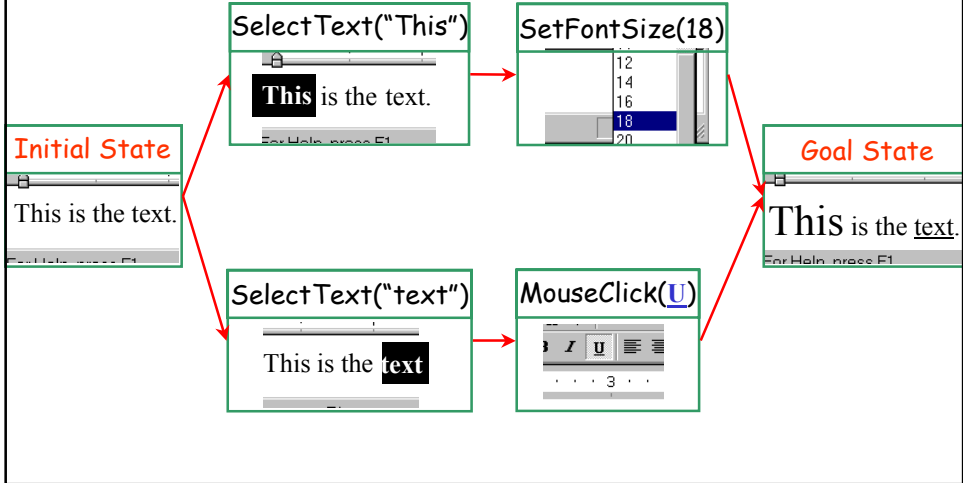
- Recent Advances in AI Planning
 - Propositional Planners
 - Very Fast
 - Based on
 - Flow-Graphs [Koehler et al. '97]
 - SAT Solving [Kautz & Selman '96]
- AI Planning, Especially Suitable for
 - Generating Test Cases
 - Automatically Generating Verification Information
 - Regression Testing

A Plan





A Plan for a GUI Task



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

Straightforward Approach

- Define One Operator for each User Action

Documents - 1000
File Edit View Ins

Menu1

Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Special...
Clear Del

Menu2

Operator :: *CUT*

Preconditions:

isCurrent(Menu2).

Effects:

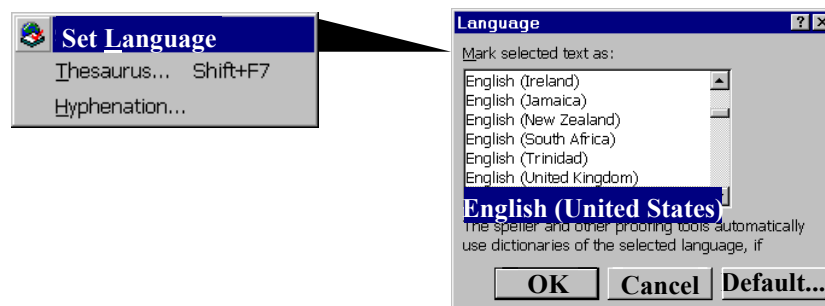
FORALL Obj in Objects
Selected(Obj)
ADD inClipboard(Obj)
DEL onScreen(Obj)
DEL Selected(Obj)
ADD isCurrent(Menu1)
DEL isCurrent(Menu2).

