Safety monitoring for dependable autonomous systems

Jérémie Guiochet

LAAS-CNRS, Université de Toulouse
France

Jeremie.guiochet@laas.fr

http://homepages.laas.fr/guiochet

Workshop on *Architecture and Safety for Autonomous Systems* @SCSSS, Stockholm, Sweden, 22-23 May 2017

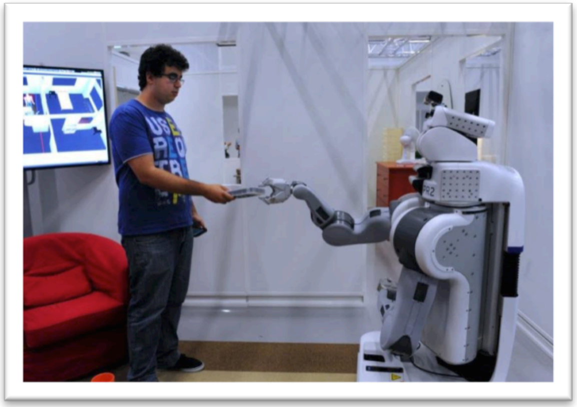
Dependable robots@laas

- Phds :
  - Execution Monitoring (2005) , Diverse task planning (2007), Robustness testing (2011), Safety monitoring (2012), Safety analysis for human-robot interactions (2015), Safety monitoring (with synthesis) (2015), Testing autonomous robots in virtual worlds (2017), Multi-level safety monitoring

- Recent collaborative 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

Autonomous systems

- Autonomy is the ability of sensing, perceiving, analyzing, communicating, planning, decision-making, and acting, to achieve assigned goals
- Autonomy level determined by
  - complexity of the mission
  - degrees of difficulty of the environment
  - levels of operator interactions
- Automatic (speed regulation) / Autonomous (cruise control)



Automatic

Autonomous

Can we trust autonomous systems ?

NB: Can we trust auto* systems ?
e.g., Toyota US trial, Tesla

Toyota Lexus, 2009

Main hazards :

- Confidence in decisional layers
  - Faults in inference mechanisms or knowledge base
  - Uncertain reaction in adverse situations (heuristics)
- Long term behavior and emerging properties (impossible to simulate/forecast)
- Integrity of localization / perception HW and SW

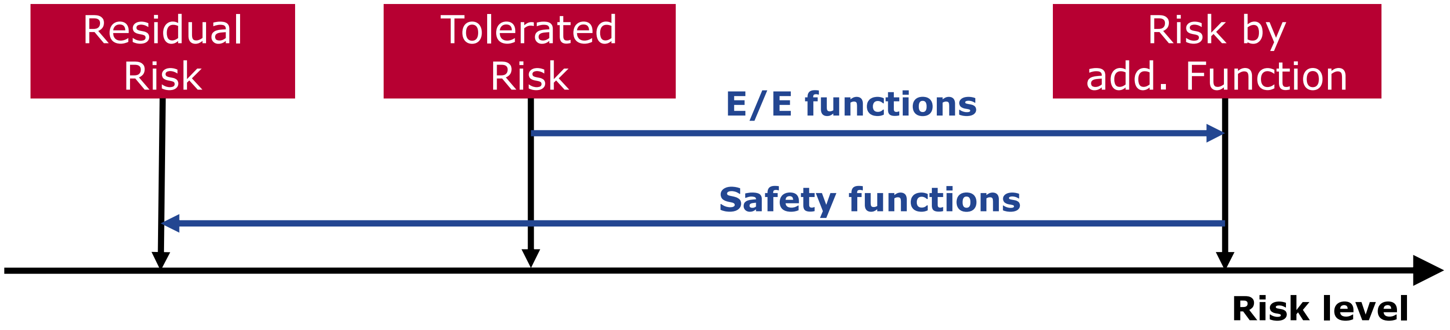
No technical standards, few regulations

- UAV regulations
- Self driving cars (new federal US Automated Vehicles Policy – September 2016)

