# Safety Rules Synthesis for Run-Time Monitoring of Autonomous Robots in Human Environment

Safety MOnitoring Framework (SMOF)
https://www.laas.fr/projects/smof/

Jérémie Guiochet
Hélène Waeselynck
David Powell

LAAS-CNRS
Université de Toulouse
Jeremie.guiochet@laas.fr
http://homepages.laas.fr/guiochet

IFIP WG 10.4 Workshop on "Smart Factories: A New Challenge for Dependable and Secure Computing" - June 24-27, 2016

# Smart factories : Back to the future

# “Robot safety” in Robotics

- Most analyses focus on physical human-robot interactions, i.e. how to control/react/plan to avoid collision or in case of physical contact
  =>no faults ! No uncertainties !


# Dependable robots@laas

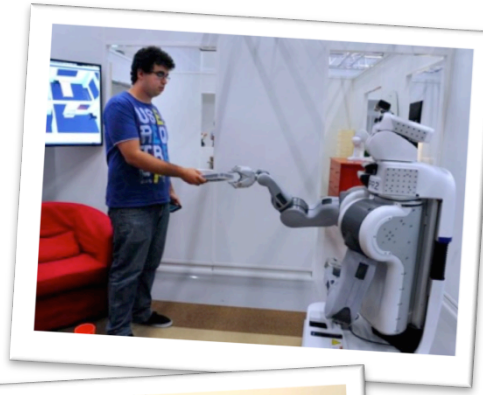
- Phds :
  - Execution Monitoring (2005) , Diverse task planning (2007), Robustness testing (2011), Safety monitoring (2012), Safety analysis of human-robot interactions (2015), Safety monitoring (with synthesis) (2015), Testing autonomous robots in virtual worlds (running), Multi-level safety monitoring (running)

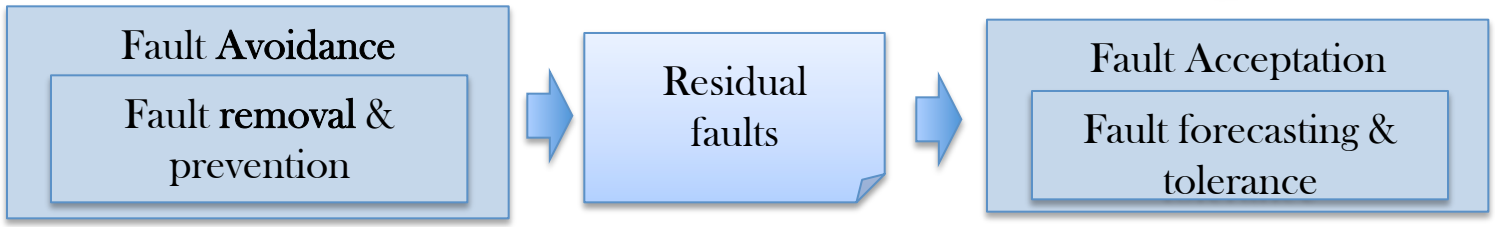
- Recent European projects :
  - CPS Engineering Labs: cyber physical systems, European H2020-ICT, 2015-2018
  - SAPHARI : Safe and Autonomous Physical Human-Aware Robot Interaction, FP7 European Project, 2011-2014
  - PHRIENDS: Physical Human-Robot Interaction: depENDability and Safety, FP6 European project, 2006-2009



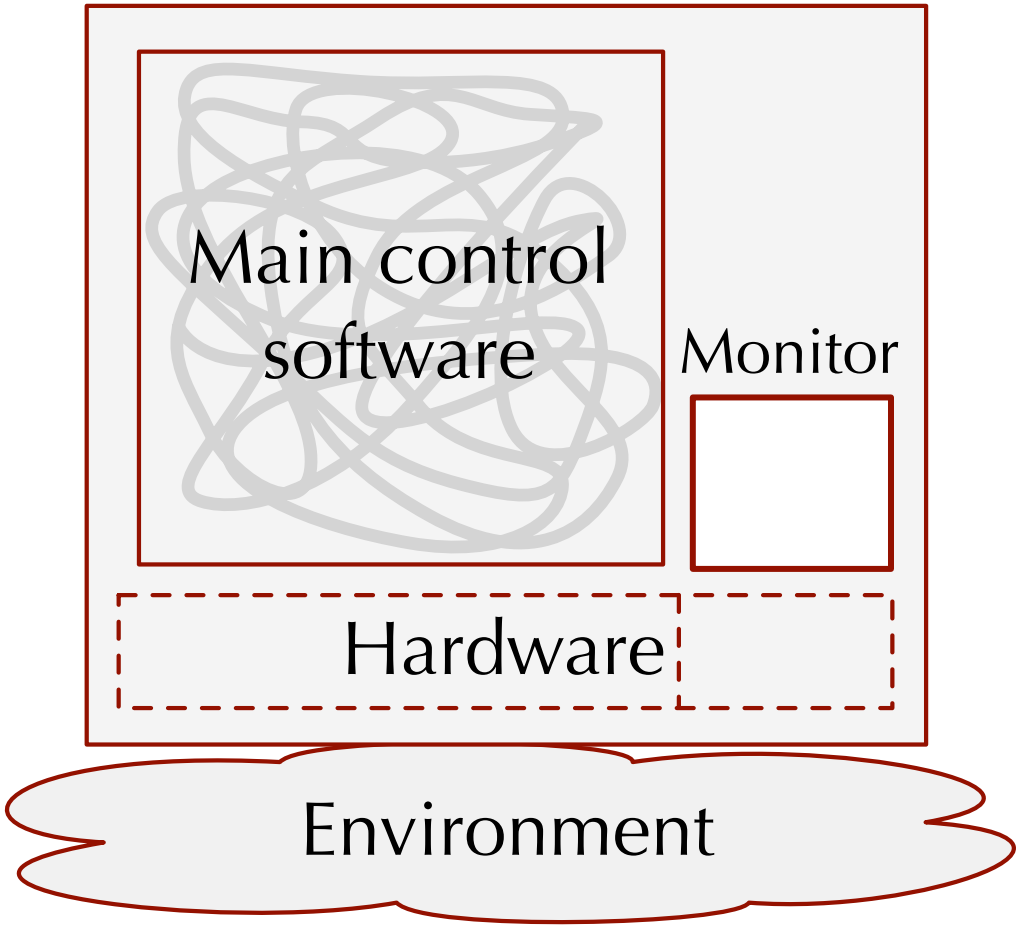
# Dependable autonomous robots

- Autonomous applications : Complex, evolving in unstructured environment, versatile, networked
- Fault model
  - Development Faults (e.g.,in autonomous SW)
  - Physical Faults (e.g., hardware)
  - Interaction Faults (e.g.,human-robot interactions)
  - Other « faults »:
    - Uncertainties (e.g., in perception, heuristics)
    - Adverse situations (e.g., unexpected hazards)





