Counterexample visualization and explanation for function block diagrams

Antti Pakonen¹, Igor Buzhinsky²,³, Valeriy Vyatkin²,⁴

¹ VTT Technical Research Centre of Finland Ltd.
² Aalto University, Finland
³ ITMO University, Russia
⁴ Luleå University of Technology, Sweden
Model checking

1. SYSTEM
2. SYSTEM MODEL
3. FORMAL PROPERTY
4. MODEL CHECKING
   Does the model satisfy the property?
   - YES
   - NO
5. COUNTER EXAMPLE
VTT customer projects

**STUK**

**Olkiluoto 3 (under construction)**
- Evaluation of NPP I&C system designs 2008-2011
- Evaluation of Olkiluoto 3 Protection System 2015
- Evaluation of Olkiluoto 3 PACS 2015

**Fortum**

**Loviisa 1 & 2 I&C modernization**
- Verification of nuclear automation 2009
- Verification of nuclear I&C in the LARA project 2012-2014
- Verification of nuclear I&C in the ELSA project 2016-2017

**Fennovoima**

**Hanhikivi 1 (decision-in-principle)**
- Model checking of functional, architecture-level I&C 2016, 2018

Digital I&C in Finnish NPPs

*Between 2008 and 2017: 43 identified design issues*
Counterexample interpretation

Trace Description: LTL Counterexample
Trace Type: Counterexample

-> State: 1.1 <-
TANK01_L_S_1 = 0
TANK01_L_S_1_FAULT = FALSE
TANK01_L_S_2 = 0
TANK01_L_S_2_FAULT = FALSE
TANK01_L_S_3 = 0
TANK01_L_S_3_FAULT = FALSE
MAN_S_RESET = FALSE
MAN_S_RESET_FAULT = TRUE
BLOCK_201_0.LCT001.last_value = FALSE
BLOCK_201_0.LCT002.last_value = FALSE
BLOCK_201_0.LCT003.last_value = FALSE
BLOCK_201_0.HCT001.last_value = FALSE
BLOCK_201_0.HCT002.last_value = FALSE
BLOCK_201_0.HCT003.last_value = FALSE
BLOCK_201_0.TOF001.clock = 10
BLOCK_201_0.TOF001.previousGoodInput = FALSE
BLOCK_201_0.TOF002.clock = 10
BLOCK_201_0.TOF002.previousGoodInput = FALSE
BLOCK_201_0.V23001.OUT = TRUE
BLOCK_201_0.V23001.votes = 0
BLOCK_201_0.V23001.faults = 3
BLOCK_201_0.V23001.fault3 = 1
BLOCK_201_0.V23001.fault2 = 1
BLOCK_201_0.V23001.fault1 = 1
BLOCK_201_0.TOF001.OUT_FAULT = TRUE
BLOCK_201_0.TOF001.OUT = FALSE
BLOCK_201_0.TOF002.OUT_FAULT = TRUE
BLOCK_201_0.TOF002.OUT = FALSE
BLOCK_201_0.TOF002.delay = 10
BLOCK_301_0.FLUSH_FAULT = TRUE
BLOCK_301_0.FLUSH = FALSE
BLOCK_301_0.REFILL_FAULT = TRUE
BLOCK_301_0.REFILL = FALSE
BLOCK_301_0.PUL001.OUT_FAULT = TRUE
BLOCK_301_0.PUL001.OUT = FALSE
BLOCK_301_0.PUL001.delay = 10
BLOCK_301_0.AND001.OUT_FAULT = TRUE
BLOCK_301_0.AND001.OUT = FALSE

-> State: 1.2 <-
MAN_S_RESET_FAULT = FALSE
BLOCK_201_0.TOF001.clock = 0
BLOCK_201_0.TOF002.clock = 0
TANK01_FLUSH_FAULT = FALSE
TANK01_FLUSH = TRUE
TANK01_REFILL_FAULT = FALSE
BLOCK_301_0.PUL001.OUT = FALSE
BLOCK_301_0.PUL001.delay = 10
BLOCK_301_0.SR001.OUT_FAULT = TRUE
BLOCK_301_0.SR001.OUT = FALSE
BLOCK_301_0.SR002.OUT_FAULT = TRUE
BLOCK_301_0.SR002.OUT = FALSE
BLOCK_301_0.AND001.OUT_FAULT = TRUE
BLOCK_301_0.AND001.OUT = FALSE

-> State: 1.2 <-
MAN_S_RESET_FAULT = FALSE
BLOCK_201_0.TOF001.clock = 0
BLOCK_201_0.TOF002.clock = 0
TANK01_FLUSH_FAULT = FALSE
TANK01_FLUSH = TRUE
TANK01_REFILL_FAULT = FALSE
BLOCK_301_0.PUL001.OUT = FALSE
BLOCK_301_0.PUL001.delay = 10
BLOCK_301_0.AND001.OUT_FAULT = TRUE
BLOCK_301_0.AND001.OUT = FALSE

