



Dániel Darvas (EN-ICE-PLC)

Formal verification of industrial control systems... at CERN

1st Developers@CERN Forum

29/09/2015

Contains joint work of B. Fernández, E. Blanco, S. Bludze, J.O. Blech, J-C. Tournier, T. Bartha, A. Vörös, I. Majzik, R. Speroni, M. Lettrich



<http://go.cern.ch/7L9h>



Source: <http://www.iphonetextgenerator.com/>

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



© Siemens AG 2014,
All rights reserved

Context – PLCs at CERN

- Programmable Logic Controllers
robust industrial computers
- Small computing capacity,
special programming languages



© Siemens AG 2014,
All rights reserved

Context – PLCs at CERN

- Programmable Logic Controllers
robust industrial computers
- Small computing capacity,
special programming languages
- **1000+ PLCs at CERN**



© Siemens AG 2014,
All rights reserved

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**

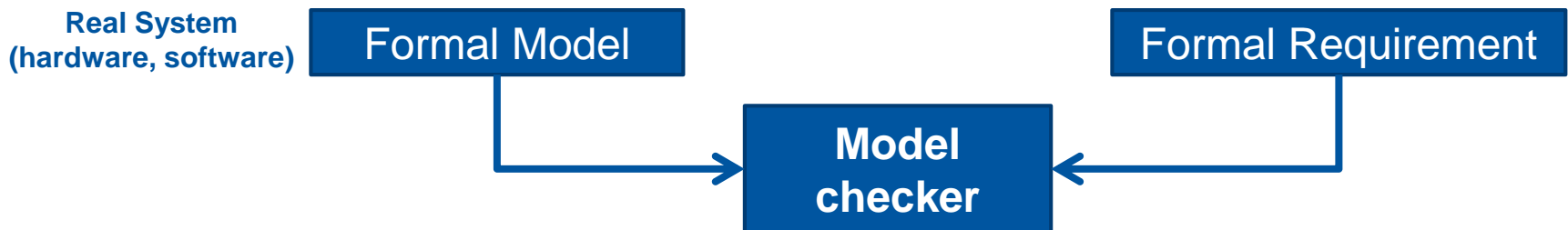
Real System
(hardware, software)

Formal Model

Formal Requirement

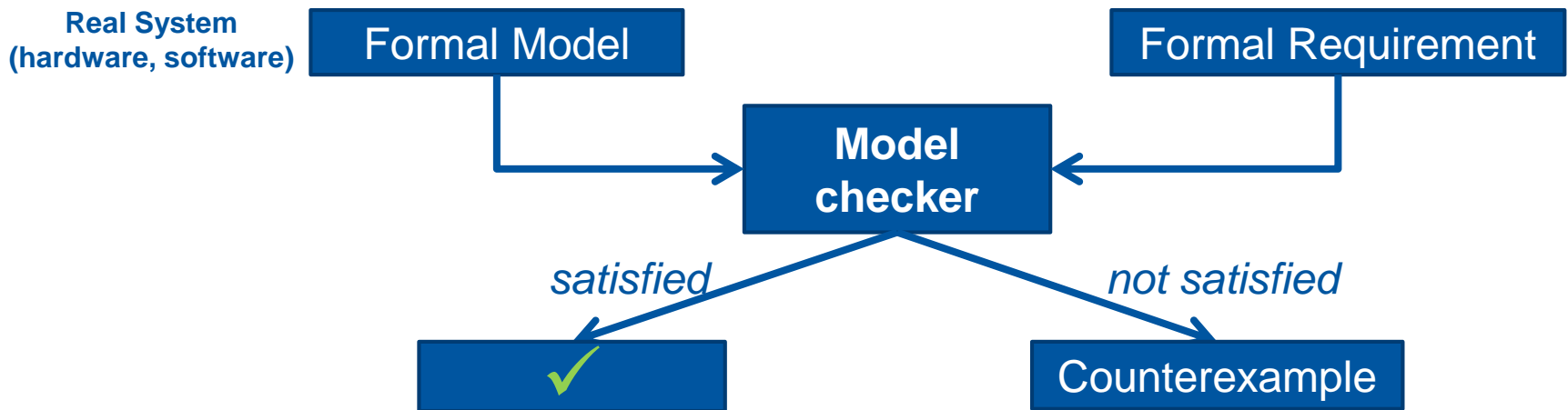
What is formal verification?

- **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**



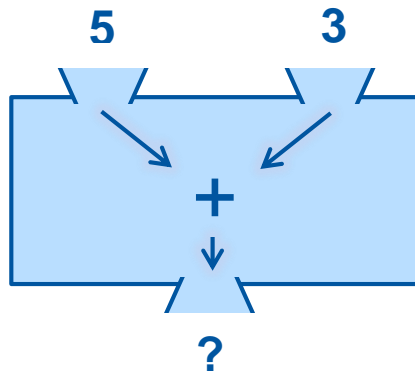
Testing vs. model checking

Testing



Testing vs. model checking

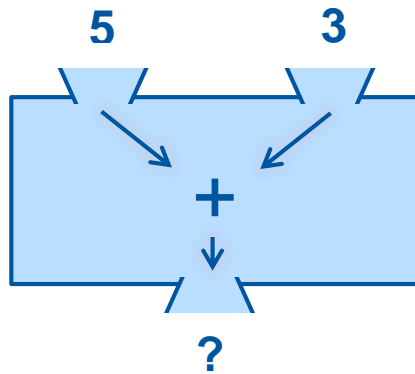
Testing



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

Testing vs. model checking

Testing

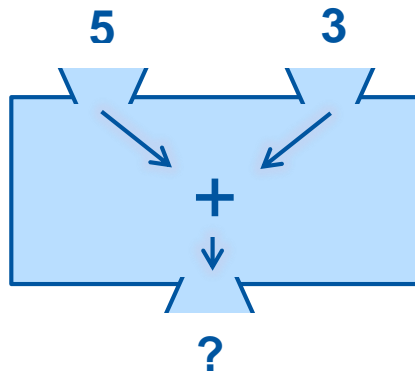


$\text{add}(5, 3) = 8 \ ?$

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

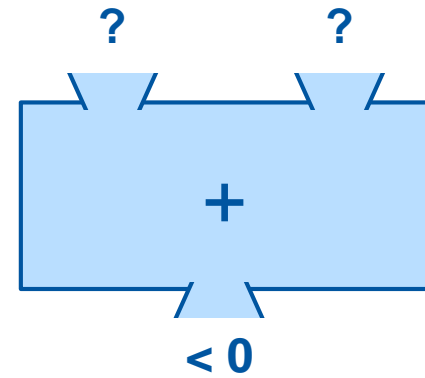
Testing vs. model checking

Testing



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

Model checking

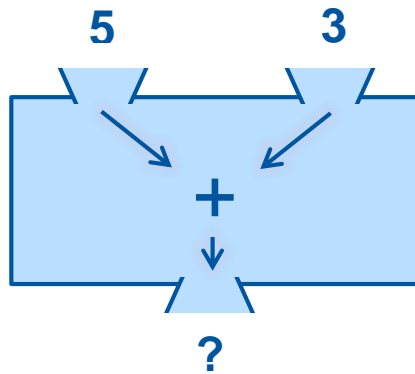


$\text{add}(*, *) < 0 \ ?$

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

Testing vs. model checking

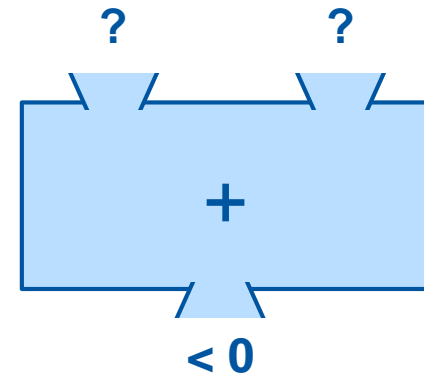
Testing



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

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

Model checking



$\text{add}(*, *) < 0$?

- 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*



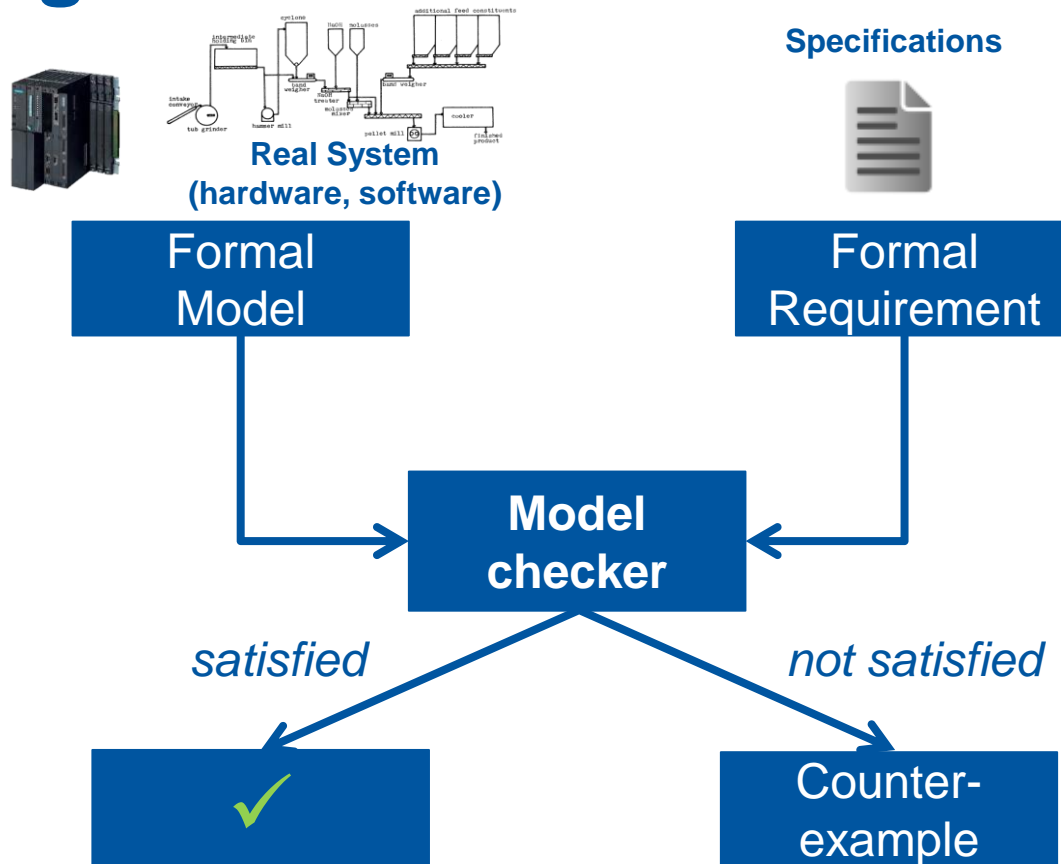
Usage of formal verification

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



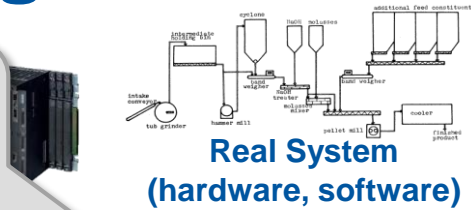
- Formal verification for **PLCs**
 - mostly in academic environment
 - not widely spread yet in industry – **too difficult!**

Challenges and answers



Challenges and answers

How to get models?



