Using model checking to improve the quality of cyber-physical systems

Jeroen Keiren
Open University of the Netherlands & Radboud Universiteit Nijmegen

11 July 2016

@jkeiren

jeroenkeiren.nl

Open Universiteit
www.ou.nl

Cyber-Physical Systems

Cyber-Physical Systems or "smart" systems are co-engineered interacting networks of physical and computational components. – NIST

Examples

- Smart grid
- Autonomous automotive systems
- Medical monitoring
- Robotics

Networks of embedded systems with physical input and output

Open Universiteit
www.ou.nl

Modelling and analysis of cyber-physical systems

Two types of behaviour in CPS:

- Continuous dynamics
- Discrete dynamics

This talk: discrete dynamics

Case study 1: IEEE 11073-20601

INTERNATIONAL STANDARD

ISO/IEEE 11073-20601

First edition
2010-05-01

Health informatics — Personal health device communication —

Part 20601:
Application profile — Optimized exchange protocol

Informatique de santé — Communication entre dispositifs de santé personnels —

Partie 20601: Profil d'application — Protocole d'échange optimisé

Reference number
ISO/IEEE 11073-20601:2010(E)

Continua HEALTH ALLIANCE

and many more

What are personal health devices?

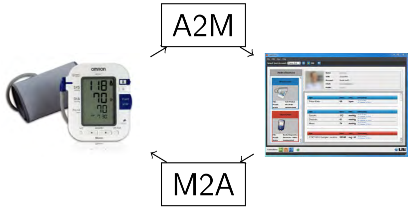
Agent devices

Manager devices

What is the purpose of IEEE 11073-20601?

"[. . .] define[. . .] a common framework for making an abstract model of personal health data available in transport-independent transfer syntax required to establish logical connections between systems and to provide presentation capabilities and services needed to perform communication tasks." [IEEE std 11073-20601]

Protocol (sketch)



Agent and manager cycle through states:

- Unassociated
- Associating
- Sending configuration
- Waiting approval
- OperatingOperating
- Disassociating

Objectives & Approach

- Assess understandability, consistency, and completeness
  - Formalise protocol in mCRL2 (ACP-like process algebra)
  - Based on standards document
- Establish correctness
  - Formulate desired properties in temporal logic (μ-calculus)
  - Use model checker to verify the properties
- Fix bugs

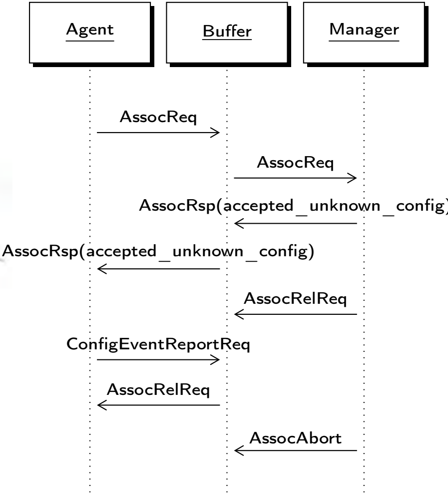
Observations on the standard

Understandability, consistency and completeness

- Requirements are not explicit
- Formalisms not introduced
- Information inconsistently duplicated between representations
  - Inconsistent terminology/abbreviations
  - Different state changes for same event
- Unexpected messages not fully treated

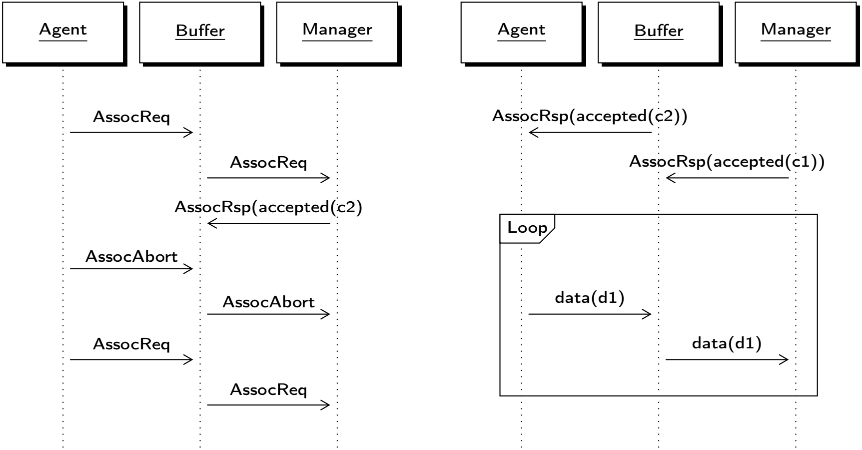
Open Universiteit
www.ou.nl

Deadlocks caused by incompleteness



Correctness

Data shall not be transmitted in inconsistent operating states.



