A Comfortable TestPlayer for Analyzing Statistical Usage Testing Strategies

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

Friedrich-Alexander-Universität
Erlangen-Nürnberg
TECHNISCHE FAKULTÄT

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

- J. A. Whittaker, J. H. Poore. **Markov Analysis of Software Specications**. ACM Transactions on Software Engineering and Methodology, 1993.
  - 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.)

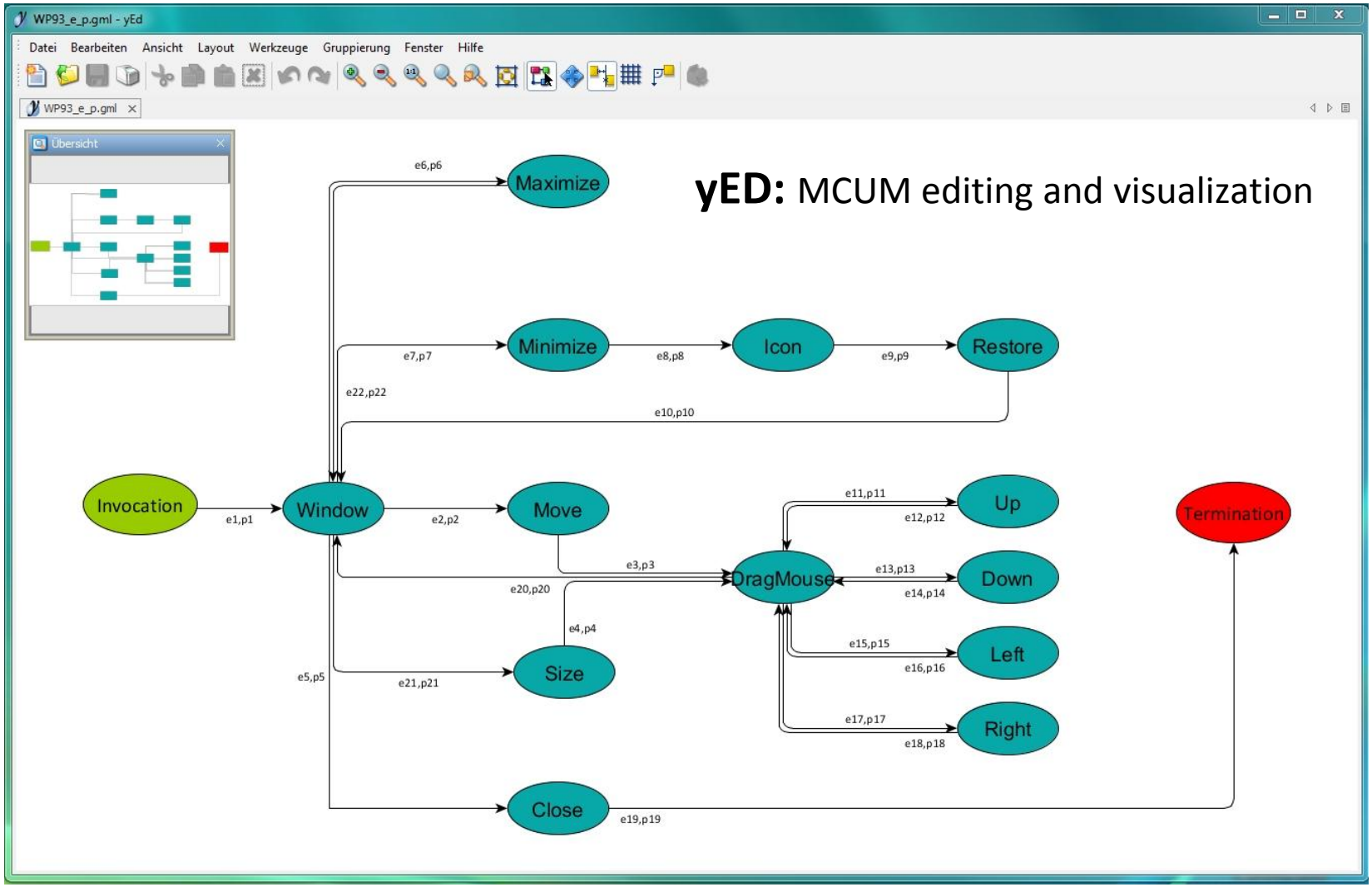
- Stacy J. Prowell. **Computations for Markov Chain Usage Models**. Technical report, Software Engineering Institute, Carnegie-Mellon University, 2000.
  - *… result of years of work by many different people, and very few results are original. **James Whittaker**, **Michael Thomason**, and **Jesse Poore** 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 .*

Short History of Statistical Usage Testing (cont.)

- Stacy J. Prowell. **Jumbl: A tool for model-based statistical testing**. HICSS, 2003.
  - *… The J Usage Model Builder Library (JUMBL) is a Java class library and set of command-line tools for working with usage models …*

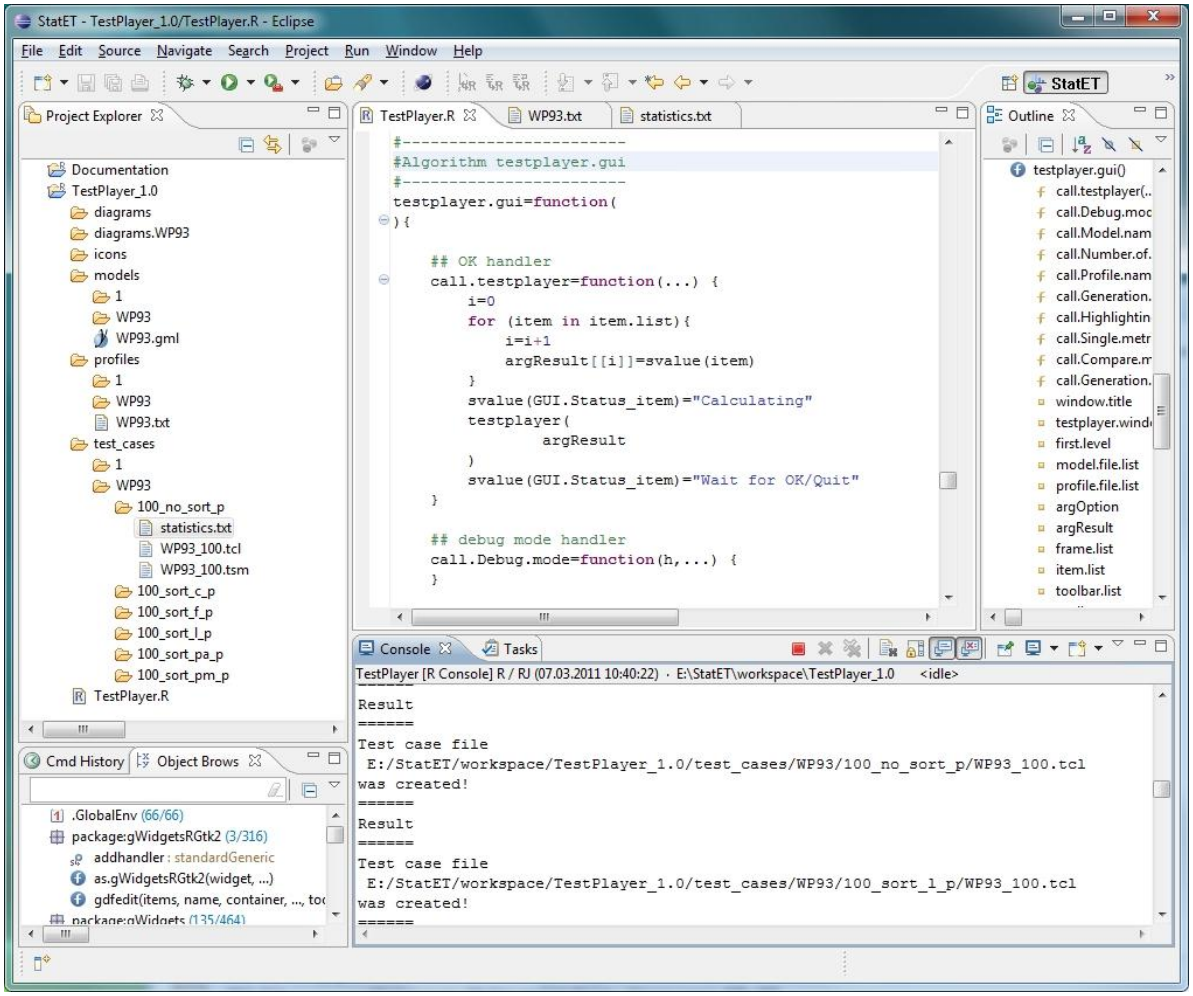
- W. Dulz and F. Zhen**. MaTeLo – Statistical Usage Testing by Annotated Sequence Diagrams, Markov Chains and TTCN-3**. IEEE QSIC, 2003.
  - *… The main focus is on automatically generating a MCUM (Markov Chain Usage Model) starting from an FDT (Formal Description Technique) in order to derive TTCN-3 (ETSI Testing and Test Control Notation version 3) compatible test case definitions …*

Tool Environment



R/Eclipse StatET

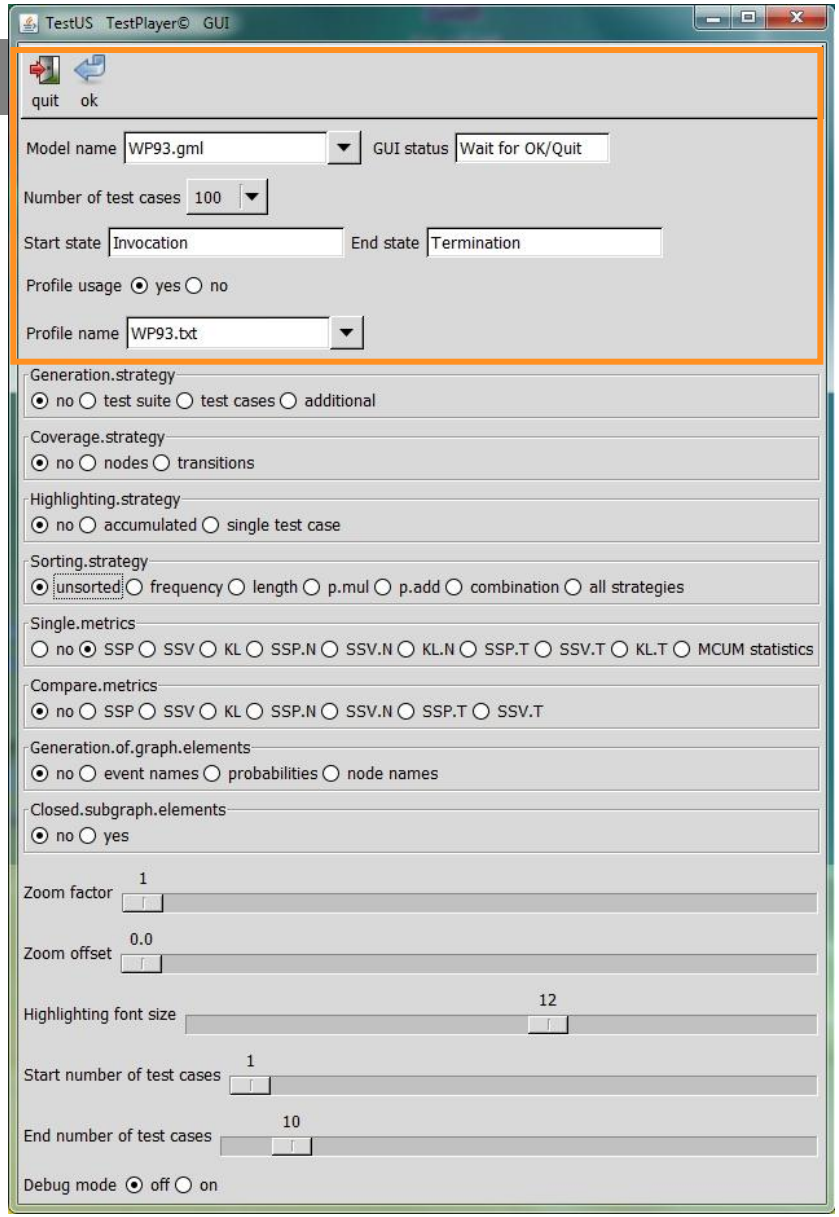
- Open source scripting language for statistical computing and graphics
- Package pool to extend the basic functionality, e.g. for GUI programming



TestPlayer User Interface

TestPlayer GUI (Gtk+)

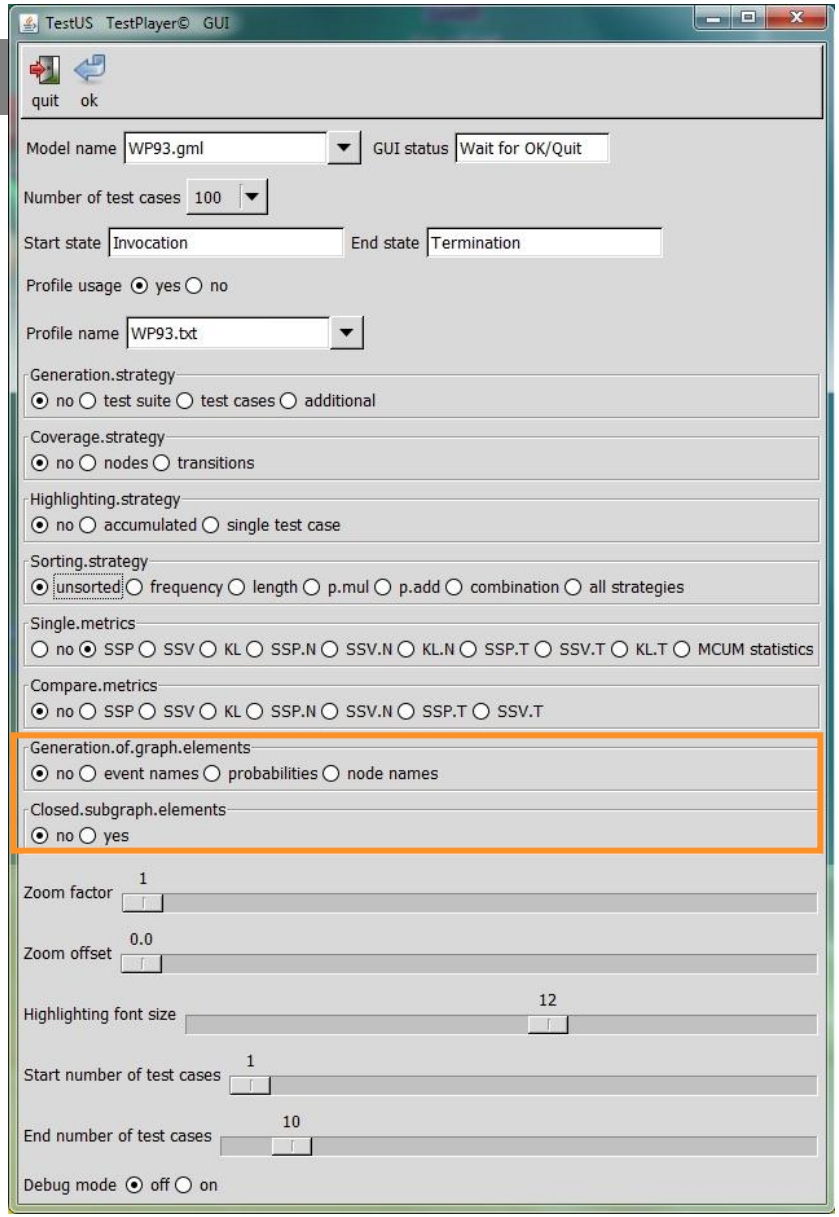
- Toolbar
  - <quit>
  - <ok>
- MCUM definition
  - <Model name>
  - <Number of test cases>
  - <Start state>
  - <End state>
  - <Profile usage>
  - <Profile name>



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

- J. A. Whittaker, J. H. Poore. **Markov Analysis of Software Specications**. ACM Transactions on Software Engineering and Methodology, 1993.