First Order Predicate Logic

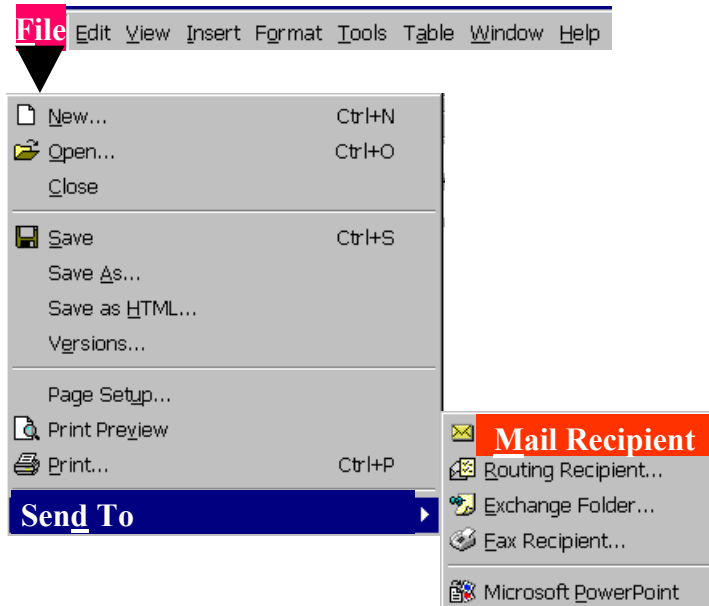
Exploit the GUI's Structure

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

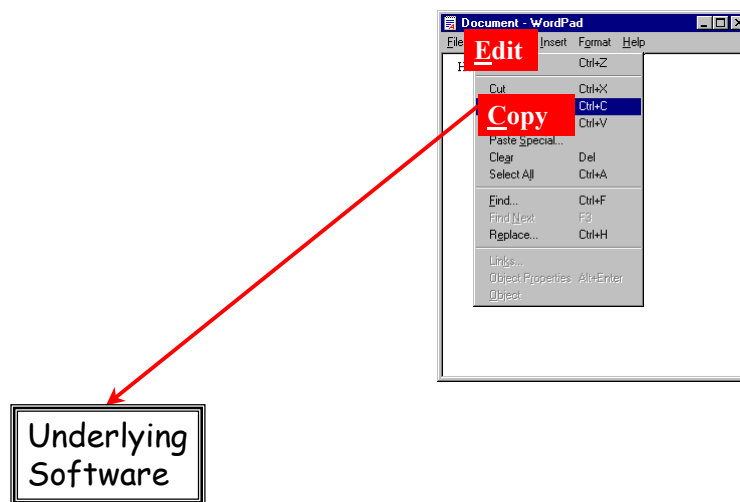
Opening Modal Windows



Opening Menus



Interacting with the Underlying Software



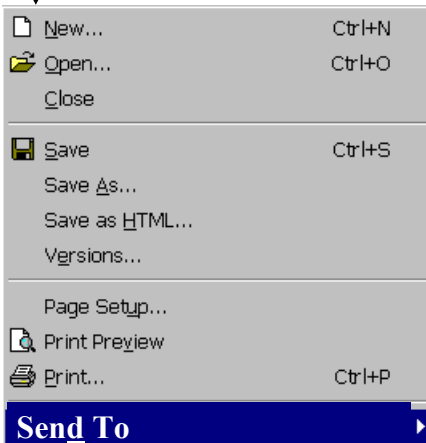
Create Hierarchical Operators

Two Types of Abstractions

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

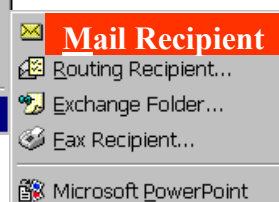
Create System-Interaction Operators

File Edit View Insert Format Tools Table Window Help

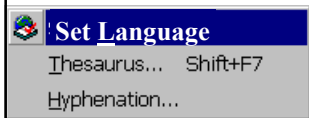


Sys-Interaction Operator:

File_SendTo_MailRecipient
= <File + SendTo + MailRecipient >



Create Abstract Operators



Straightforward Approach

Main GUI's Operator Set

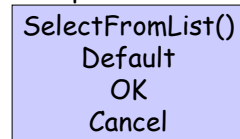


Using Abstraction

Main GUI's Operator Set

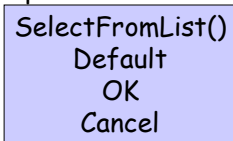


Language Window's Operator Set



Create Abstract Operators

Language Window's Operator Set



Define Abstraction

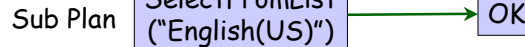


SetLanguage()

Abstract Operator



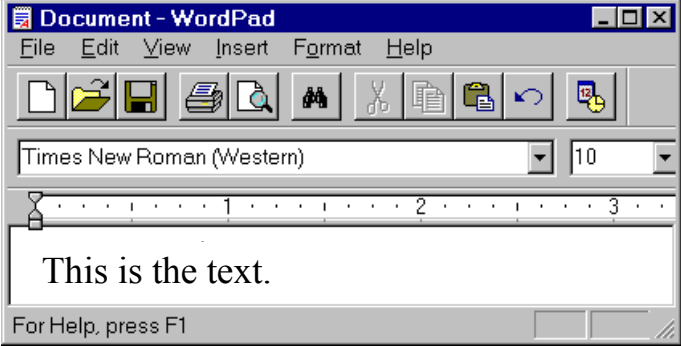
Planner



Effects of Exploiting the GUI's Structure

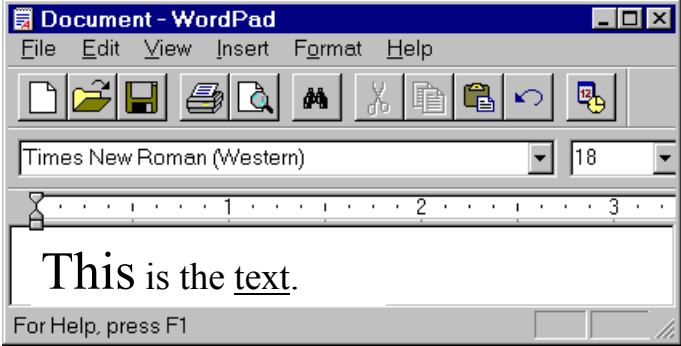
- Reduction in Planning Operators
 - 325 operators 32 operators
 - Ratio 10:1 for MS WordPad
 - 20:1 for MS Word
- System Automatically Determines the System-interaction and Abstract Operators