<table>
<thead>
<tr>
<th>State</th>
<th>Transition</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A</td>
<td>TANK01_L_S_1 = 0, TANK01_L_S_1_FAULT = FALSE, TANK01_L_S_2 = 0, TANK01_L_S_2_FAULT = FALSE, TANK01_L_S_3 = 0, TANK01_L_S_3_FAULT = FALSE, MAN_S_RESET = FALSE, MAN_S_RESET_FAULT = TRUE, BLOCK_201_0.LCT001.last_value = FALSE, BLOCK_201_0.LCT002.last_value = FALSE, BLOCK_201_0.LCT003.last_value = FALSE, BLOCK_201_0.HCT001.last_value = FALSE, BLOCK_201_0.HCT002.last_value = FALSE, BLOCK_201_0.HCT003.last_value = FALSE, BLOCK_201_0.TOF001.clock = 10, BLOCK_201_0.TOF001.previousGoodInput = FALSE, BLOCK_201_0.TOF002.clock = 10, BLOCK_201_0.TOF002.previousGoodInput = FALSE, BLOCK_201_0.V23001.OUT = TRUE, BLOCK_201_0.V23001.votes = 0, BLOCK_201_0.V23001.faults = 3, BLOCK_201_0.V23001.fault3 = 1, BLOCK_201_0.V23001.fault2 = 1, BLOCK_201_0.V23001.fault1 = 1, BLOCK_201_0.TOF001.OUT_FAULT = TRUE, BLOCK_201_0.TOF001.OUT = FALSE, BLOCK_201_0.TOF002.OUT_FAULT = TRUE, BLOCK_201_0.TOF002.OUT = FALSE, BLOCK_201_0.TOF002.delay = 10</td>
</tr>
<tr>
<td>1.2</td>
<td>B</td>
<td>MAN_S_RESET_FAULT = FALSE, BLOCK_201_0.TOF001.clock = 0, BLOCK_201_0.TOF002.clock = 0, TANK01_FLUSH_FAULT = FALSE, TANK01_FLUSH = TRUE, TANK01_REFILL_FAULT = FALSE, BLOCK_301_0.PUL001.OUT = FALSE, BLOCK_301_0.PUL001.delay = 10, BLOCK_301_0.AND001.OUT_FAULT = TRUE, BLOCK_301_0.AND001.OUT = FALSE</td>
</tr>
</tbody>
</table>

VTT 2018
Challenge: identifying the root of the failure

Industrial examples:

<table>
<thead>
<tr>
<th>Trace length (steps)</th>
<th>Model variables</th>
<th>Variables in property</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>410</td>
<td>1</td>
</tr>
<tr>
<td>44</td>
<td>105</td>
<td>5</td>
</tr>
<tr>
<td>33</td>
<td>142</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>779</td>
<td>10</td>
</tr>
<tr>
<td>23</td>
<td>109</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>289</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>91</td>
<td>2</td>
</tr>
<tr>
<td>22</td>
<td>151</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>300</td>
<td>1</td>
</tr>
</tbody>
</table>

Questions for analyst:

Where (in the model or the in the trace) does the failure occur?

What values on the trace cause it to falsify the specification?
Model checking

1. SYSTEM
   → SYSTEM MODEL
   → MODEL CHECKING
   → YES
   → COUNTER EXAMPLE

2. SYSTEM PROPERTY
   → FORMAL PROPERTY
   → DOES THE MODEL SATISFY THE PROPERTY?
   → NO
   → Model or specification is incorrect.
   → Scenario is irrelevant in actual context.
   → Can a more serious scenario be found?
Counterexample visualization (1/3)

Variable tables


State diagrams

Sequence diagrams

"Model view"


Counterexample visualization (3/3)

Domain specific visualizations


Automated analysis of counterexamples

Counterexample minimization
Analysing common features in within sets of traces
Obtaining shortest counterexamples

Important value assignment based on theory of causal models [1]

Proposed toolset

1. Model animation

2. Property animation

3. Important value highlighting
Model animation

MODCHK
Developed at VTT, in use since 2014

<table>
<thead>
<tr>
<th>Signal type and status</th>
<th>Visualization in model view</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary TRUE, valid</td>
<td></td>
</tr>
<tr>
<td>binary TRUE, invalid</td>
<td>---</td>
</tr>
<tr>
<td>binary FALSE, valid</td>
<td>20</td>
</tr>
<tr>
<td>binary FALSE, invalid</td>
<td>0</td>
</tr>
<tr>
<td>analogue 20, valid</td>
<td>20</td>
</tr>
<tr>
<td>analogue 0, invalid</td>
<td>0</td>
</tr>
</tbody>
</table>
Property animation + important value highlight

https://github.com/igor-buzhinsky/nusmv_counterexample_visualizer
Compact vs. full mode

https://github.com/igor-buzhinsky/nusmv_counterexample_visualizer
Evaluation of the tools based on industrial examples

Property **animation** usefulness:

- **7% excellent** (directly points out root cause)
- **14% great** (provides clues that lead to the root cause)
- **79% good** (accurately points out the first failure)
- **0% fair** (no added value)

Important value **highlighting** usefulness:

- **7% excellent**
- **28% great**
- **65% good**
- **0% fair**
Conclusions

Model and property animation are both very useful. Highlighting of important values – also very useful

Practical tools should offer a range of methods.

**Model checking** – a well-established and integral I&C verification method in the Finnish nuclear industry!

https://www.vttresearch.com/modelchecking/