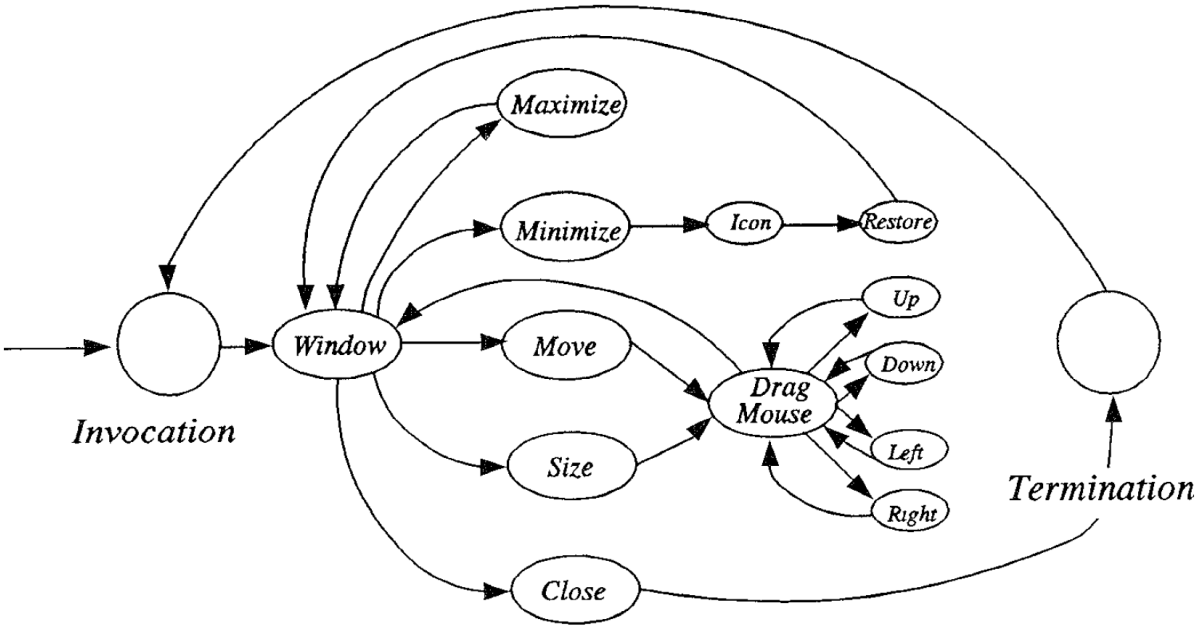
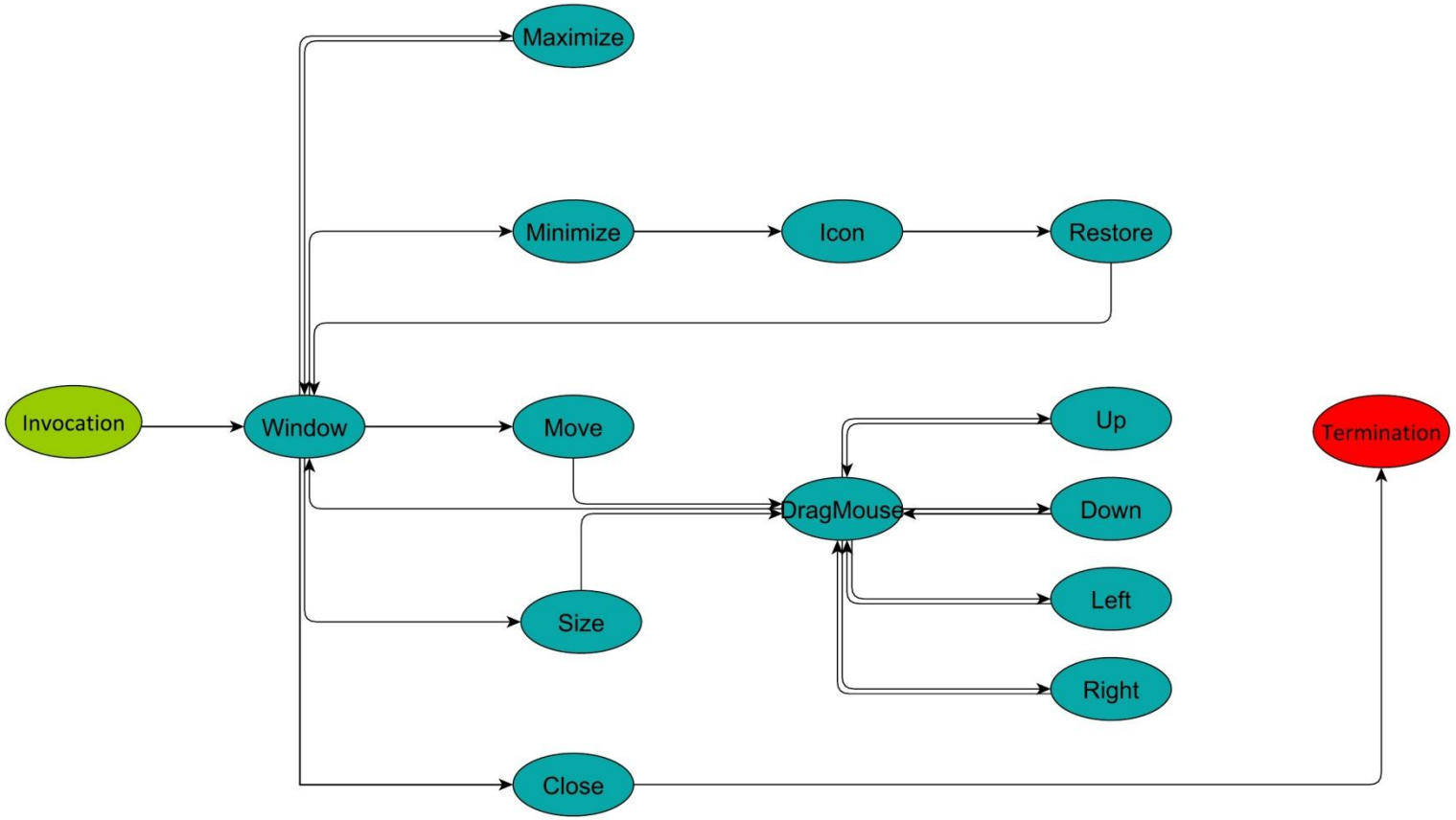
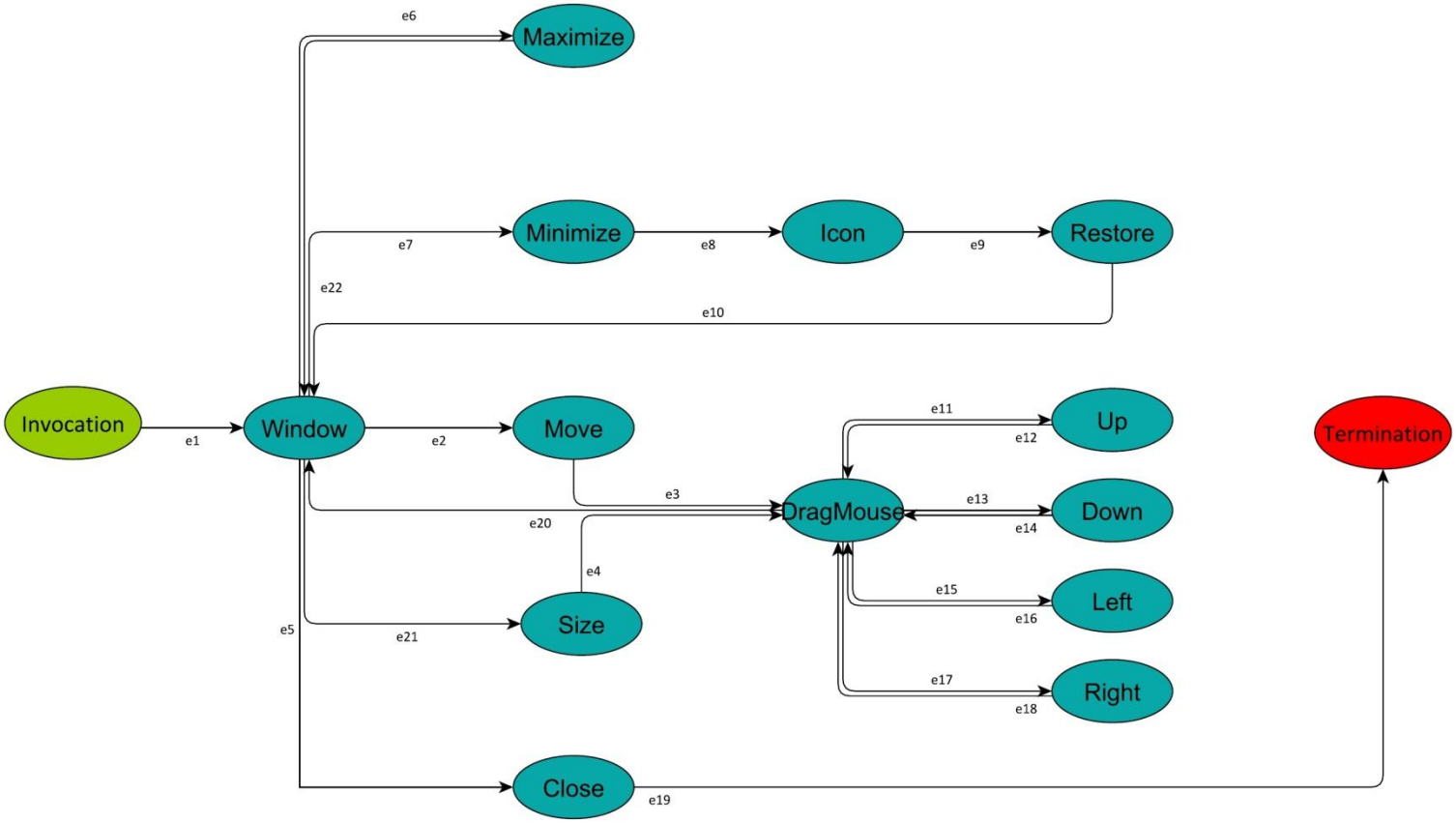