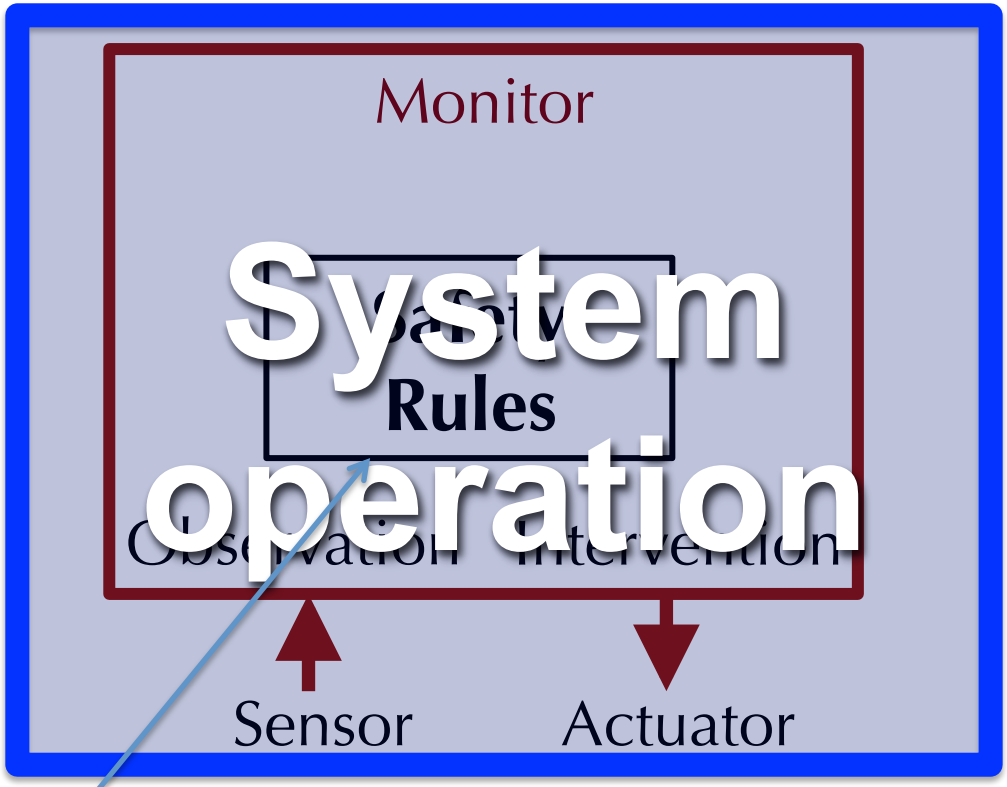
# Active safety monitor



# Safety Rules

Properties required from the monitor:

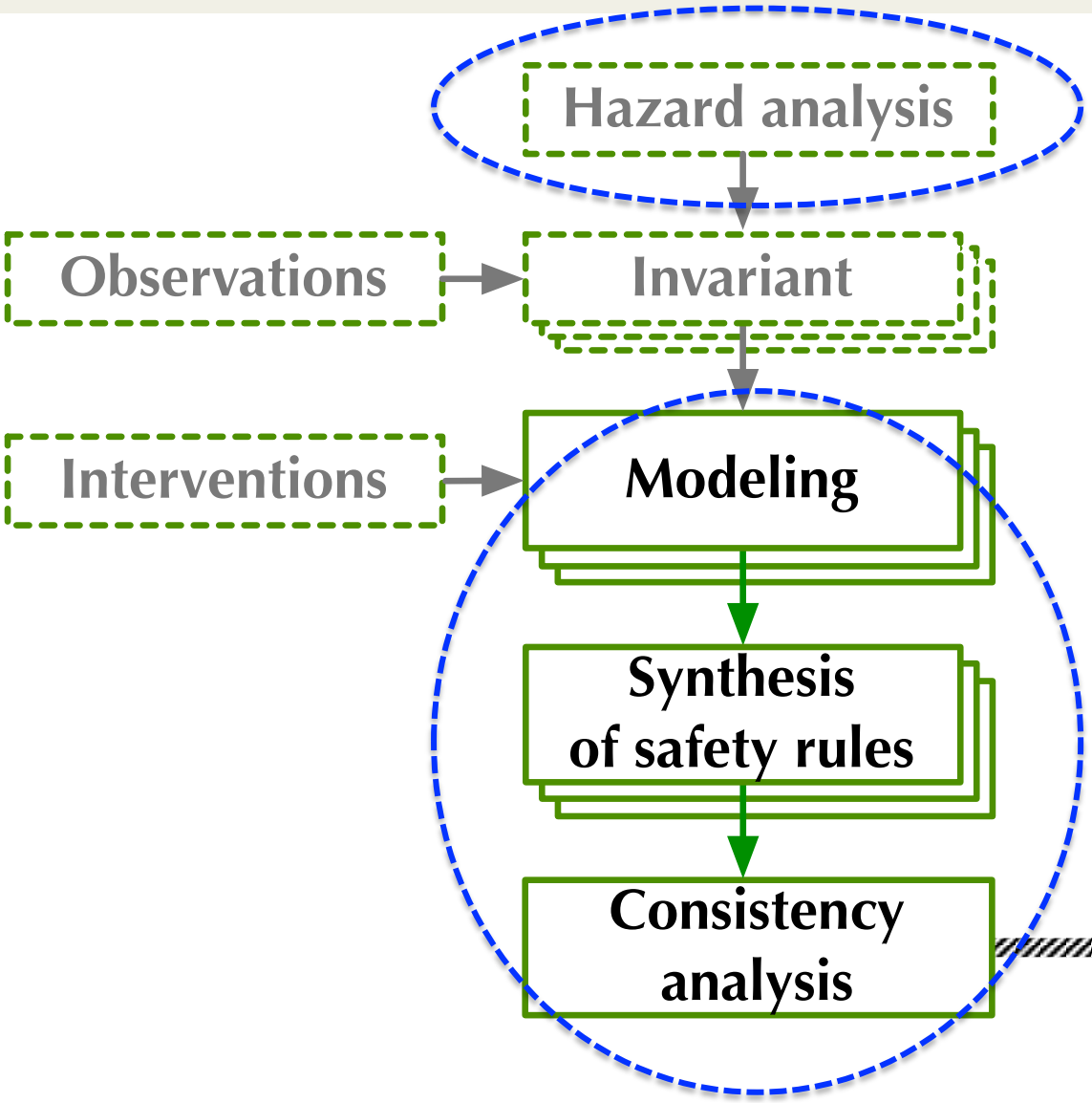
- Safety
- Permissiveness




System design

Specification of the safety rules

# Method overview



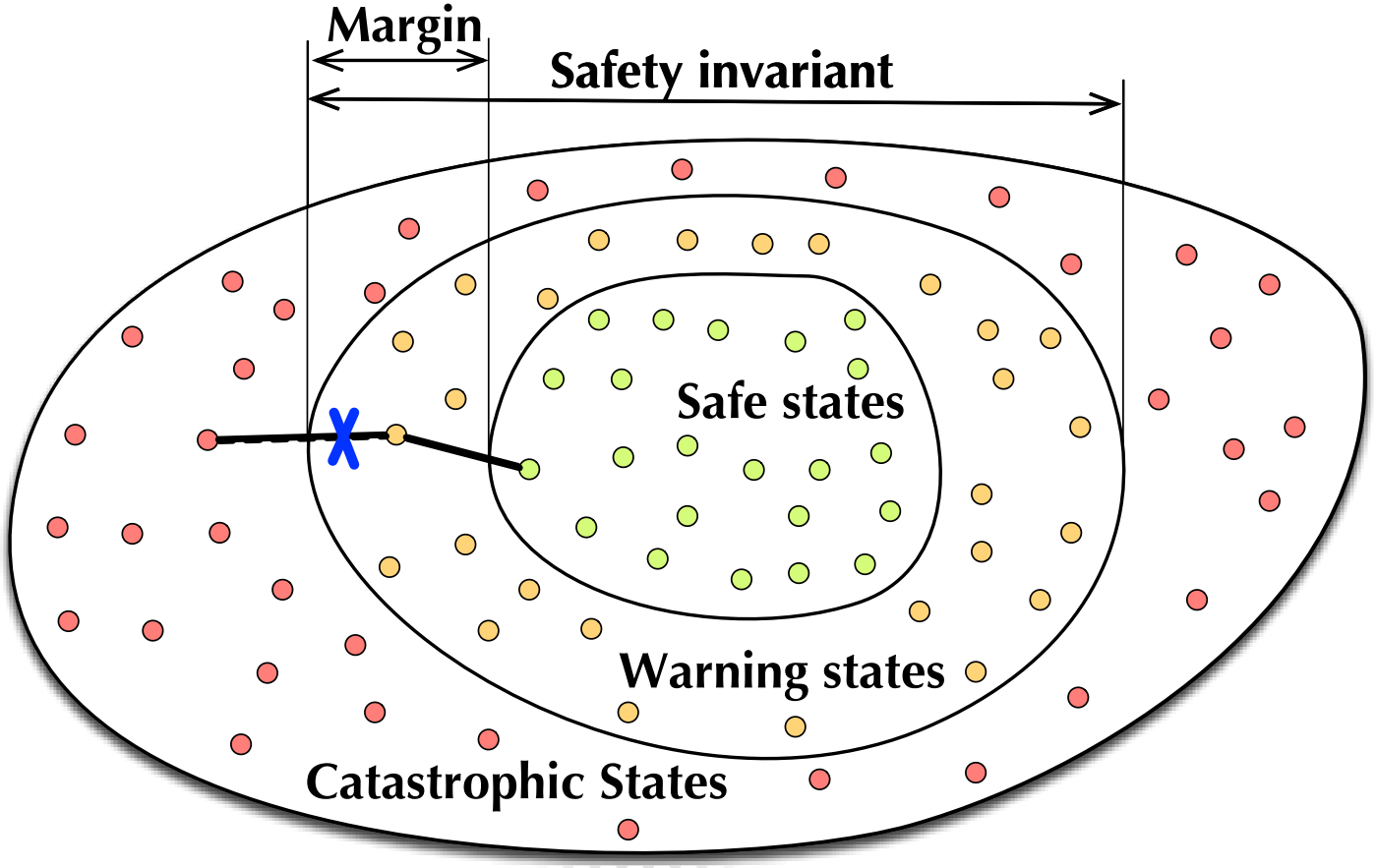

HAZOP-UML approach

SMOF = Modeling template + tools



8

# Concepts: margin, warning states

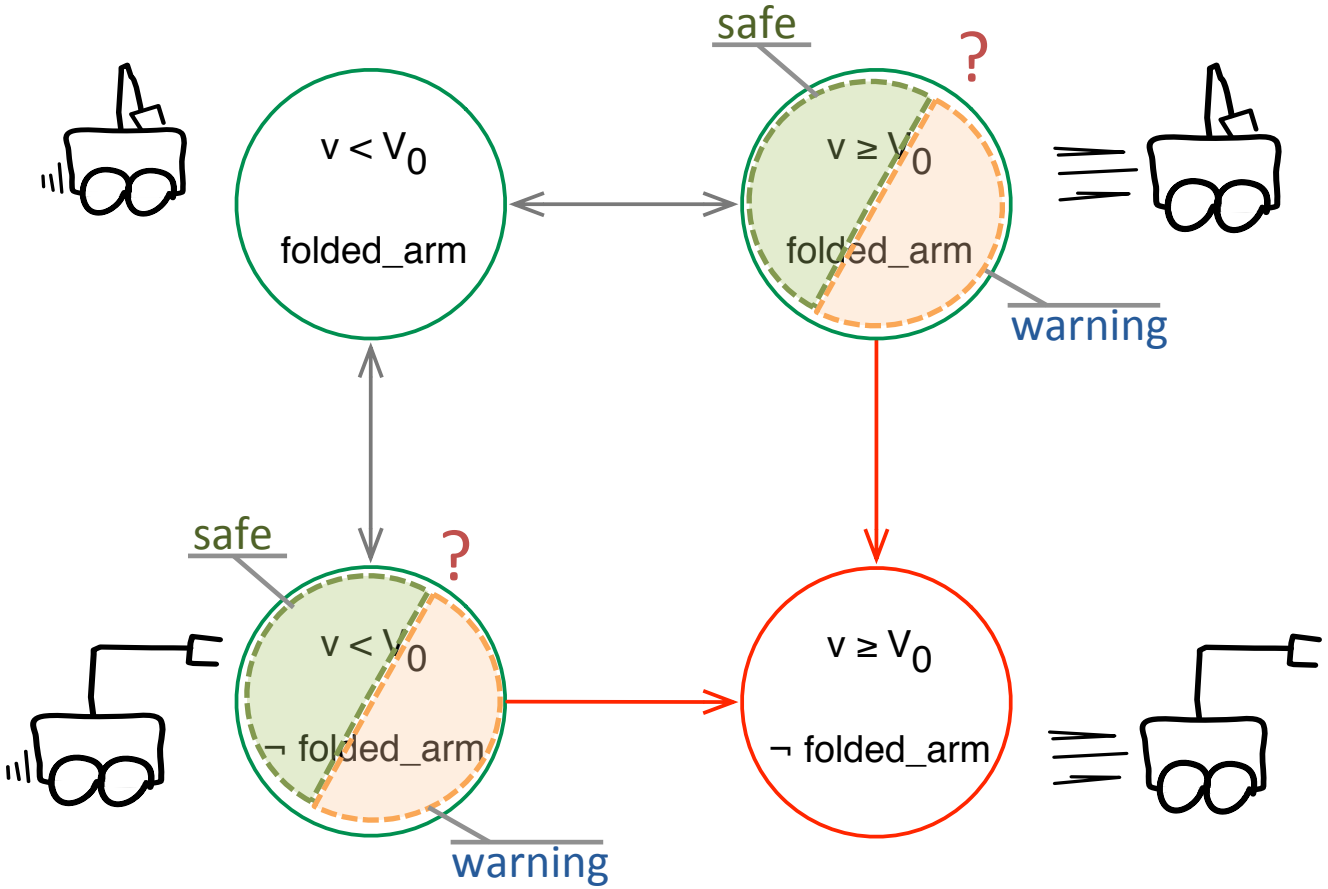