Observations

- Omissions and inconsistencies cause easy to fix bugs
- Session setup contains a severe bug
- Agent and manager devices can transfer data in inconsistent configurations
- Rest of the protocol should be verified

Lessons learned

- Formal modelling allows detection of problems in short timespan
- Standards should provide clear requirements
- Bugs cannot be fixed without breaking standard conformance
- Formal verification should be part of the development process of every communication standard

Unverified standards contain subtle, hard to find errors!

Open Universiteit
www.ou.nl

Case study 2: CERN

source: CERN

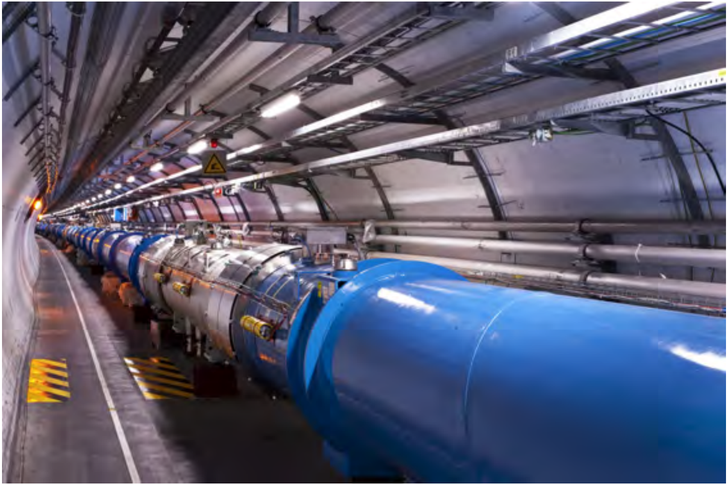
Open Universiteit
www.ou.nl

Large Hadron Collider

source: CERN

Open Universiteit
www.ou.nl

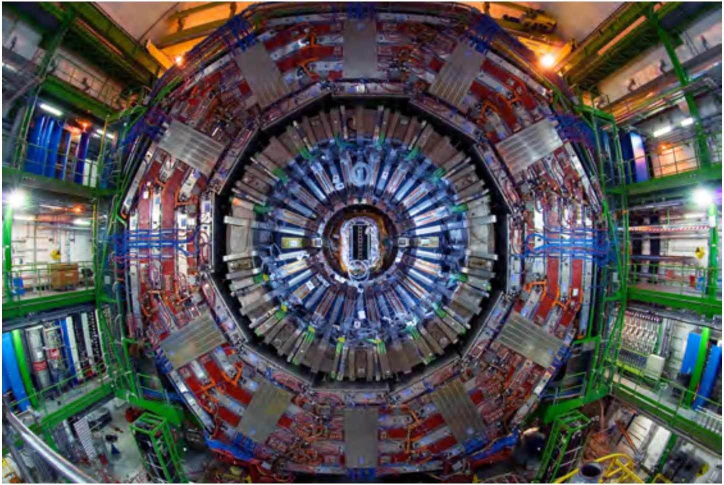