Fig 5. Structural phase—Constructing the usage Markov chain

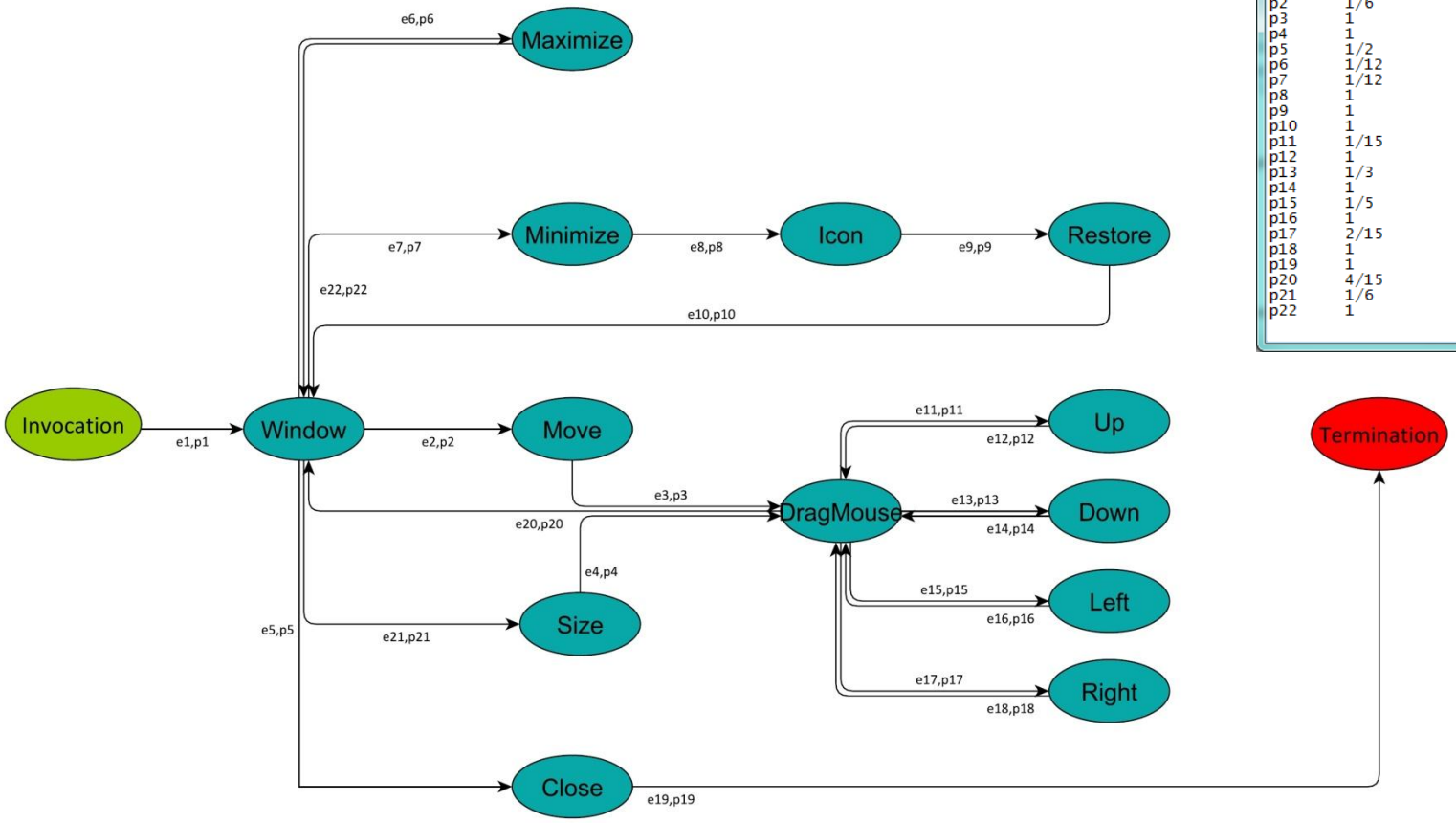
Uncomplete MCUM



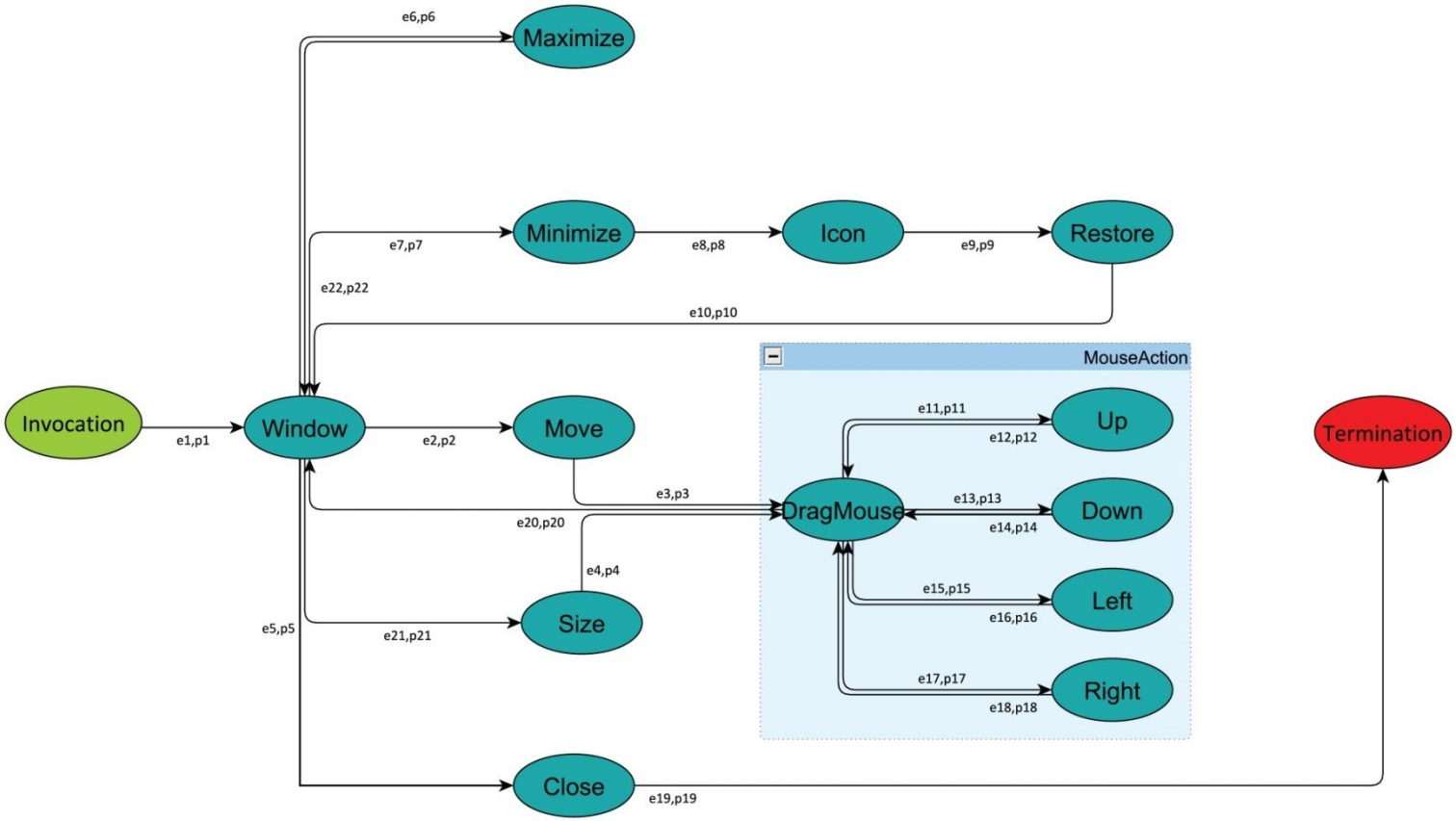
Extended MCUM



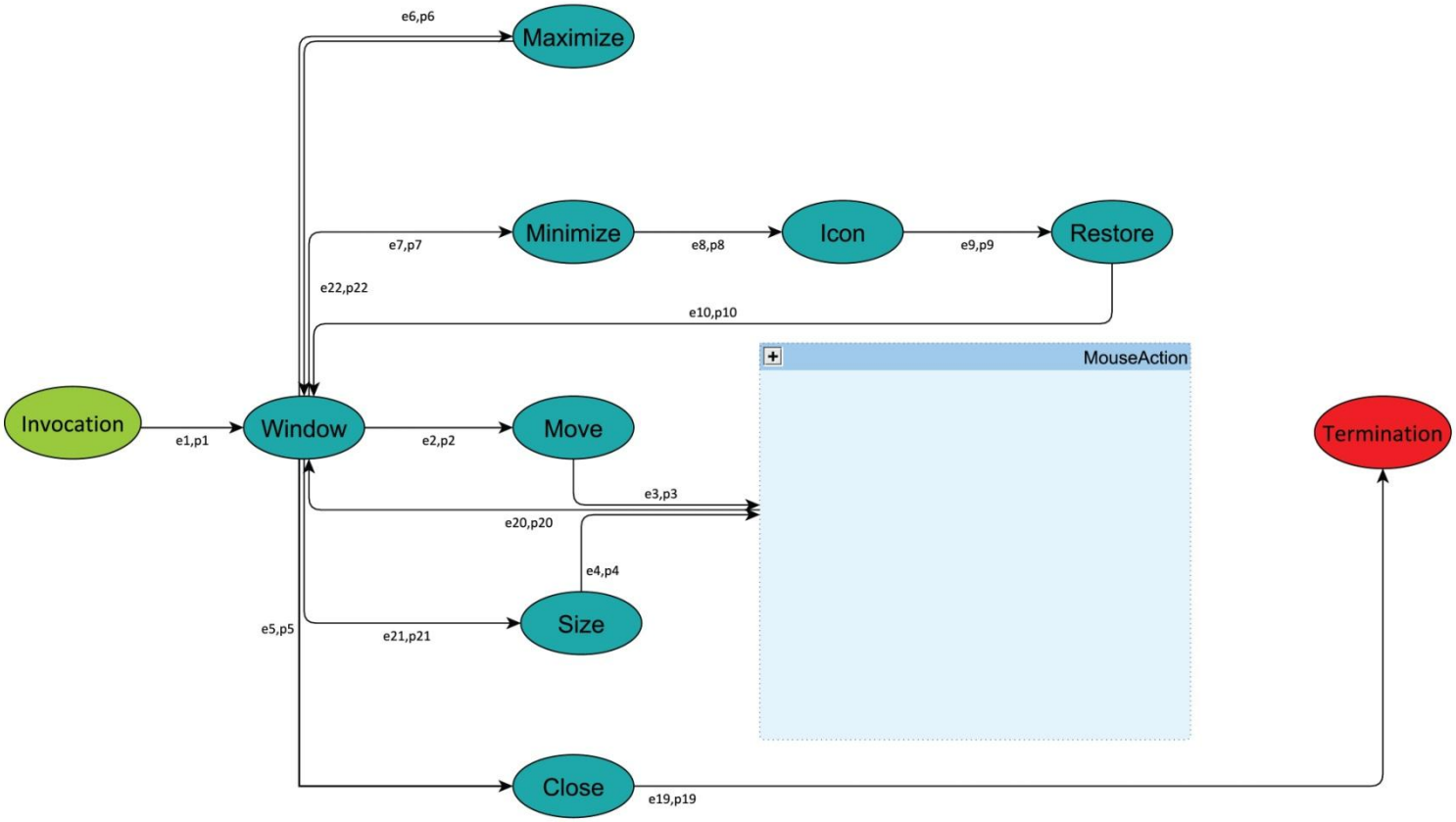
Completed MCUM



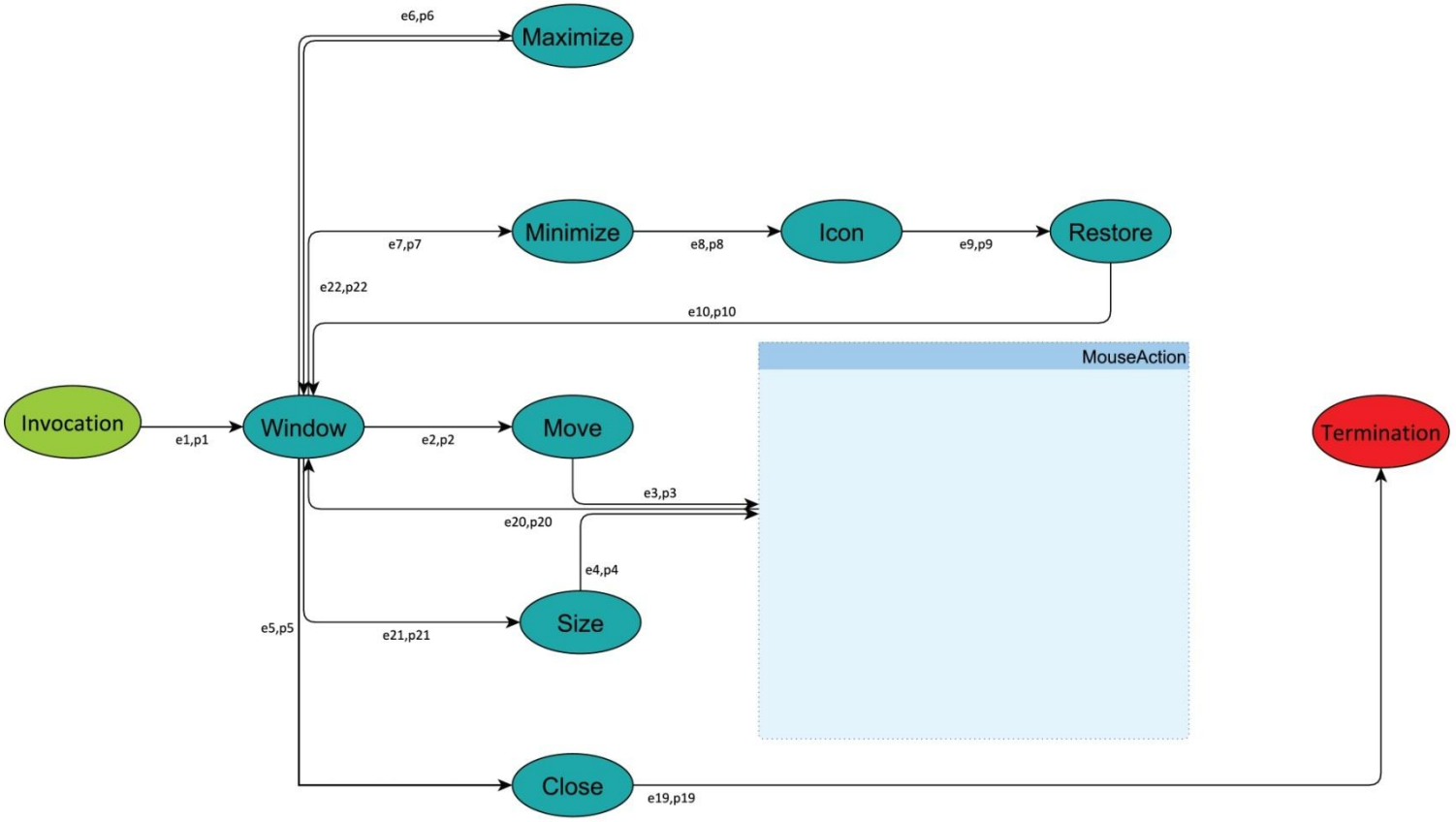
Open hierarchical MCUM



Closed hierarchical MCUM



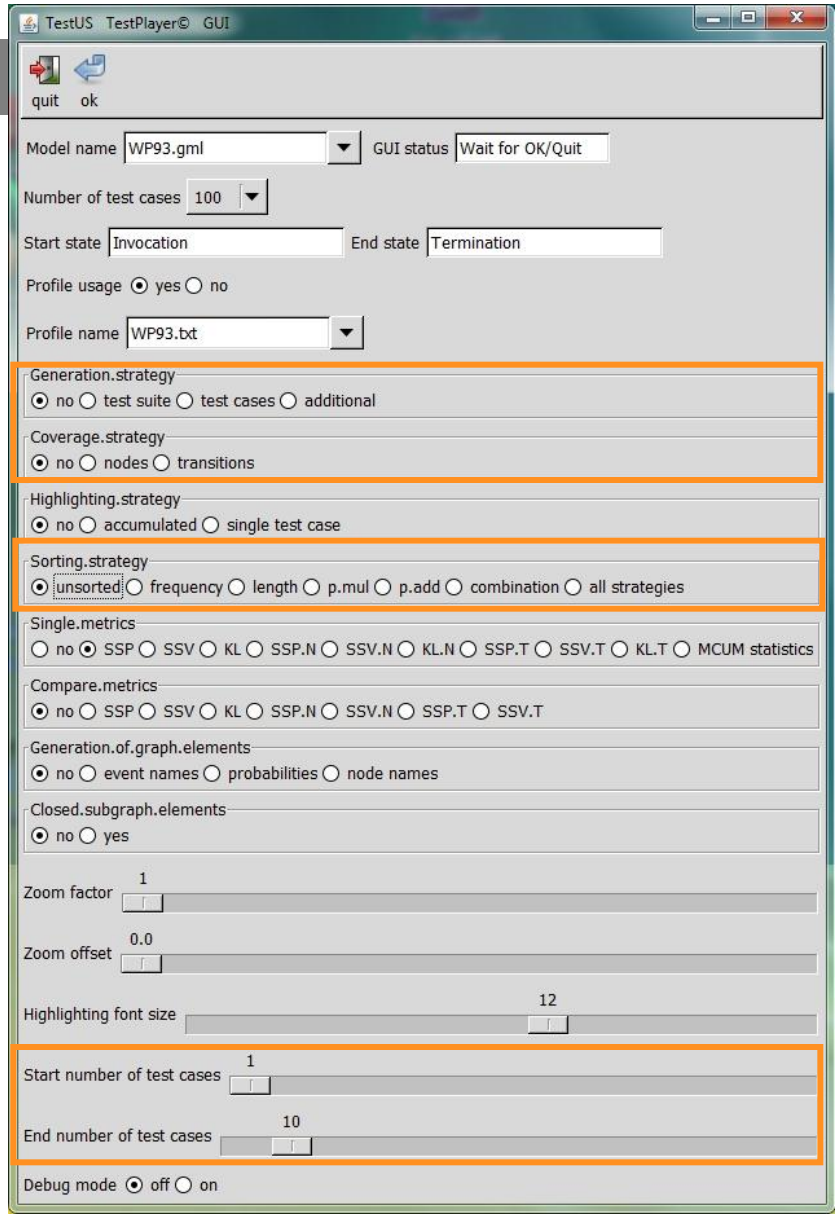
Pruned hierarchical MCUM



TestPlayer User Interface

TestPlayer GUI (cont.)

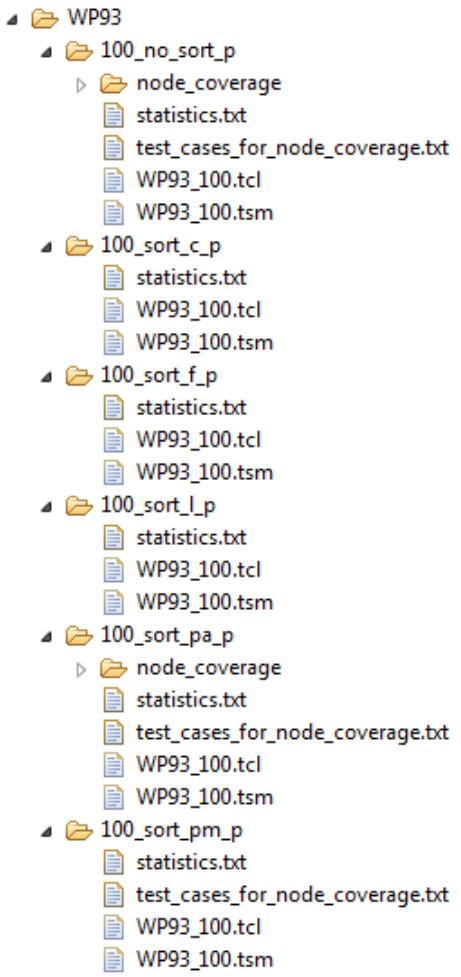
- Test case/suite generation
  - <Generation.strategy>
  - <Coverage.strategy>
    - <nodes>
    - <transitions>
  - <Sorting.strategy>
  - <Start.number> and <End.number> of generated test cases



TestPlayer User Interface

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



Test case generation

test_cases_for_node_coverage.txt - Editor

Datei Bearbeiten Format Ansicht ?

