A Comfortable TestPlayer for Analyzing Statistical Usage Testing Strategies

Winfried Dulz

(Department of Computer Science 7,
University of Erlangen-Nuremberg, Germany)
Outline

- **Motivation**
  - Short History of Statistical Usage Testing

- **Tool Environment**
  - yED
  - R/Eclipse StatET

- **TestPlayer User Interface**
  - Generation of Test Cases
  - Visualization of Test Cases

- **Evaluation of Test Suites**
  - Intra Test Suite Analysis
  - Inter Test Suite Analysis

- **Final Remarks**
Short History of Statistical Usage Testing


Reviewer Leo G. Egan, Jr.: *The authors introduce a new method to analyze software specifications before design and coding begin. They assume all programmers and systems analysts are conversant with and comfortable with Markov techniques and higher mathematical analyses in general, which traditionally is not the case. I would not recommend this paper to all programmers and analysts. I would make it known and available to systems engineers who have the responsibility of designing systems that are hardware-intensive and especially software-intensive.*
Short History of Statistical Usage Testing (cont.)


  ... result of years of work by many different people, and very few results are original. James Whittaker, Michael Thomason, and Jesse Poore did the original work on Markov chain usage models. Gwen Walton’s research applied mathematical programming techniques to set model probabilities under testing constraints. Jenny Morales and Dave Pearson investigated combining information across tests to improve reliability measurements. Kirk Sayre’s research provided many new and useful analytical results, and provided a framework for simulation and partition testing. Walter Gutjahr demonstrated how a Markov chain could be modified to bias test generation toward low-use critical function, and how the bias could be removed in the results.
Motivation

Short History of Statistical Usage Testing (cont.)

  
  ... The J Usage Model Builder Library (JUMBL) is a Java class library and set of command-line tools for working with usage models ...

  
  ... The main focus is on automatically generating a MCUM (Markov Chain Usage Model) starting from an FDT (Formal Description Technique) in order to derive TTCN-3 (ETSI Testing and Test Control Notation version 3) compatible test case definitions ...
Tool Environment

yED: MCUM editing and visualization
Tool Environment

R/Eclipse StatET

- Open source scripting language for statistical computing and graphics
- Package pool to extend the basic functionality, e.g. for GUI programming
TestPlayer User Interface

TestPlayer GUI (Gtk+)

- **Toolbar**
  - <quit>
  - <ok>

- **MCUM definition**
  - <Model name>
  - <Number of test cases>
  - <Start state>
  - <End state>
  - <Profile usage>
  - <Profile name>
TestPlayer User Interface

TestPlayer GUI (cont.)

- MCUM extensions
  - <Generation.of.graph.elements>
    - <node names>
    - <event names>
    - <probabilities>
  - <Closed.subgraph.elements> in hierarchical models
Running example


Fig 5. Structural phase—Constructing the usage Markov chain
TestPlayer User Interface

Uncomplete MCUM

Invocation -> Window

Window -> Move

Move -> DragMouse

DragMouse -> Size

Size -> Close

Window -> Maximize

Maximize -> Minimize

Minimize -> Icon

Icon -> Restore

Restore -> Up

Up -> Down

Down -> Left

Left -> Right

Right -> Termination
TestPlayer User Interface

Extended MCUM

Invocation → Window → Move → DragMouse → Size → Close

Maximize → Minimize → Icon → Restore

Up → Down → Left → Right

Termination
TestPlayer User Interface

Completed MCUM

Invocation → Window → Move → DragMouse → Size → Close

Maximize → Minimize → Icon → Restore

Up → Down → Left → Right

Termination

Winfried Dulz
FAU University Erlangen-Nuremberg
Open hierarchical MCUM
Closed hierarchical MCUM
Pruned hierarchical MCUM
TestPlayer User Interface

TestPlayer GUI (cont.)

- Test case/suite generation
  - <Generation.strategy>
  - <Coverage.strategy>
    - <nodes>
    - <transitions>
  - <Sorting.strategy>
  - <Start.number> and <End.number> of generated test cases
Test case generation

- <Sorting.strategy>
  - <unsorted>
  - <frequency>
  - <length>
  - <p.mul>
  - <p.add>
  - <combination>

- Test case list
  - <xyz>_<aaa>.tcl
    - <xyz>: model
    - <aaa>: number of sorted test cases
TestPlayer User Interface

Test case generation

- Test case file: ...
  - Invocation: ...
  - Number of relevant nodes (non-zero frequency): 15
- Test case file: ...
  - Invocation: ...
  - Number of relevant transitions (non-zero frequency): 22
- Source entropy of the profile: 0.1846
- Total number of generated test cases for the test suite: 263
- Number of unique test cases in the test suite: 100
- Mean length of a test case for the profile: 4.8667
- Mean length of a test case in the test suite: 10.1086
- Kullback-Leibler divergence between MCMC and test suite: 0.0011
- Mean weighted transition deviation between MCMC and test suite: 0.1706
- Number of test cases needed to cover all nodes: 3
- Number of test cases needed to cover all transitions: 124
Test case visualization