Large Hadron Collider



source: CERN

Open Universiteit
www.ou.nl

CMS



source: CERN

Open Universiteit
www.ou.nl

CMS detector: Scale

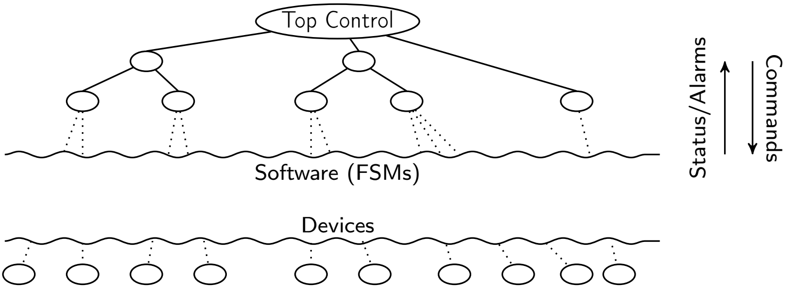
source: CERN

Control software

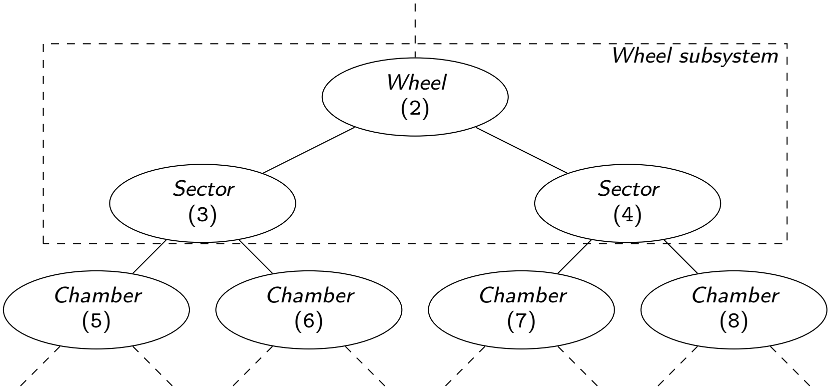
source: CERN

Open Universiteit
www.ou.nl

Control software: Global structure

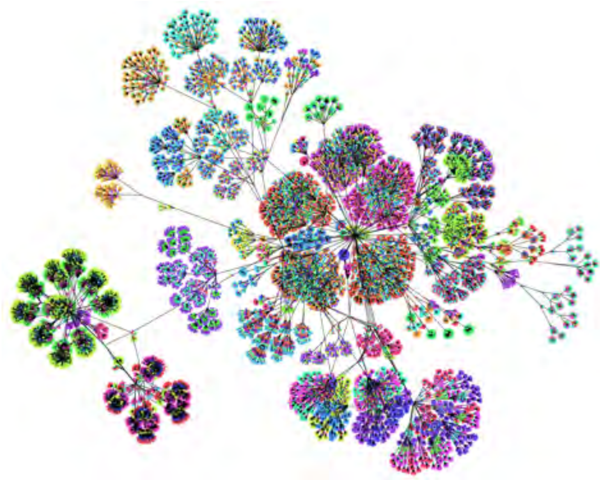


Control software: Local structure



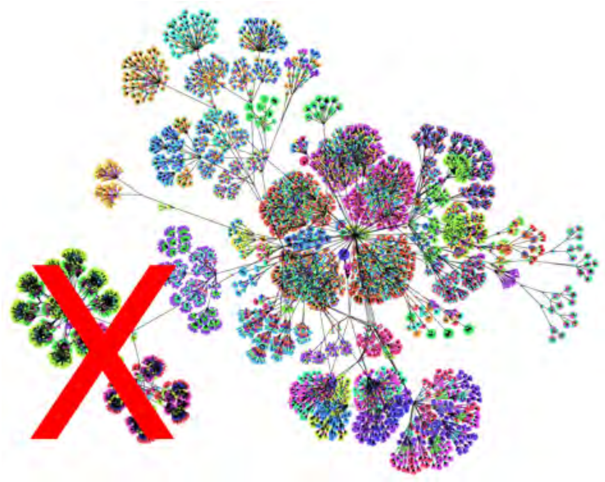
Open Universiteit
www.ou.nl

Control software: Complexity



Open Universiteit
www.ou.nl

Problem: Unresponsive subsystems



Open Universiteit
www.ou.nl

Methodology

1. Understand/define semantics SML
2. Identify desirable properties
3. Verify properties
4. Automate verification
5. Optimise verification (dedicated tooling)
6. Integrate tooling into IDE