Initial State

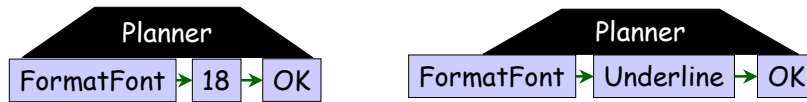
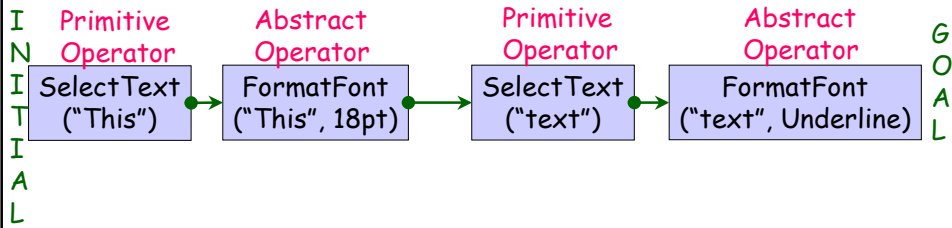


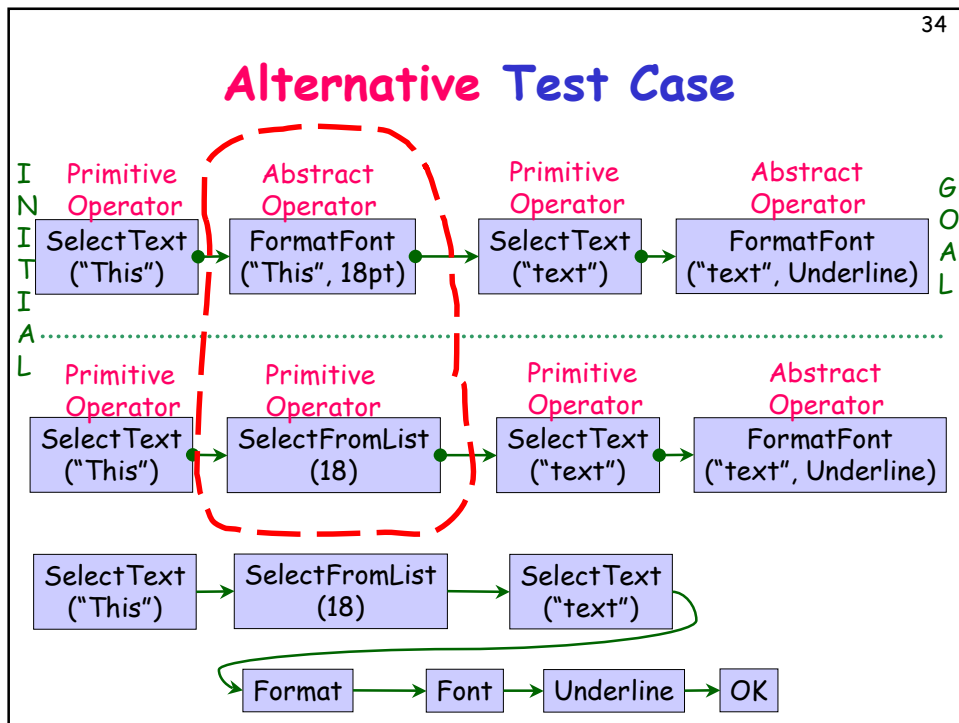
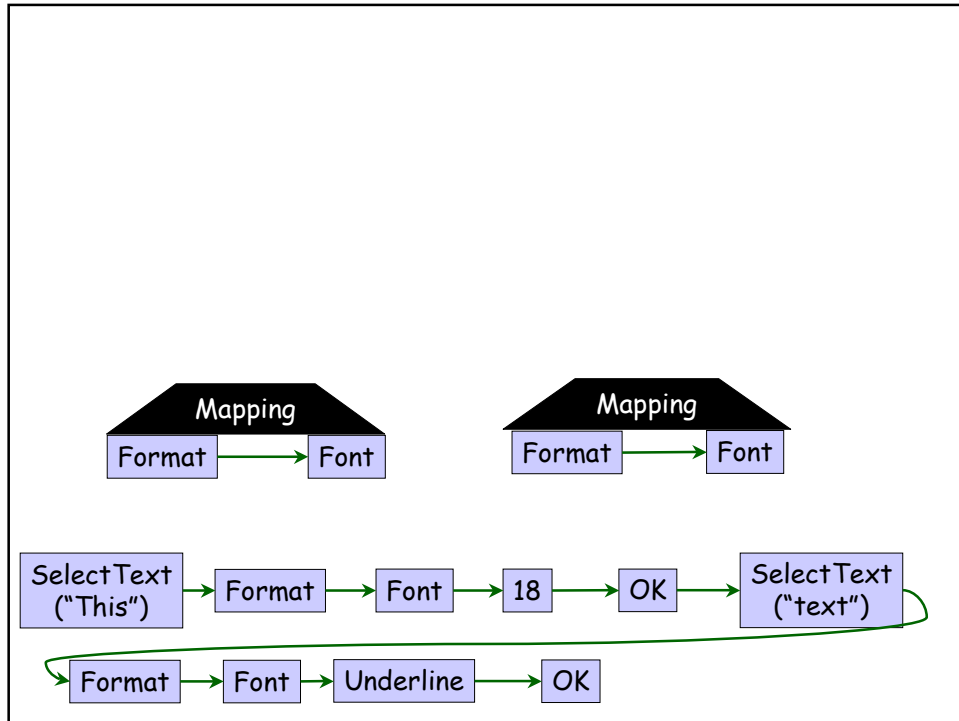
30