Tesla, 2016

A popular form of fault tolerance: active safety monitoring

- Run-time monitoring of the system + actions to keep it in a safe state
- Implemented in most industrial processes as a “safety function”

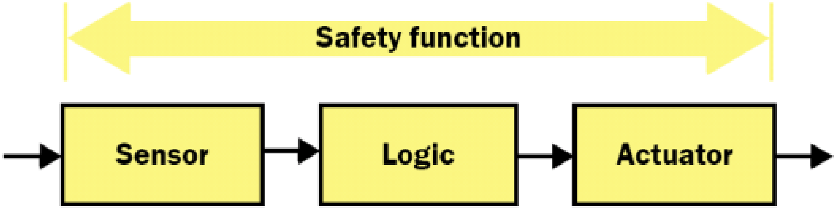


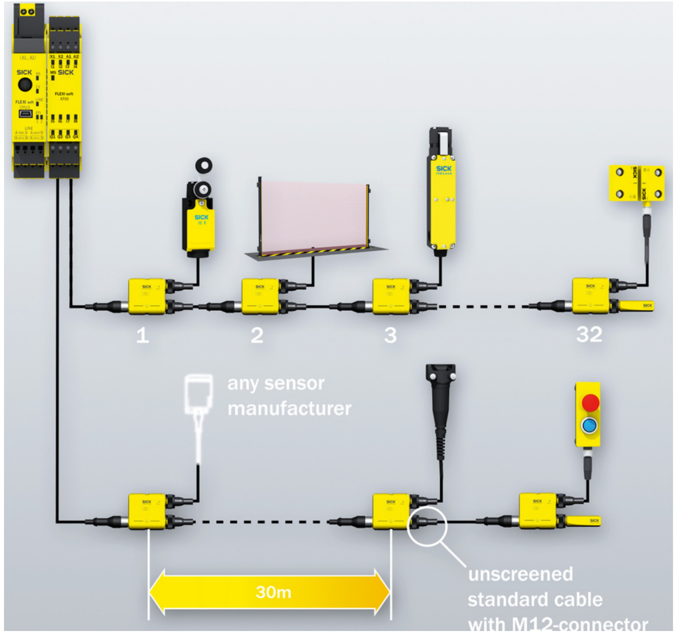
Source: IEC 61508:2010

Safety monitors for advanced applications: two main issues

- Safety layers with required *integrity level* to guarantee *safety properties* (runtime verification)
  - Issue#1 : Integrity of the HW and SW (perception/ control/actuators)
    - Standardized approaches (e.g. ISOIEC61508, or ISO26262 or ISO13849) -> more complex perception and reaction functions… Applicability ?
  - Issue#2 : Safety rules identification
    - Multifunction and autonomous systems -> Complex rules that could be non consistent. Research approaches (e.g. use of formal tools to synthetize safety rules) -> Applicability ?

