deep se
dependable evolvable pervasive software engineering group

Istituto di Tecnologie Industriali e Automazione
Consiglio Nazionale delle Ricerche

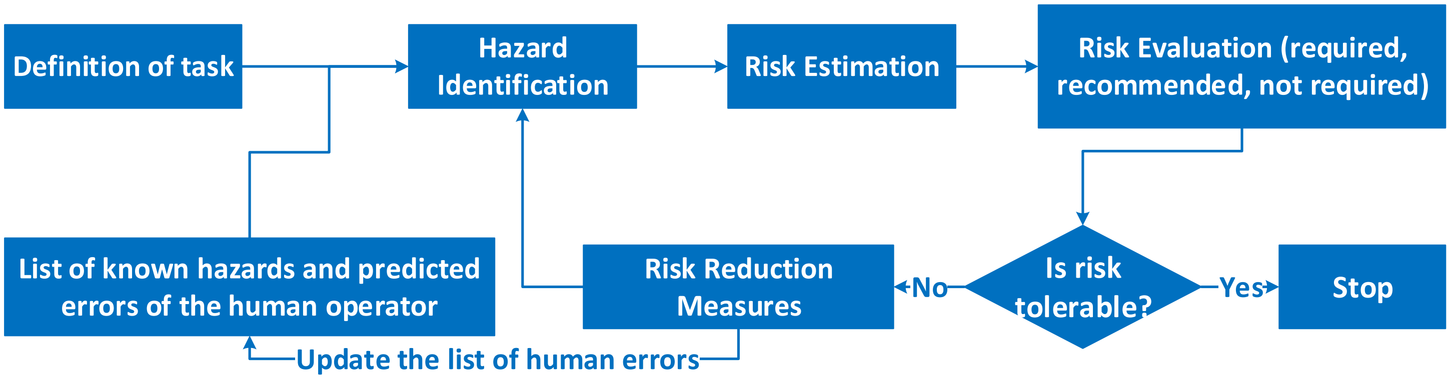
SAFER-HRC: Safety Analysis through Formal vERification in Human-Robot Collaboration

Mehrnoosh Askarpour[1], Dino Mandrioli[1], **Matteo Rossi[1]**, Federico Vicentini[2]
[1]Dipartimento di Elettronica e Informazione, Politecnico di Milano
[2]Istituto di Tecnologie Industriali e Automazione, CNR

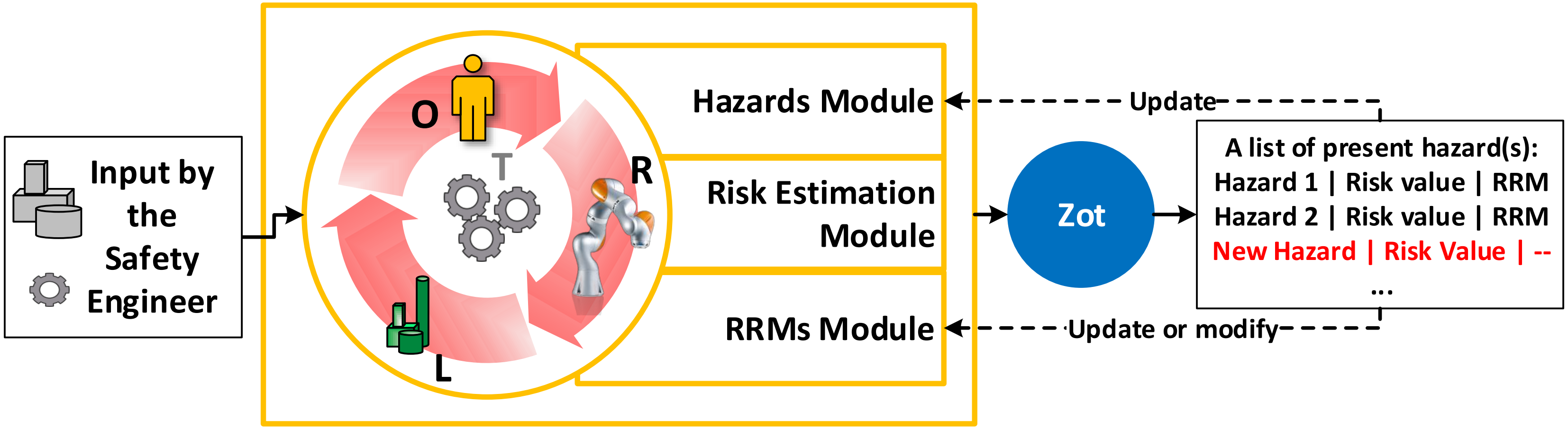
Introduction

- Human-Robot Collaborative (HRC) applications are novel applications in which humans and robots **intentionally** interact
  - dangerous situations for humans can arise
- Safety assessment: determine that no hazardous situations exist or, if they exist, their level of risk is acceptable
  - a deep understanding of possible interactions between humans and robots is necessary
- Approach based on formal methods: exploit formal verification techniques to exhaustively analyze human-robot interactions