- A safety rule assigns interventions to warning states
- A **strategy** is a set of safety rules intended to ensure an invariant

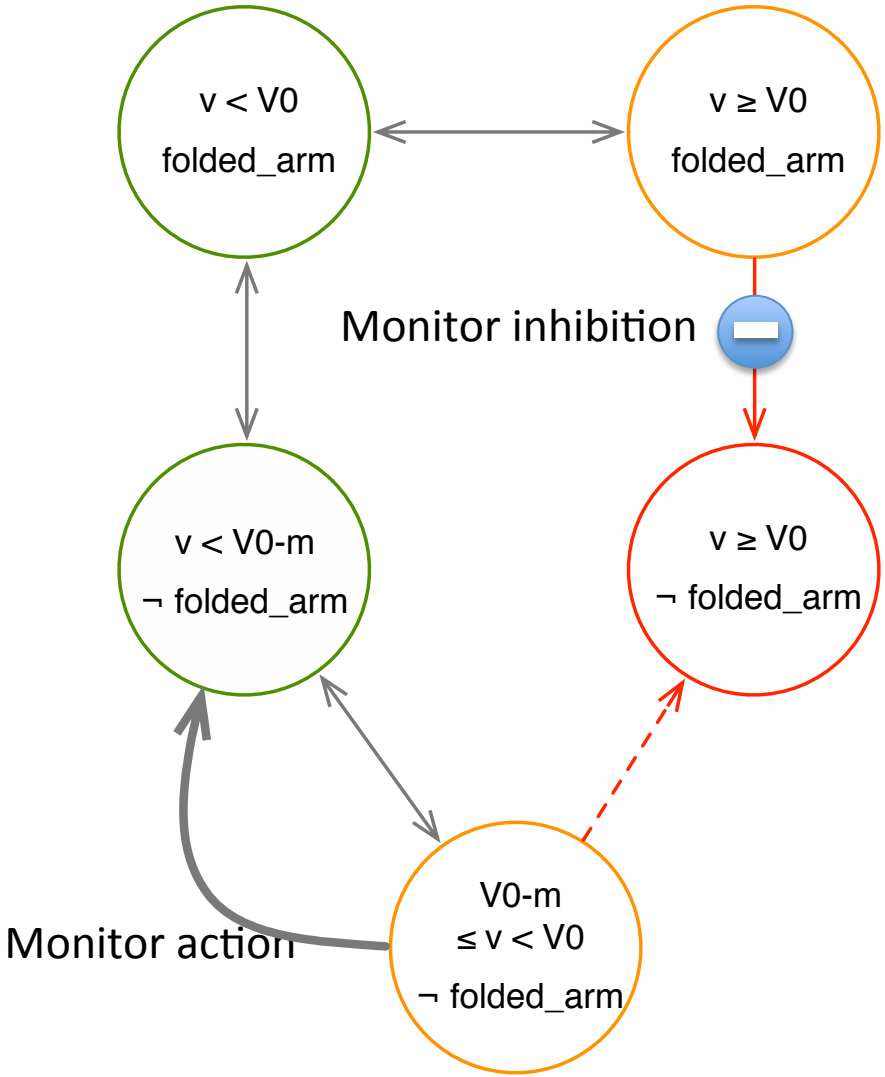
# Toy example

- Cata = ( $v \geq V_0 \wedge \neg$ *folded_arm*)

Safety invariant SI1 :
( $v < V_0 \vee$ *folded_arm*)