Issue#1 : Monitor HW/SW integrity





Issue#1 : Monitor HW/SW integrity

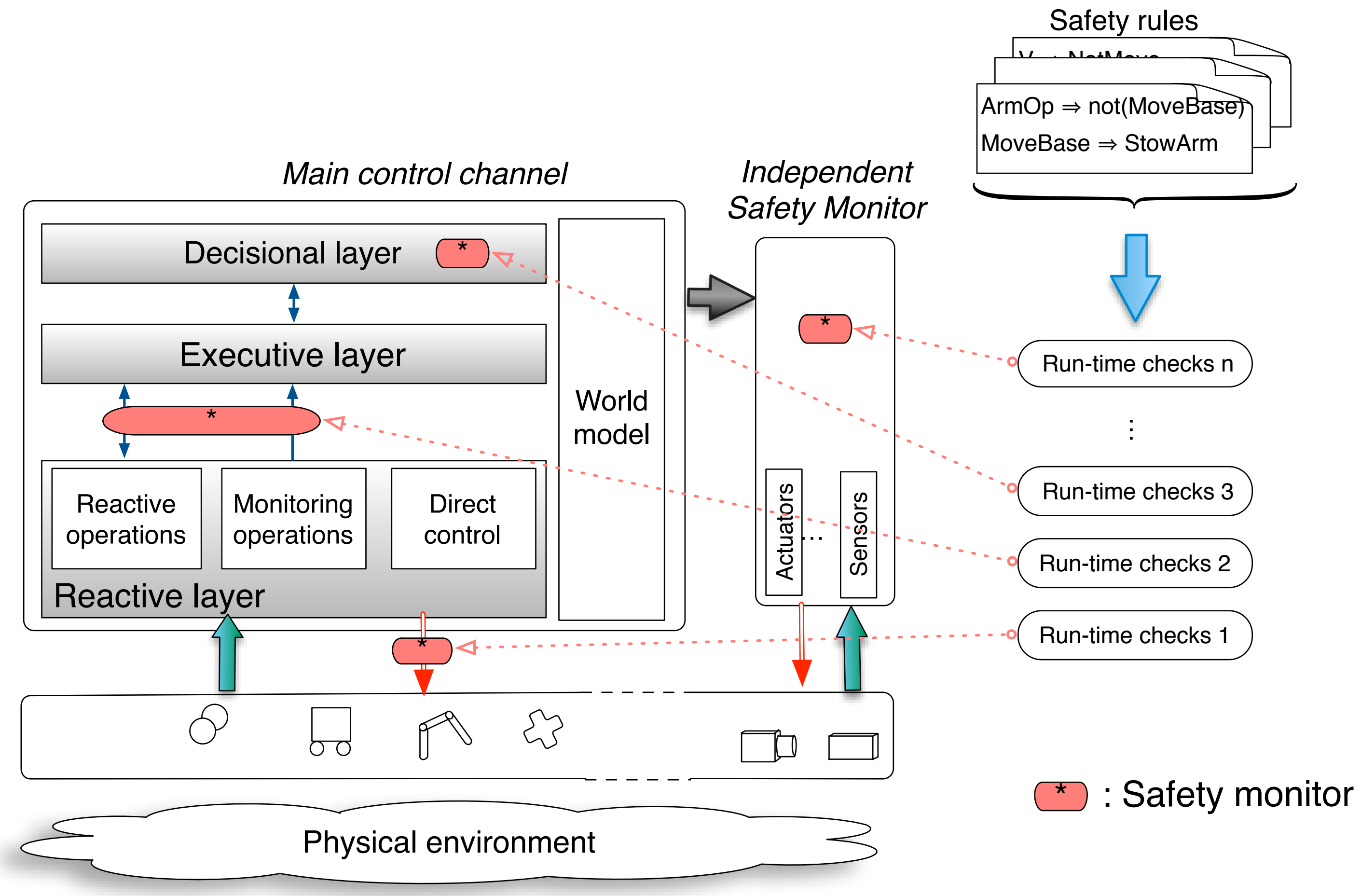
- Yes but…
  - Sensors: lasers, video, 3D perception, video
  - Logic: video treatment, optimization algorithms
  - Actuators: variable stifness actuators in robotics
- Complexity too high, low ressources (place, power, etc.)
- For now robotics designer stick to the "EU Machinary directive" with basic safety functions (e.g. High speed -> remove power)