Open Universiteit
www.ou.nl

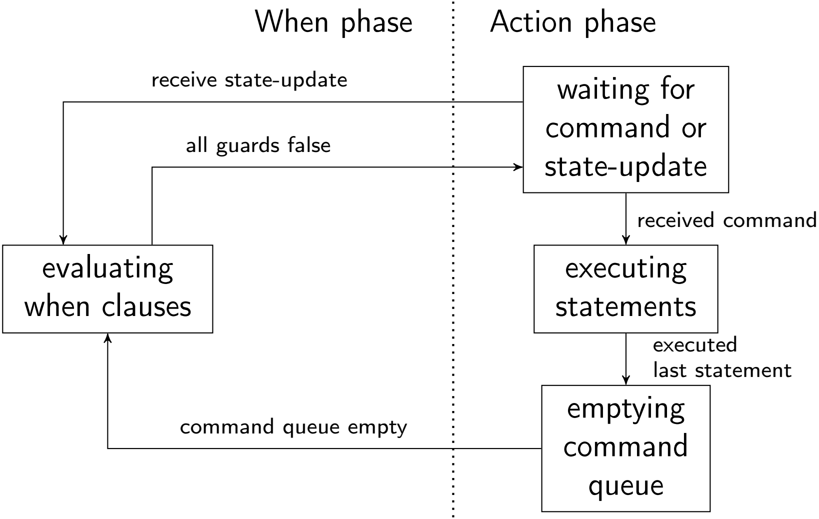
State manager language

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

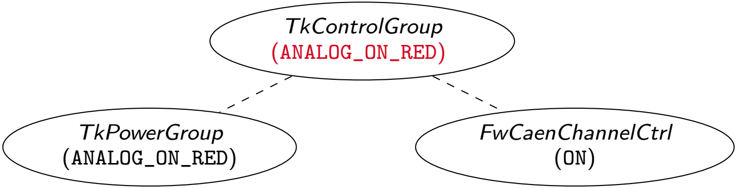


Stabilisation: Livelocks

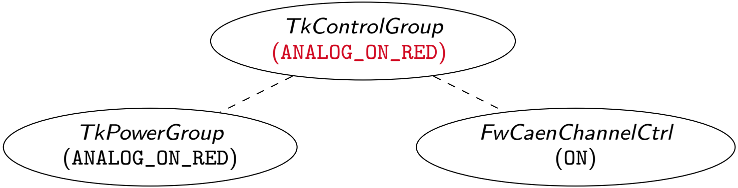
```
state: ANALOG_ON_RED
 ...
 when ( $ANY$TkPowerGroup not_in_state
        DIGITAL_ON_RED )
      move_to LVMIXED
 ...
state: LVMIXED
 ...
 when ( $ALL$FwCaenChannelCtrl in_state ON  and
        $ALL$TkPowerGroup in_state ANALOG_ON_RED )
      move_to ANALOG_ON_RED
 ...
```

Open Universiteit
www.ou.nl

Stabilisation

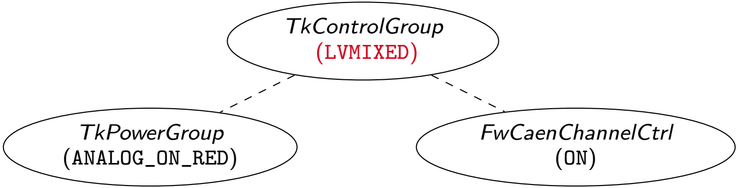


Stabilisation



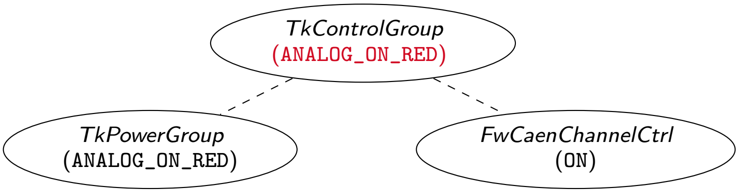

```
state: ANALOG_ON_RED
...
when ( $ANY$TkPowerGroup not_in_state
       DIGITAL_ON_RED )
     move_to LVMIXED
...
```

Stabilisation




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

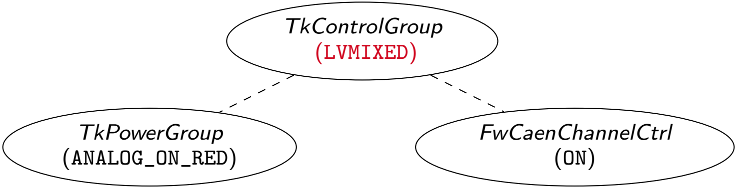
Stabilisation




```
state: ANALOG_ON_RED
...
when ( $ANY$TkPowerGroup not_in_state
       DIGITAL_ON_RED )
     move_to LVMIXED
...
```

Open Universiteit
www.ou.nl

Stabilisation

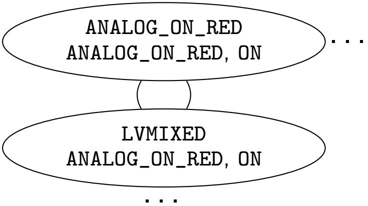



```
state: LVMIXED
 ...
 when ( $ALL$FwCaenChannelCtrl in_state ON  and
        $ALL$TkPowerGroup in_state ANALOG_ON_RED )
      move_to ANALOG_ON_RED
 ...
```

Livelocks

Formalisation

$$\mathcal{F} + children \rightarrow \mathcal{M}$$



Lemma

$\mathcal{F}$ contains a loop of $move\_to$ actions iff $\mathcal{M}$ contains loops

Livelocks

Translation to SAT

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
   - move-to steps parents can take
3. loop condition
   - parent must be able to return to its starting state

Open Universiteit
www.ou.nl

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!