# Toy example (2)



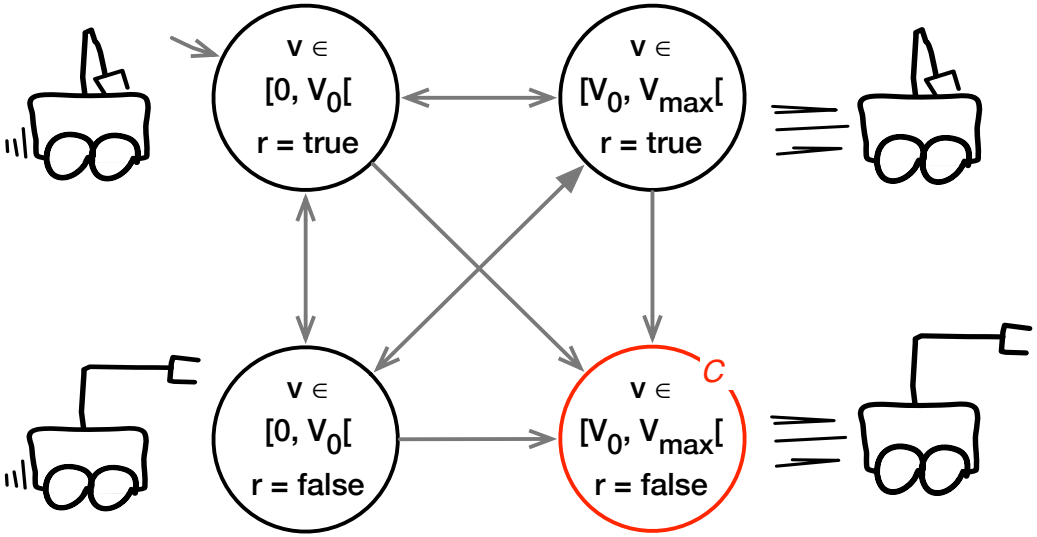

**Safety strategy**

Safety rule SR1 (inhibition):
$(v \geq V_0 \wedge folded\_arm) \rightarrow next(arm)=folded\_arm$

Safety rule SR2 (action):
$V_0\text{-}\theta < v \leq V_0 \wedge \neg\, folded\_arm \rightarrow brake$

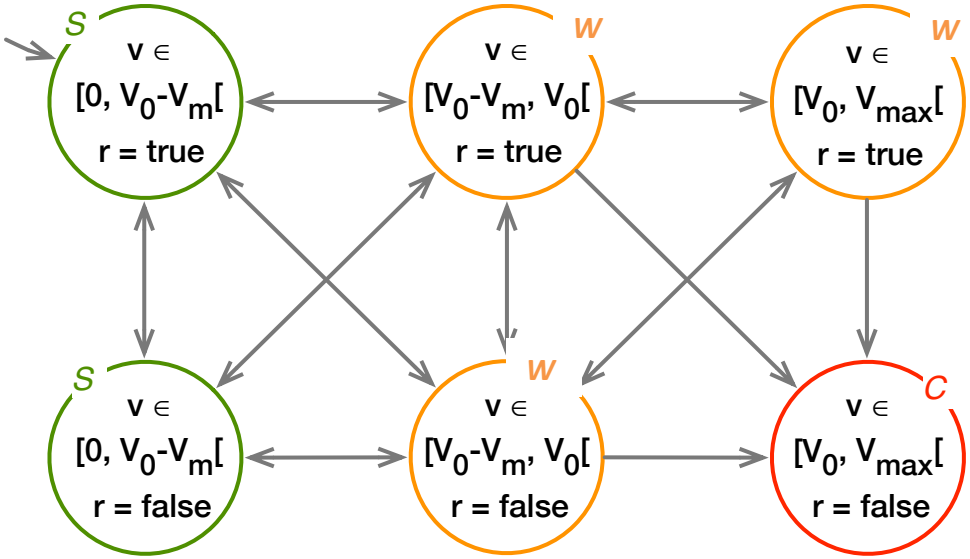
A safety strategy
for safety invariant SI1

# Toy example (3)



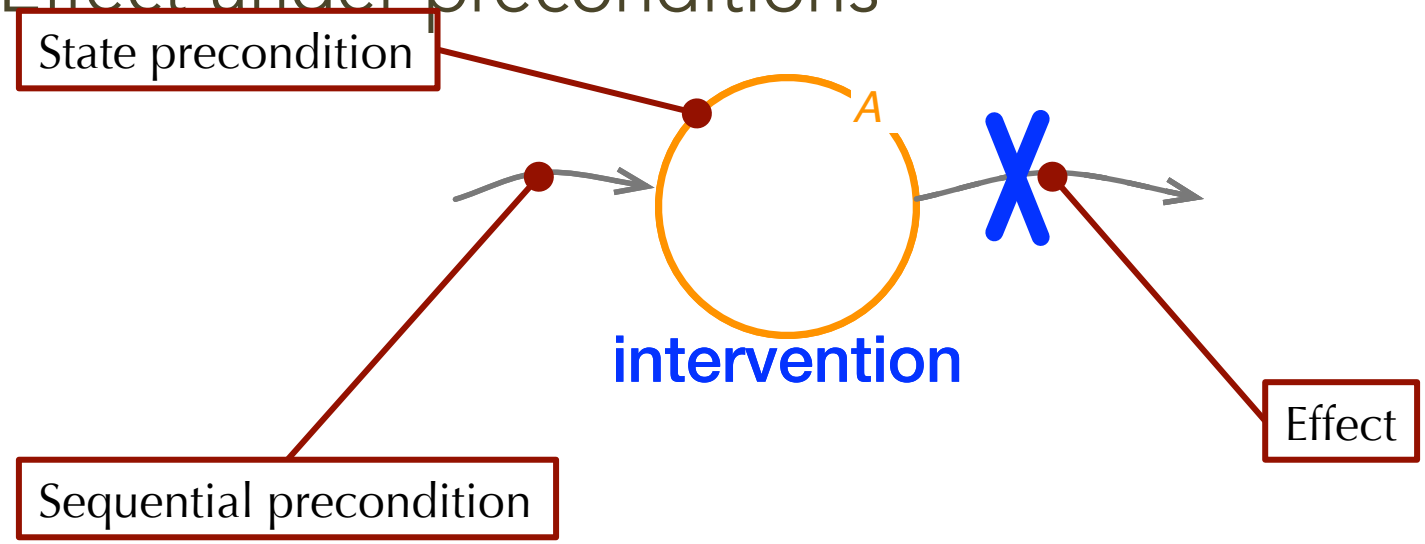

Margin on velocity

3 warning states



# Interventions

- Ability of the monitor to constrain the system behavior
- E.g.: engage platform brakes, lock the arm position
- Effect under preconditions