Issue#1 : Monitor HW/SW integrity

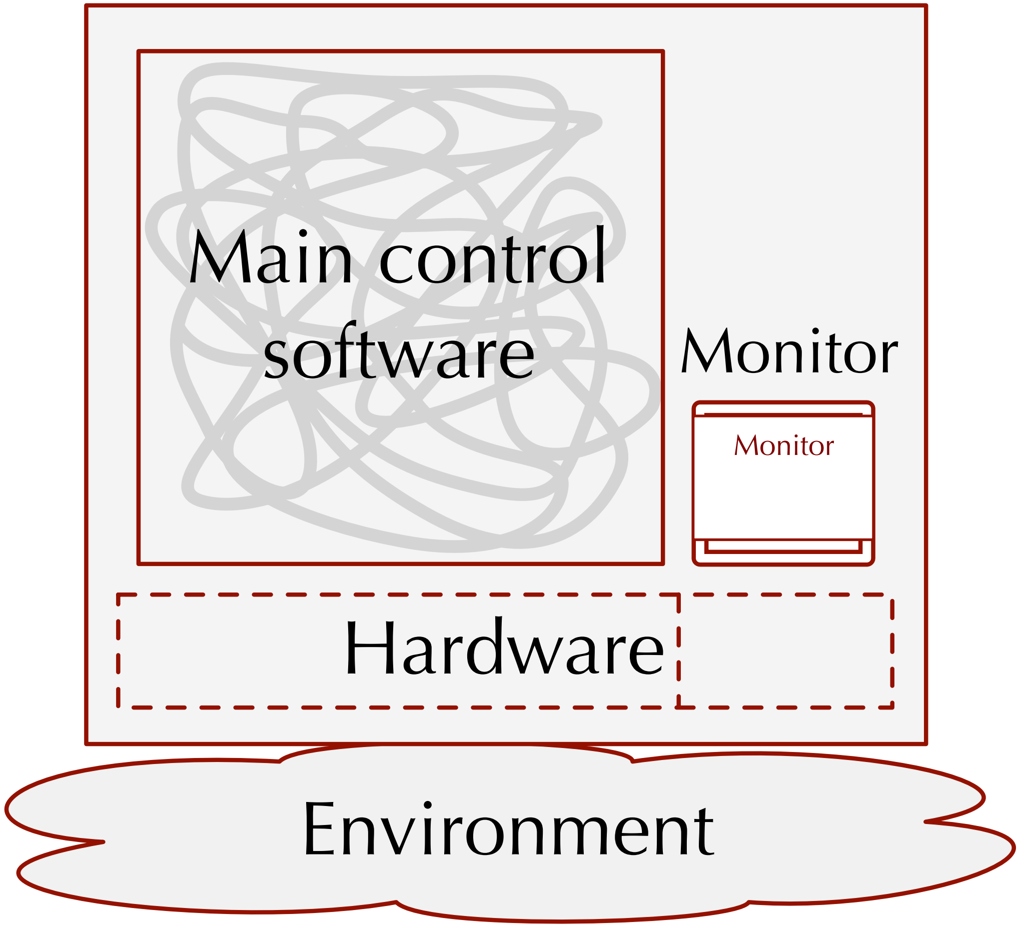
- Certification of safety monitors is a compromise
  - A simple but certifiable monitor
  - A complex but not certifiable monitor

	Hazardous situations coverage	HW SW Certification
Simple monitor	No	Yes
Complex monitor	Yes	No