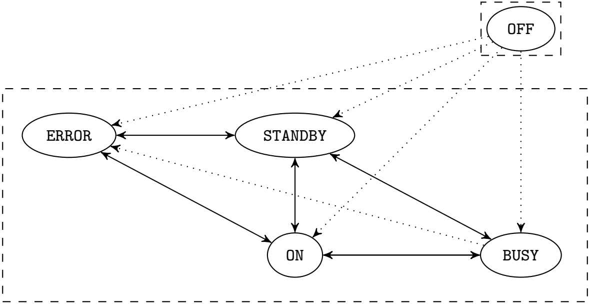
Open Universiteit
www.ou.nl

Results: Livelocks

- Full system checking in 79 seconds
- 1 302 FSMs have looping potential
- Most not observed/short lived
- Outages of control system traced back to detected problems:

```
...
6/11/2011 15:23:57 - [PIXELBARREL_BMI_S7] in state [ANALOG_ON_RED]
6/11/2011 15:23:57 - [PIXELBARREL_BMI_S7] in state [LVMIXED]
6/11/2011 15:23:57 - [PIXELBARREL_BMI_S7] in state [ANALOG_ON_RED]
...
6/11/2011 15:38:08 - [PIXELBARREL_BMI_S7] in state [ANALOG_ON_RED]
6/11/2011 15:38:08 - [PIXELBARREL_BMI_S7] in state [LVMIXED]
6/11/2011 15:38:08 - [PIXELBARREL_BMI_S7] in state [ANALOG_ON_RED]
6/11/2011 15:38:08 - [PIXELBARREL_BMI_S7] in state [LVMIXED]
...
```

Unreachable states

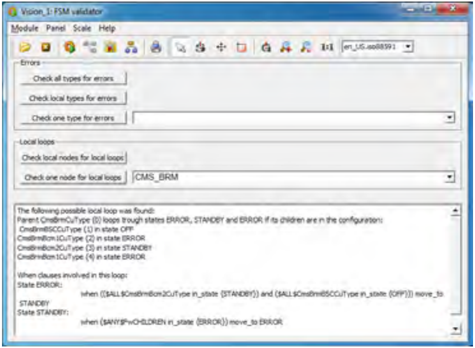


Results: Reachability

- 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



Open Universiteit
www.ou.nl

Lessons learned

- Generic tools needed to develop understanding
- Huge system, yet verifiable due to careful analysis
- Specialised tools needed for 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

Verification of Cyber Physical Systems

Cyber physical systems

Networks of embedded systems with physical input and output

- Modelling and verification well-understood
- Possible for huge systems

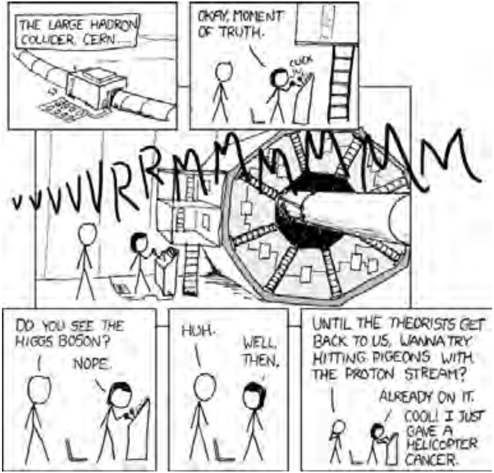
- Modelling and verification using differential equations
- Well-studied

- Combining discrete- and continuous dynamics in single formalism
- Modelling and verification using: hybrid automata
- Currently studying this with Prof. Cleaveland

Thank you

Joint work with:

- Martijn Klabbers (LaQuSo)
- Yi Ling Hwong (CERN)
- Vincent Kusters (TU/e, CERN, ETH-Z)
- Sander Leemans (TU/e, CERN)
- Tim Willemse (TU/e)



source: `http://xkcd.com/401`

Open Universiteit
www.ou.nl