Specifications



Formal Model

Formal Requirement

Model checker

satisfied

not satisfied

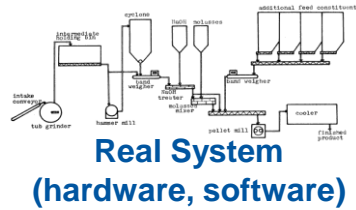


Counter-example

Challenges and answers

How to get models?

Automated generation



Specifications



Formal Model

Formal Requirement

Model checker

satisfied

not satisfied

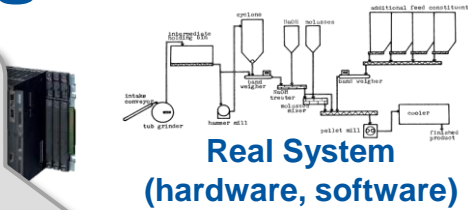


Counter-example

Challenges and answers

How to get models?

Automated generation



Formal Model

Specifications



How to formalize requirements?

Formal Requirement

Model checker

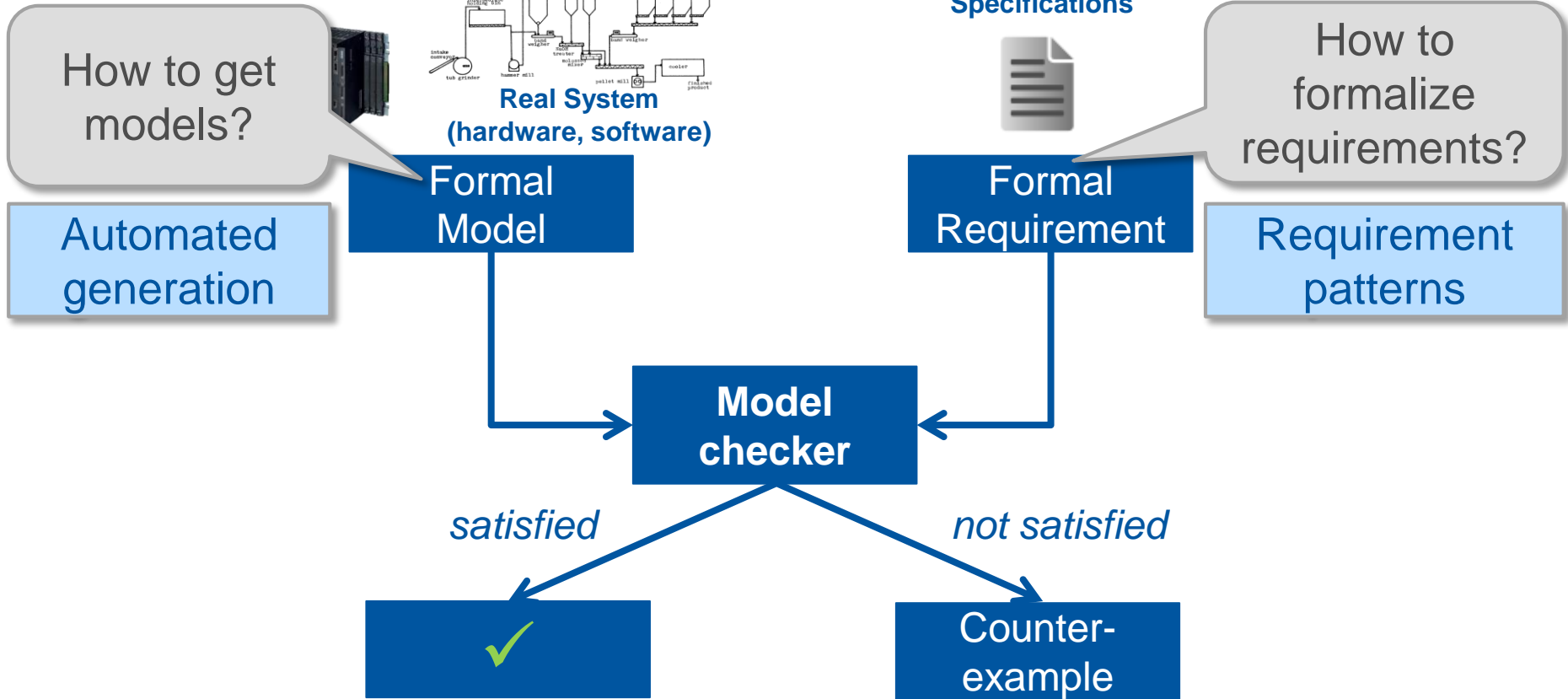
satisfied

not satisfied

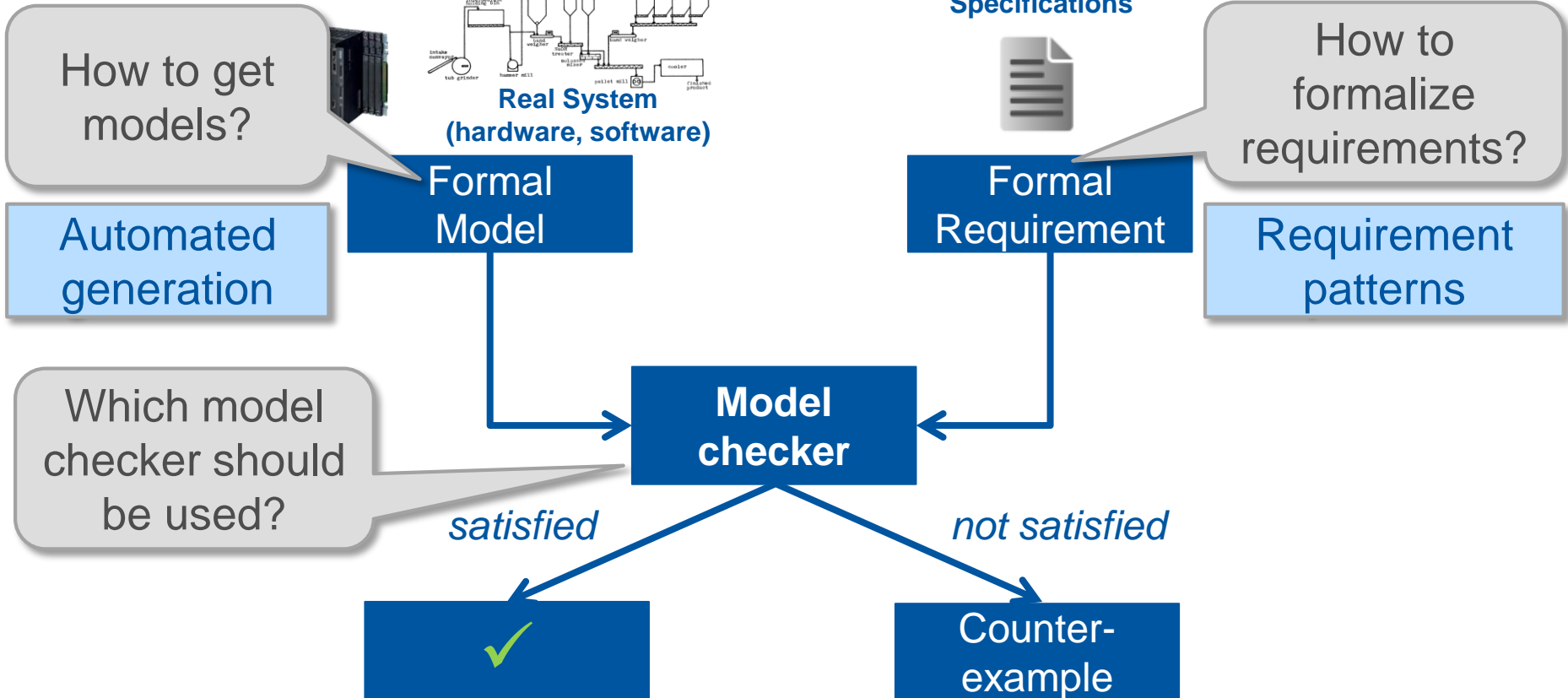


Counter-example

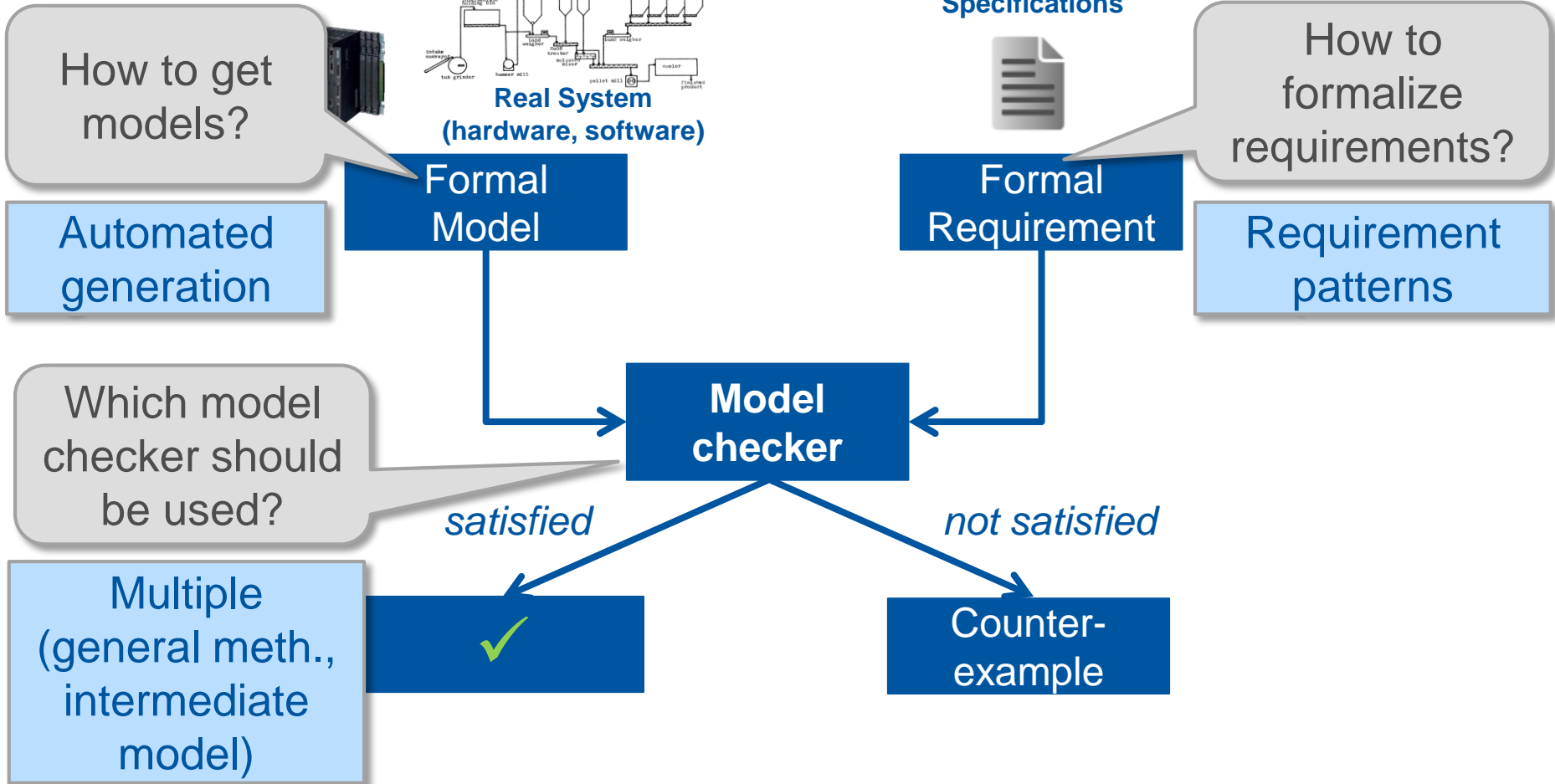
Challenges and answers



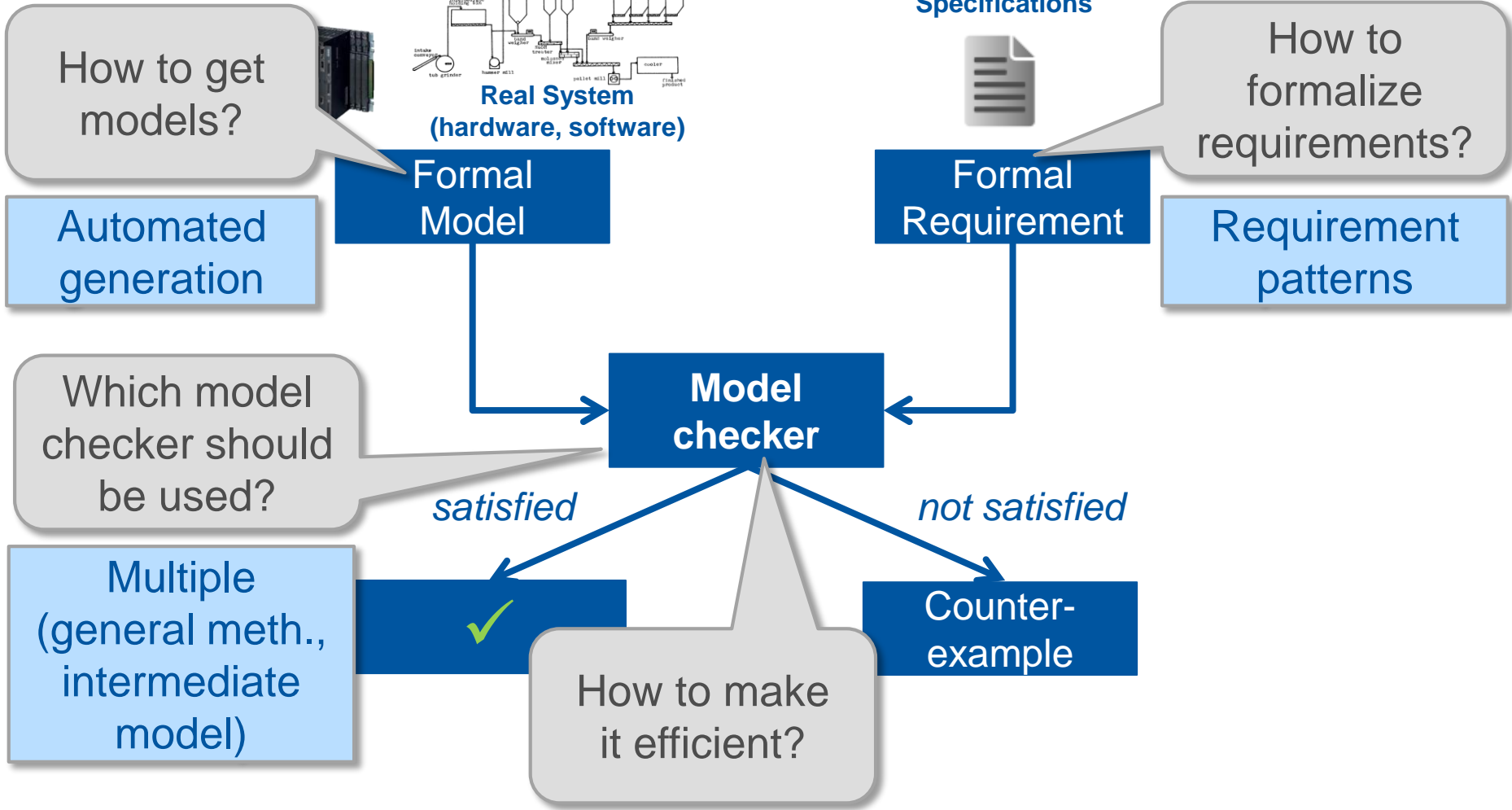
Challenges and answers



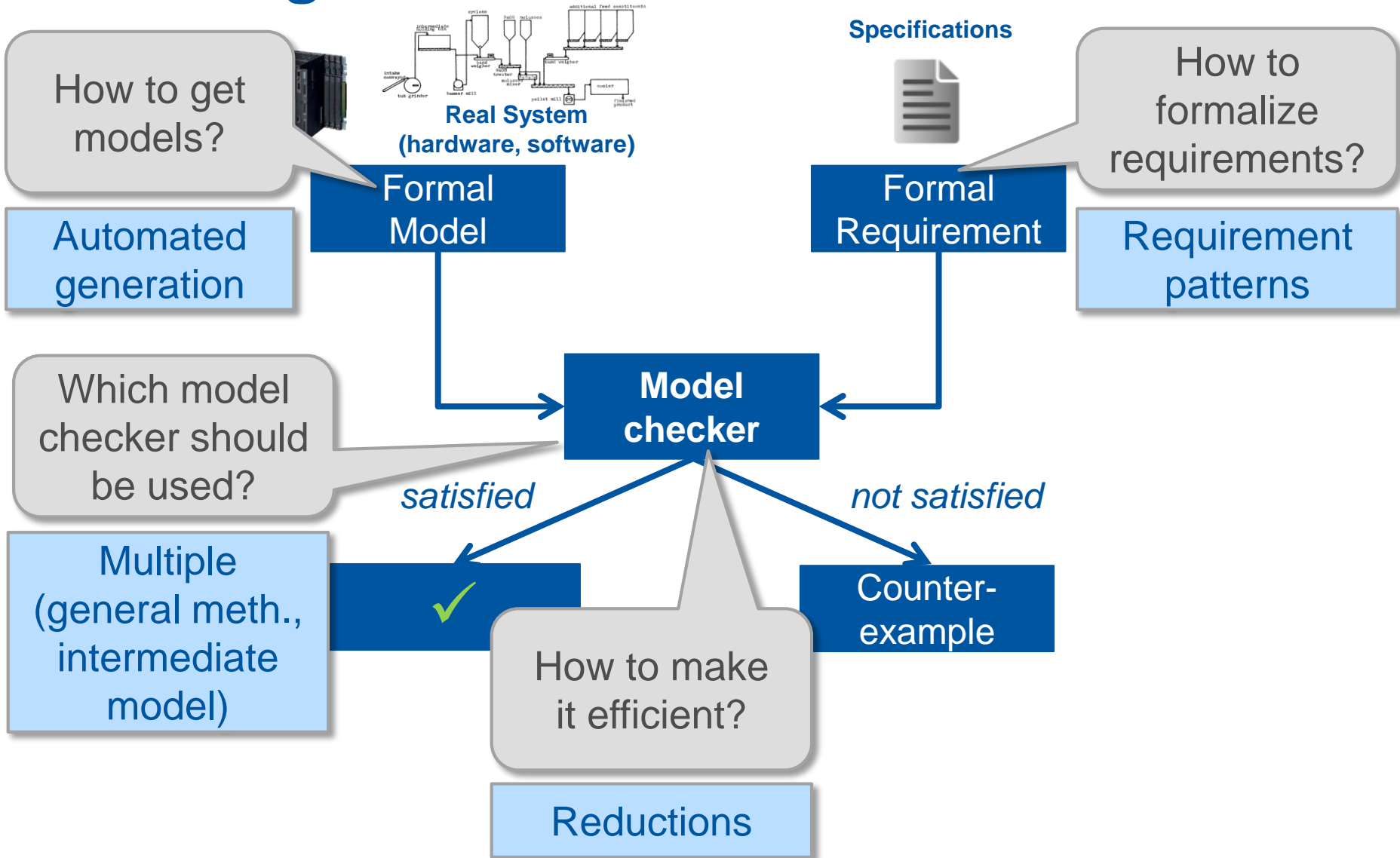
Challenges and answers



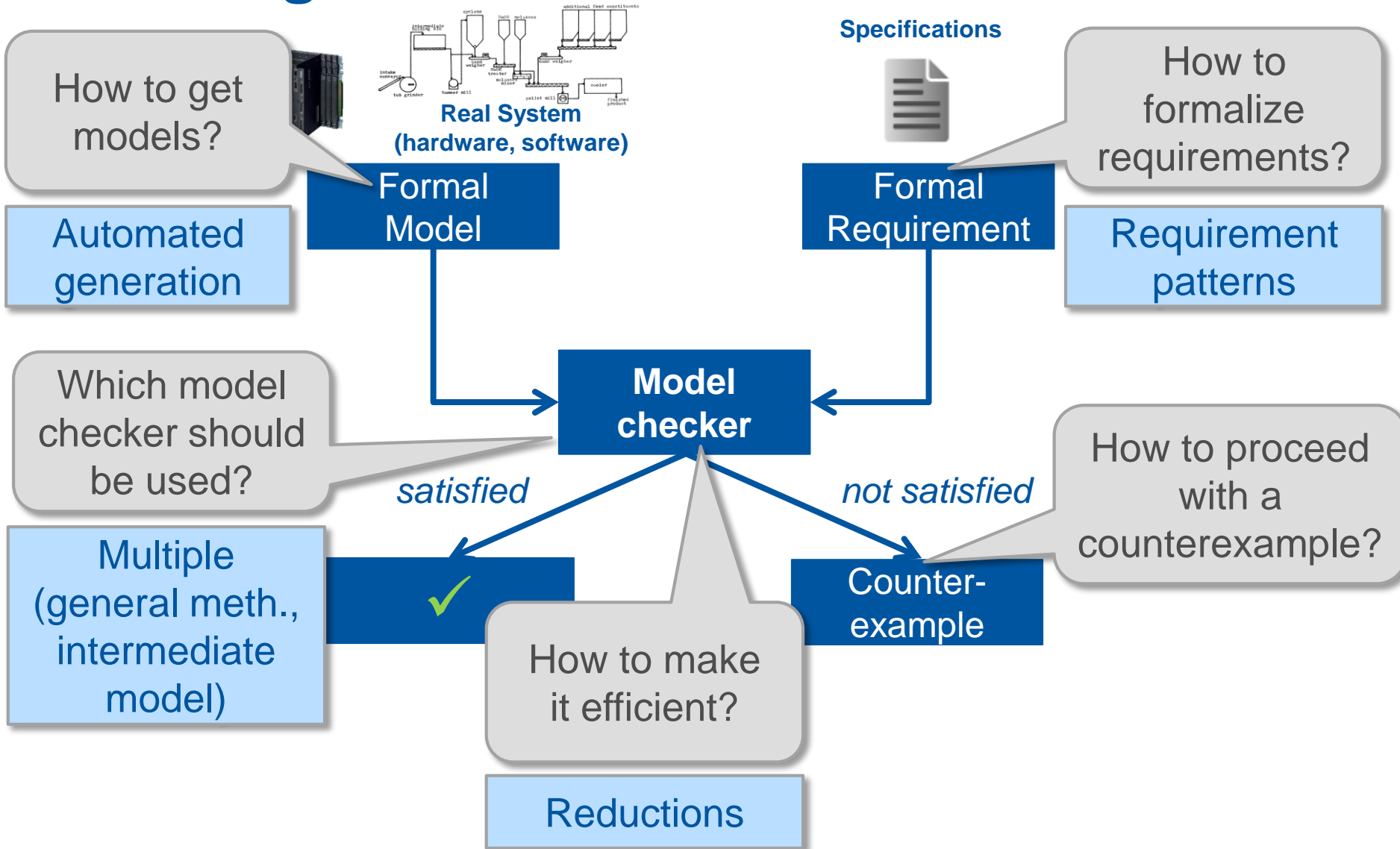
Challenges and answers



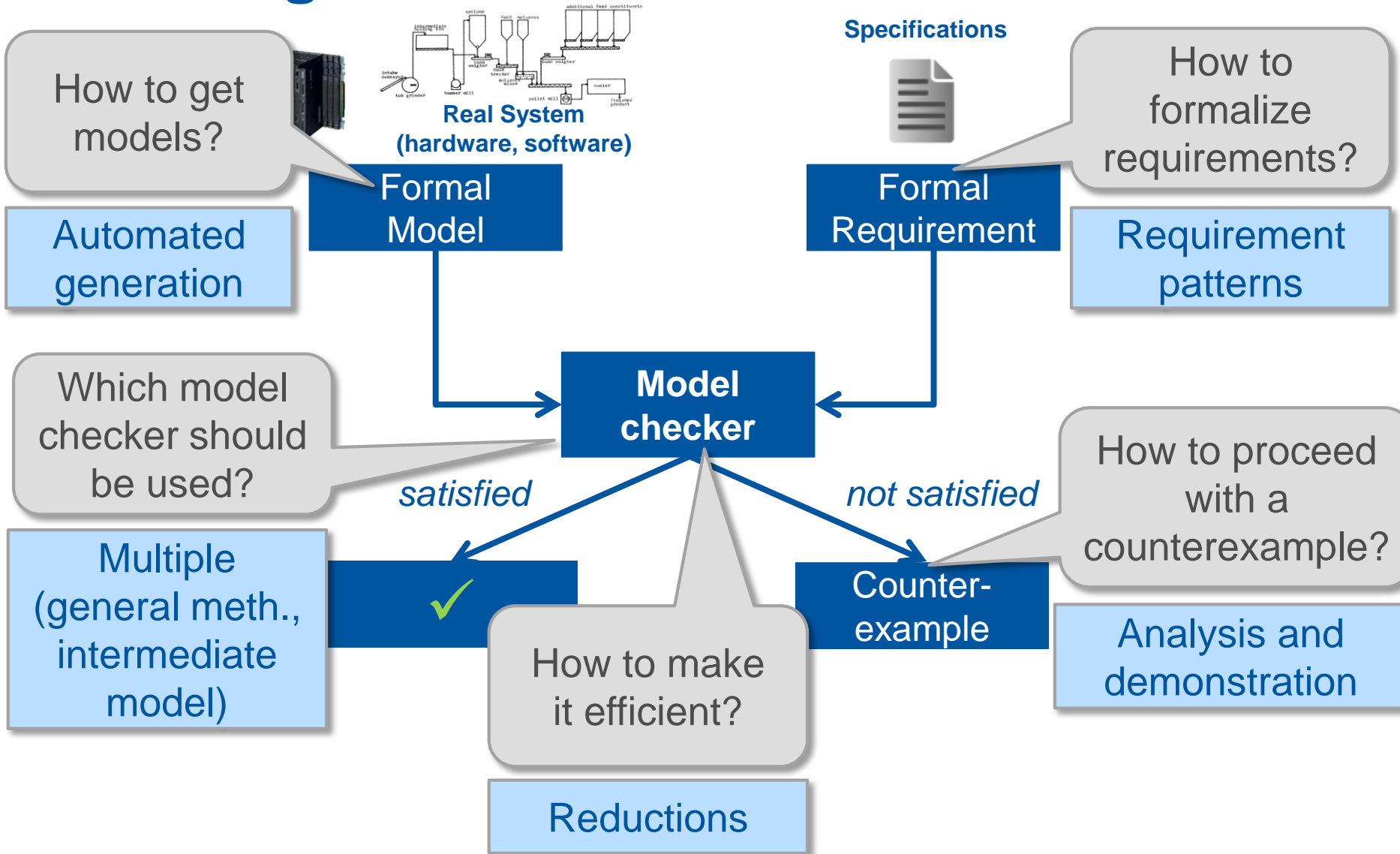
Challenges and answers



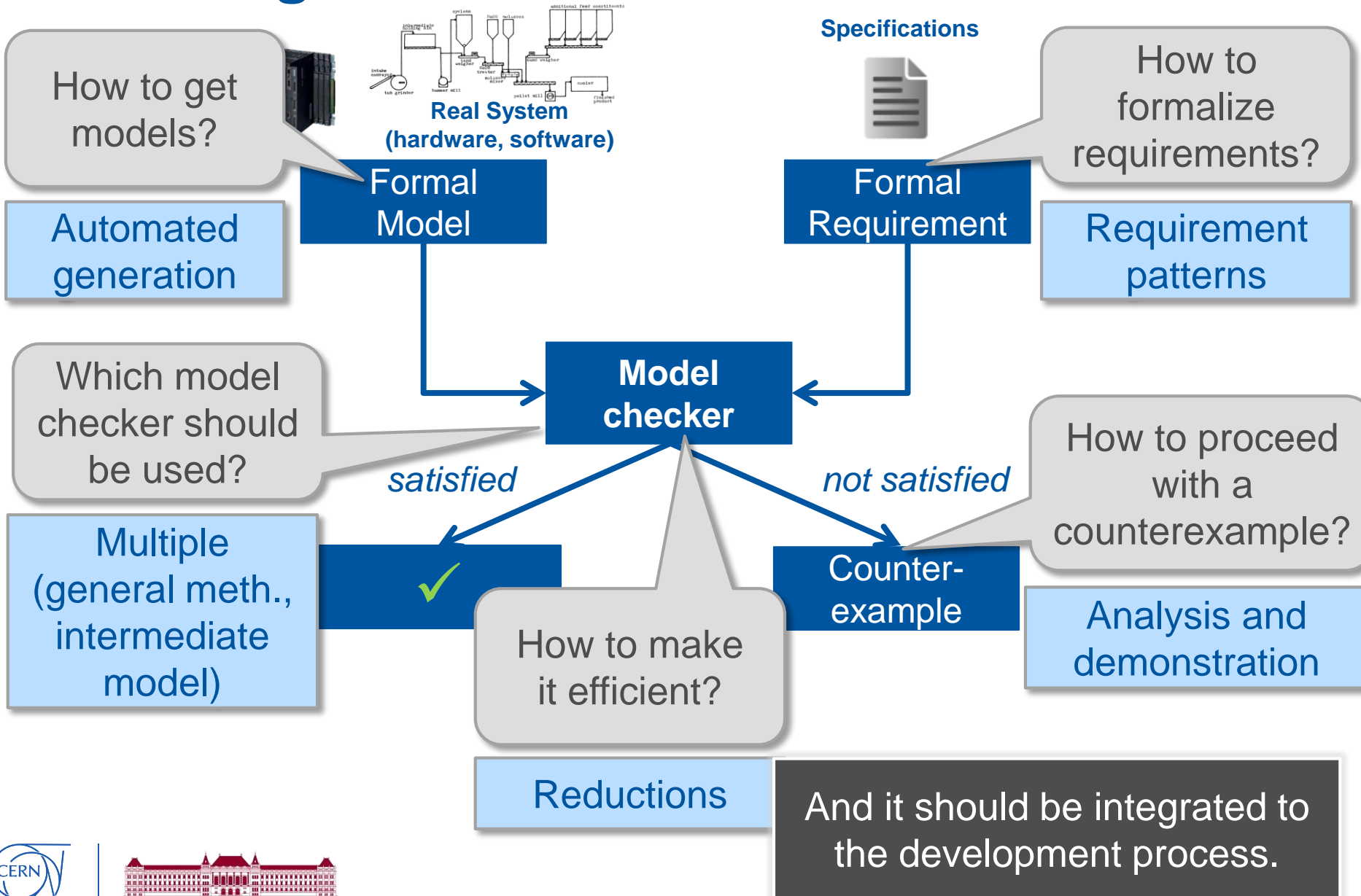