Classic risk assessment approach



Overview of SAFER-HRC



Background: The TRIO logic

- TRIO is a first-order linear temporal logic
  - with a metric notion of time
  - the time domain can be discrete or dense
  - here we focus on a discrete time domain
- The TRIO specification of a system consists of a set of TRIO formulae
  - The formulae state how items are constrained and how they vary over time
- Automated formal verification carried out through the Zot bounded satisfiability checker

TRIO temporal operators

Operator	Definition
$\text{Past}(F, t)$	$t \geq 0 \wedge \text{Dist}(F, -t)$
$\text{Futr}(F, t)$	$t \geq 0 \wedge \text{Dist}(F, t)$
$\text{Alw}(F)$	$\forall d : \text{Dist}(F, d)$
$\text{AlwP}(F)$	$\forall d > 0 : \text{Past}(F, d)$
$\text{AlwF}(F)$	$\forall d > 0 : \text{Futr}(F, d)$
$\text{SomF}(F)$	$\exists d > 0 : \text{Futr}(F, d)$
$\text{SomP}(F)$	$\exists d > 0 : \text{Past}(F, d)$
$\text{Lasted}(F, t)$	$\forall d \in (0, t] : \text{Past}(F, d)$
$\text{Lasts}(F, t)$	$\forall d \in (0, t] : \text{Futr}(F, d)$
$\text{WithinP}(F, t)$	$\exists d \in (0, t] : \text{Past}(F, d)$
$\text{WithinF}(F, t)$	$\exists d \in (0, t] : \text{Futr}(F, d)$
$\text{Since}(F, G)$	$\exists d > 0 : \text{Lasted}(F, d) \wedge \text{Past}(G, d)$
$\text{Until}(F, G)$	$\exists d > 0 : \text{Lasts}(F, d) \wedge \text{Futr}(G, d)$

ORL-module

- L: definition of the layout of the cell
  - subdivision in regions, their adjacency
- O: constraints on the operator
  - operator's body also divided in regions
  - avoid unnatural shapes, e.g.:

$$\mathrm{Alw}(head_{reg} = arms_{reg} \vee \mathrm{adj}(head_{reg}, arms_{reg}))$$

- R: constraints on the robot
  - possible relative position of robot elements, e.g.:

$$\mathrm{Alw}(\mathrm{R1}_{reg} = \mathrm{R2}_{reg} \vee \mathrm{adj}(\mathrm{R1}_{reg}, \mathrm{R2}_{reg}))$$

Tasks' model

- A task is broken down into *actions*
  - an action can be carried out by either the operator, or the robot
- Each action is described by 3 sets of formulae:
  - *pre-conditions*, which must hold when the action starts executing
  - *safety constraints*, which hold throughout the execution of the action
  - *post-conditions*, which hold at the end of the execution
- An action can be in one of several states: *not started*, *waiting*, *executing*, *paused*, *done*

Action sequencing

- Pre- and Post-conditions govern the order and possible parallelism in which actions can occur

- Example:

$$a_{screwdrive,preC} \Rightarrow a_{movetopallet,sts} = \mathrm{dn} \wedge a_{bringwp,sts} = \mathrm{dn}$$

Robot vs. Operator actions

- Robot is deterministic: as soon as it can execute an action, it does so:

$$a_{i,pmr} = ro \wedge \ a_{i,sts} = \mathrm{wt} \wedge a_{i,safC} \Rightarrow \mathrm{Futr}(a_{i,sts} = \mathrm{exe}, 1)$$

- Operators, on the other hand, are nondeterministic by nature (but we assume they cannot wait forever):

$$a_{i,pmr} = op \wedge a_{i,sts} = \mathrm{wt} \Rightarrow \mathrm{WithinF}(a_{i,sts} = \mathrm{exe} \vee a_{i,sts} = \mathrm{ns}, \Delta)$$