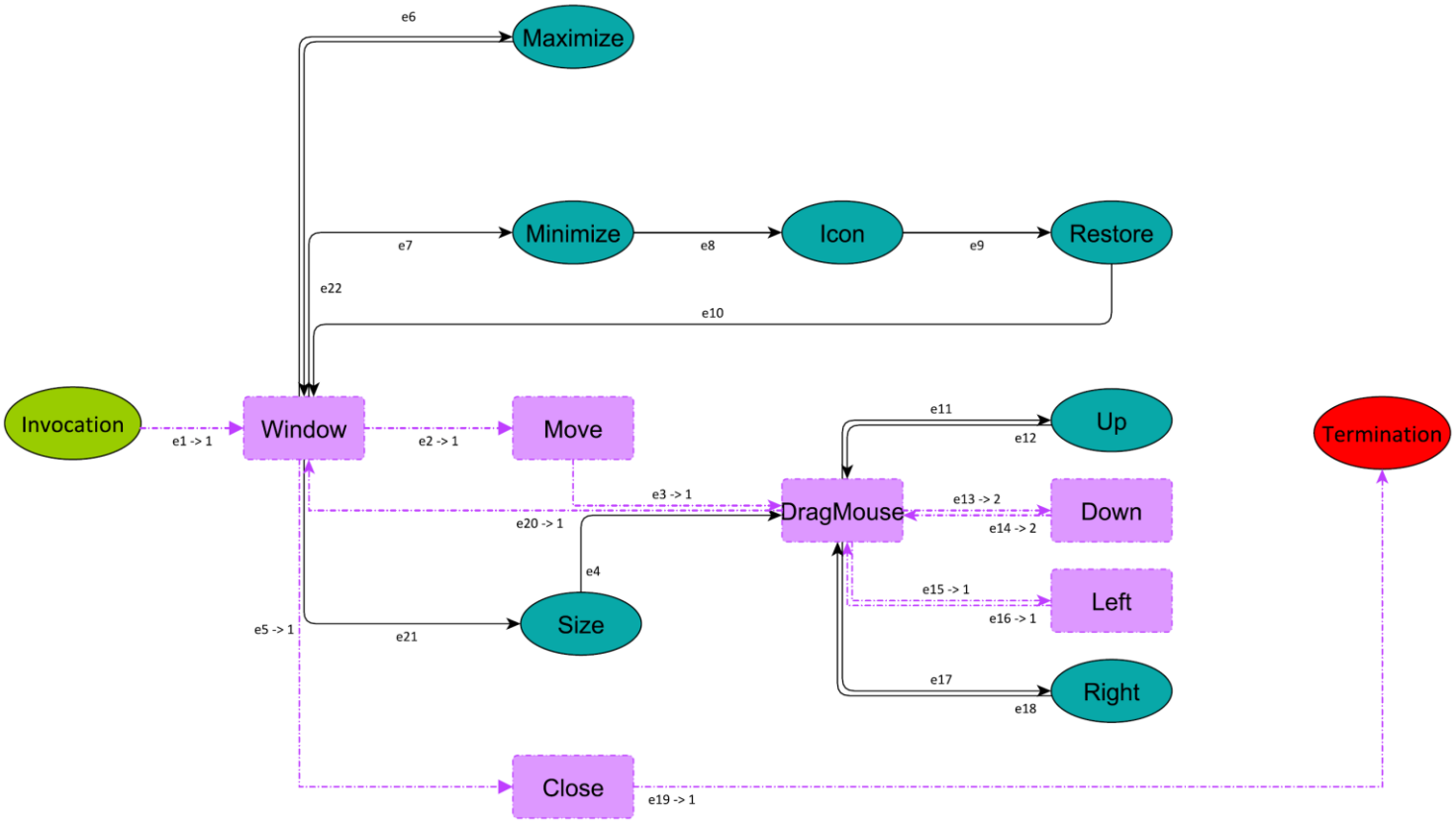
```
============================
Test cases for node coverage
============================
Test case file:  F:/StatET/workspace/TestPlayer_1.0/test_cases/WP93/100_no_sort_p/WP93_100.tcl
Generation date:  Sat Mar 12 00:54:06 2011
Number of test cases:  3

==============
TEST CASE[ 1 ]
(l= 12 , pm= 0.00049 , pa= 7.8 , f= 1 )
==============
Invocation
  ->   e1
Window
  ->   e2
Move
  ->   e3
DragMouse
  ->   e13
Down
  ->   e14
DragMouse
  ->   e13
Down
  ->   e14
DragMouse
  ->   e15
Left
  ->   e16
DragMouse
  ->   e20
Window
  ->   e5
Close
  ->   e19
Termination
================
END OF TEST CASE
================

==============
TEST CASE[ 2 ]
(l= 18 , pm= 4.4e-07 , pa= 11 , f= 1 )
==============
Invocation
  ->   e1
Window
  ->   e2
Move
  ->   e3
DragMouse
  ->   e15
Left
  ->   e16
DragMouse
  ->   e13
Down
  ->   e14
```

statistics.txt - Editor

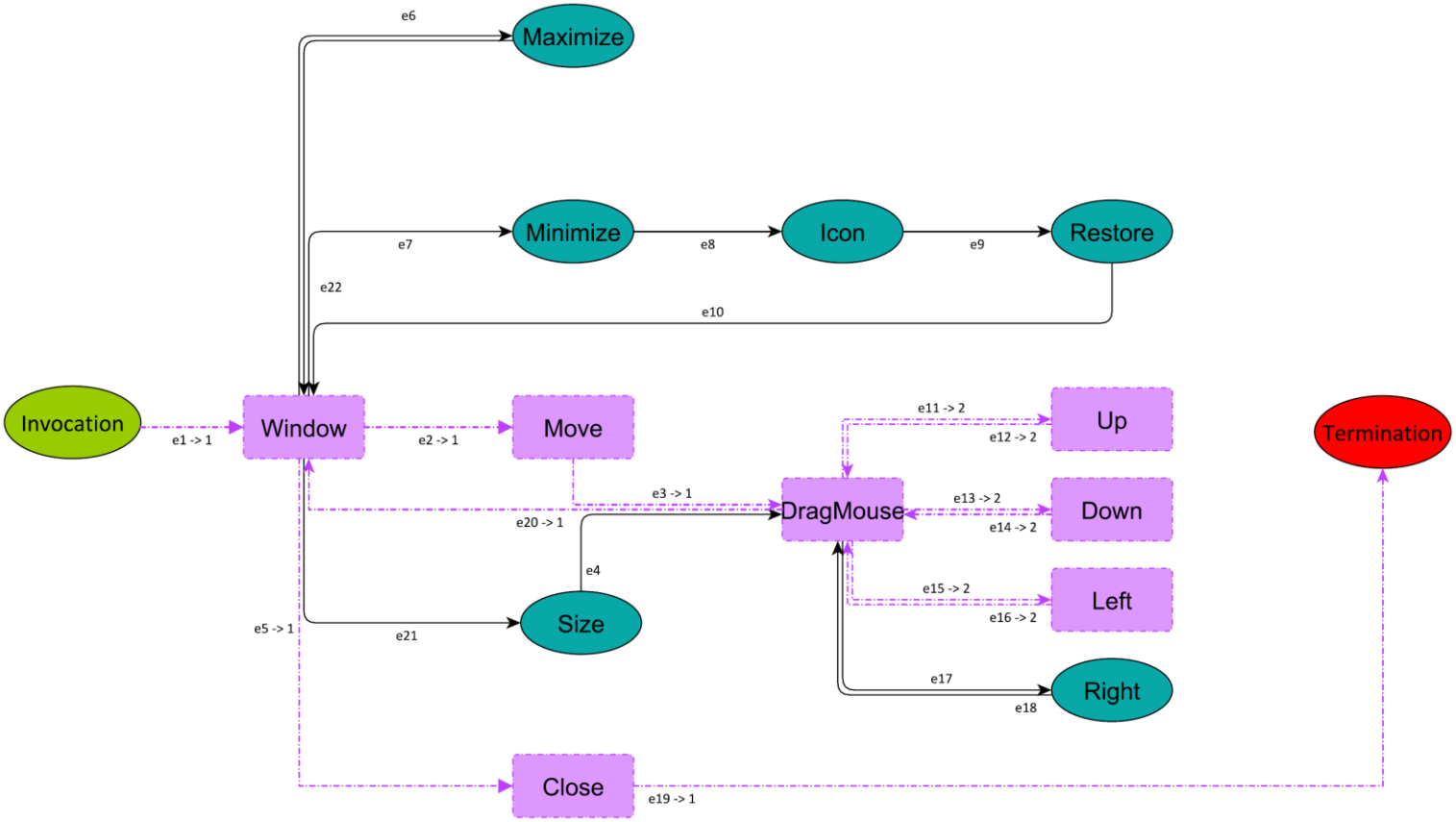
Datei Bearbeiten Format Ansicht ?

```
============================
Test suite characteristics
============================
Test case file:
        F:/StatET/workspace/TestPlayer_1.0/test_cases/WP93/100_no_sort_p/WP93_100.tcl
Generation date:
        Tue Mar 08 21:21:53 2011
Probabilities:
        p1=1
        p2=1/6
        p3=1
        p4=1
        p5=1/2
        p6=1/12
        p7=1/12
        p8=1
        p9=1
        p10=1
        p11=1/15
        p12=1
        p13=1/3
        p14=1
        p15=1/5
        p16=1
        p17=2/15
        p18=1
        p19=1
        p20=4/15
        p21=1/6
        p22=1
Number of relevant nodes(non zero frequency):
        15
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:
        9.6667
Mean length of a test case in the test suite:
        10.1065
Kullback/Leibler divergence between MCUM and test suite:
        0.0111
Mean weighted transition deviation between MCUM and test suite:
        0.1766
Number of test cases needed to cover all nodes:
        3
Test cases that cover all nodes:
        1 2 4
Number of test cases needed to cover all transitions:
        3
Test cases that cover all transitions:
        1 2 4
```