# Modeling with SMOF

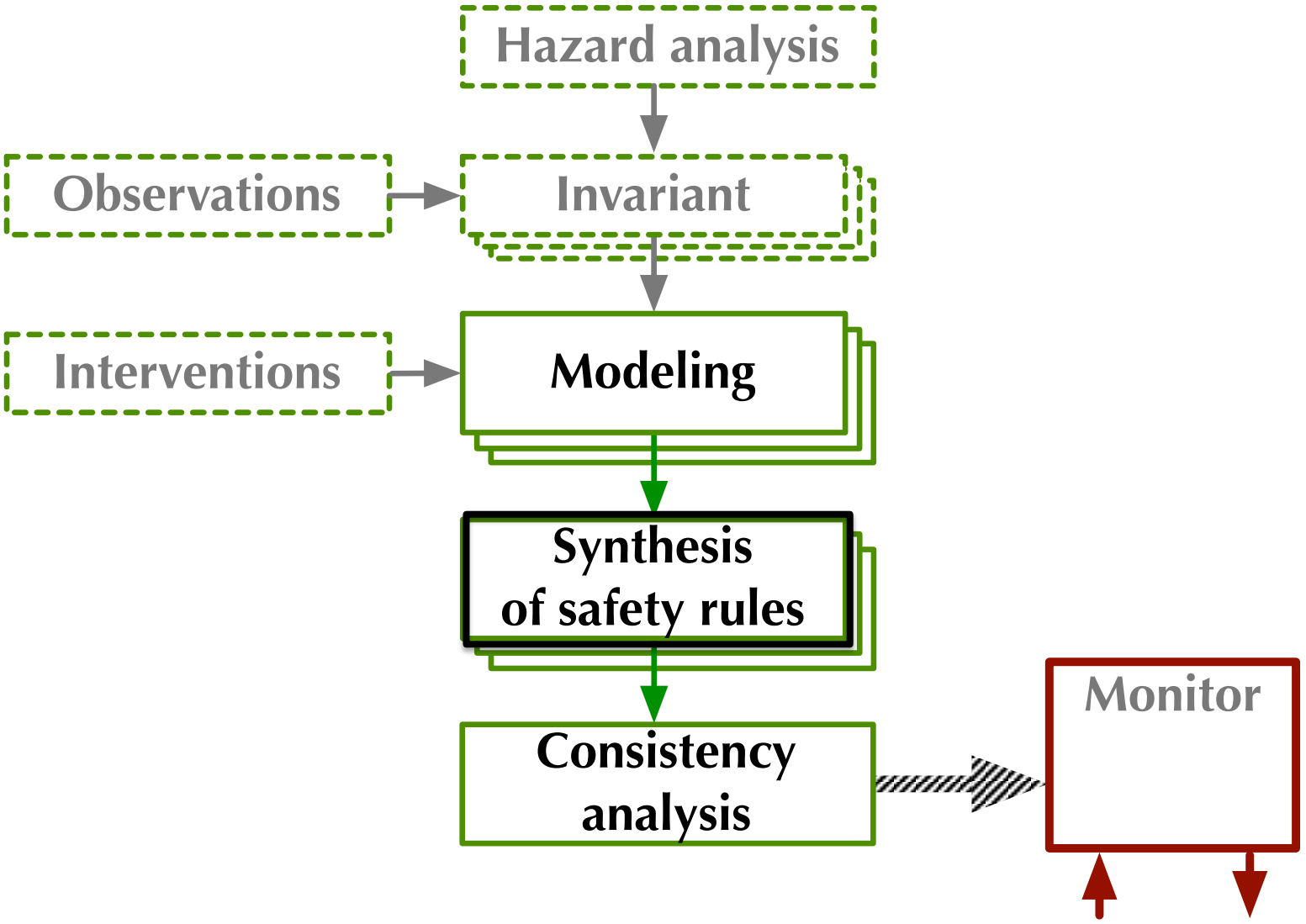
- NuSMV
- Modeling template:
  - Predefined parts
  - Parts to be edited by the user
  - Generated parts

```
VAR
pf_vel: Continuity(0,2,0);
arm_pos : Continuity(0,1,1);

DEFINE cata:= (pf_vel=2 & arm_pos=0);

VAR
brake : Intervention(TRUE, pf_vel!=0, flag_brake, next(pf_vel)=pf_vel!=2);
lock_arm : Intervention(arm_pos=1, TRUE, flag_lock_arm, next(arm_pos)=1);
```

# Method


Hazard analysis
Observations
Invariant
Interventions
Modeling
Synthesis
of safety rules
Consistency
analysis
Monitor