Challenges and answers



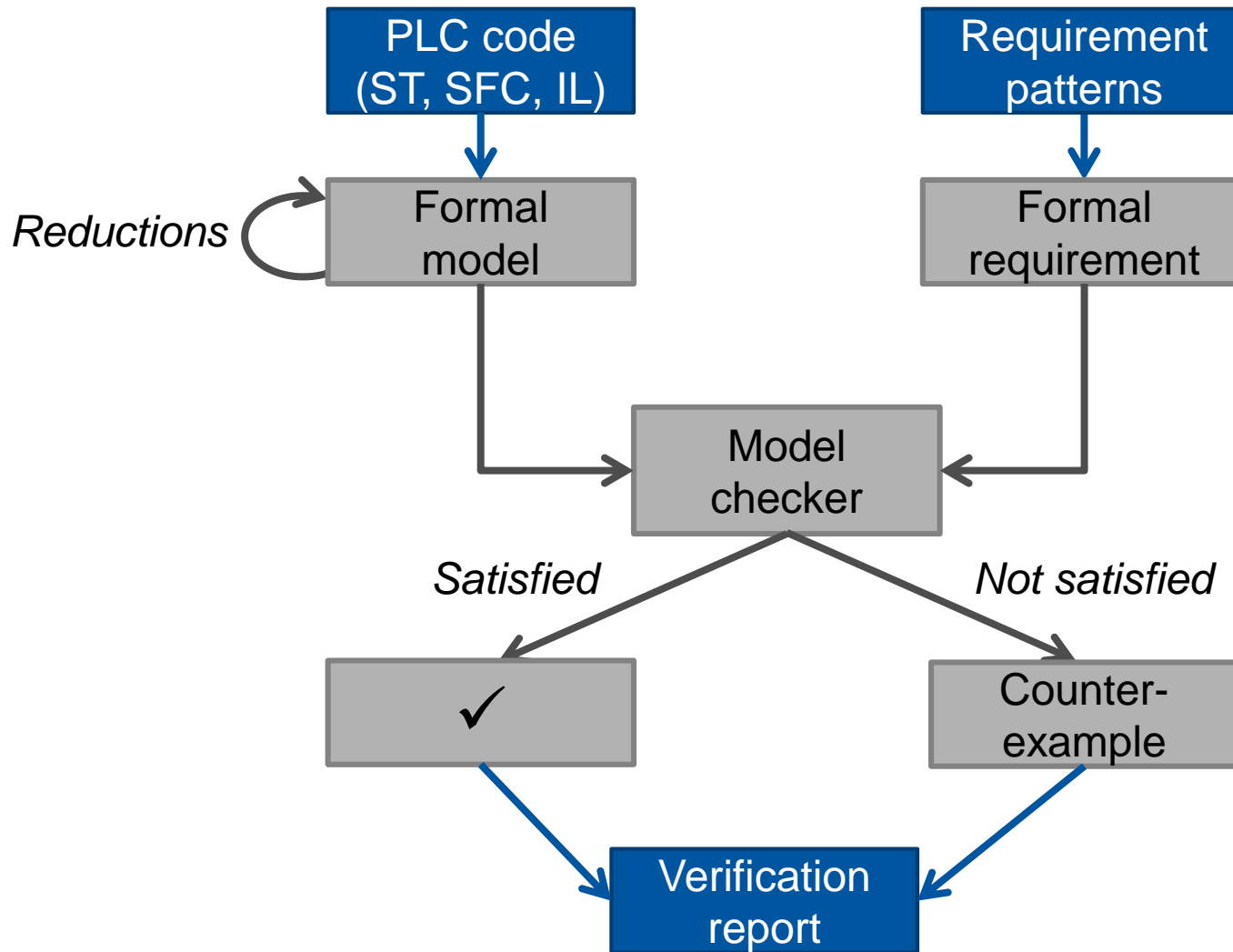
Challenges and answers



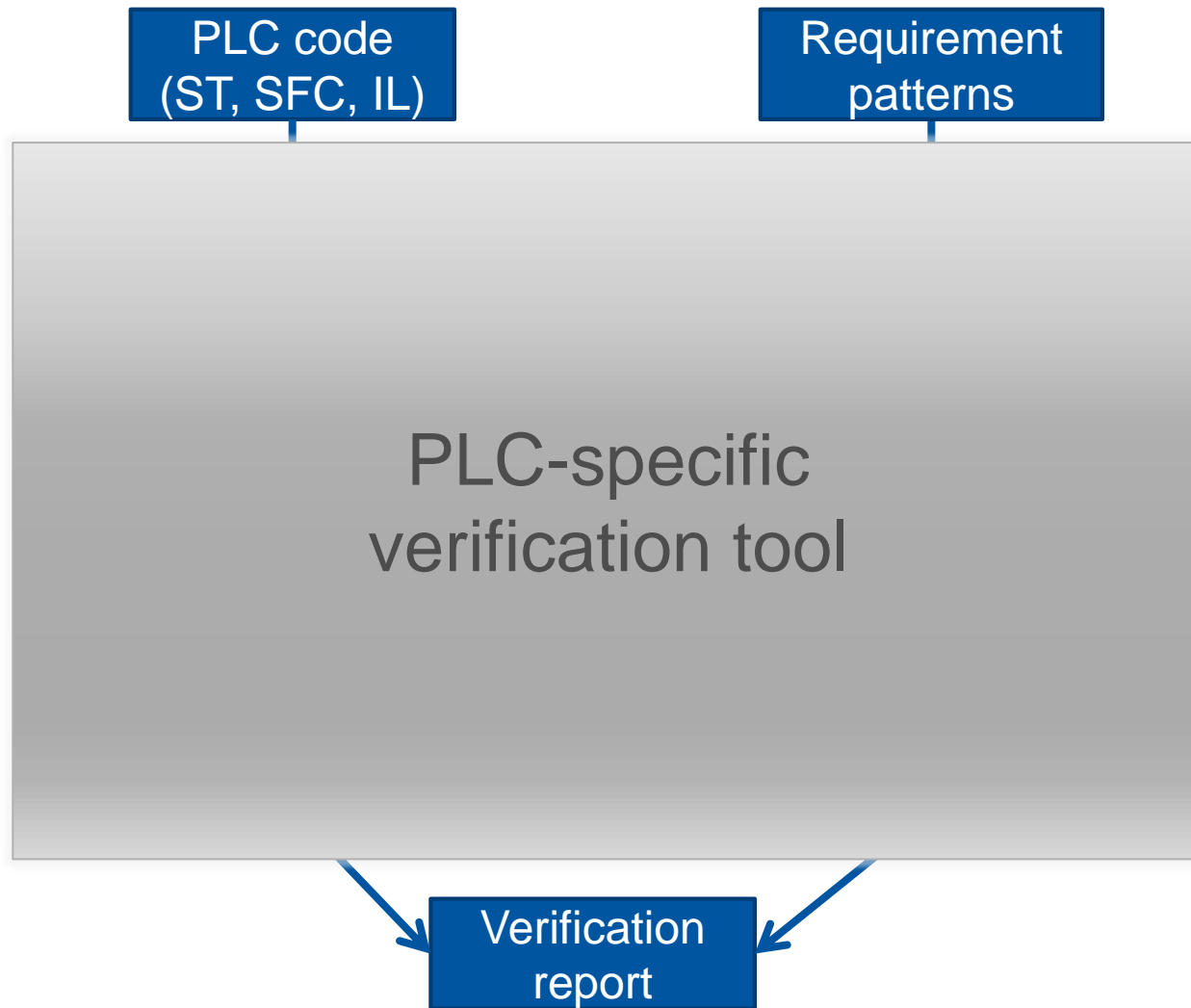
Challenges and answers



Model checking (extended workflow for PLCs)



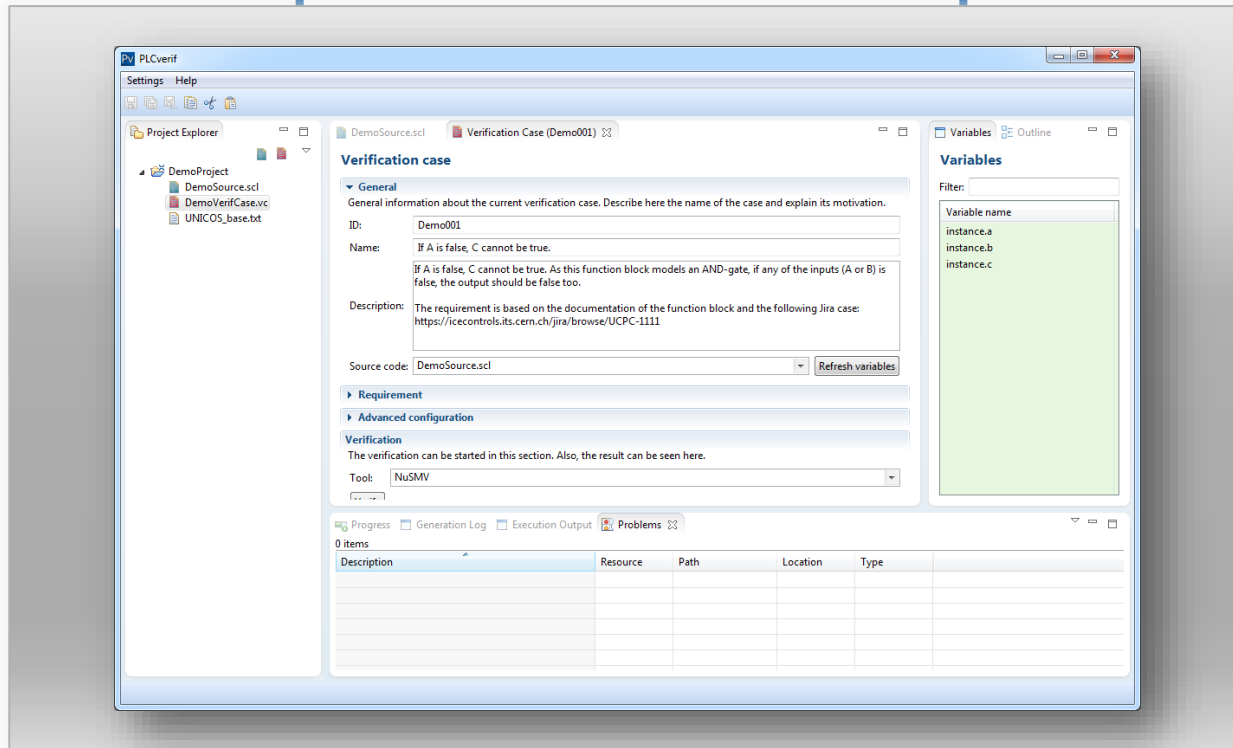
Model checking (extended workflow for PLCs)



Model checking (extended workflow for PLCs)

PLC code
(ST, SFC, IL)

Requirement
patterns



Verification
report

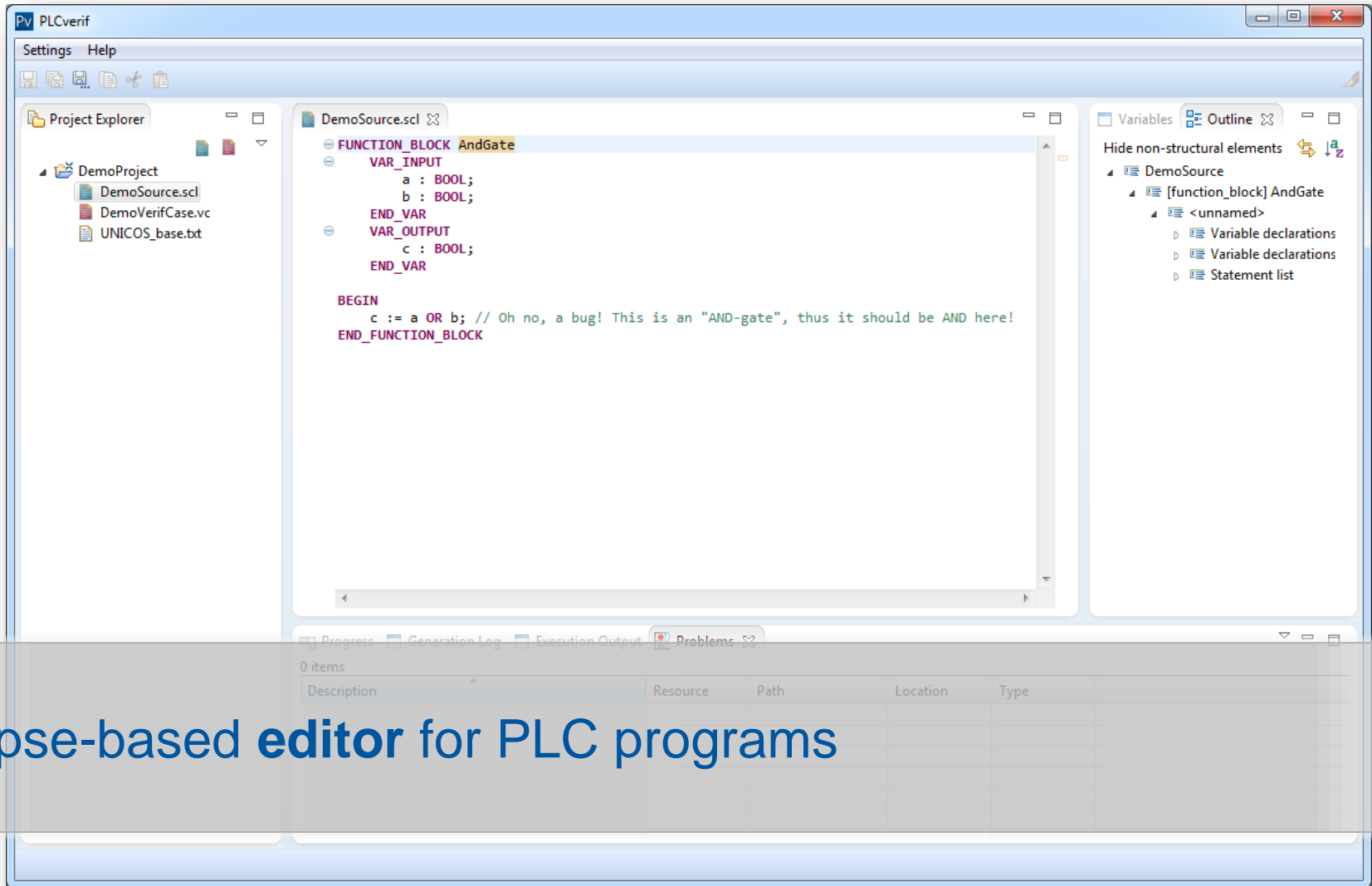
Model checking in practice (at CERN)



The PLCverif tool



The PLCverif tool



Eclipse-based **editor** for PLC programs

The PLCverif tool

The screenshot displays the PLCverif application window. The main area is titled "Verification case" and contains several sections:

- General:** 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 code: DemoSource.scl; Refresh variables button.
- Requirement:** (Collapsed)
- Advanced configuration:** (Collapsed)
- Verification:** The verification can be started in this section. Also, the result can be seen here.; Tool: NuSMV.