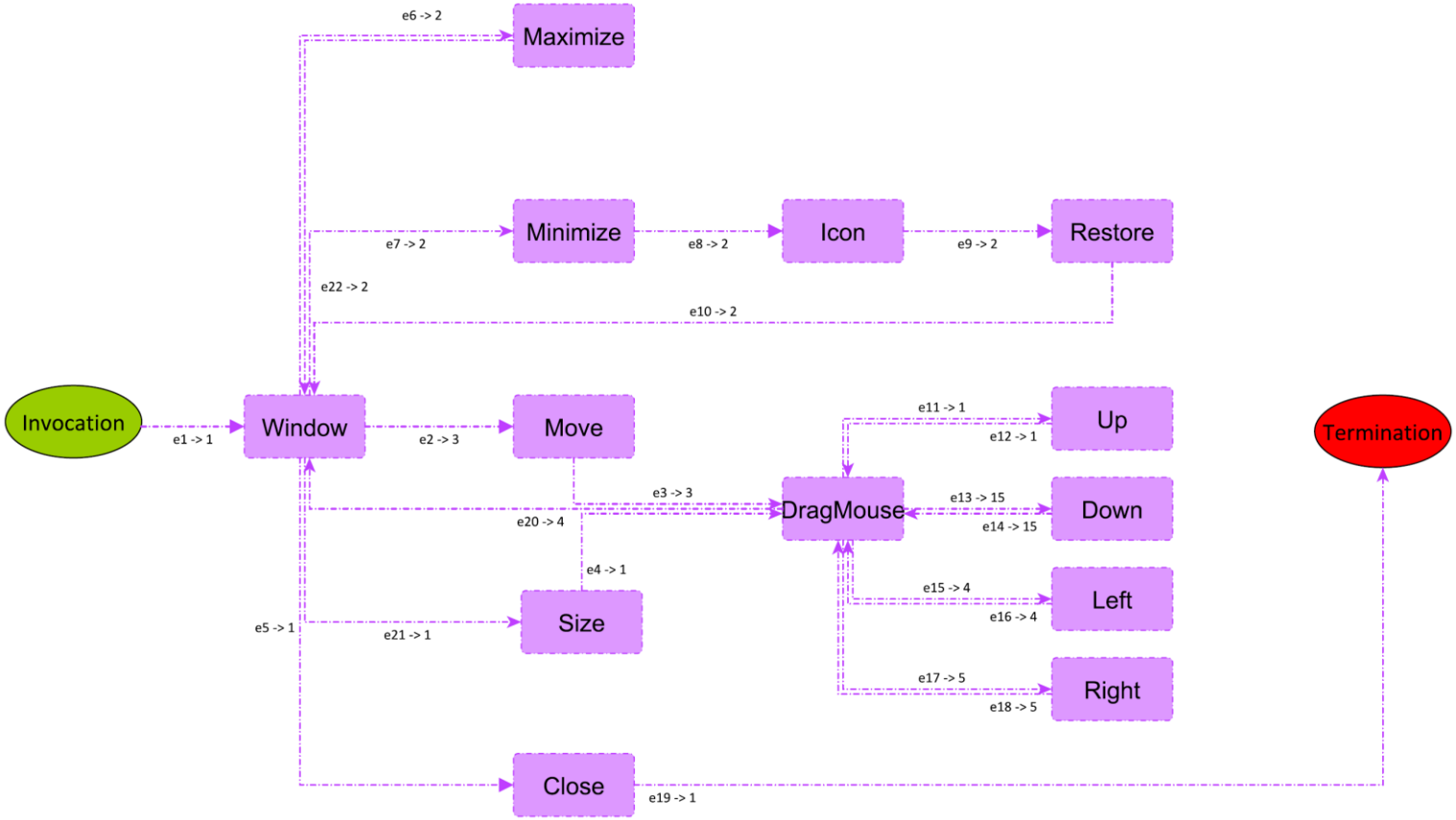
Test case visualization



Test case visualization



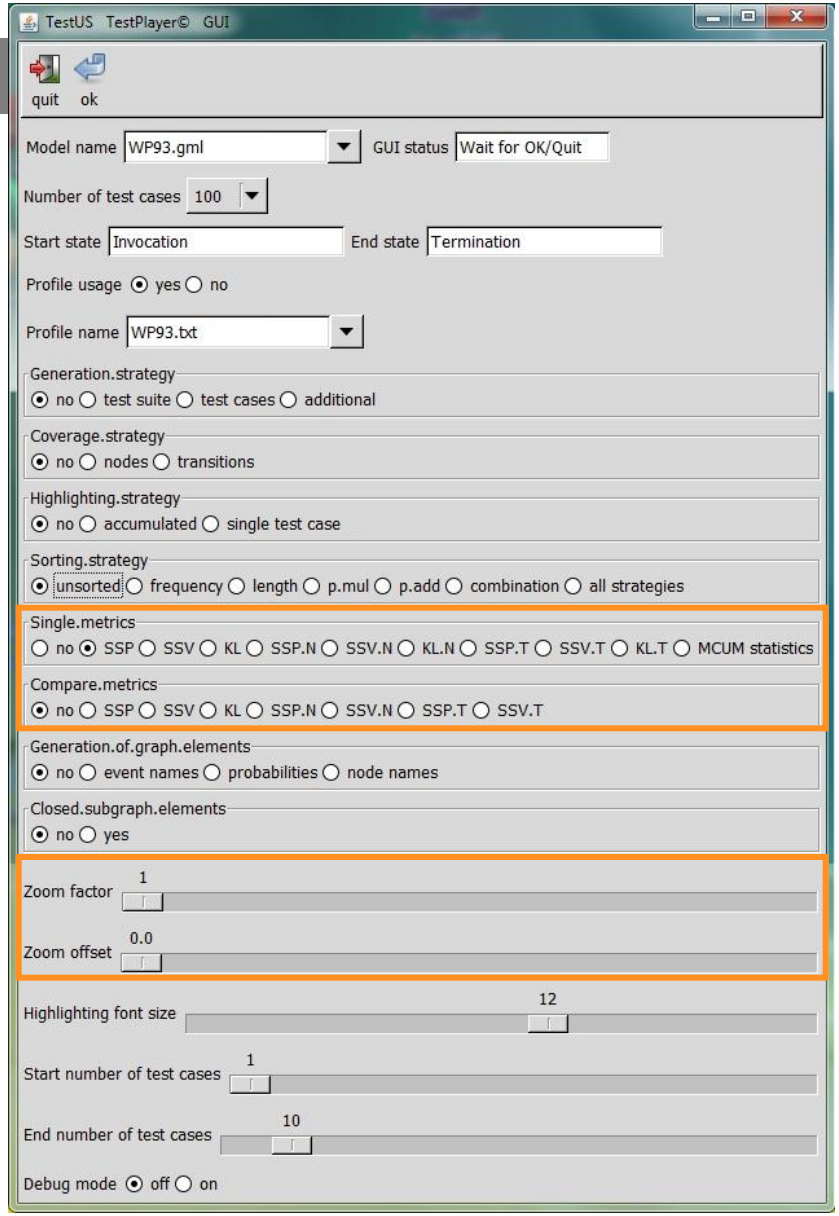
Test case visualization



TestPlayer User Interface

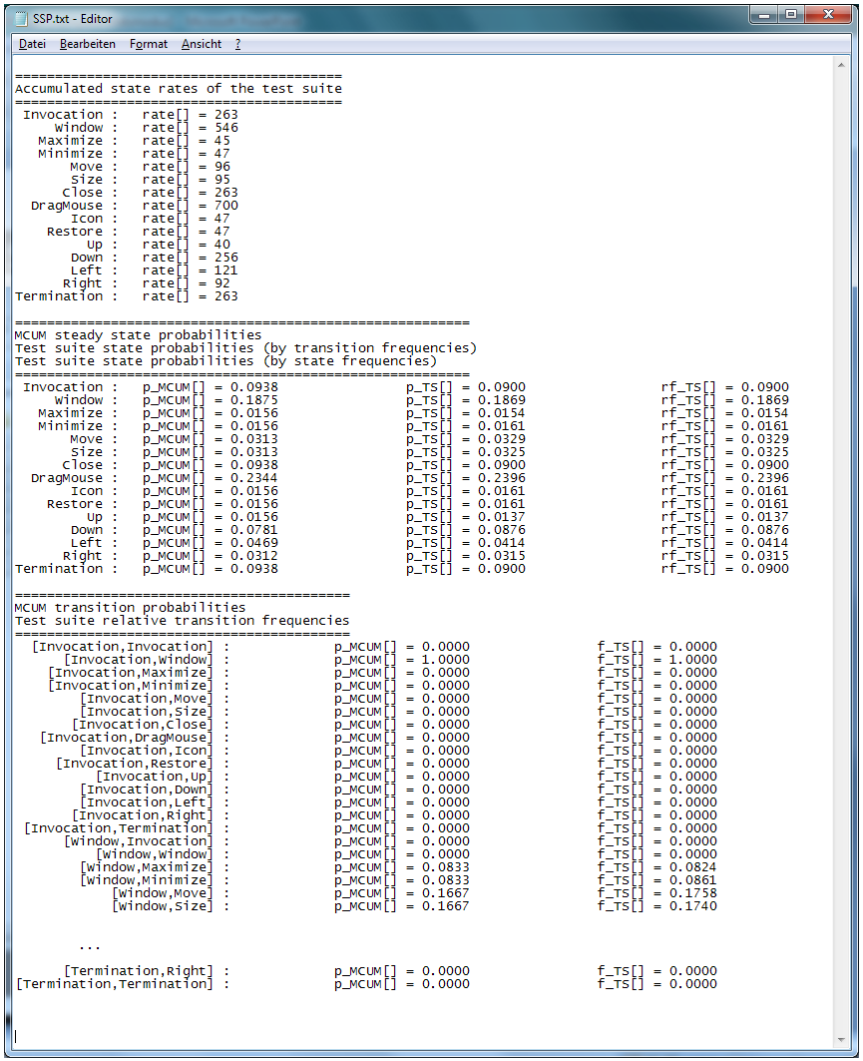
TestPlayer GUI (cont.)

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



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

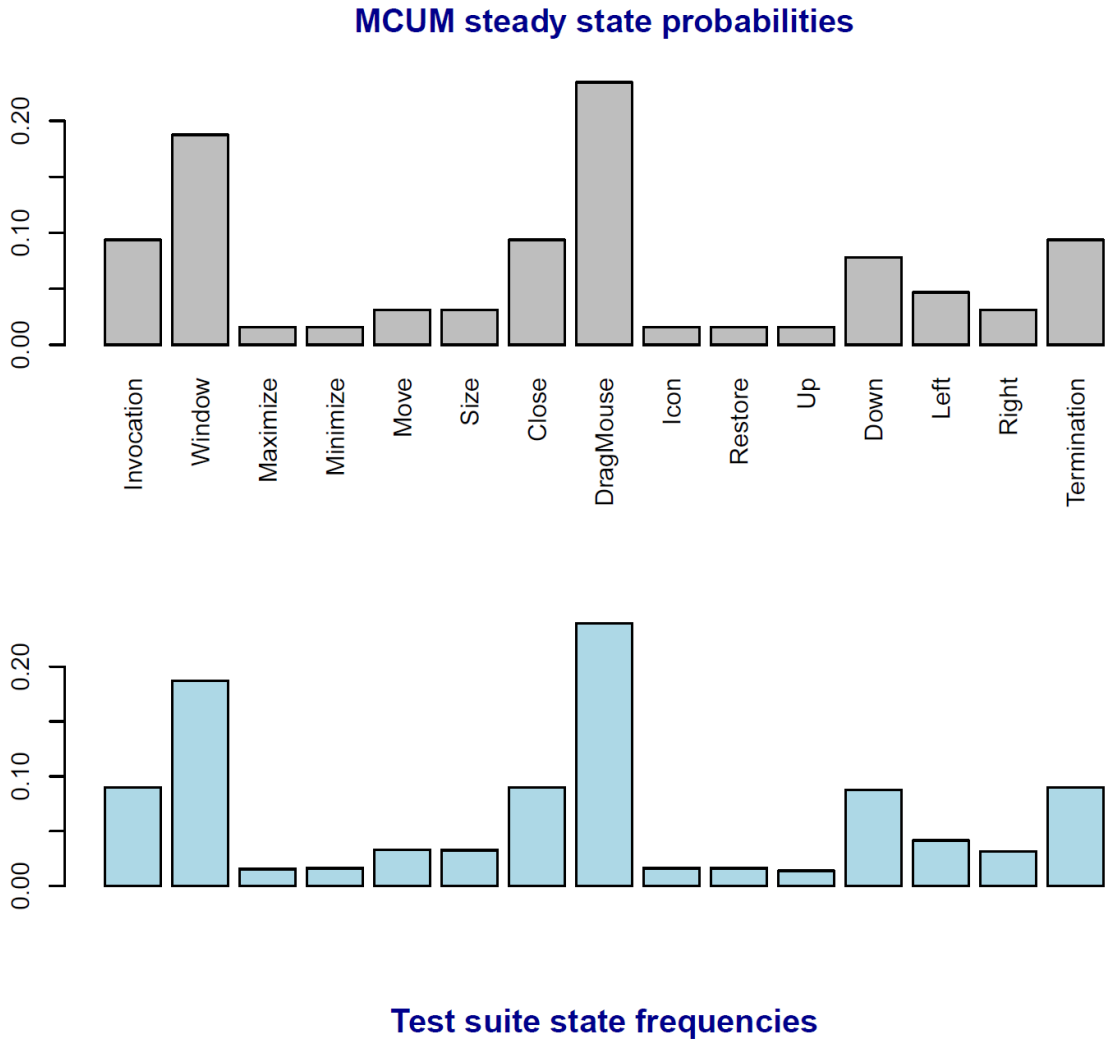


SSP.txt - Editor

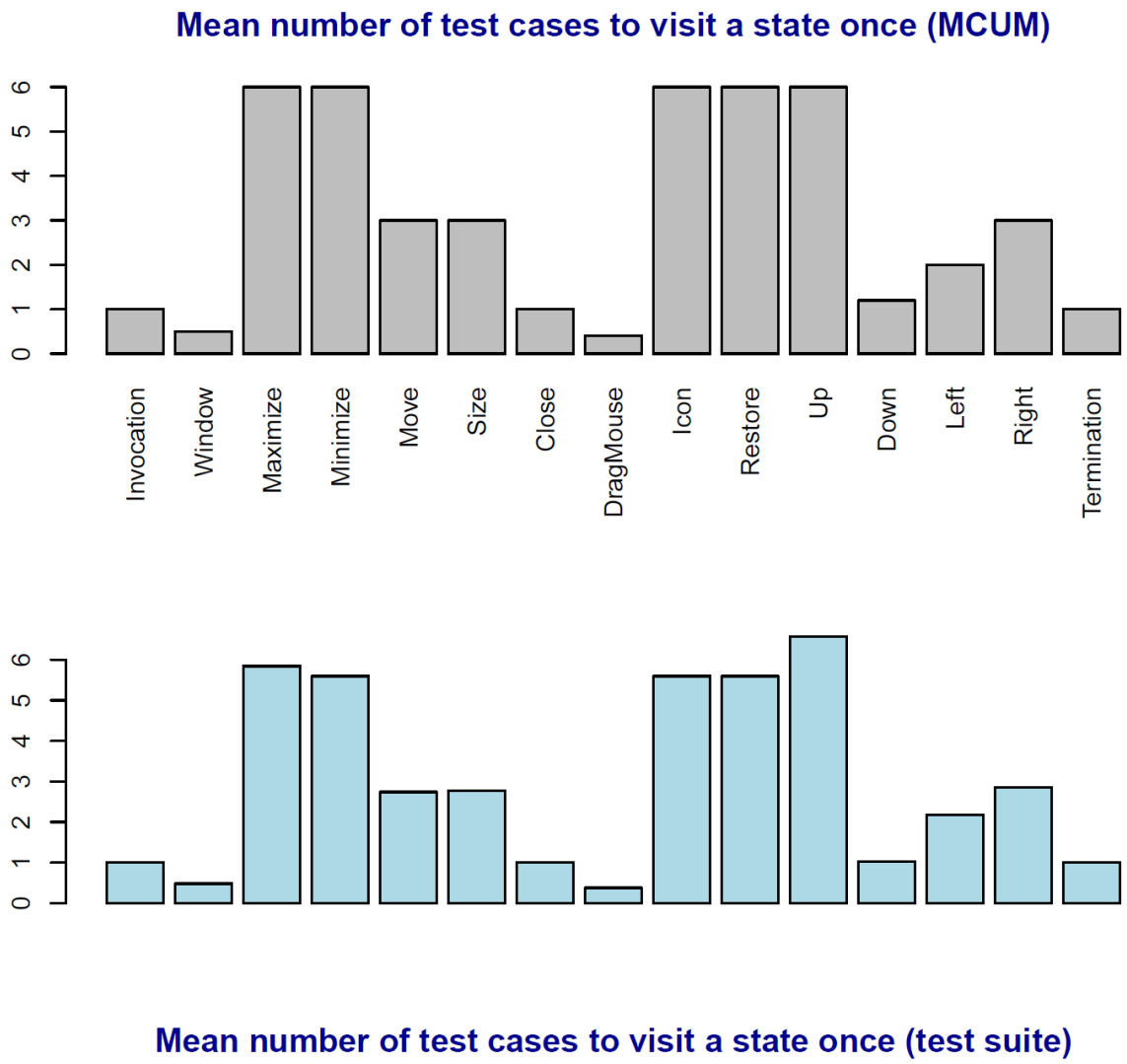
Datei Bearbeiten Format Ansicht ?

```
========================================
Accumulated state rates of the test suite
========================================
 Invocation :   rate[] = 263
     Window :   rate[] = 546
   Maximize :   rate[] = 45
   Minimize :   rate[] = 47
       Move :   rate[] = 96
       Size :   rate[] = 95
      Close :   rate[] = 263
  DragMouse :   rate[] = 700
       Icon :   rate[] = 47
    Restore :   rate[] = 47
         Up :   rate[] = 40
       Down :   rate[] = 256
       Left :   rate[] = 121
      Right :   rate[] = 92
Termination :   rate[] = 263

=========================================================
MCUM steady state probabilities
Test suite state probabilities (by transition frequencies)
Test suite state probabilities (by state frequencies)
=========================================================
 Invocation :   p_MCUM[] = 0.0938              p_TS[] = 0.0900              rf_TS[] = 0.0900
     Window :   p_MCUM[] = 0.1875              p_TS[] = 0.1869              rf_TS[] = 0.1869
   Maximize :   p_MCUM[] = 0.0156              p_TS[] = 0.0154              rf_TS[] = 0.0154
   Minimize :   p_MCUM[] = 0.0156              p_TS[] = 0.0161              rf_TS[] = 0.0161
       Move :   p_MCUM[] = 0.0313              p_TS[] = 0.0329              rf_TS[] = 0.0329
       Size :   p_MCUM[] = 0.0313              p_TS[] = 0.0325              rf_TS[] = 0.0325
      Close :   p_MCUM[] = 0.0938              p_TS[] = 0.0900              rf_TS[] = 0.0900
  DragMouse :   p_MCUM[] = 0.2344              p_TS[] = 0.2396              rf_TS[] = 0.2396
       Icon :   p_MCUM[] = 0.0156              p_TS[] = 0.0161              rf_TS[] = 0.0161
    Restore :   p_MCUM[] = 0.0156              p_TS[] = 0.0161              rf_TS[] = 0.0161
         Up :   p_MCUM[] = 0.0156              p_TS[] = 0.0137              rf_TS[] = 0.0137
       Down :   p_MCUM[] = 0.0781              p_TS[] = 0.0876              rf_TS[] = 0.0876
       Left :   p_MCUM[] = 0.0469              p_TS[] = 0.0414              rf_TS[] = 0.0414
      Right :   p_MCUM[] = 0.0312              p_TS[] = 0.0315              rf_TS[] = 0.0315
Termination :   p_MCUM[] = 0.0938              p_TS[] = 0.0900              rf_TS[] = 0.0900

========================================
MCUM transition probabilities
Test suite relative transition frequencies
========================================
  [Invocation,Invocation] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
      [Invocation,Window] :          p_MCUM[] = 1.0000          f_TS[] = 1.0000
    [Invocation,Maximize] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
    [Invocation,Minimize] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
        [Invocation,Move] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
        [Invocation,Size] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
       [Invocation,Close] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
   [Invocation,DragMouse] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
        [Invocation,Icon] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
     [Invocation,Restore] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
          [Invocation,Up] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
        [Invocation,Down] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
        [Invocation,Left] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
       [Invocation,Right] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
 [Invocation,Termination] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
      [Window,Invocation] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
          [Window,Window] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
        [Window,Maximize] :          p_MCUM[] = 0.0833          f_TS[] = 0.0824
        [Window,Minimize] :          p_MCUM[] = 0.0833          f_TS[] = 0.0861
            [Window,Move] :          p_MCUM[] = 0.1667          f_TS[] = 0.1758
            [Window,Size] :          p_MCUM[] = 0.1667          f_TS[] = 0.1740

          ...

     [Termination,Right] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
[Termination,Termination] :          p_MCUM[] = 0.0000          f_TS[] = 0.0000
```

Evaluation of Test Suites



Evaluation of Test Suites



Evaluation of Test Suites

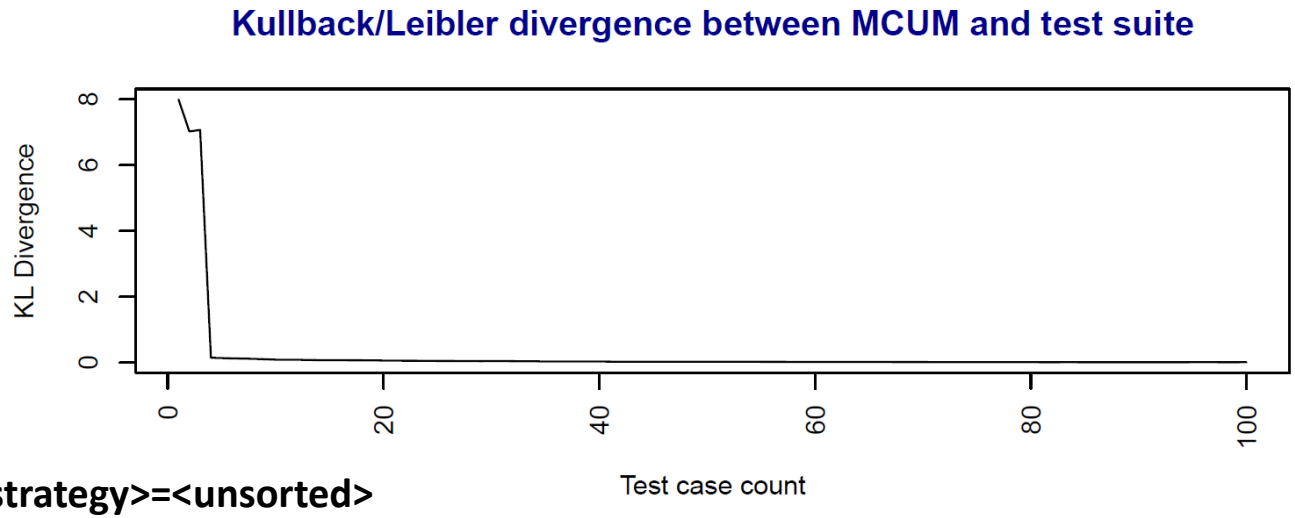
- KL
  - Kullback/Leibler divergence (KL) between MCUM and test suite vs mean weighted transition deviation (MWT) between MCUM and test suite