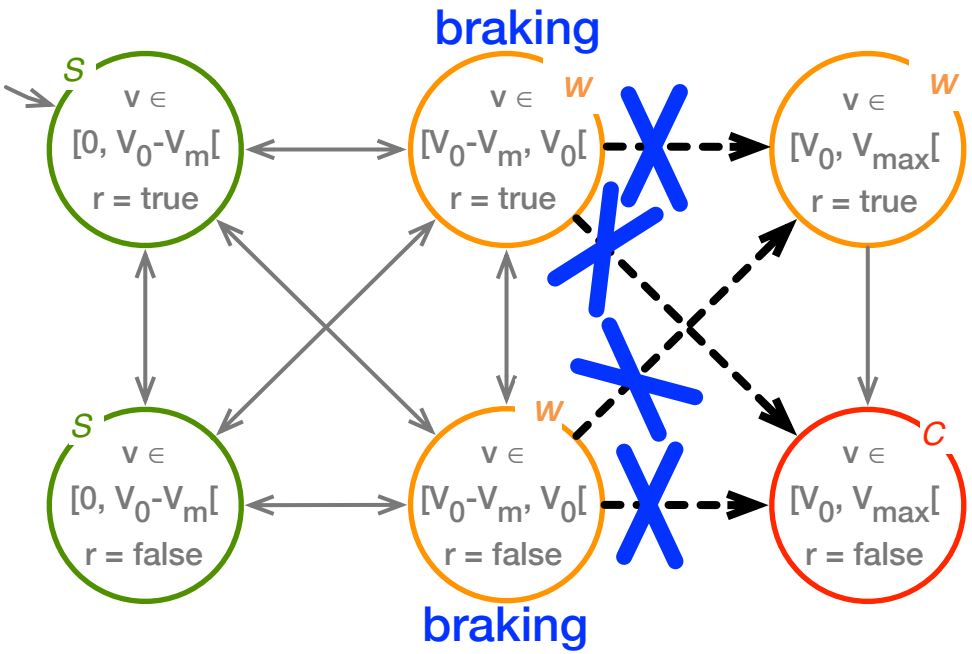
# Strategies

- Association

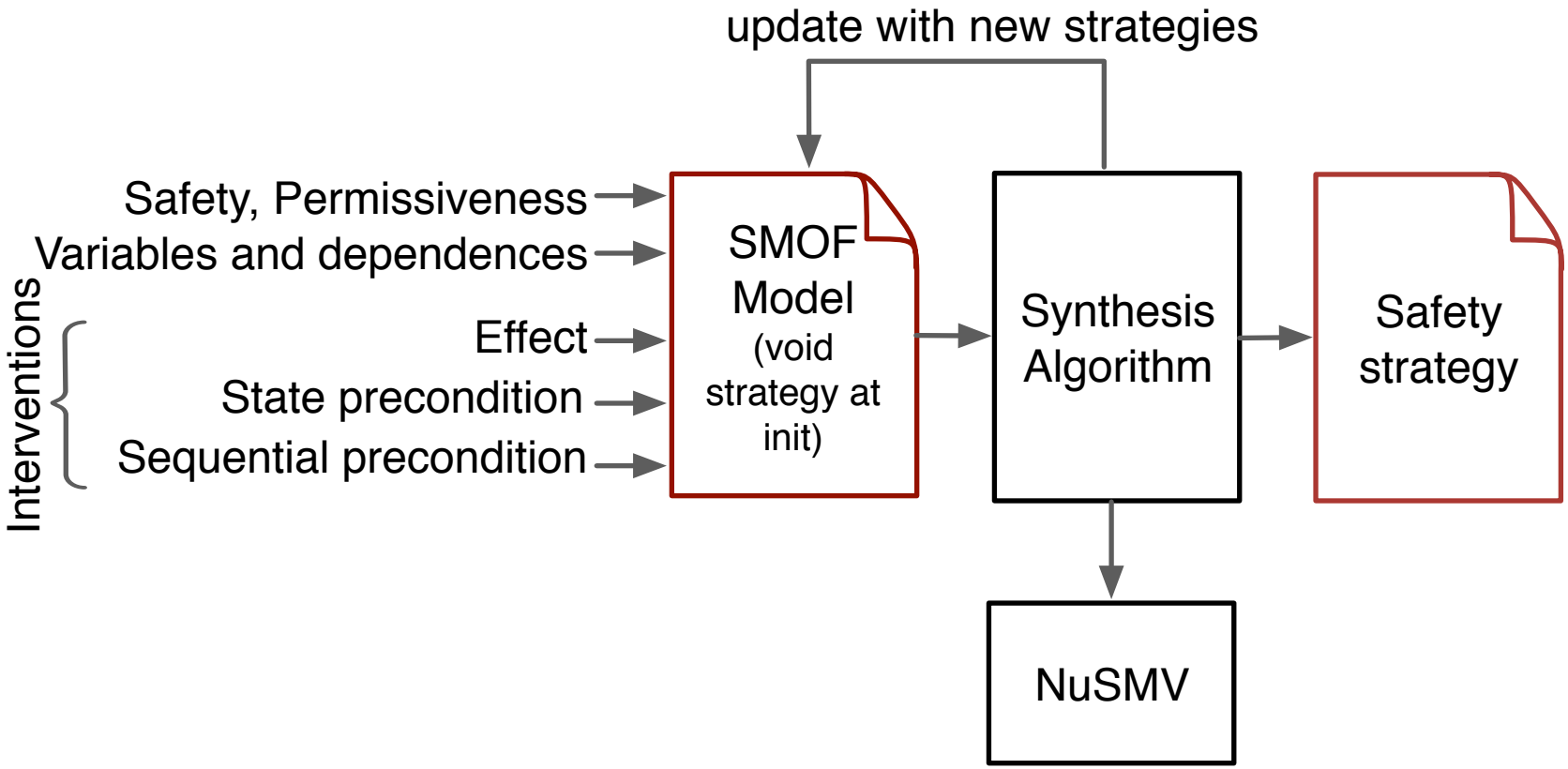
  Warning state – combination of interventions

- Required properties:
  - **Safe**: catastrophic states are not reachable
  - **Permissive**: non-catastrophic states are reachable

This strategy is safe,
but not permissive !



# Synthesis of strategies

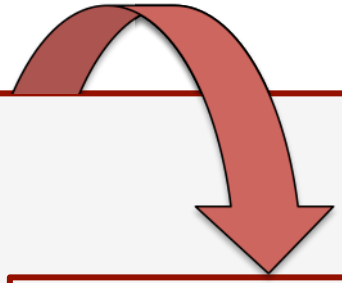


# Exemplary result

```
VAR
pf_vel: Continuity(0,2,0);
arm_pos : Continuity(0,1,1);

DEFINE cata:= (pf_vel=2 & arm_pos=0)
--Safety property
INVARSPEC !cata

-- Intervention(precondition, flag,
VAR
brake : Intervention(TRUE, pf_vel!=0
lock_arm : Intervention(arm_pos=1, T
```



```
-- Warning states
DEFINE flag_st_1 := arm_pos = 0 & pf_vel=1;
DEFINE flag_st_2 := arm_pos = 1 & pf_vel=1;
DEFINE flag_st_3 := arm_pos = 1 & pf_vel=2;
-------------------------------------------------------------

-- Strategy definition
DEFINE flag_brake := flag_st_2 | flag_st_3 ;
DEFINE flag_lock_arm := flag_st_1 ;
```

