1

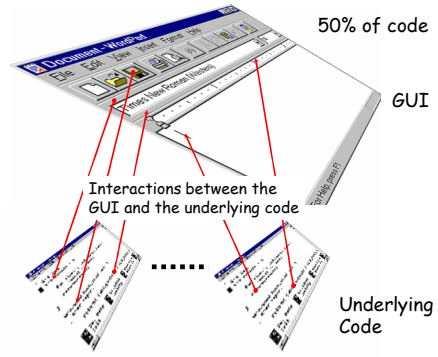
Coverage Criteria for GUI Testing

Atif M. Memon*, Mary Lou Soffa, Martha E. Pollack
atif@cs.umd.edu
Dept. of Computer Science
University of Pittsburgh

*now at the University of Maryland

2

Research focus



3

GUI Test Case

- Sequence of Events
  - [IEEE TSE Feb '01]
- Not just individual events

4

Coverage Criteria

- Two purposes
  - Test data selection criteria
    - Rules used to select test cases
  - Test data adequacy criteria
    - Rules used to determine how much testing has been done
- Common Examples for Conventional Software
  - Statement coverage
  - Branch coverage
  - Path coverage

} Structural Representation of the Code

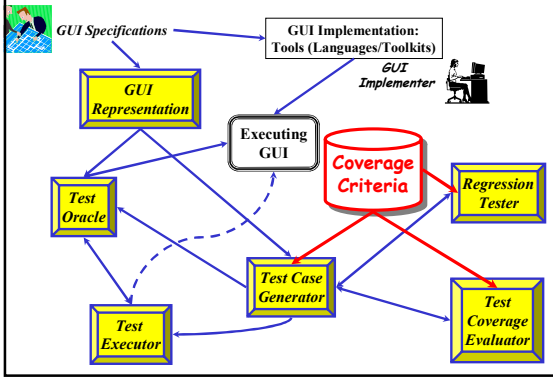
5

Coverage Criteria for GUIs

- Cannot use code-based coverage
  - Source code not always available
  - Event-based input
    - Different level of abstraction
- Our Contribution
  - Hierarchical structure of the GUI in terms of events
  - Coverage criteria based on events

6

Role of the Coverage Criteria



Outline

7

- GUI Definition
- Representation of GUIs
- Coverage Criteria
- Case Study
- Conclusions

GUI Definition

8

- Hierarchical
- Graphical Front-end
- Accepts User-generated and System-generated events
- Fixed sets of events
- Deterministic Output
- State of the GUI is the set of Objects and their Properties

GUI Representation

9

- Motivation
  - GUI testing needs a "Unit of Testing"
    - Manageable
    - Test the unit comprehensively
    - Test interactions among units
  - GUIs are created using library elements
    - Need to test these elements before packaging them for reuse
      - Certain level of confidence that the element has been adequately tested
    - User of these elements should be able to test the element in its context of use

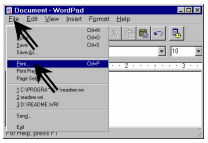
Model GUI Hierarchically

10

- Hierarchy
  - GUIs are decomposed into a hierarchy of components
  - Hierarchical decomposition makes testing intuitive and efficient
  - Several hierarchical views of GUIs
  - We examine Modal Dialogs to create the hierarchical model

Modal Windows in GUIs

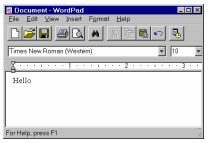
11

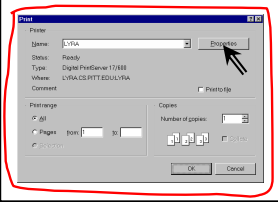


Main

Modal Windows in GUIs

12





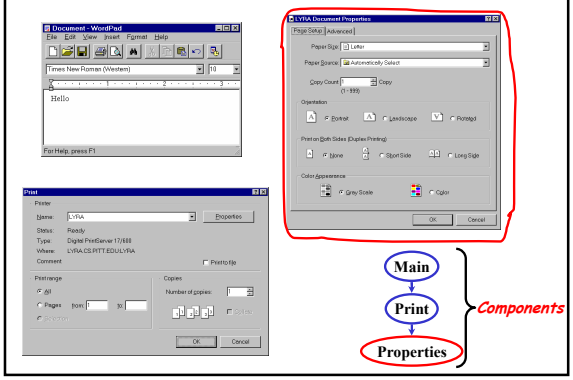
Main

invokes

Print

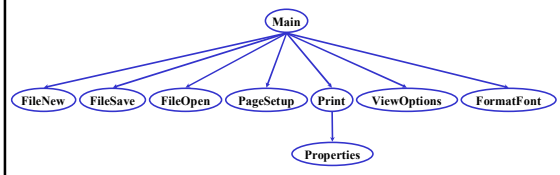
Modal Windows in GUIs

13



Integration Tree

14

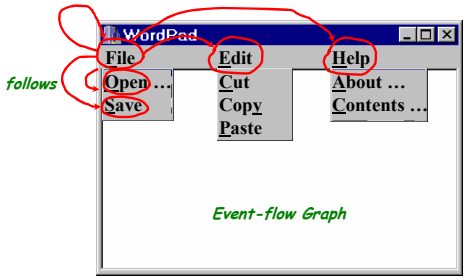


Definition: Integration tree is a triple $\langle N, R, B \rangle$

- N is the set of components in the GUI
- $R \in N$ is a designated component called the ***Main*** component
- B is the set of directed edges showing the invokes relation between components, i.e., $(C_x, C_y) \in B$ iff C_x invokes C_y.