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

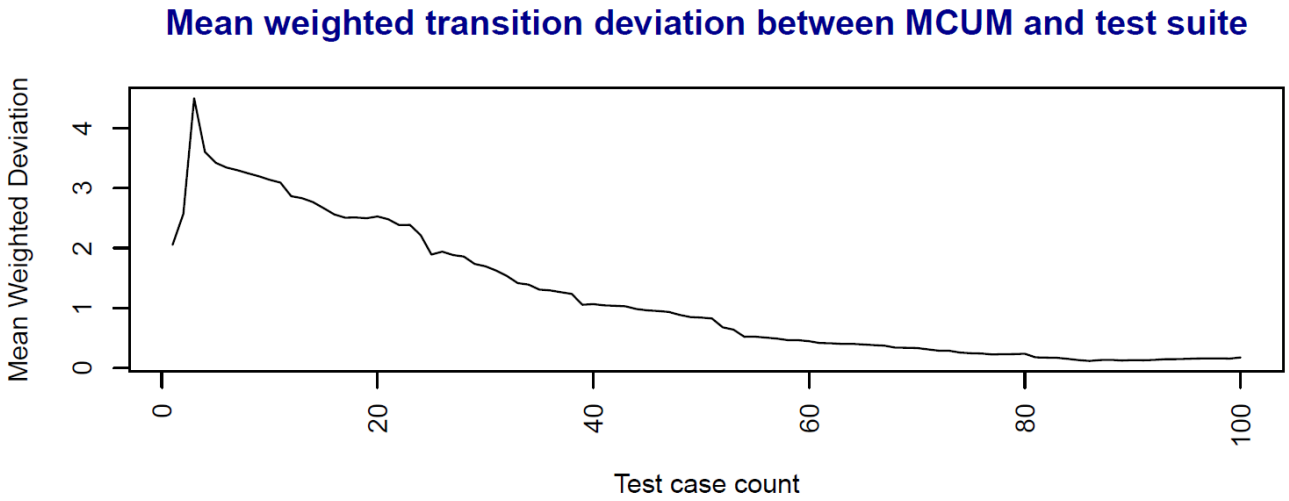
$$MWT(U,T) = \frac{\sum_i \sum_j \|u_{i,j} - t_{i,j}\|}{\max(u_{i,j}) - \min(u_{i,j})}$$

$$\overline{K(U,T)} = \sum_i \pi_i \sum_j u_{i,j} \cdot \log\left(\frac{u_{i,j}}{\epsilon - \epsilon \cdot (sgn(t_{i,j})) + t_{i,j}}\right)$$

