SMOF
Safety Monitoring Framework

Jérémie Guiochet
LAAS-CNRS
Université de Toulouse
France
jeremie.guiochet@laas.fr
http://homepages.laas.fr/guiochet
https://www.laas.fr/projects/smofo/

WORKSHOP GdR MAC/GdR Rob, CNAM, Paris, 23 mai 2019
This is it ...
Hazards

• Internal faults in decisional layers
  – Inference design or programming faults
  – Knowledge: neglected situations, bad decision criteria, programming faults, etc.

• Uncertainties in perception, decision, and action functions (SW+HW)

• Unexpected adverse situations
  – Environmental conditions (light, obstacles, surface, etc.)
  – Human actions (attacks, unexpected behavior, etc.)
  – Other systems (multi-robots, multi-agents)
Dependability concepts

Attributes
- Availability
- Reliability
- Safety
- Confidentiality
- Integrity
- Maintainability

Security

Means
- Fault Prevention
- Fault Tolerance
- Fault Removal
- Fault Forecasting

Threats
- Faults
- Errors
- Failures

Dependability

Jean-Claude Laprie

A. Avižienis, J.-C. Laprie, B. Randell, C. Landwehr, 2004
## Means for dependability

<table>
<thead>
<tr>
<th>Means</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preventing occurrence or introduction of faults</td>
<td>Fault Prevention</td>
</tr>
<tr>
<td>Reducing the presence of faults</td>
<td>Fault Removal</td>
</tr>
<tr>
<td>Estimating the present number, the future incidence and the likely consequences of faults</td>
<td>Fault Forecasing</td>
</tr>
<tr>
<td>Delivering correct service in spite of faults</td>
<td>Fault Tolerance</td>
</tr>
</tbody>
</table>

- **Fault Avoidance**
- **Fault Acceptance**
Fault removal and tolerance with verification

- **Requirements / Analysis / Design**
- **System behavior model**
- **System properties**
- **Model checking**
  - Formal verification of the model (model execution)
- **Test**
  - Verification of the effective execution of the code
- **Active safety monitoring**
  - Run-time monitoring

- **Safety Monitoring Framework**

Example binary numbers: 1110001, 1100010, 1010100, 0101001
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”

FAQs

What are safety functions?
The safety function defines how safety measures will reduce the risk. A safety function must be defined for each hazard that has not been eliminated by design measures. It is necessary to provide a precise description of the safety function to achieve the required safety with reasonable effort. The type and number of components required for the function are derived from the definition of the safety function.

Examples of safety functions:

- Temporarily preventing access (Door interlock with locking at the injection mold machine)
- Initiating a stop (Access protection at a vulcanizing system for tires)
- Initiating a stop and preventing start (Access protection and presence detection on a turntable system)
- Differentiating people/material (Access protection in an automatic transport system for autochassis)
- Initiating a stop (Hazardous area protection at a automated guided system)
- Preventing entry/access permanently
- Retaining parts/substances/radiation
- Preventing start
- Preventing an unexpected start
- Monitoring machine parameters
- Disabling safety functions manually and for a limited time
- Combining or switching safety functions
- Initiating a stop if a defined speed is exceeded
- Initiating a stop if a defined moving direction is breech
- Safe brake control

More information:
- Brochure “Guide Safe Machinery”
Active independant safety monitor

Environment

Hardware

Monitor

Main control software
Properties required from the monitor:
- Safety
- Permissiveness

⇒ Specification of the safety rules
• A safety rule assigns interventions to warning states
• A **strategy** is a set of safety rules intended to ensure an invariant
Toy example

The robot arm \((r)\) must be folded (true) when the platform velocity \((v)\) is greater than \(V_0\)

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

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

State precondition

Sequential precondition

Effect
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!
Consistency analysis

Synthesis of safety rules

Modeling

Invariant

HAZOP-UML approach

SMOF = Modeling template + tools

Process

Hazard analysis

Observations

Interventions

Consistency analysis

Monitor
Synthesis of strategies

Interventions → Synthesis algorithm → Safe and permissive strategies
Warning states → addition of a strategy → Verification of properties → NuSMV
Model → Model
Branch & bound algo
Modeling with SMOF

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

```plaintext
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);
```
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, effect)
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);

-- 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;
SMOF extensions

SMOF returns:
No solution

No safe strategy

Change safety req. ← User defined

Change observation(s) ← User defined

Change intervention(s)

Change perm. req.

Permissiveness tuning

Diagnosis

Intervention suggestion

User defined

L. Masson (University of Toulouse)

Interactive Elicitation of Safety Rules

02/21/2019 14 / 45
A case study from Kuka

- Mobile platform with an articulated arm
  KUKA GmbH
  SAPHARI-FP7

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

- HAZOP Analysis
  - 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

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


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) – open source: https://www.laas.fr/projects/smof/

- Level of expertise impact model expression, and thus synthesis
- Monitoring limited to the functional level