Issue#2: Safety monitors rules



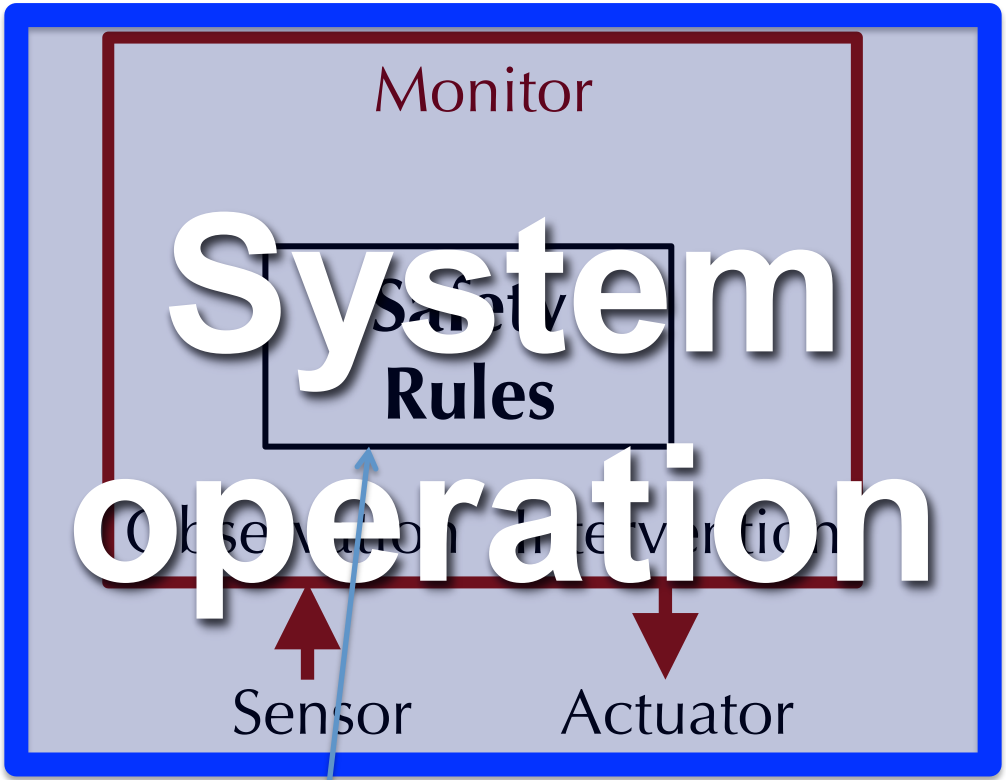
An example of a solution for issue#2: Active independant safety monitor



Safety Rules

Properties required from the monitor:

- Safety
- Permissiveness



Monitor

Safety Rules

Observation

Intervention

Sensor

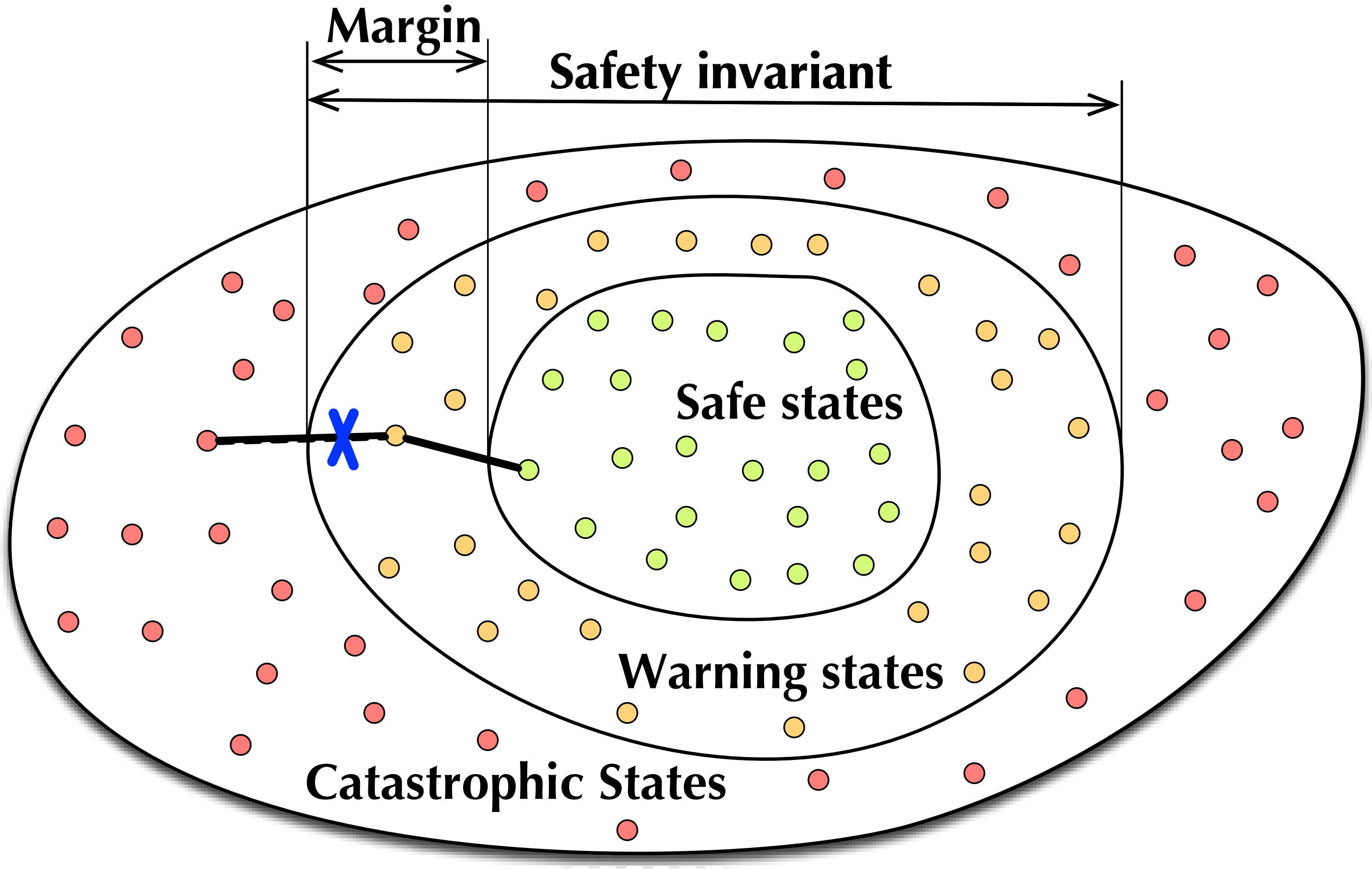
Actuator

System operation

System design

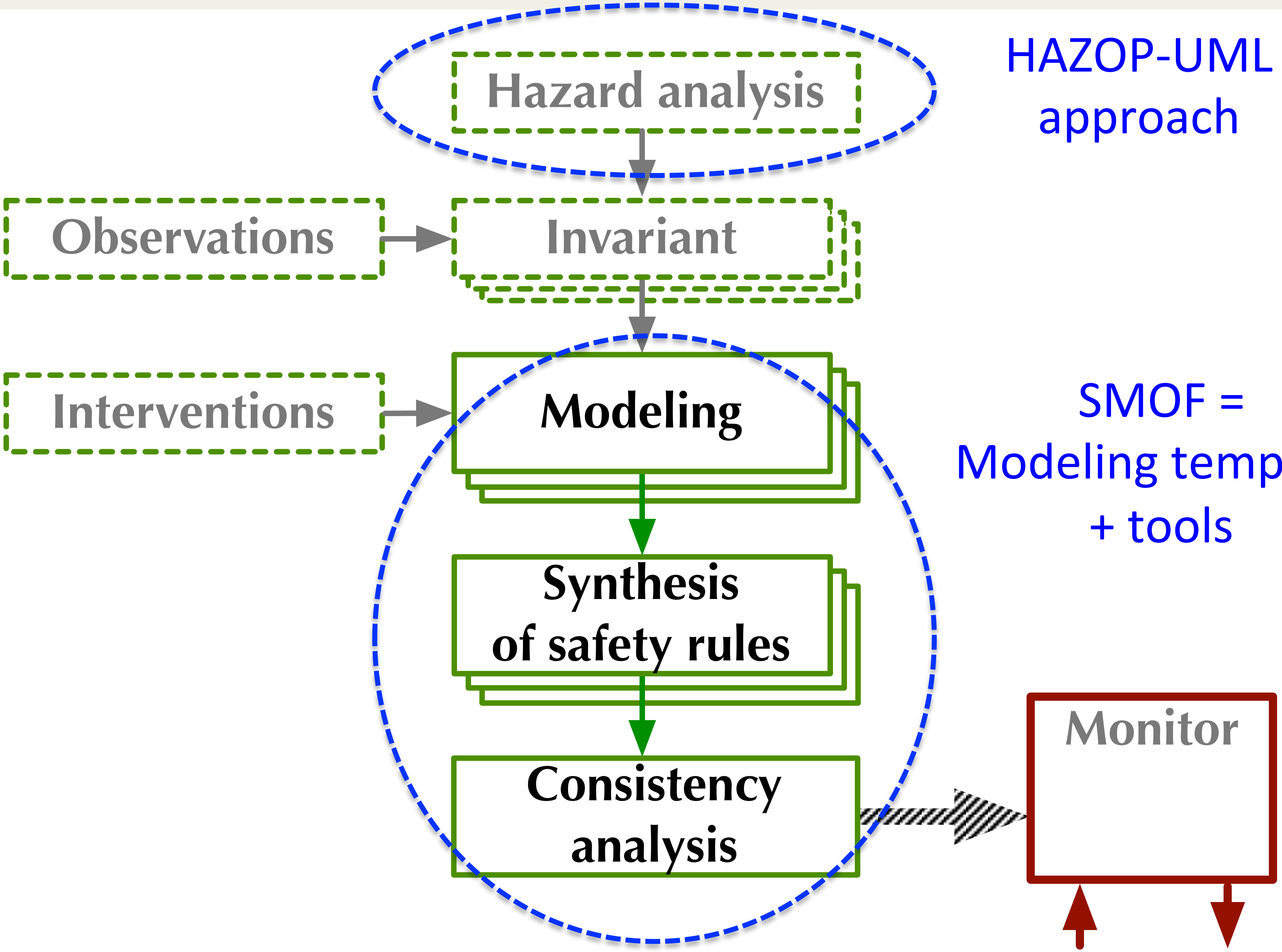
⇒ Specification of the safety rules

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

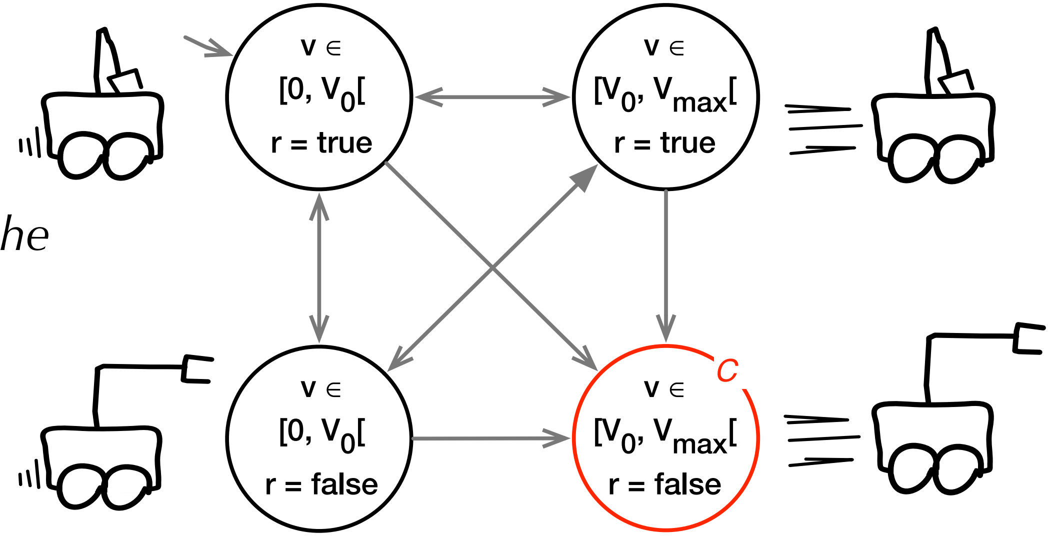
Method



HAZOP-UML approach

SMOF = Modeling template + tools

Toy example

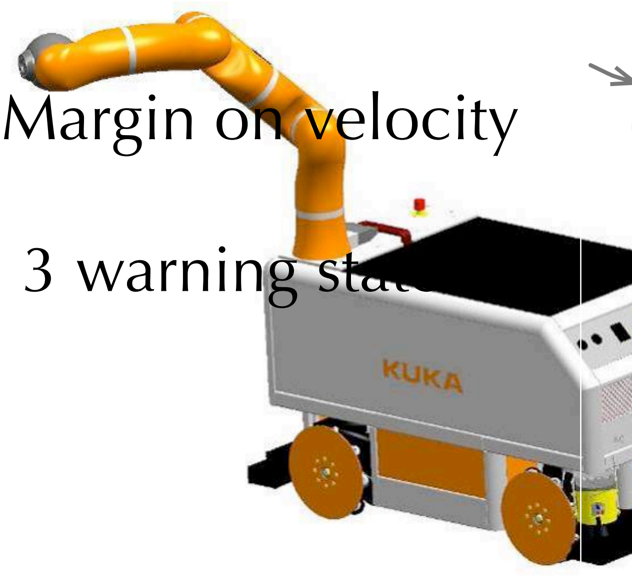

v ∈ [0, V_0[r = true
v ∈ [V_0, V_{max}[r = true
v ∈ [0, V_0[r = false
C
v ∈ [V_0, V_{max}[r = false