On the right side, there is a "Variables" panel with a filter input and a list of variable names: instance.a, instance.b, and instance.c.

At the bottom, there are tabs for Progress, Generation Log, Execution Output, and Problems. Below these is a table with columns: Description, Resource, Path, Location, Type.

Defining **verification cases** (requirement, fine-tuning, etc.)
No model checker-related things or temporal logic expressions

The PLCverif tool

PLCverif — Verification report



Generated at Mon Jul 07 15:19:22 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. $A = \text{false} \ \& \ C = \text{true}$ is impossible at the end of the PLC cycle.
Result:	Not satisfied

Tool: nusmv

Total runtime (until getting the verification results): 212 ms

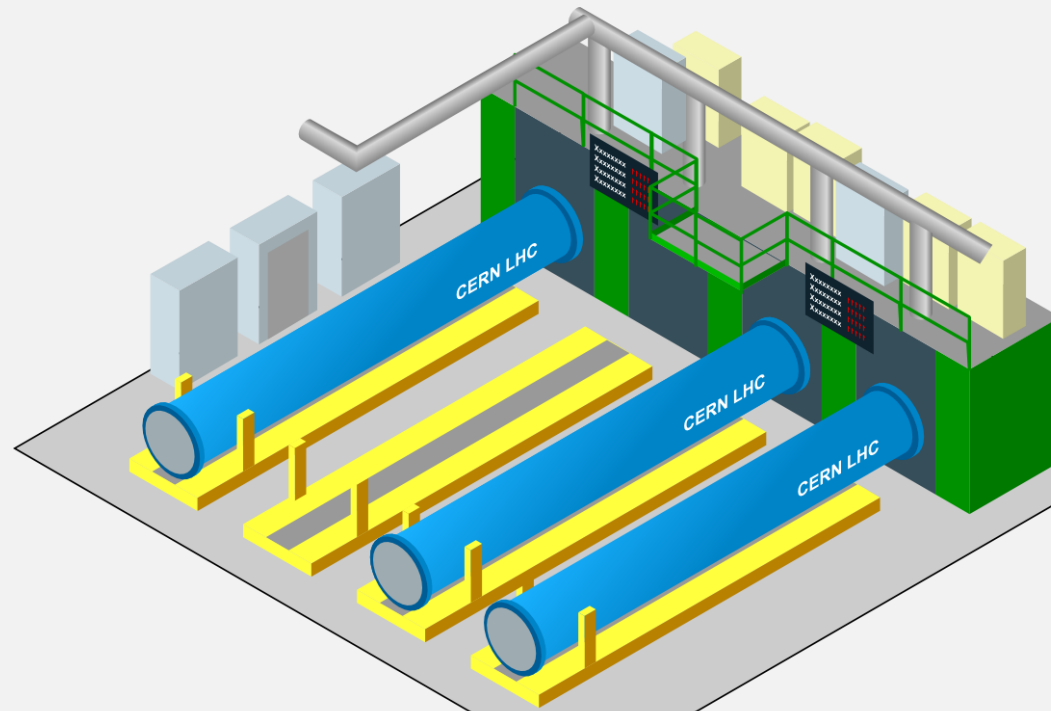
Total runtime (incl. visualization): 361 ms

Counterexample

	Variable	End of Cycle 1
Input	a	FALSE
Input	b	TRUE
Output	c	TRUE

Click-button verification,
verification **report** with the analysed **counterexample**

Example – SM18 safety system



SM18



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

SM18



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

Core:



SM18



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

Core:

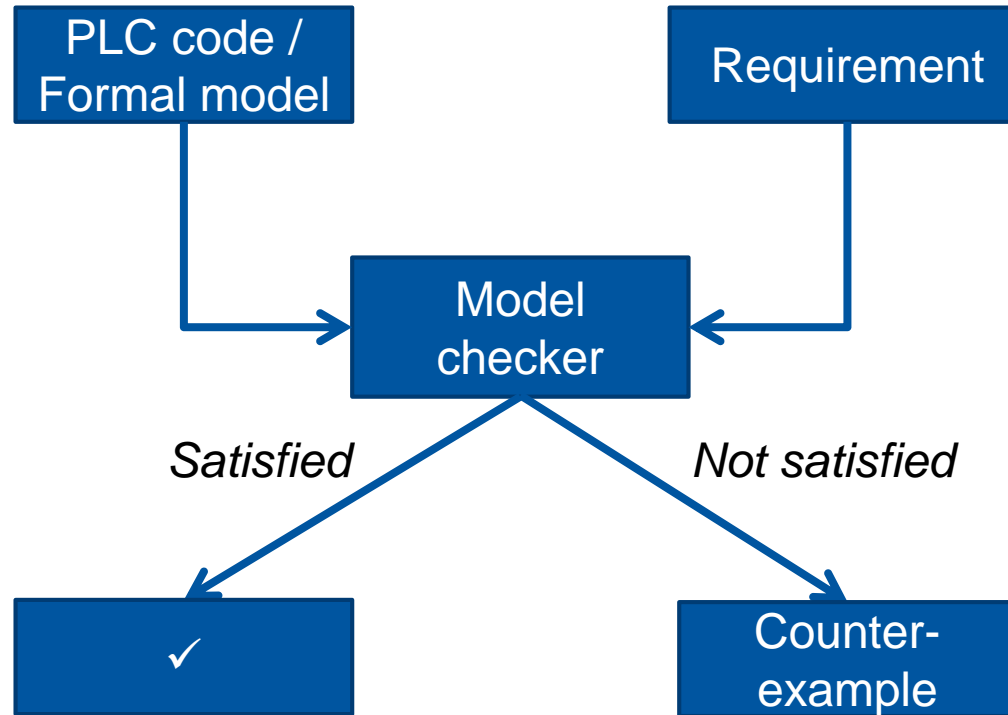
selected test
switch statuses
current voltages
cryo conditions

