Finding bugs in the LHC: Verification methods for PLC programs

Alpine Verification Meeting
18–21/09/2017, Visegrád

Contains joint work with B. Fernández, E. Blanco, Gy. Sallai, I. Majzik, R. Speroni, M. Lettrich

http://go.cern.ch/8gsR
Programmable Logic Controllers

- Crucial parts of industrial control systems

- PLCs: robust industrial computers for control tasks

- Special domain
  - Special programming languages
  - Different background knowledge
  - Smaller user base
Special and numerous PLC languages

**ST**

```
IF NOT (x = TRUE OR y = FALSE) THEN
    r1 := TRUE;
END_IF;
```

```
r2 := (a >= b);
```

**IL (Siemens)**

```
A ( O x
ON Y )
NOT r1
S
L a
L b
>=$((r2)
```

**LD**

```
x
NOT (S)
```

```
Y
```

```
>=
IN1
IN2
```

```
r1
```

```
r2
```
We need good quality!

- **1000+ PLCs** at CERN
- **Critical** and/or **expensive** systems are operated

Cryogenics  
Vacuum  
Gas  
Detector control

Better quality  
Higher availability

Reasonable effort  
development, training

Photos: © CERN
We are working with prototypes anyways

- Not everything can be anticipated

66kV transformer short

- At around 5:30 Friday morning a marten (“Martes foina”) jumped on the transformer, creating an electrical arc on an 18kV pole.
- The air ionisation expanded and jumped to other 18kV poles and finally the 66kV.
- The 66kV switches opened as expected.
- The short circuit perturbed briefly the whole 66kV CERN network.

“How to Survive a UFO Attack”
B. Auchmann, J. Ghini, L. Grob, G. Iadarola, A. Lechner, G. Papotti
Evian, December 15, 2015
State of the art:
Verification of PLC programs
Options for PLC program verification

Subjective list of possible verification methods:

- Static code analysis *(code smells, …)*
- Testing *(module tests, integration tests, …)*
- Formal verification *(model checking)*
State of the practice

- Lack of static analysis tools
- Old development environments

High FP ratio, many missed problems
State of the practice

- Practically the only verification method used

- **ISA/ANSI/IEC 62381:**
  - **FAT:** Factory Acceptance Test
  - **SAT:** Site Acceptance Test

- Typically **no module testing, no specific tool support**

- (Module testing required for SIL1+ safety systems)
State of the practice
State of the practice
PLCverif:
New verification methods for PLCs at CERN
Goals (originally)
What do we want?

Formal verification!

How do we want it?

Without much effort
Without understanding model checkers
Without writing CTL/LTL formulae
Without using command line interface
Without reading the manual
Quickly
Goals

**Lightweight verification**
*Keep it simple, stupid and usable*

**Flexible tool**
*Model checking, testing, static analysis*

**Reusable language infrastructure**
*Lowering the entry barrier*

**Reduced development effort**
*Integrating external tools*
Overall idea

Lightweight verification
Flexible tool

Reusable lang. infrastructure
Reduced development effort

PLC code
Requirements
Formal model
External verification tools
Verification report
Overall idea

Lightweight verification
Flexible tool

Reusable lang. infrastructure
Reduced development effort

PLC code

Formal model

Static analysis
External verification tools
Verification report

Requirements

Visualization
Overall idea

Lightweight verification
Flexible tool

Reusable lang. infrastructure
Reduced development effort

PLC code

Requirements

Formal model

Static analysis

External verification tools

Verification report

Visualization
Overall idea

Lightweight verification
Flexible tool

Reusable lang. infrastructure
Reduced development effort

Formal model

Static analysis
nuXmv
UPPAAL
CBMC
JUnit

Verification report

Visualization

Reduction

STL code
SCL code
...

Req. pattern
Assertions
...

PLCverif features

- Helps **unit** and **regression** testing

![Diagram](image)

**Testing**

Work done together with Gyula Sallai
PLCverif features

- PLC code
- Requirement patterns
- Control Flow Automata
- Support for multiple model checkers
- ITS-GAL representation
- nuXmv representation
- Report
- Fixed English sentences with gaps to fill
- Making verification feasible
- Human-readable verification report
- Full automated workflow

Read more: D. Darvas et al. PLCverif: A tool to verify PLC programs based on model checking techniques. ICALEPCS 2015. doi: 10.18429/JACoW-ICALEPCS2015-WEPGF092
Example results
Results

- Experimental phase, very preliminary results

- Issues found in our **well-established library**
  - *Read, but not written variables*
  - *Incorrect logic expressions*

```
IF (((E_MOffR AND (MMoSt OR FoMoSt OR SoftLDSt))
    OR (AuOffRSt AND AuMoSt)
    OR (LDSt AND PHLDCmd AND HOffRSt)
    OR (FE_PulseOn AND PPulse AND NOT POutOff) AND EnRstartSt)
    OR (E_FuStopI AND NOT PFsPosOn)) THEN

MOnRSt := FALSE;
```

Missing parentheses around AND expression nested in OR: `(FE_PulseOn AND PPulse AND NOT POutOff) AND EnRstartSt`.
Results

- Automated unit testing using **Jenkins** for our base object library
  - Formal verification with **CBMC**, **nuXmv**, … too

![Test Result: (root)](image)

- **Test Result Trend**
  - **Expected**: /OnOff-Siemens/case3_expected.csv
  - **Actual**: /OnOff-Siemens/case3_output.csv

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Duration</th>
<th>Fail</th>
<th>Skip</th>
<th>Pass</th>
<th>(diff)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB_HYST</td>
<td>0 ms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>OnOff-Siemens</td>
<td>0 ms</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>TON</td>
<td>0 ms</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Total differences: 7
SO YOU ARE USING JENKINS

TELL ME HOW GREAT IT IS TO AUTOMATICALLY EXECUTE A JOB ON EACH COMMIT
Results

- Model checking is **more and more accepted and used** at CERN
  - Of course, **performance** may still be a problem
  - **Environment modelling** is a challenge too

- **Use cases**
  - **Library of base blocks**
    - Deeply hidden problems found in code used for 10+ years in production, in 200+ PLCs
  - **Various magnet testing safety controllers**
    - Several safety issues found
  - **ITER collaboration**
    - Ongoing verification of a critical communication protocol’s implementation for their fusion reactor

Read more: D. Darvas et al. *Formal verification of safety PLC based control software*. iFM 2016. doi: 10.1007/978-3-319-33693-0_32
Results – tooling

- **SCL code** and **verification case editor**
- **One-click verification**
- **Multiple model checkers** under hood
- **Verification report**


http://cern.ch/plcverif
Future work

What is next?

- More development of PLCverif (KT-funded project)
  - Goal: “Making the tool production-ready”
  - More stable, more generic, more open

- Integrating automated unit testing in the real development workflow

- Analysis of static analysis rules
Conclusion

- **Static analysis**: preliminary work, but very promising
- **Testing**: dedicated support for automated unit testing
- **Model checking**: often feasible, requiring acceptable resources and knowledge

- Big impact on PLC verification by introducing **lightweight verification methods**
- **Industrial application** of model checking is interesting, desired and feasible

- Important to **specifically target usage domains**
http://go.cern.ch/8gsR

Get the presentation!
Model checking at CERN


- D. Darvas et al. **Formal verification of safety PLC based control software.** Integrated Formal Methods (LNCS 9681), pp. 508-522, Springer, 2016. [http://doi.org/10.1007/978-3-319-33693-0_32](http://doi.org/10.1007/978-3-319-33693-0_32)