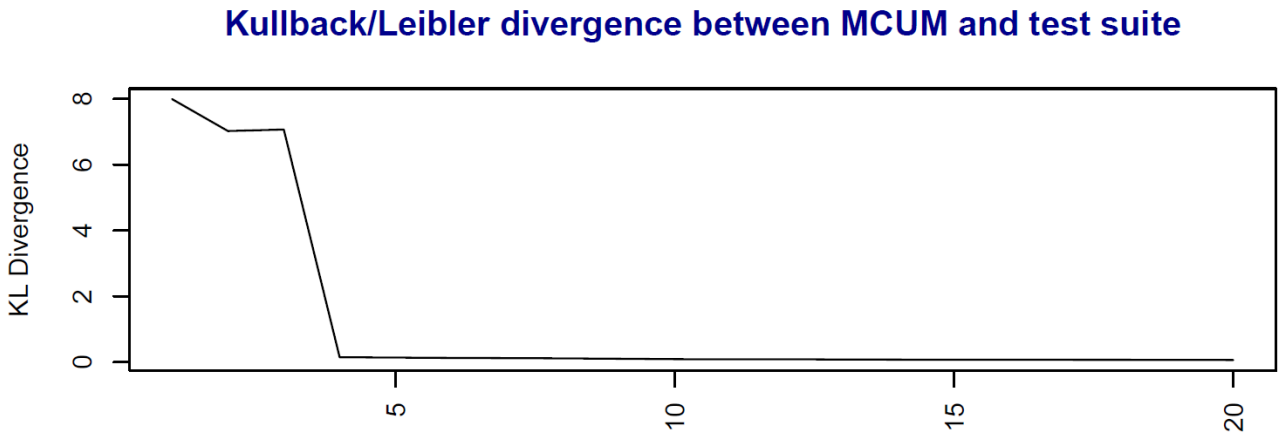
Evaluation of Test Suites



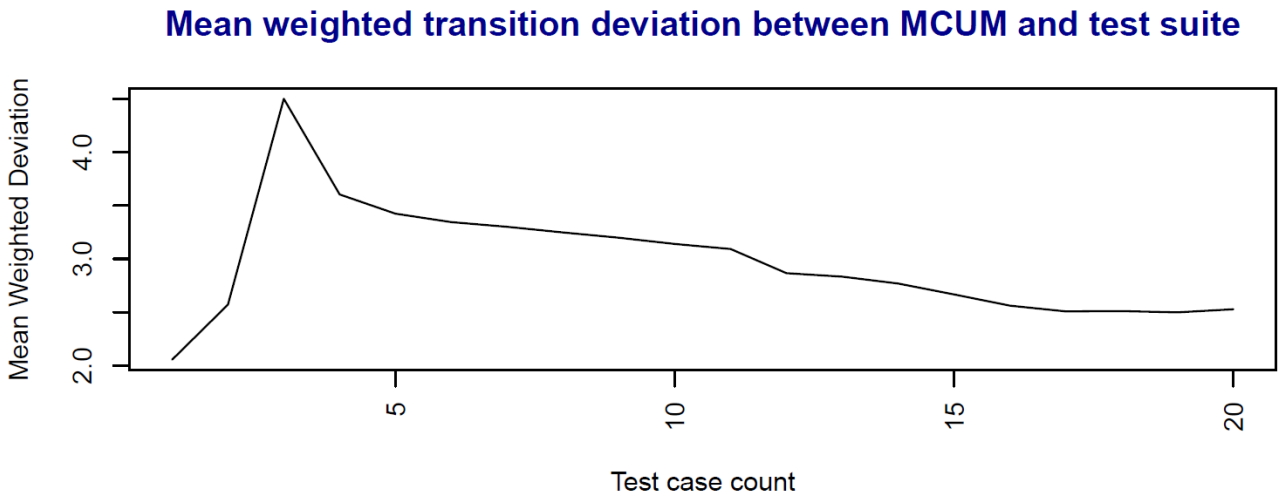
<Sorting.strategy>=<unsorted>



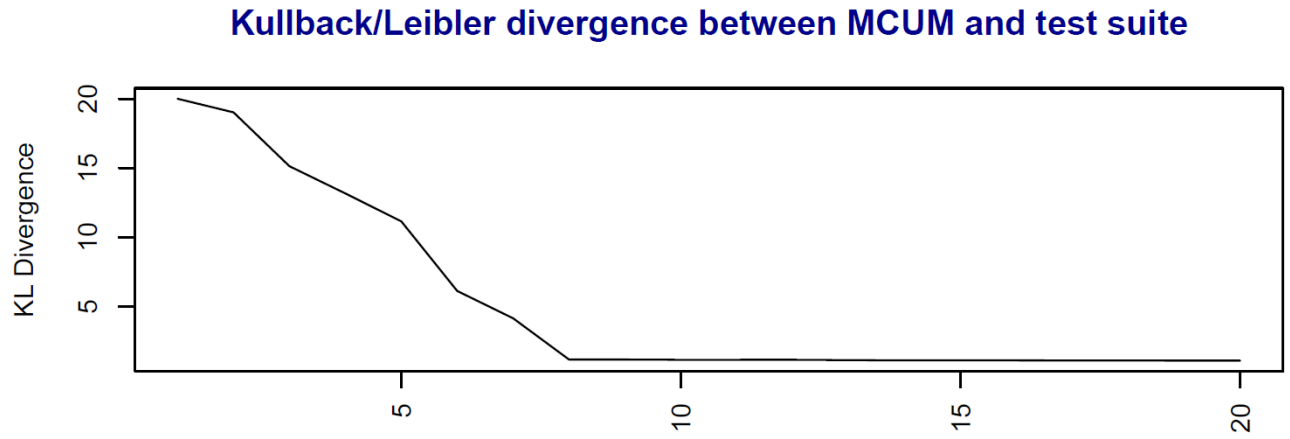
Evaluation of Test Suites



<Sorting.strategy>=<unsorted> **<Zoom factor>=<5>**

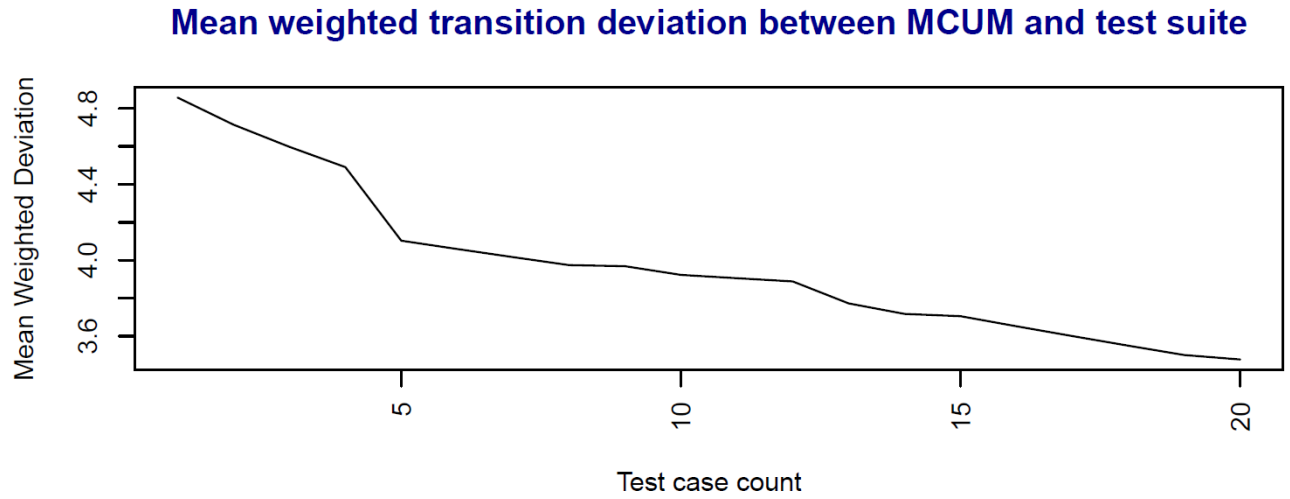


Evaluation of Test Suites

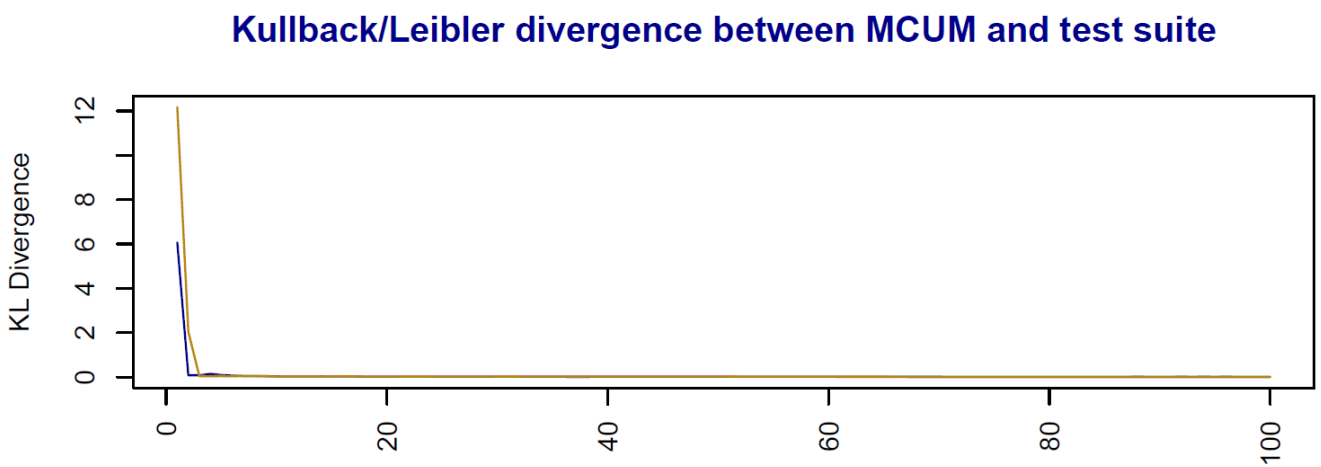


<Sorting.strategy>=<length>

<Zoom factor>=<5>

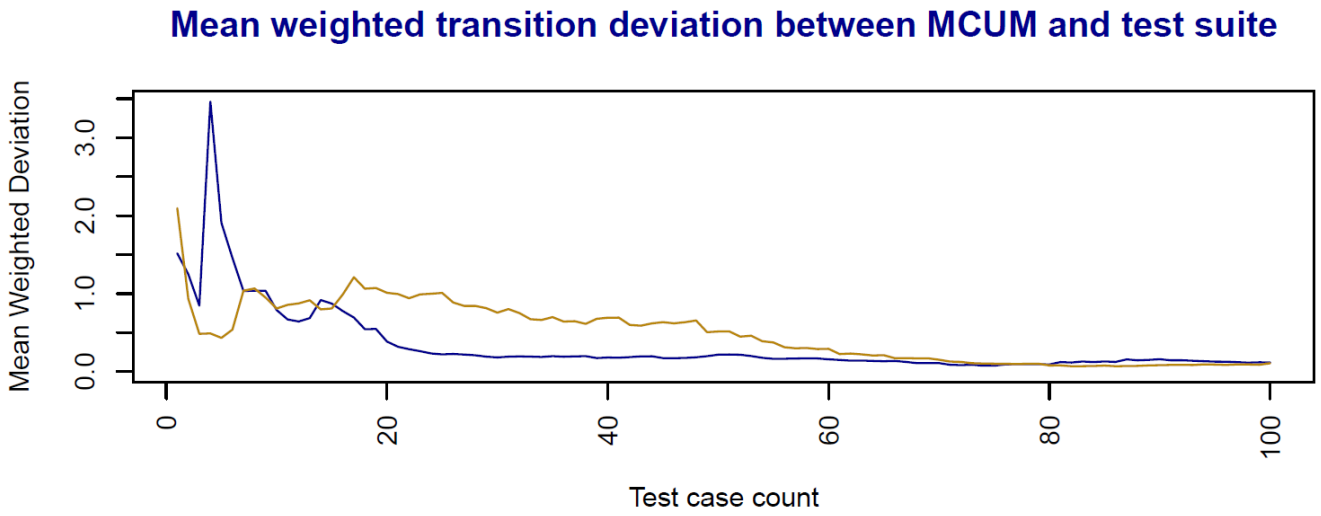


Evaluation of Test Suites

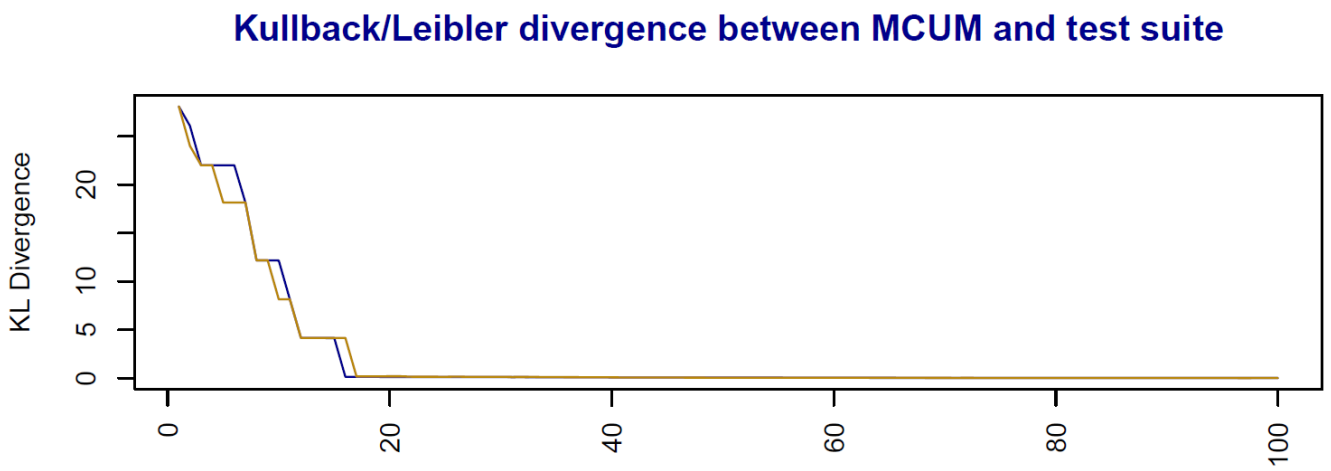


<Sorting.strategy>=<unsorted>

<Number of test cases>=<100>

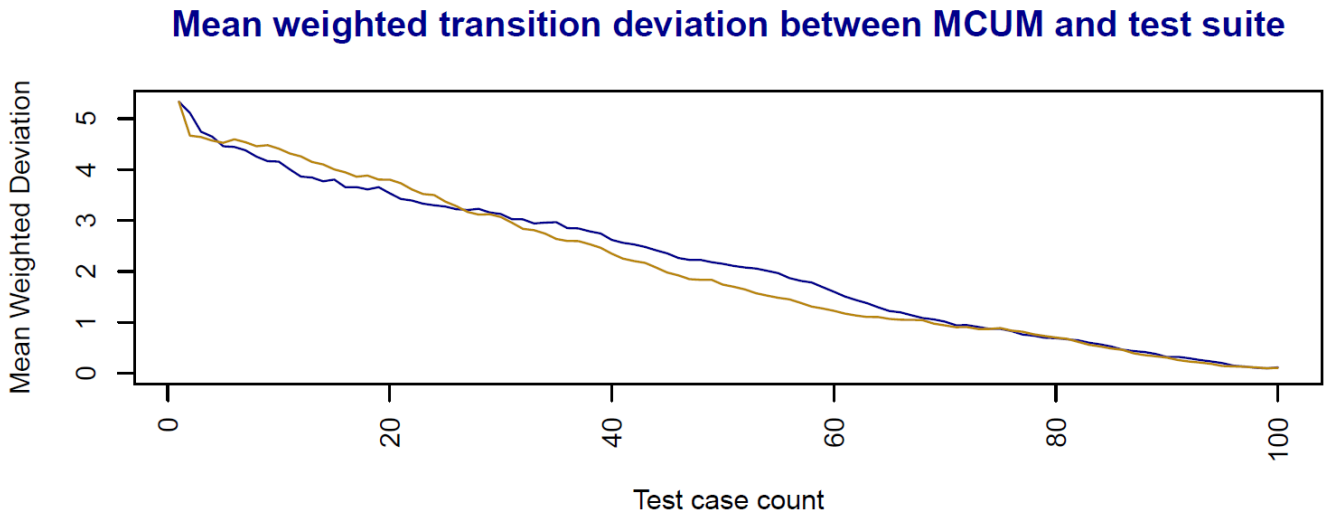


Evaluation of Test Suites

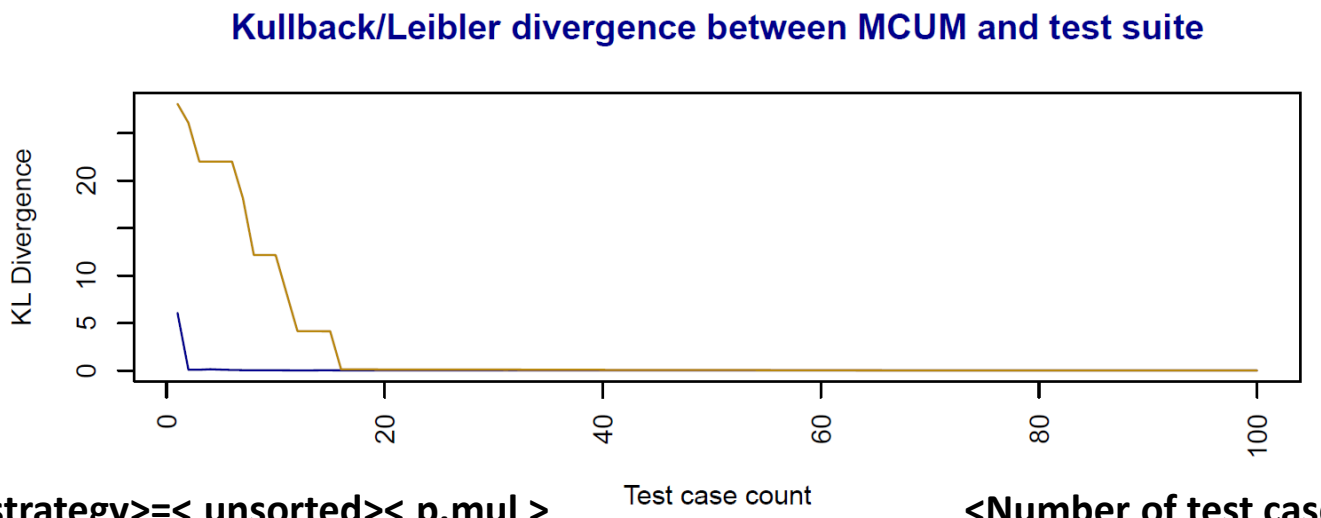


<Sorting.strategy>=< p.mul >