SM18 PLCSE
safety logic

test allowed

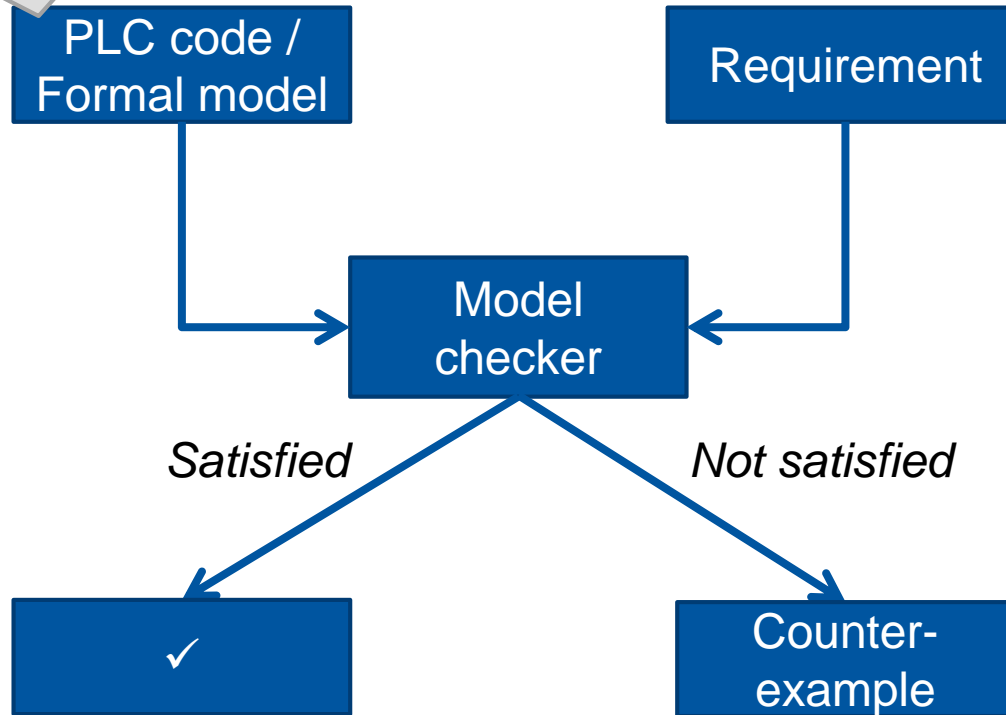
Safety-critical,
can be dangerous

Model checking workflow for SM18



Model checking workflow for SM18

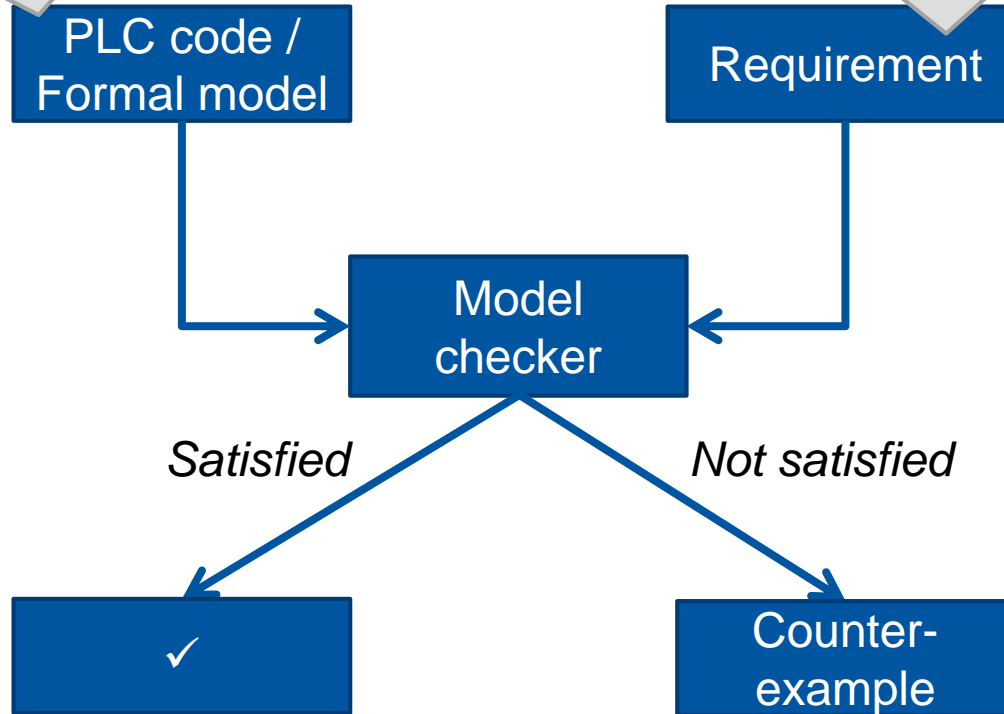
SM18 PLCSE
Safety Logic



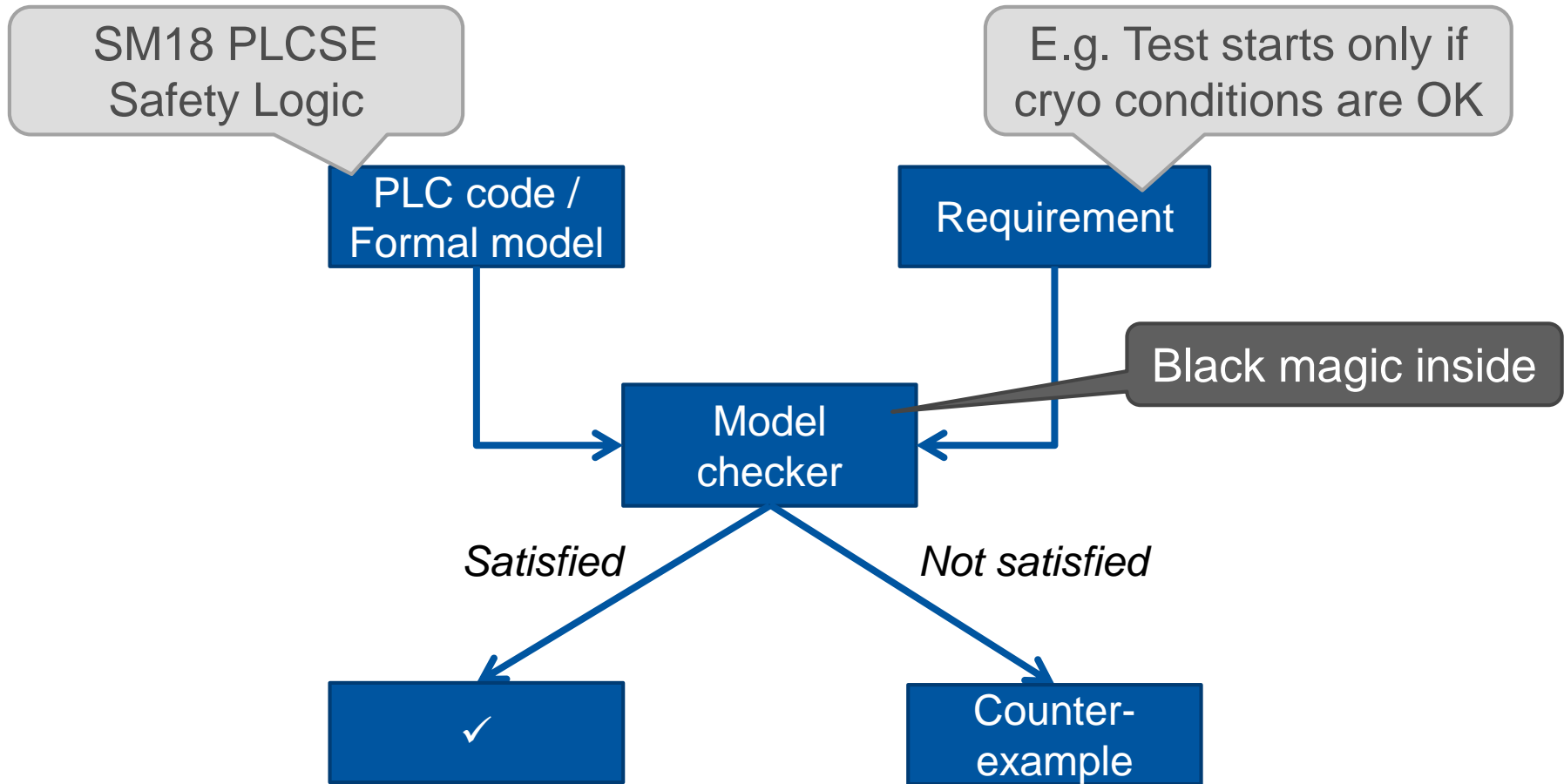
Model checking workflow for SM18

SM18 PLCSE
Safety Logic

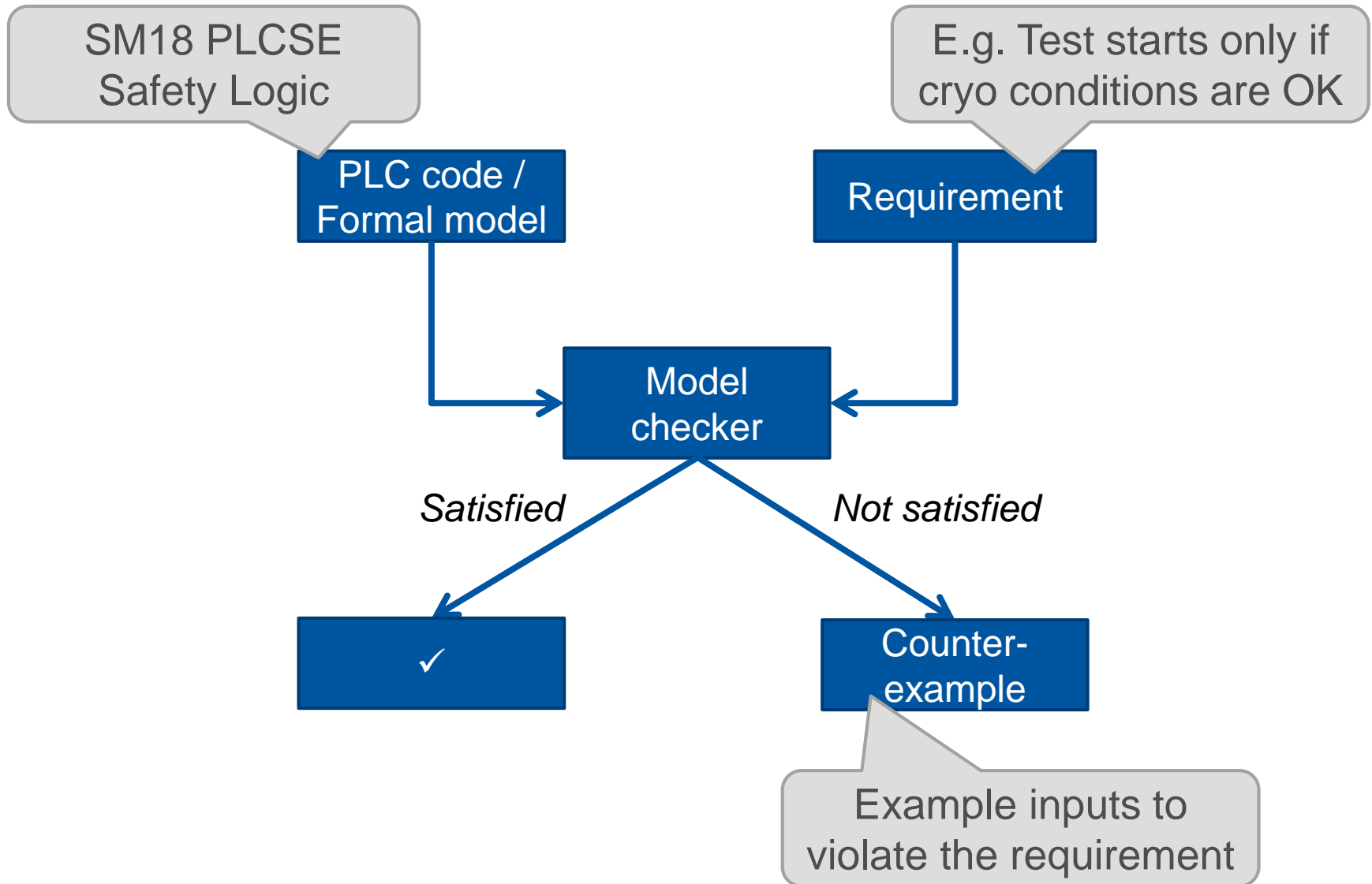
E.g. Test starts only if
cryo conditions are OK



Model checking workflow for SM18



Model checking workflow for SM18





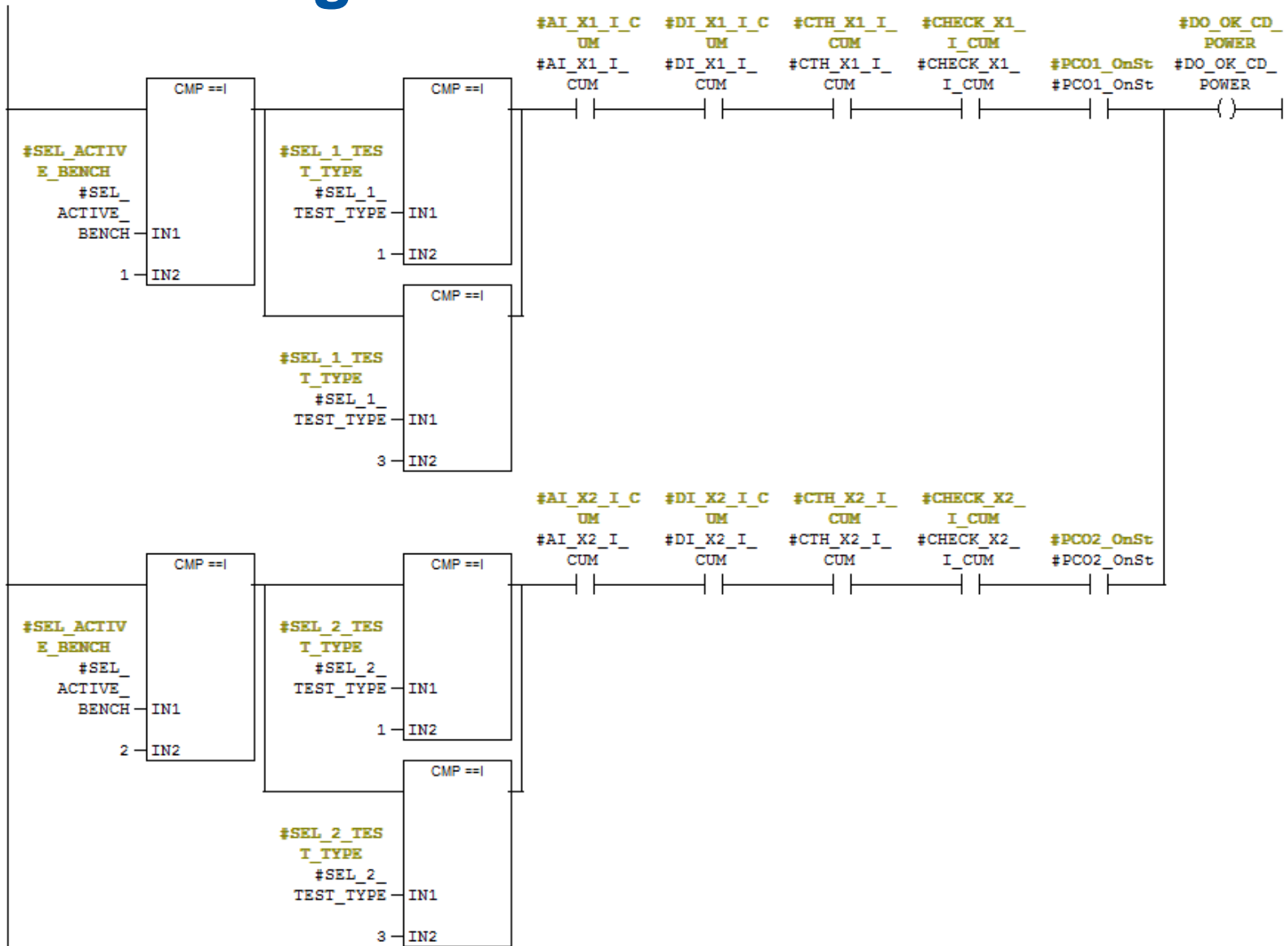
TBC_ACTIVE_BENCH
TBC_SWITCH_MAIN
TBC_POLARITY_MAIN
TBC_SWITCH_CD
TBC_SWITCH_EF
TBC_HV_TEST
TBC_SWITCH_QH
TBC_MAGNET_PHASE
TBC_INTERCON
TBC_FLASHBOX_ADJ_POWER
TBC_V_QH1
TBC_V_QH2
TBC_V_QH3
TBC_V_QH4
TBC_V_LEAD_A
TBC_V_LEAD_B
TBC_V_LEAD_C
TBC_V_LEAD_D
TBC_V_LEAD_E
TBC_V_LEAD_F
TBC_I_MAIN
TBC_I_CD
TBC_I_EF
TBC1_SWITCH_MAIN
TBC1_CABLE_TEMP
TBC1_CABLE_WATER
TBC1_INTERC_QH_CONN
TBC1_SWITCH_CD
TBC1_SWITCH_EF
TBC2_SWITCH_MAIN
TBC2_CABLE_TEMP
TBC2_CABLE_WATER
TBC2_INTERC_QH_CONN
TBC2_SWITCH_CD
TBC2_SWITCH_EF
TBC_SWITCH_MAIN_CC
TBC_SWITCH_CD_CC
TBC_SWITCH_EF_CC
TBC_POWER_QH
TBC_SWITCH_QH_HF
TBC_SWITCH_QH_LF
TBC_STATUS_PC_MAIN
TBC_STATUS_PC_AUX
TBC_POL_MAIN_A
TBC_POL_MAIN_B
TBC1_FT_LEAD_A
TBC1_FT_LEAD_B
TBC1_LEAD_AUX
TBC1_T_MAG
TBC1_ANTICRYO
TBC1_CRYO_1_9K
TBC1_CRYO_4_5K
TBC1_CRYO_HV
TBC1_CRYO_20K
TBC1_CRYO_300K
TBC1_CRYO_300KAIR
TBC2_FT_LEAD_A
TBC2_FT_LEAD_B
TBC2_LEAD_AUX
TBC2_T_MAG
TBC2_ANTICRYO
TBC2_CRYO_1_9K
TBC2_CRYO_4_5K
TBC2_CRYO_HV
TBC2_CRYO_20K
TBC2_CRYO_300K
TBC2_CRYO_300KAIR

