Analysing Hierarchical Control Software application to CERN’s detector control systems

Jeroen Keiren (@jkeiren)

IPA Spring Days, Groesbeek, May 14, 2013
Background
CERN
Background
LHC

source: CERN
Background

LHC

source: CERN
Background

CMS

source: CERN
Background

Scale

source: CERN
Control software

source: CERN
Control software
Global structure

Diagram:
- Top Control
- Software (FSMs)
- Devices

Flow:
- Status/Alarms
- Commands
Control software
Local structure
Control software
Complexity
Problem
Unresponsive subsystems
Methodology

1. Understand/define semantics SML
2. Identify desirable properties
3. Verify properties
4. Automate verification
5. Develop dedicated tooling
6. Integrate tooling into IDE
State manager language

Example (SML)

class: $FWPART_*TOP*RPC_Chamber_CLASS$

  state: OFF

    when (($ANY$FwCHILDREN in_state ERROR) or
           ($ANY$FwCHILDREN in_state TRIPPED))
      move_to ERROR

...

action: STANDBY

  do STANDBY $ALL$RPC_HV
  do ON $ALL$RPC_LV
Processing in a state machine

**When phase**
- receive state-update
  - all guards false
- evaluating when clauses
  - command queue empty

**Action phase**
- waiting for command or state-update
  - received command
- executing statements
  - executed last statement
- emptying command queue

/ Department of Mathematics and Computer Science
state: ANALOG_ON_RED
...
when ( $ANY$TkPowerGroup not_in_state DIGITAL_ON_RED )
  move_to LVMIXED
...
state: LVMIXED
...
when ( $ANY$TkPowerGroup in_state {ON, HVMIXED} )
  move_to HVMIXED
when ( $ALL$FwCaenChannelCtrl in_state ON and
  $ALL$TkPowerGroup in_state ANALOG_ON_RED )
  move_to ANALOG_ON_RED
...
Livelocks

Example

- TkControlGroup
  - TkPowerGroup (ANALOG_ON_RED)
  - FwCaenChannelCtrl (ON)
Livelocks
Example

```
state: ANALOG_ON_RED
...
when ( $ANY$TkPowerGroup not_in_state DIGITAL_ON_RED )
  move_to LVMIXED
...
```
state: LVMIXED
...
when ( $ALL$FwCaenChannelCtrl in_state ON and
      $ALL$TkPowerGroup in_state ANALOG_ON_RED )
move_to ANALOG_ON_RED
...

Department of Mathematics and Computer Science
state: ANALOG_ON_RED

... when ( $ANY$TkPowerGroup not_in_state DIGITAL_ON_RED )
move_to LVMIXED

...
state: LVMIXED

... when ( $\text{ALL}$FwCaenChannelCtrl in_state ON and $\text{ALL}$TkPowerGroup in_state ANALOG_ON_RED ) move_to ANALOG_ON_RED

...
$\mathcal{F} + children \rightarrow \mathcal{M}$
Livelocks
Formalisation

\[ \mathcal{F} + \text{children} \rightarrow \mathcal{M} \]

- ANALOG\_ON\_RED
- ANALOG\_ON\_RED, ON
- LVMIXED
- ANALOG\_ON\_RED, ON
Lemma

\( \mathcal{F} \) contains a loop of \texttt{move\_to} actions iff \( \mathcal{M} \) contains loops
Existence of loop in $\mathcal{F}$ as satisfiability formula $\varphi_{\mathcal{F}}$: 
Livelocks
Translation to SAT

Existence of loop in $\mathcal{F}$ as satisfiability formula $\varphi_{\mathcal{F}}$:

1. state constraints
Existence of loop in $\mathcal{F}$ as satisfiability formula $\varphi_{\mathcal{F}}$:

1. state constraints
   - each FSM is always in exactly one state
   - children do not change state in when-phase

2. transition relation
Existence of loop in $\mathcal{F}$ as satisfiability formula $\varphi_{\mathcal{F}}$:

1. state constraints
2. transition relation
3. loop condition
Livelocks
Translation to SAT

Existence of loop in $\mathcal{F}$ as satisfiability formula $\varphi_{\mathcal{F}}$:

1. state constraints
2. transition relation
   - move-to steps parents can take
3. loop condition
Existence of loop in $\mathcal{F}$ as satisfiability formula $\varphi_\mathcal{F}$:

1. state constraints
2. transition relation
3. loop condition
Livelocks
Translation to SAT

Existence of loop in $\mathcal{F}$ as satisfiability formula $\varphi_{\mathcal{F}}$:

1. state constraints
2. transition relation
3. loop condition
   - parent must be able to return to its starting state
Theorem

There is a loop in $\mathcal{F}$ iff $\varphi_{\mathcal{F}}$ is satisfiable
Livelocks
Translation to SAT

Theorem

There is a loop in $\mathcal{F}$ iff $\varphi_{\mathcal{F}}$ is satisfiable

SAT encoding will find child states if loop exists!
Livelocks
Results

- Full system checking in 79 seconds
- 1302 FSMs have looping potential
- Most not observed/short lived
- Outages of control system traced back to detected problems:

...  
...

...  
...

/ Department of Mathematics and Computer Science
Reachability Sketch
Reachability Results

- Full system checked in 18 minutes
- 903 FSMs have reachability issues
- Partly due to clever programmer tricks
- Real problems typically due to copy/paste
Implementation

- Full automated translation to mCRL2
- Dedicated translations to SMT for described problems
- Integration of dedicated tools in IDE
Conclusions

- CERN: eager to improve software
- Generic solutions
- Huge system, yet effective verification
- Real-life problems detected
- Diagnostics ensure quick fixing

“We should have had these tools at the start of the LHC project” — CMS engineer
Future work

- Scale model checking techniques
- Verify larger subsystems
- Continue developing dedicated tooling
- Integrate additional tooling in IDE
- Verify standard SCADA systems
Thank you

source: http://xkcd.com/401