Goal State



Test Case

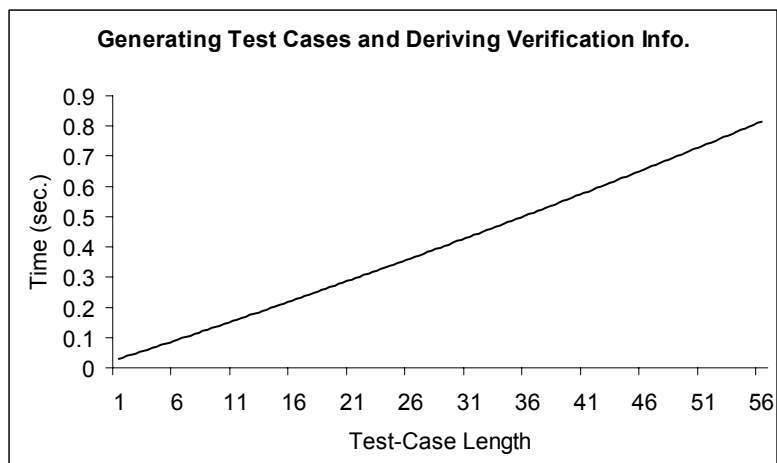




Methods to Generate **Alternative** Test Cases

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

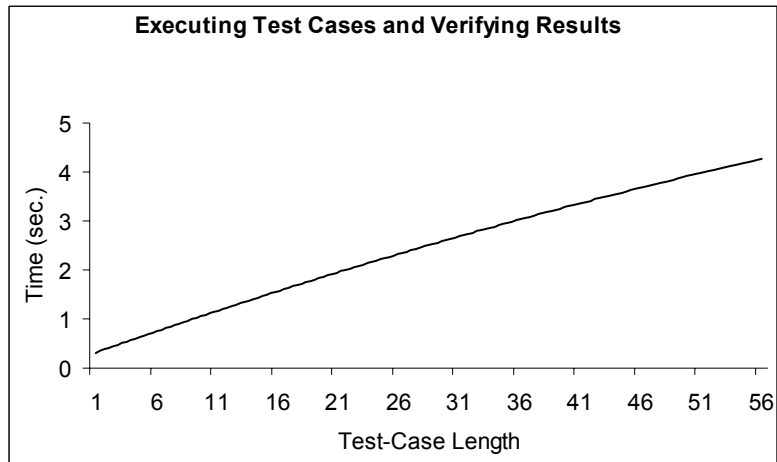
Experimental Results



(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

Task No.	Single Level		Hierarchical	
	Plan Length	Time (sec.)	Plan Length	Time (sec.)
1	18	8.93	3	0.11
2	20	47.62	4	0.18
3	24	189.87	5	0.14
4	26	3312.72	6	7.18
5	-	-	3	0.1
6	-	-	4	13.01

Experimental Results (contd...)



Test Oracles & Regression Testing

- Exploiting Planning Algorithms to get Expected Behavior of GUI
- Using Hierarchical GUI Model for Efficient Regression Testing

Concluding Remarks

- Overview of PATHS & Current Status
- Planning & Test Case Generation
- Automatic Generation of Preconditions and Effects from GUI Specifications