<Number of test cases>=<100>

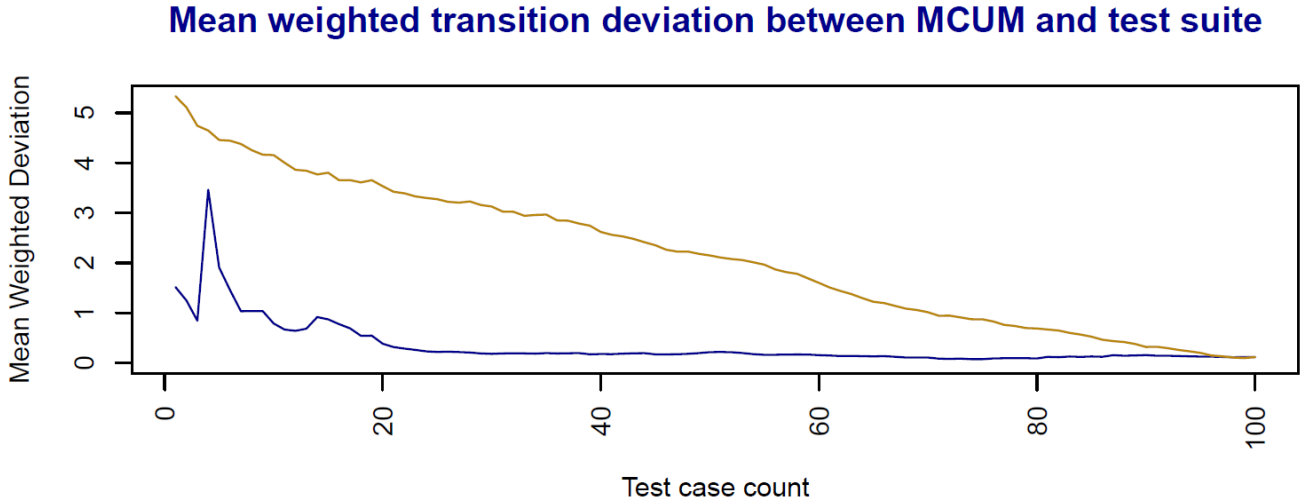


Evaluation of Test Suites



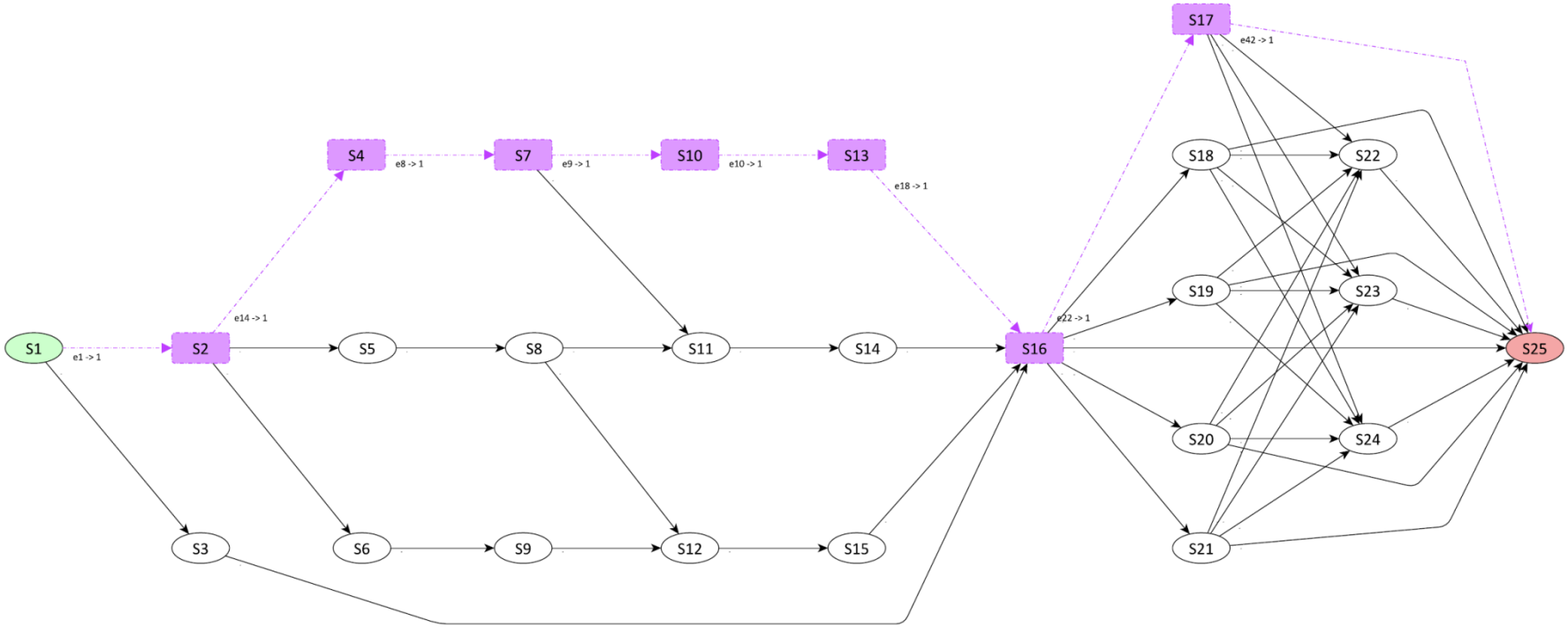
<Sorting.strategy>=< unsorted>< p.mul >

<Number of test cases>=<100>



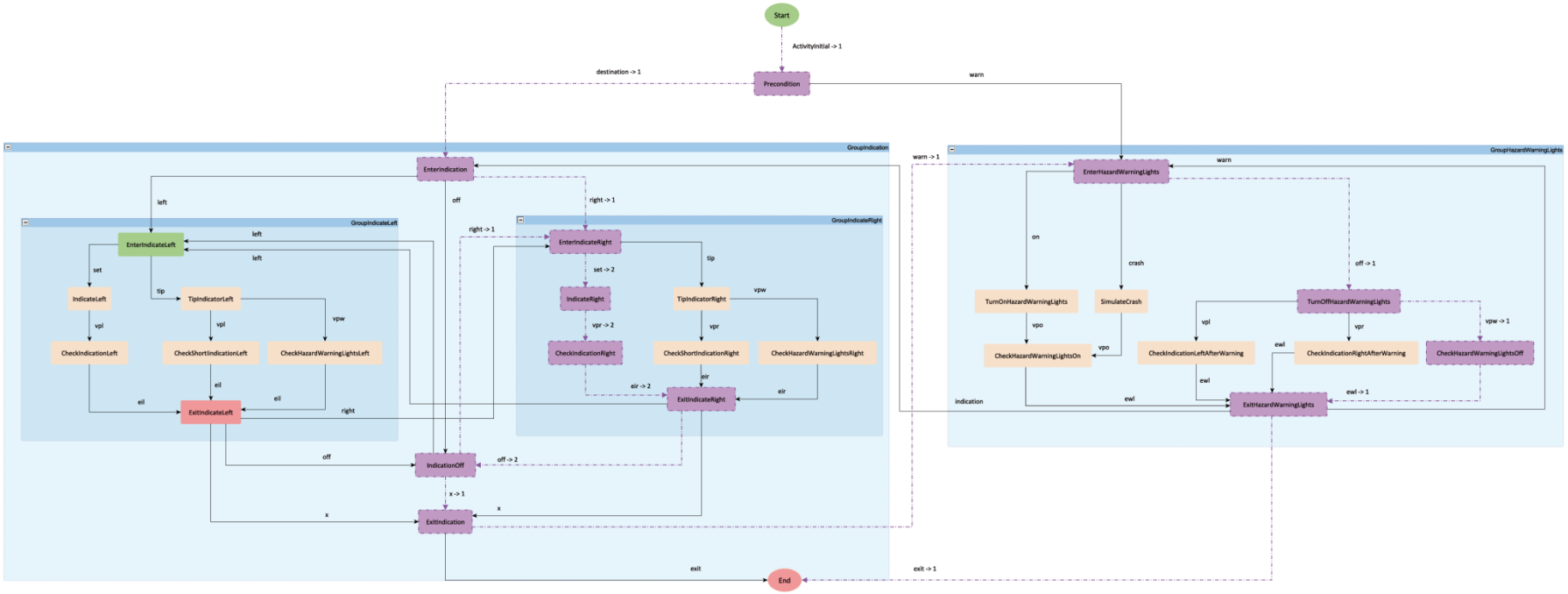
Final Remarks

- Other models



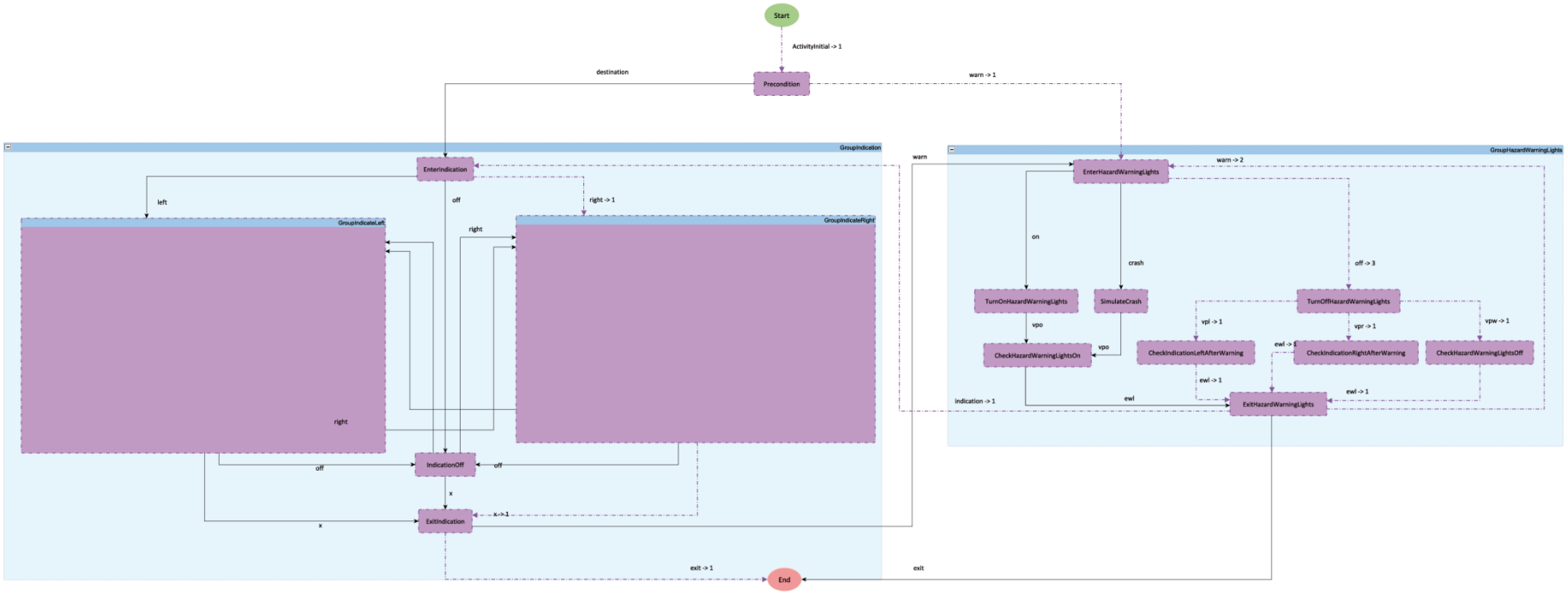
Final Remarks

- Other models



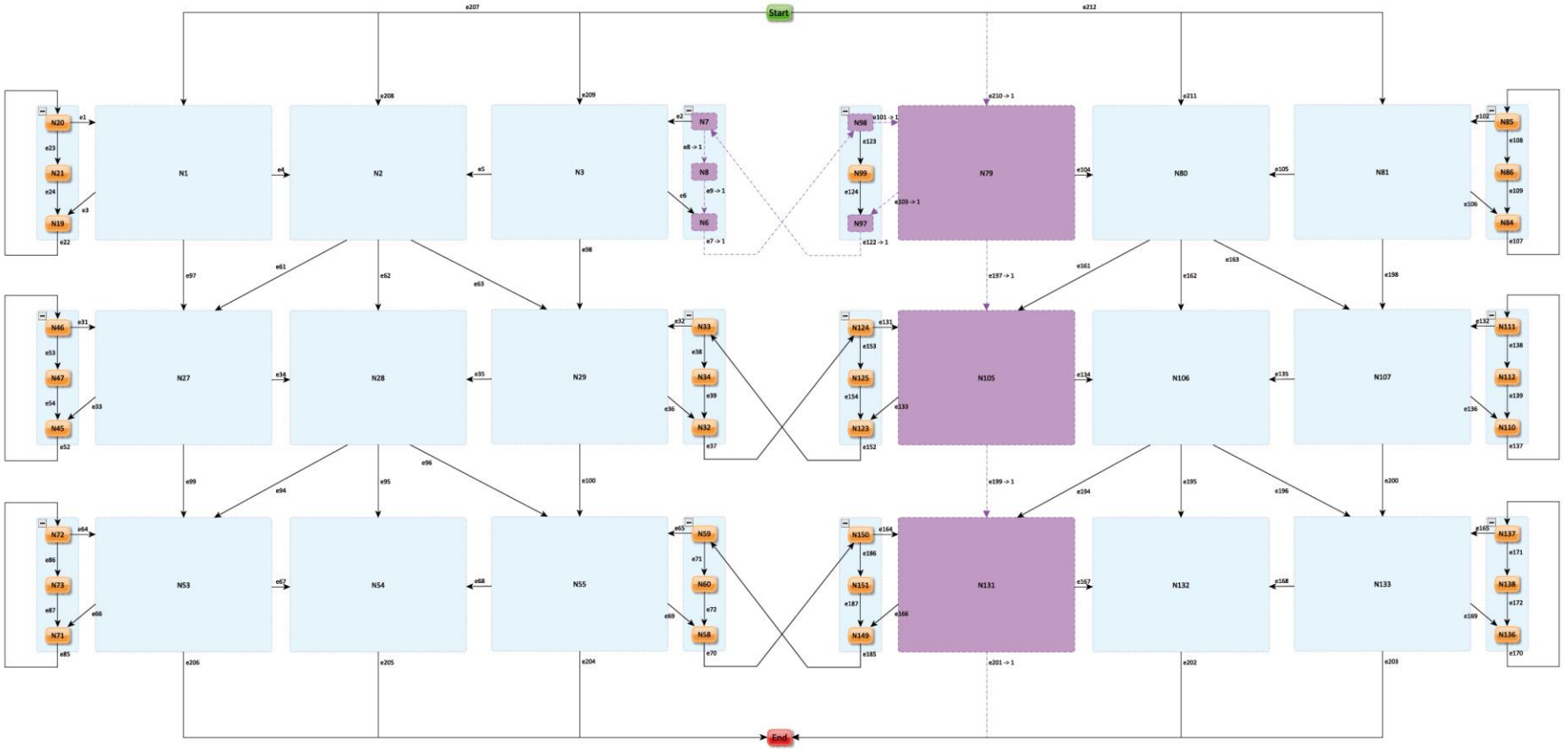
Final Remarks

- Other models



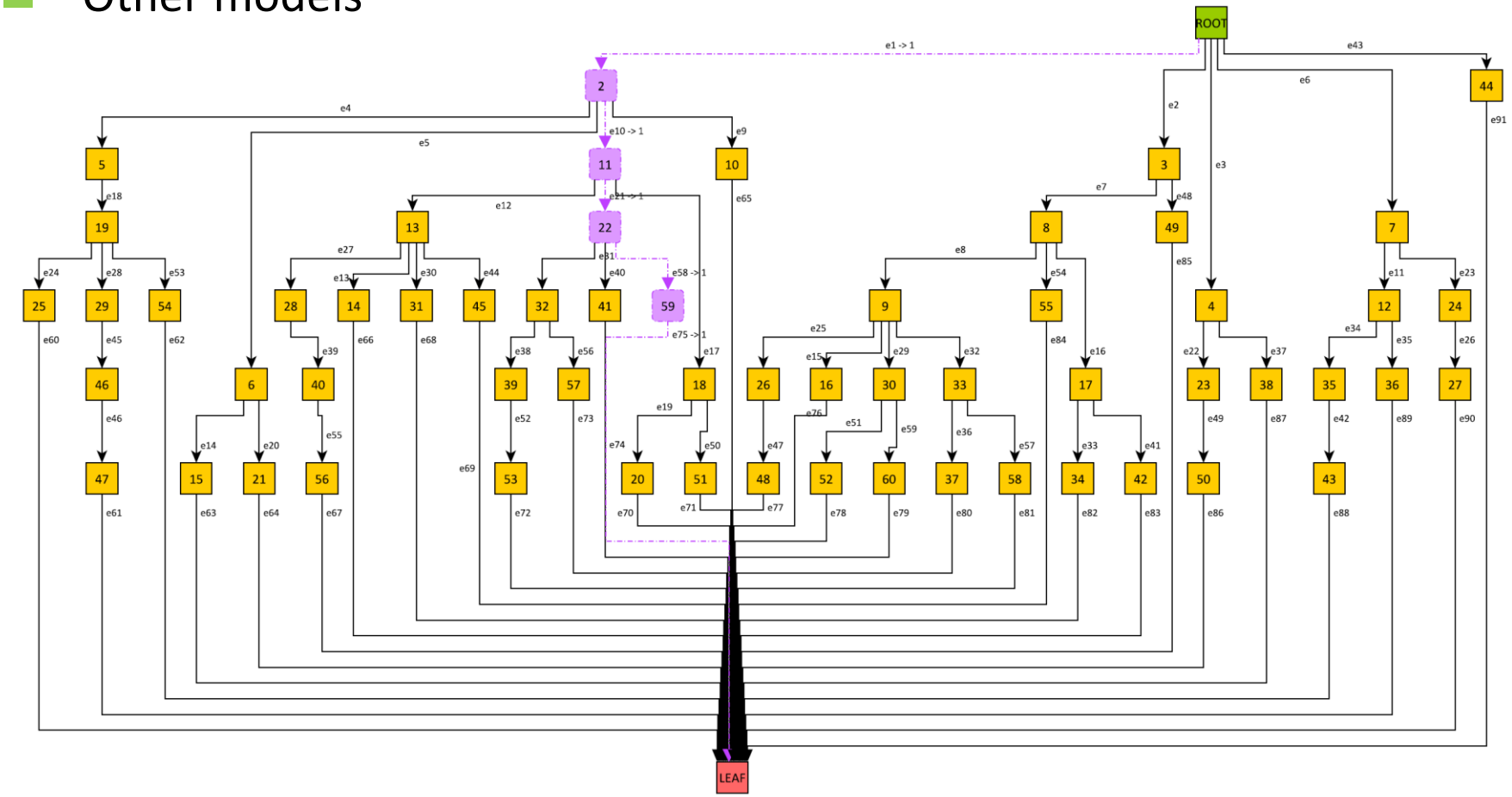
Final Remarks

- Other models



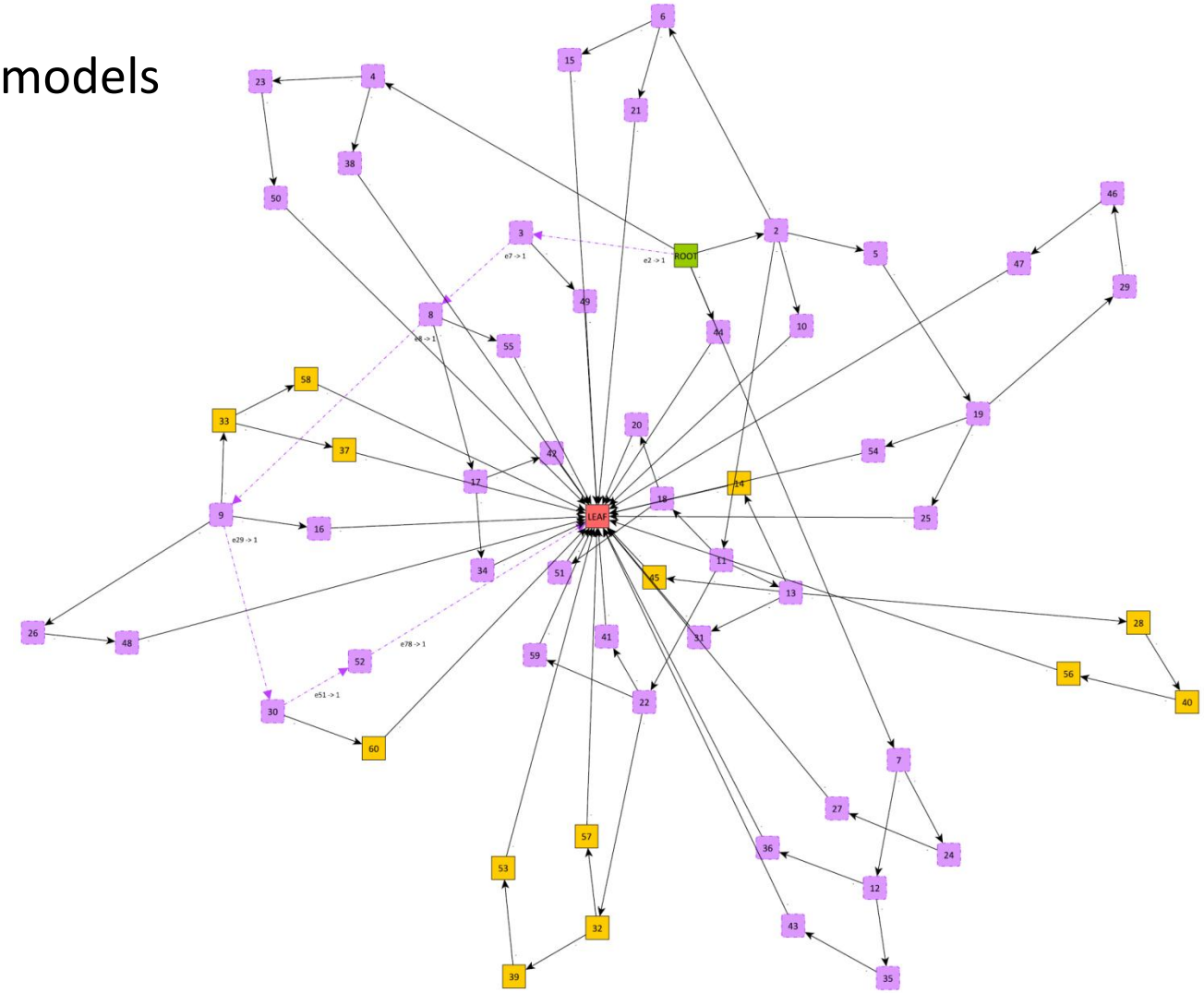
Final Remarks

- Other models



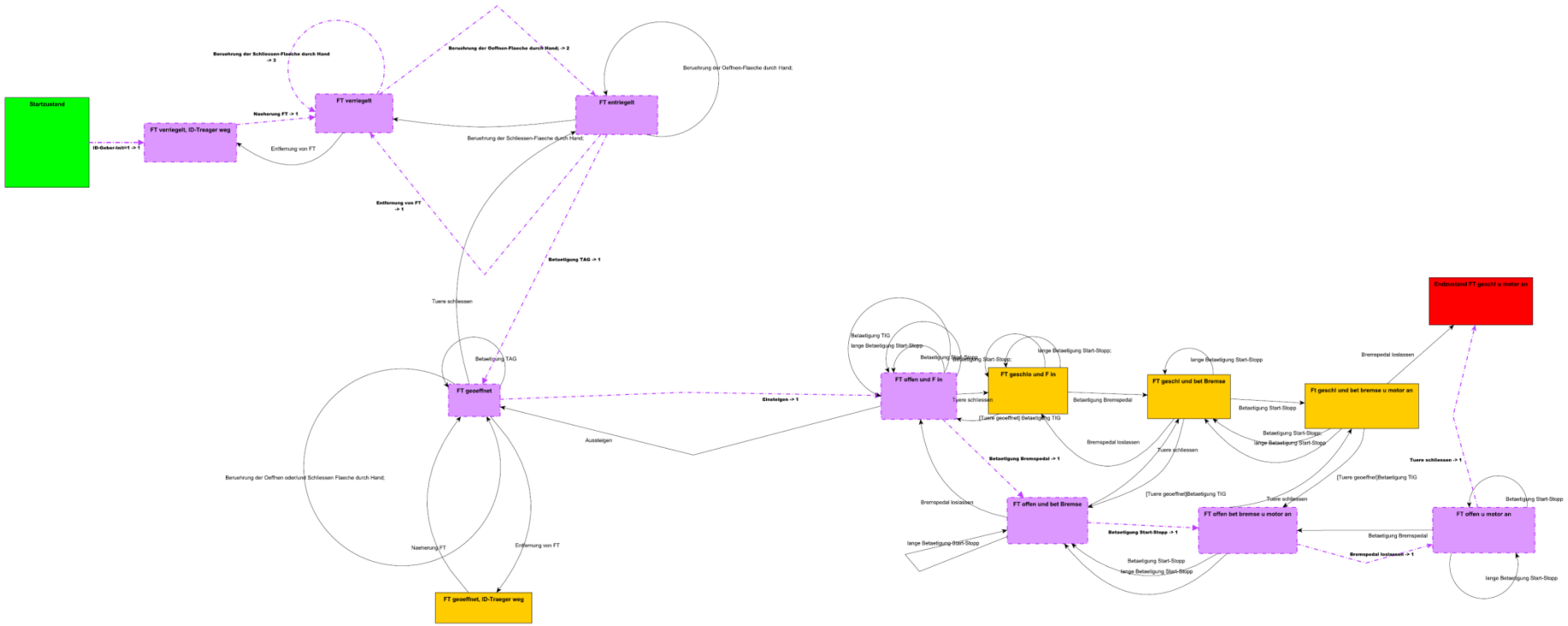
Final Remarks

- Other models



Final Remarks

- Other models



- Thank you for the attention
- Further information:

www.testus.eu

Questions ?