SM18 PLCSE
safety logic

TBC1_INTERC
TBC1_INTERC_POWER
TBC2_INTERC
TBC2_INTERC_POWER
TBC_INTERC_CC
TBC_FLASHBOX_ADJ_ON
TBC_CRYO_I_BELOW_2KA
TBC1_CRYO_ACTIVE_BENCH
TBC2_CRYO_ACTIVE_BENCH
TBC1_HV_OK_300KAIR
TBC1_HV_OK_COLD
TBC2_HV_OK_300KAIR
TBC2_HV_OK_COLD
TBC_OK_CD_POWER
TBC_OK_EF_POWER
TBC_OK_MAIN_POWER
TBC1_OK_FOR_TEST
TBC2_OK_FOR_TEST



Ladder Diagram



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

The image displays two overlapping screenshots illustrating a verification workflow. The top screenshot shows the Jenkins web interface, and the bottom screenshot shows an Outlook email containing a verification report.

Jenkins Interface

Build Queue
No builds in the queue.

Build Executor Status

- master
 - 1 Idle
 - 2 Idle
- ddarvas-plcverif-jenkins1 (offline)
- mletriclinux00
 - 1 [PLCverif SM18 SVN](#) #31

Build History Table

S	W	Name ↓	Last Success	Last Failure	Last Duration
		AssemblePLCverif	58 min - #28	6 days 22 hr - #16	2 min 34 sec
		PLCverif OnOff SVN	47 min - #8	59 min - #7	4 min 10 sec
		PLCverif SM18 SVN	3 days 19 hr - #29	1 hr 29 min - #30	1 hr 20 min
		PLCverif SVN Template	N/A	N/A	N/A
		PLCverifUploadAndVerify	1 day 14 hr - #51	1 day 14 hr - #50	6.5 sec

Outlook Email: HTMLSummary.html

Size: 58 KB
Last changed: 18 September 2015

Message: HTMLSummary.html (58 KB)

Some pictures have been blocked to help prevent the sender from identifying your computer. Open this item to view the pictures.

DO_FLASHBOX_ADJ_ON_req1 : DO_FLASHBOX_ADJ_ON_req1

4. DO_FLASHBOX_ADJ_ON = (((SEL_1_TEST_TYPE >= 0ud8_5 AND SEL_1_TEST_TYPE <= 0ud8_9 AND SEL_ACTIVE_BENCH = 0ud8_1 AND (PCO1_OnSt OR PCO2_OnSt)) OR (SEL_2_TEST_TYPE >= 0ud8_5 AND SEL_2_TEST_TYPE <= 0ud8_9 AND SEL_ACTIVE_BENCH = 0ud8_2 AND (PCO1_OnSt OR PCO2_OnSt)))) is always true at the end of the PLC cycle.

Satisfied Total: 19082 ms* (MChk: 250 ms) [Open the Verification Report](#)

DO_FLASHBOX_ADJ_ON_req2 : DO_FLASHBOX_ADJ_ON_req2

1. If DO_FLASHBOX_ADJ_ON is true at the end of the PLC cycle, then (SEL_FLASHBOX_ADJ_POWER = 0ud8_1) should always be true at the end of the same cycle.

Satisfied Total: 10489 ms* (MChk: 256 ms) [Open the Verification Report](#)

DO_INTERC_CC_req2 : DO_INTERC_CC_req2

4. DO_INTERC_CC = (SEL_SWITCH_INTERCON = 0ud8_3) is always true at the end of the PLC cycle.

Satisfied Total: 8499 ms* (MChk: 98 ms) [Open the Verification Report](#)

DO_INTERC_CC_req3 : DO_INTERC_CC_req3 (safety)

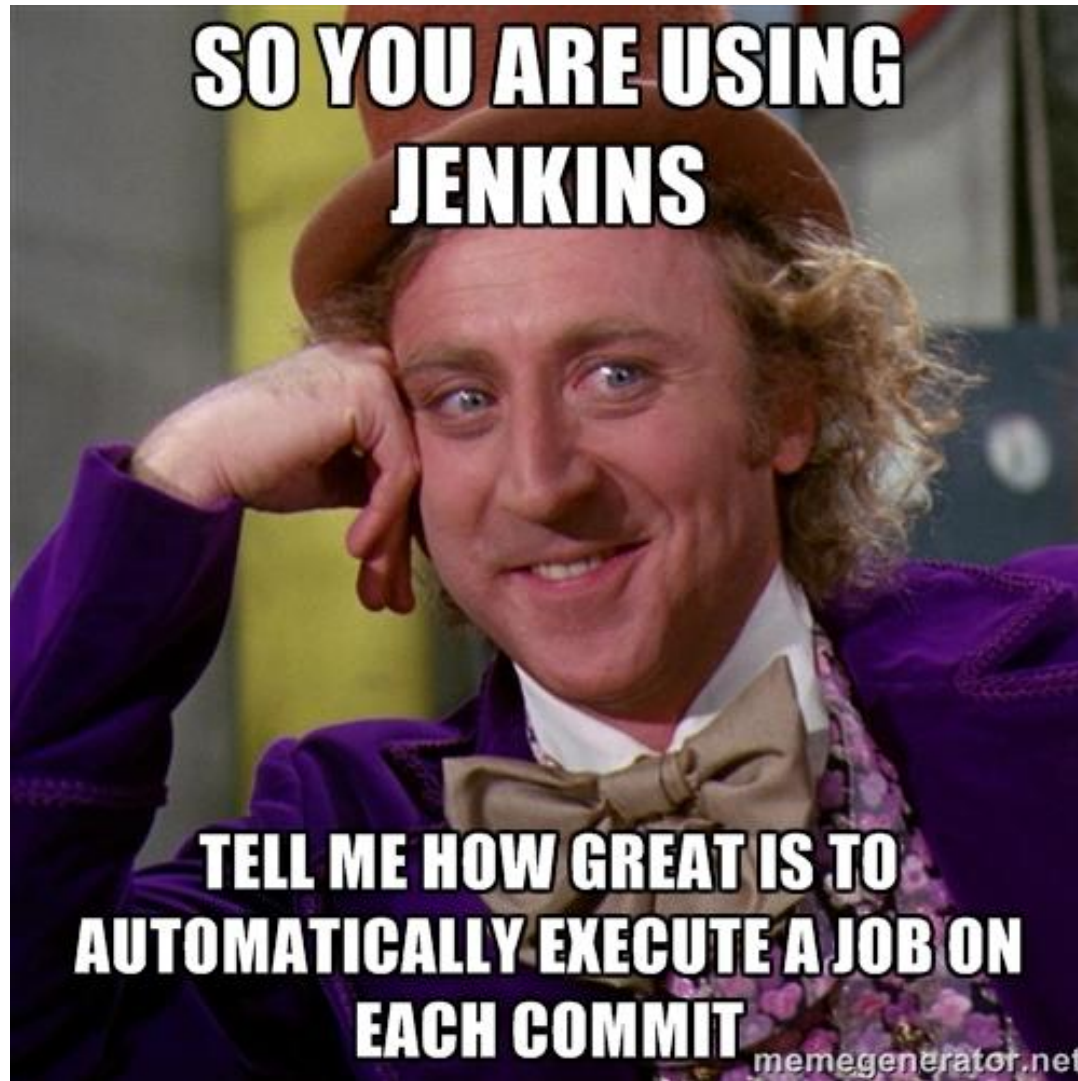
1. If DO_INTERC_CC is true at the end of the PLC cycle, then SEL_SWITCH_INTERCON = 0ud8_3 and SEL_SWITCH_INTERCON = 0ud8_3 should always be true at the end of the same cycle.

Satisfied Total: 9784 ms* (MChk: 160 ms) [Open the Verification Report](#)

See more about Jenkins PLCverif.



Verification workflow in practice



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

Summary

<http://go.cern.ch/7L9h>
<http://cern.ch/plcverif>



Summary

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

<http://go.cern.ch/7L9h>
<http://cern.ch/plcverif>



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

<http://go.cern.ch/7L9h>
<http://cern.ch/plcverif>



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>





Messages

Edit

Formal verification?

Yes, boring stuff

Academic

Just survive until the next pres :)



Send





Messages

Edit

Formal verification?

Yes, boring stuff

Academic

Just survive until the next pres :)

Formal verification is great!



Send





Messages

Edit

Formal verification?

Yes, boring stuff

Academic

Just survive until the next pres :)

Formal verification is great!

Well... At least now it's over.



Send





www.cern.ch