Representing a Component

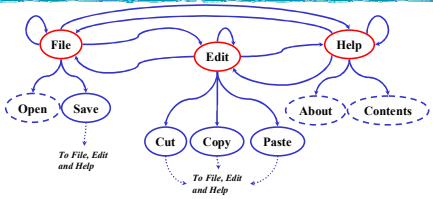
15



Definition: Event e_x follows e_y iff e_x can be performed immediately after e_y.

Event-flow Graph

16



Definition: Event-flow graph is a 4-tuple $\langle V, E, B, I \rangle$

- V is the set of vertices, representing events,
- E is the set of directed edges, showing the follows relationship,
- B is the set of events first available (shown in red),
- I is the set of events that invoke other components (dotted lines).

Classifying Events

17

- Classification
  - A new classification of events aids in creating the hierarchical model of the GUI
    - Opening modal windows
      - Restricted-focus events
    - Closing modal windows
      - Termination events
    - Opening modeless windows
      - Unrestricted-focus events
    - Opening menus
      - Menu-open events
    - Interacting with underlying software
      - System-interaction events

Coverage Criteria

18

- Intuitively
  - Each component is a unit of testing
  - Test events within each component
    - Intra-component coverage criteria
  - Test events across components
    - Inter-component coverage criteria

Coverage Criteria

19

- Intra-component Coverage
  - Event coverage
    - Individual events
    - Each node in the event-flow graph
  - Event-interaction coverage
    - Each pair of events
    - Each edge in the event-flow graph
  - Length-n event sequence coverage
    - Sequences of events
    - Bounded by length
      - Length-1 event sequences
      - Length-2, length-6 event sequences
    - Paths in the event-flow graph

Coverage Criteria

20

- Inter-component Coverage
  - Invocation coverage
    - Invoke each component
    - Each restricted-focus event
  - Invocation-termination coverage
    - Invoke each component and terminate it
    - Restricted-focus event followed by a termination event
  - Inter-component length-n coverage
    - Longer sequences from one component to another
    - Bounded by length

Case Study

21

- Purpose
  - To determine:
    - How many test cases do we need to test WordPad
    - Correlation between event and code-based coverage
    - How well did our planning-based approach [ICSE '99] do
- Experimental design
  - GUI: our version of MS WordPad (36 modal windows, 362 events)
  - Hardware platform: 350 MHz Pentium based machine, 256 MB RAM

Test Cases for WordPad

22

	Event-sequence Length							
Component Name	**1'**	**2'**	**1**	**2**	**3**	**4**	**5**	**6**
Main			56	791	14354	255720	4490626	78385288
FileOpen			10	80	640	5120	40960	327680
FileSave			10	80	640	5120	40960	327680
Print			12	108	972	8748	78732	708588
Properties			13	143	1573	17303	190333	2093663
PageSetup			11	88	704	5632	45056	360448
FormatFont			9	63	441	3087	21609	151263
Print+Properties	1	2		13	260	3913	52520	663013
Main+FileOpen	1	2		10	100	1180	17160	278760
Main+FileSave	1	2		10	100	1180	17160	278760
Main+PageSetup	1	2		11	110	1298	18876	306636
Main+FormatFont	1	2		9	81	909	13311	220509
Main+Print+Properties				12	145	1930	28987	466578

Results

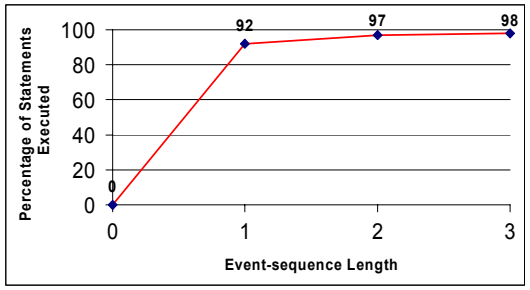
Correlation between Event-based & Code-based Coverage

23

- Code Instrumentation
- Generated all event sequences up to length 3. Total test cases: 21,659
- Executed all 21,659 cases and obtained execution traces
- Statement coverage

Correlation between Event-based & Code-based Coverage

24



Results

Evaluating the Planning Approach

25

- Used our earlier-developed planning-based approach
- 500 test cases of different lengths

Evaluating the Planning Approach

26

	Event-sequence Length							
Component Name	**1'**	**2'**	**1**	**2**	**3**	**4**	**5**	**6**
Main			88	41	10.92	0.36	0.03	0.00
FileOpen			90	56	17.50	0.72	0.06	0.05
FileSave			90	41	20.63	1.27	0.47	0.02
Print			92	34	32.20	9.00	3.92	0.19
Properties			92	45	27.59	1.80	0.97	0.06
PageSetup			91	49	25.43	2.56	0.66	0.06
FormatFont			89	37	39.00	13.67	0.66	0.06
Print+Properties	100	0		46	51.15	8.18	3.87	0.05
Main+FileOpen	100	0		40	11.00	10.17	1.30	0.16
Main+FileSave	100	0		20	13.00	8.64	1.26	0.28
Main+PageSetup	100	0		45	60.91	4.31	1.94	0.08
Main+FormatFont	100	0		33	28.40	5.17	0.97	0.10
Main+Print+Properties				50	38.62	6.37	0.65	0.09

Results

Future Work

27

- GUI's Structure and its Testability
- Apply Criteria to
  - Object-oriented Software
  - Component-based Software
  - Reactive Software