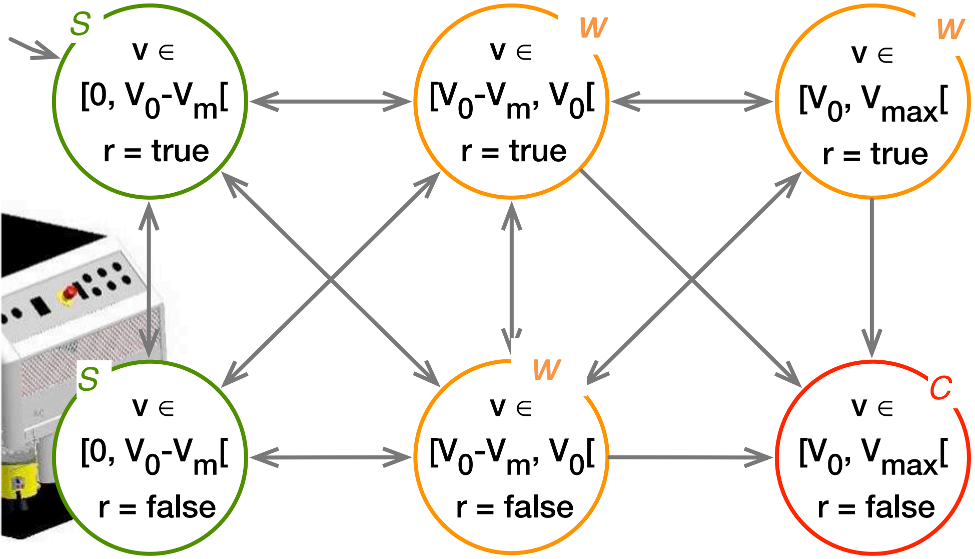
The robot arm must be folded when the platform velocity is greater than V_0

$$(r = \text{true}) \vee (v < V_0)$$

Margin on velocity

3 warning sta




S
v ∈ [0, V_0-V_m[r = true
W
v ∈ [V_0-V_m, V_0[r = true
W
v ∈ [V_0, V_{max}[r = true
S
v ∈ [0, V_0-V_m[r = false
W
v ∈ [V_0-V_m, V_0[r = false
C
v ∈ [V_0, V_{max}[r = false

Applicability of safety rules synthesis

Source code of the synthesis algorithm : https://www.laas.fr/projects/smof/

- In FP7-SAPHARI project (robotic co-worker)
  - 10 rules with maximum 3 variables
- In H2020 CPSELAB project (airport light measurement mobile robot)
  - All rules ok, except one rule with more than 8 variables -> no synthesis (but the tool was used to check a rule consistence)

Conclusion

- Safety monitors as "certified safety function" might be a good solution (when no guarantee can be delivered for the main autonomous controller)
- 2 main open issues
  - HW SW integrity
  - Safety rules identification

Biblio

- Open source project and scientific publications available at:

https://www.laas.fr/projects/smof/