Dániel Darvas (EN-ICE-PLC)

Formal verification of industrial control systems… at CERN

1st Developers@CERN Forum
29/09/2015


http://go.cern.ch/7L9h
Formal verification?

Yes, boring stuff

Academic

Just survive until the next pres :)
Context – CERN

- PLCs for controlling vacuum, cryogenics, CV, etc. systems
Context – CERN

- PLCs for controlling vacuum, cryogenics, CV, etc. systems
- Failures might have negative impact
Context – CERN

- PLCs for controlling vacuum, cryogenics, CV, etc. systems
- Failures might have negative impact
- Increasing complexity without decreasing quality?
Context – PLCs at CERN

- Programmable Logic Controllers
  robust industrial computers
Context – PLCs at CERN

- Programmable Logic Controllers
  robust industrial computers

- Small computing capacity,
  special programming languages
Context – PLCs at CERN

- Programmable Logic Controllers
  *robust industrial computers*

- Small computing capacity,
  special programming languages

- **1000+ PLCs** at CERN
Goal

- To improve the quality by eliminating bugs
  - Complementing automated and manual testing
Goal

- To **improve the quality** by eliminating bugs
  - Complementing automated and manual testing

- Apply **model checking** to find “**high quality**” bugs
Goal

- To **improve the quality** by eliminating bugs
  - Complementing automated and manual testing

- Apply **model checking** to find "**high quality**" bugs

- **Integrate** formal verification to the development process
What is formal verification?
What is formal verification?

- **Formal verification**: mathematically sound methods to check properties of specifications / implementations / …
What is formal verification?

- **Formal verification**: mathematically sound methods to check properties of specifications / implementations / …

- **Model checking**
  - Automated formal verification method
  - Checks **all possible executions** (contrarily to testing)
  - Goal: prove correctness OR **find hidden/rare problems**
What is formal verification?

- **Formal verification**: mathematically sound methods to check properties of specifications / implementations / ...

- **Model checking**
  - *Automated* formal verification method
  - Checks **all possible executions** (contrarily to testing)
  - Goal: prove correctness OR **find hidden/rare problems**
What is formal verification?

- **Formal verification**: mathematically sound methods to check properties of specifications / implementations / …

- **Model checking**
  - *Automated* formal verification method
  - Checks all possible executions (contrarily to testing)
  - Goal: prove correctness OR find hidden/rare problems
What is formal verification?

- **Formal verification**: mathematically sound methods to check properties of specifications / implementations / …

- **Model checking**
  - Automated formal verification method
  - Checks all possible executions (contrarily to testing)
  - Goal: prove correctness OR find hidden/rare problems

```
Real System (hardware, software) → Formal Model → Model checker
                                    ^
                                    v
                                    satisfied not satisfied

                                         √

                                    Formal Requirement → Counterexample
```

---

CERN

MÜEGYETEM 1782
Testing vs. model checking

Testing
Testing vs. model checking

Testing

\[ \text{add}(5, 3) = 8 \]
Testing vs. model checking

**Testing**

Inputs are known, outputs are checked

\[
\text{add}(5, 3) = 8
\]
Testing vs. model checking

**Testing**

```
+  
5 3
?  
```

\[ \text{add}(5, 3) = 8 \ ? \]

- **Inputs are known**, outputs are checked

**Model checking**

```
+  
?  ?
```

\[ \text{add}(\ast, \ast) < 0 \ ? \]
Testing vs. model checking

**Testing**

```
5 + 3
```

add(5, 3) = 8 ?

- Inputs are known, outputs are checked

**Model checking**

```
? + ?
```

add(*, *) < 0 ?

- E.g. the possibility of an **output combination** is checked.
- Can be used in other ways too.
Usage of formal verification
Usage of formal verification

- Used both in **industry** and **academia**
  - typically when the *cost of failure is high*
Usage of formal verification

- Used both in **industry** and **academia**
  - typically when the *cost of failure is high*
Usage of formal verification

- Used both in **industry** and **academia**
  - typically when the *cost of failure is high*

- Formal verification for **PLCs**
  - mostly in academic environment
  - not widely spread yet in industry – **too difficult!**
Challenges and answers

Real System (hardware, software)

Formal Model

Formal Requirement

Specifications

Model checker

satisfied

Counter-example

not satisfied
Challenges and answers

How to get models?

Real System (hardware, software)

Formal Model

Specifications

Formal Requirement

Model checker

satisfied

not satisfied

Counter-example
Challenges and answers

How to get models?