- Invocation
  - Window
  - Move
  - Size
- DragMouse
  - Up
  - Down
  - Left
  - Right
- Maximize
- Minimize
- Icon
- Restore
- Close

 Edges:
- e1 \rightarrow 1
- e2 \rightarrow 1
- e3 \rightarrow 1
- e4
- e5 \rightarrow 1
- e6
- e7
- e8
- e9
- e10
- e11
- e12
- e13 \rightarrow 1
- e14 \rightarrow 1
- e15 \rightarrow 1
- e16 \rightarrow 1
- e17
- e18
- e19 \rightarrow 1
- e20 \rightarrow 1
- e21
- e22
Test case visualization
TestPlayer User Interface

Test case visualization

Invocation → Window → Move → DragMouse → Size → Close

Invocation → Minimize × 2

Invocation → Maximize × 2

Minimize → Icon → Restore

Window

DragMouse

Move

Size

Close

Up

Down

Left

Right

Termination
TestPlayer User Interface

TestPlayer GUI (cont.)

- Test suite evaluation
  - <Single.metrics>
    - <SSP>
    - <SSV>
    - <KL>
    - <SSP.N>, <SSV.N>, <KL.N>
    - <SSP.T>, <SSV.T>, <KL.T>
  - <MCUM statistics>
  - <Compare.metrics>

Winfried Dulz
FAU University Erlangen-Nuremberg
Evaluation of Test Suites

- **SSP**
  - steady state probabilities of the MCUM vs. state frequencies of the test suite

- **SSV**
  - mean number of test cases to visit a state once in the MCUM vs. mean number of test cases to visit a state once in the test suite
Evaluation of Test Suites

MCUM steady state probabilities

Test suite state frequencies
Evaluation of Test Suites

Mean number of test cases to visit a state once (MCUM)

Mean number of test cases to visit a state once (test suite)
Kullback/Leibler divergence (KL) between MCUM and test suite vs mean weighted transition deviation (MWT) between MCUM and test suite

\[
KL(U, T) = \sum_i \pi_i \sum_j u_{i,j} \cdot \log \left( \frac{u_{i,j}}{t_{i,j}} \right)
\]

\[
MWT(U, T) = \frac{\sum_i \sum_j \| u_{i,j} - t_{i,j} \|}{\max(u_{i,j}) - \min(u_{i,j})}
\]

\[
K(U, T) = \sum_i \pi_i \sum_j u_{i,j} \cdot \log \left( \frac{u_{i,j}}{\epsilon - \epsilon \cdot (\text{sgn}(t_{i,j})) + t_{i,j}} \right)
\]
Evaluation of Test Suites

Kullback/Leibler divergence between MCUM and test suite

<Sorting.strategy>=<unsorted>

Mean weighted transition deviation between MCUM and test suite
Evaluation of Test Suites

Kullback/Leibler divergence between MCUM and test suite

\[ <\text{Sorting.strategy}> = \text{<unsorted>} \]

\[ <\text{Zoom factor}> = \text{<5>} \]

Mean weighted transition deviation between MCUM and test suite

Winfried Dulz
FAU University Erlangen-Nuremberg

29
Evaluation of Test Suites

Kullback/Leibler divergence between MCUM and test suite

\[ \text{KL Divergence} \]

Test case count

\(<\text{Sorting.strategy}> = <\text{length}>\)

<Zoom factor> = <5>

Mean weighted transition deviation between MCUM and test suite

\[ \text{Mean Weighted Deviation} \]

Test case count
Evaluation of Test Suites

Kullback/Leibler divergence between MCUM and test suite

<Sorting.strategy>=<unsorted>

<Number of test cases>=<100>

Mean weighted transition deviation between MCUM and test suite
Evaluation of Test Suites

Kullback/Leibler divergence between MCUM and test suite

\[ \text{Sorting.strategy} = \text{p.mul} \]

\[ \text{Number of test cases} = 100 \]

Mean weighted transition deviation between MCUM and test suite
Evaluation of Test Suites

Kullback/Leibler divergence between MCUM and test suite

<Sorting.strategy>=< unsorted>< p.mul >

<Number of test cases>=<100>

Mean weighted transition deviation between MCUM and test suite
Final Remarks

- Other models
Other models
Final Remarks

- Other models
Final Remarks

- Other models
Final Remarks

- Other models
Final Remarks

- Other models
Other models
Final Remarks

- Thank you for the attention
- Further information: www.testus.eu

Questions ?