

Determining the Adequacy of GUI Test Cases

Atif M. Memon (atif@cs.umd.edu) (<http://www.cs.umd.edu/~atif>)



Department of
Computer Science

Coverage Criteria

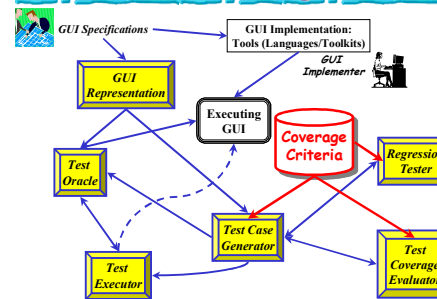
- Two purposes
 - Test data selection criteria
 - Rules used to select test cases
 - Test data adequacy criteria
 - Rules used to determine how much testing has been done
- Common Examples for Conventional Software
 - Statement coverage
 - Branch coverage
 - Path coverage

Structural
Representation
of the Code

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

Role of the Coverage Criteria



GUI Representation

- 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

- 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

Coverage Criteria

- 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

- 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

- 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

Test Cases for WordPad

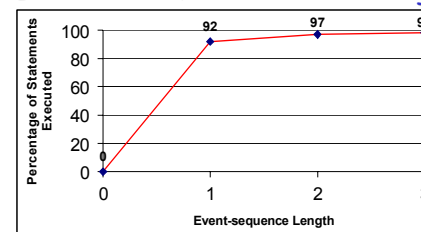
Component Name	Event-sequence Length					
	1	2	3	4	5	6
Main	56	791	14354	255720	4490626	78385288
FileOpen	10	80	640	5120	40960	327680
FileSave	10	80	640	5120	40960	327680
Print	12	108	972	8748	78732	708588
Properties	13	143	1573	17303	190333	2093603
PageSetup	11	88	704	5632	45056	360448
FormatFont	9	63	441	3087	21609	151263
Print+Properties	1	2	13	260	3913	52520
Main+FileOpen	1	2	10	100	1180	17160
Main+FileSave	1	2	10	100	1180	17160
Main+PageSetup	1	2	11	110	1298	18876
Main+FormatFont	1	2	9	81	909	13311
Main+Print+Properties			12	145	1930	28987

Results

Correlation between Event-based & Code-based Coverage

- 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



Results

Further Reading: Publications

- Atif M. Memon, Mary Lou Soffa and Martha E. Pollack, Coverage Criteria for GUI Testing, 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT International Symposium on the Foundations of Software Engineering (FSE-9), Vienna University of Technology, Austria, Sept. 10-14, 2001.
- Atif M. Memon, A Comprehensive Framework for Testing Graphical User Interfaces, PhD Dissertation, University of Pittsburgh, Pittsburgh, PA, July 2001.