Automated generation

Real System (hardware, software)

Specifications

Formal Model

Formal Requirement

Model checker

satisfied

Counter-example

not satisfied
Challenges and answers

How to get models?
Automated generation

Real System (hardware, software)

Formal Model

Specifications

How to formalize requirements?

Formal Requirement

Model checker

satisfied

Counter-example

not satisfied
Challenges and answers

How to get models?
Automated generation

Formal Model

Real System (hardware, software)

Formal Requirement

Specifications

How to formalize requirements?
Requirement patterns

Model checker

satisfied

Counter-example

not satisfied
Challenges and answers

How to get models?
Automated generation

How to formalize requirements?
Requirement patterns

Which model checker should be used?

Real System (hardware, software)

Formal Model

Formal Requirement

Specifications

How to get models?
Automated generation

Which model checker should be used?

Model checker

Satisfied

Counter-example

Not satisfied
Challenges and answers

How to get models?
- Automated generation

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to formalize requirements?
- Requirement patterns

Real System
(hardware, software)

Specifications

Formal Model

Formal Requirement

Model checker

satisfied

not satisfied

Counter-example

Specifications

How to formalize requirements?
Challenges and answers

How to get models?
- Automated generation

How to formalize requirements?
- Formal model
- Formal requirement

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

How to make it efficient?
- Counter-example

Formal model

How to get models?
- Automated generation

How to formalize requirements?
- Formal model
- Formal requirement

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
- Automated generation

Model checker

How to formalize requirements?
- Requirement patterns

Which model checker should be used?
- Multiple (general meth., intermediate model)

How to make it efficient?
- Counter-example

Real System
(hardware, software)

Specifications

How to formalize requirements?
- Requirement patterns

How to get models?
Challenges and answers

- How to get models?
  - Automated generation

- Which model checker should be used?
  - Multiple (general method, intermediate model)

- How to formalize requirements?
  - Requirement patterns

- How to make it efficient?
  - Reductions

- How to get models?
  - Formal Model

- How to formalize requirements?
  - Formal Requirement

- Model checker
  - satisfied
  - not satisfied

- Counter-example

- Specifications

Real System (hardware, software)
Challenges and answers

- How to get models?
  - Automated generation

- Which model checker should be used?
  - Multiple (general meth., intermediate model)

- How to formalize requirements?
  - Requirement patterns

- How to proceed with a counterexample?
  - Reductions

- How to make it efficient?
  - Counterexample

- How to get models?
  - Formal Model

- How to formalize requirements?
  - Formal Requirement

- How to proceed with a counterexample?
  - Model checker

Real System (hardware, software)
Challenges and answers

- How to get models?
  - Automated generation

- Which model checker should be used?
  - Multiple (general meth., intermediate model)

- How to formalize requirements?
  - Requirement patterns

- How to make it efficient?
  - Analysis and demonstration

- How to proceed with a counterexample?
  - Reductions

Real System (hardware, software)

Specifications

Formal Model

Formal Requirement

Model checker

satisfied

not satisfied

Counter-example
Challenges and answers

How to get models?
Automated generation

Which model checker should be used?
Multiple (general meth., intermediate model)

How to formalize requirements?
Requirement patterns

How to proceed with a counterexample?
Analysis and demonstration

And it should be integrated to the development process.

How to make it efficient?
Reductions

Formal Model

Real System (hardware, software)

Specifications

Model checker

Counter-example

How to get models?

How to formalize requirements?

How to proceed with a counterexample?

And it should be integrated to the development process.
Model checking (extended workflow for PLCs)

- PLC code (ST, SFC, IL)
- Formal model
- Requirement patterns
- Formal requirement
- Model checker
- Satisfied
- Not satisfied
- Counter-example
- Verification report

Reductions
Model checking (extended workflow for PLCs)

- PLC code (ST, SFC, IL)
- Requirement patterns
- PLC-specific verification tool
- Verification report
Model checking (extended workflow for PLCs)

PLC code (ST, SFC, IL)

Requirement patterns

Verification report

- PLC code (ST, SFC, IL)
- Requirement patterns
- Verification report
Model checking in practice (at CERN)
The PLCverif tool
The PLCverif tool

Eclipse-based **editor** for PLC programs
The PLCverif tool

Defining verification cases (requirement, fine-tuning, etc.)

No model checker-related things or temporal logic logic expressions
The PLCverif tool

PLCverif — Verification report

Generated at Mon Jul 07 15:10:22 CEST 2014 | PLCverif v2.0.1 | (C) CERN EN-ICE-PLC | Show/hide expert details

ID:  Demo001
Name:  If A is false, C cannot be true.
Description:  If A is false, C cannot be true. As this function block models an AND-gate, if any of the inputs (A or B) is false, the output should be false too.