# A case study from FP7-SAPHARI

SAPHARI
Safe and Autonomous Physical Human-Aware Robot Interaction

- Mobile platform with an articulated arm
  KUKA LWR III arm + omnirob


- Safety Monitor can:
  - Block the arm
  - Engage the platform brakes

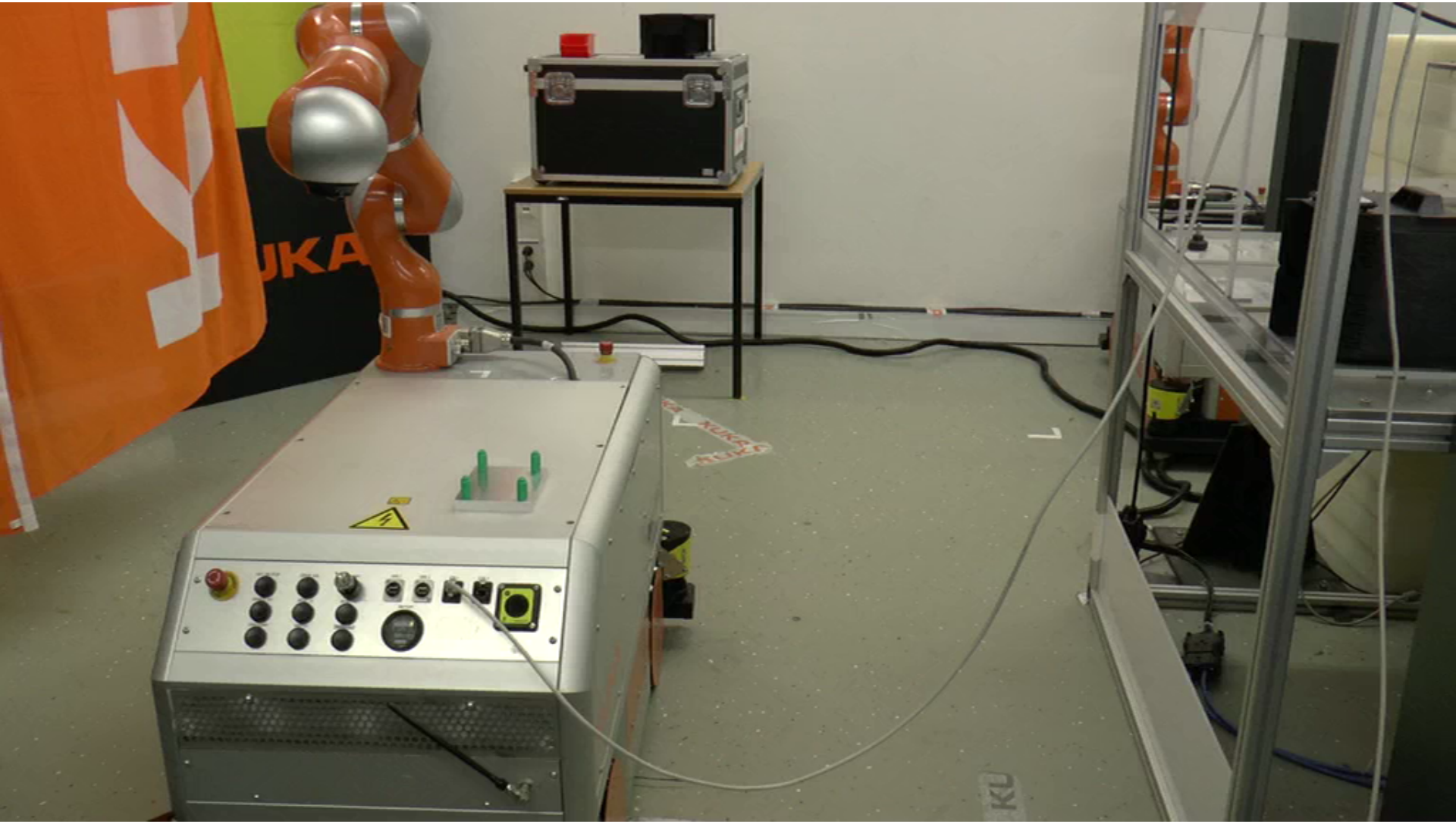
- Hazard Analysis with HAZOP-UML
  - 100 lines with a non-zero severity
  - 13 invariants, including:

"The robot arm must not be extended beyond the platform footprint when the platform moves."

# Case study safety invariants

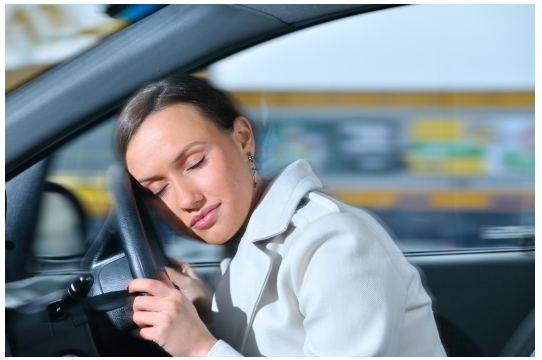
| | |
|---|---|
| SI1 | The velocity of robot arm must not be greater than $V_0$. |
| SI2 | The velocity of robot platform must not be greater than $V_1$. |
| SI3 | The robot must not enter the restricted area. |
| SI4 | The robot platform must not collide with a human. |
| SI5 | The robot arm must not be extended beyond the platform footprint when the platform moves. |
| SI6 | A gripped box must not be tilted more than $\alpha_0$. |
| SI7 | A collision between a human and the robot arm must not hurt the human. |
| SI8 | The velocity of any point of the robot must not be greater than $V_2$. |
| SI9 | The robot arm must not drop a box. |
| SI10 | The robot arm must not clamp human parts. |
| SI11 | The robot gripper must not clamp human parts. |
| SI12 | The robot must not override boxes laid on tables, shelves and robot storage. |
| SI13 | The robot must follow the hand-guiding. |

# The safety monitor in action



# To conclude

- Autonomous robotic functions development Vs Trusting autonomous systems
  - E.g. planners and residual fault
  - High recent new developments in robotics… new challenges for dependability means
- Active Safety monitoring
  - A method/tool
  - Rigorous development
- Needs for high integrity level (certification) of the monitor
  - E.g. KUKA: triplicated safety monitor board, MISRA standard for software
- Explore distributed active monitoring and safety rule synthesis in autonomous architectures (decisional level to functional level)



# Conclusion

+ SMOF provides a systematic and formal approach for the expression of safety rules

+ Dev. of a tool (no combinatorial explosion of the algorithm with acceptable performance )

- Level of expertise impact model expression, and thus synthesis

- Monitoring limited to the functional level

Future directions : several warning regions, interventions and observation located at different layers (hardware and software) with different integrity levels