The requirement is based on the documentation of the function block and the following Jira case:
https://icecontrols.its.cern.ch/jira/browse/UCPC-1111

Source file:  DemoSource.scl
Requirement:  3. A = false & C = true is impossible at the end of the PLC cycle.
Result:  Not satisfied

Tool:  nusmv
Total runtime (until getting the verification results):  212 ms
Total runtime (incl. visualization):  361 ms

Counterexample

<table>
<thead>
<tr>
<th>Variable</th>
<th>End of Cycle 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input a</td>
<td>FALSE</td>
</tr>
<tr>
<td>Input b</td>
<td>TRUE</td>
</tr>
<tr>
<td>Output c</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Click-button verification, verification report with the analysed counterexample
Example – SM18 safety system
Goal: ensuring safety by allowing/forbidding tests
Goal: ensuring safety by allowing/forbidding tests

Core:

selected test  →  SM18 PLCSE
switch statuses  →  safety logic
current voltages  →  test allowed
cryo conditions  →  

© CERN
**Goal:** ensuring **safety** by allowing/forbidding tests

**Core:**

- selected test
- switch statuses
- current voltages
- cryo conditions

Safety-critical, can be dangerous

**SM18 PLCSE safety logic**

test allowed
Model checking workflow for SM18

- PLC code / Formal model
- Requirement
- Model checker

- Satisfied
- Not satisfied

- Counter-example
Model checking workflow for SM18

PLC code / Formal model

Requirement

Model checker

Satisfied

Counter-example

Not satisfied
Model checking workflow for SM18

- SM18 PLCSE Safety Logic
- PLC code / Formal model
- Requirement

- Satisfied
- Not satisfied

- E.g. Test starts only if cryo conditions are OK
- Counter-example

- Satisfied
  - ✓
SM18 PLCSE Safety Logic

PLC code / Formal model

Model checker

Satisfied

E.g. Test starts only if cryo conditions are OK

Requirement

Black magic inside

Counter-example

Not satisfied
Model checking workflow for SM18

- SM18 PLCSE Safety Logic
- PLC code / Formal model
- Requirement
- Model checker
- Example inputs to violate the requirement

E.g. Test starts only if cryo conditions are OK

Satisfied
- ✔

Not satisfied
- Counter-example
SM18 PLCSE safety logic

selected test
switch statuses
current voltages
cryo conditions

test allowed
Excerpt from the work of R. Speroni
Problems found *(before putting in production!)*
Problems found (before putting in production!)

Requirement misunderstanding
- Recognised while specifying requirements
Problems found *(before putting in production!)*

Requirement misunderstanding
- Recognised while specifying requirements

Functionality problems
- “The [magnet] test should start, but it doesn’t.”
Problems found *(before putting in production!)*

**Requirement misunderstanding**
- Recognised while specifying requirements

**Functionality problems**
- “The [magnet] test should start, but it doesn’t.”

**Safety problems**
- “The [magnet] test **should NOT start**, but it does.”
Problems found *(before putting in production!)*

**Requirement misunderstanding**
- Recognised while specifying requirements

**Functionality problems**
- “The [magnet] test should start, but it doesn’t.”

**Safety problems**
- “The [magnet] test **should NOT start**, but it does.”
- Some really hidden:
  - **65536 input combinations** for 1 magnet test scenario start should be allowed in 2 of them
Verification workflow in practice
Verification workflow in practice

So you are using Jenkins

Tell me how great is to automatically execute a job on each commit

Based on still from Willy Wonka & the Chocolate Factory,
Source: memegenerator.net
Summary

- “Formal verification is not relevant to industry.” FALSE!
Summary

- “Formal verification is not relevant to industry.” FALSE!

- First steps to apply formal verification to PLCs
  - Interesting bugs found (with joint effort)
  - Critical parts can be checked
  - Complementary to testing
Summary

- "Formal verification is not relevant to industry." FALSE!

- First steps to apply formal verification to PLCs
  - Interesting bugs found (with joint effort)
  - Critical parts can be checked
  - Complementary to testing

- Still long way to go
  - Improving the performance
  - Formal specification

http://go.cern.ch/7L9h
http://cern.ch/plcverif
Formal verification?
Yes, boring stuff
Academic
Just survive until the next pres :)
Formal verification?

Yes, boring stuff

Academic

Just survive until the next pres :)

Formal verification is great!
Formal verification?
Yes, boring stuff
Academic
Just survive until the next pres :)
Formal verification is great!
Well... At least now it's over.