Hazard definition

- Suitable predicates are introduced in the model to highlight hazardous situations
- Example:

$$hazard_{headinjury} \Leftrightarrow head_{reg} = \mathrm{EE}_{reg} \wedge \neg(mode_{robot} = idle) \wedge \mathrm{EE}_{type} = screwdriver$$

Risk scoring

Consequences	Severity SE	Class CI (Fr+Pr+Av)					Frequency Fr		Probability Pr		Avoidance Av	
		3-4	5-7	8-10	11-13	14-15						
Death, loss of eye or arm	4						<=1h	5	Very high	5		
Permanent injury, loss of fingers	3						>1h to <=24h	5	Likely	4		
Reversible injury, medical attention	2						>24h to <=2w	4	Possible	3	Impossible	5
Reversible injury, first aid	1						>2w to <=1y	3	Rarely	2	Possible	3
							>1y	2	Negligible	1	Likely	1

Black area = RRM required
Gray area = RRM recommended
White area = RRM not required

$$hazard_{headinjury} \wedge (14 \leq CI_{headinjury} \leq 15) \rightarrow risk_x = 2$$

Risk Reduction Measures

- Hazards whose risk level is not negligible must be countered to reduce the risk level
- This can be done in many ways, and the choice rests with the safety engineer
- RRMs result in modifications to the model
  - e.g., modifications to the layout, such as adding barriers, which prevent certain regions to be reached by the operator
- Example of formalization of RRM:

$$\begin{aligned} RRM_{headinjury} \Leftrightarrow\ & \mathrm{R1}_{reg} = \mathrm{futr}(\mathrm{R1}_{reg}, 1) \land \mathrm{R2}_{reg} = \mathrm{futr}(\mathrm{R2}_{reg}, 1) \land \\ & \mathrm{EE}_{reg} = \mathrm{futr}(\mathrm{EE}_{reg}, 1) \end{aligned}$$

$$hazard_{headinjury} \Rightarrow \mathrm{Until}_w(RRM_{headinjury}, \neg hazard_{headinjury})$$

Formal Verification-based risk assessment

- Checking whether known hazards, with unacceptable risk levels, can occur in the system corresponds to checking property:

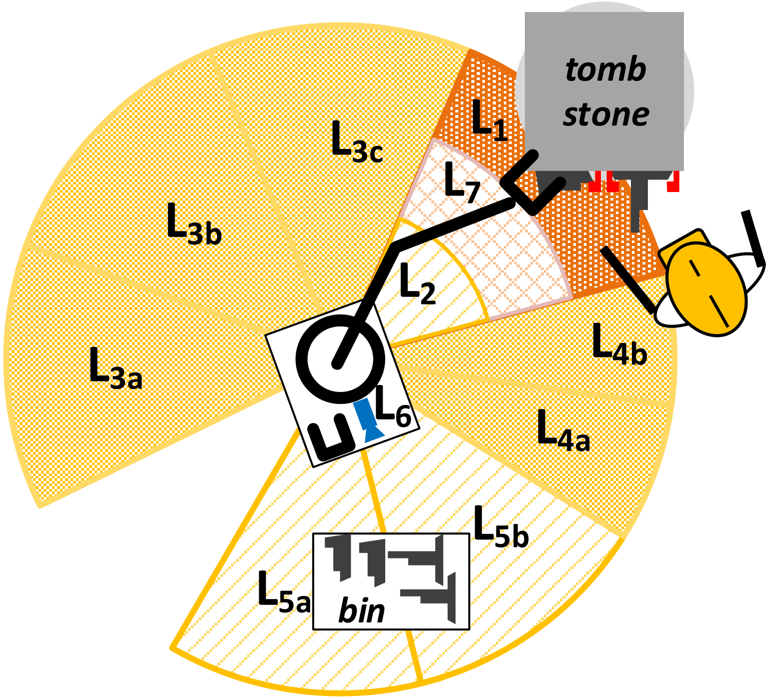
$$\mathrm{Alw}(\forall x(risk_x = 0))$$

  - if the property does not hold, the formal verification tool returns a trace (i.e., a sequence of actions) in which the risk exceeds the "negligible" level
- Checking whether it is possible that some safety constraint is violated, but no hazard is highlighted
  - i.e., some unconsidered hazard can occur

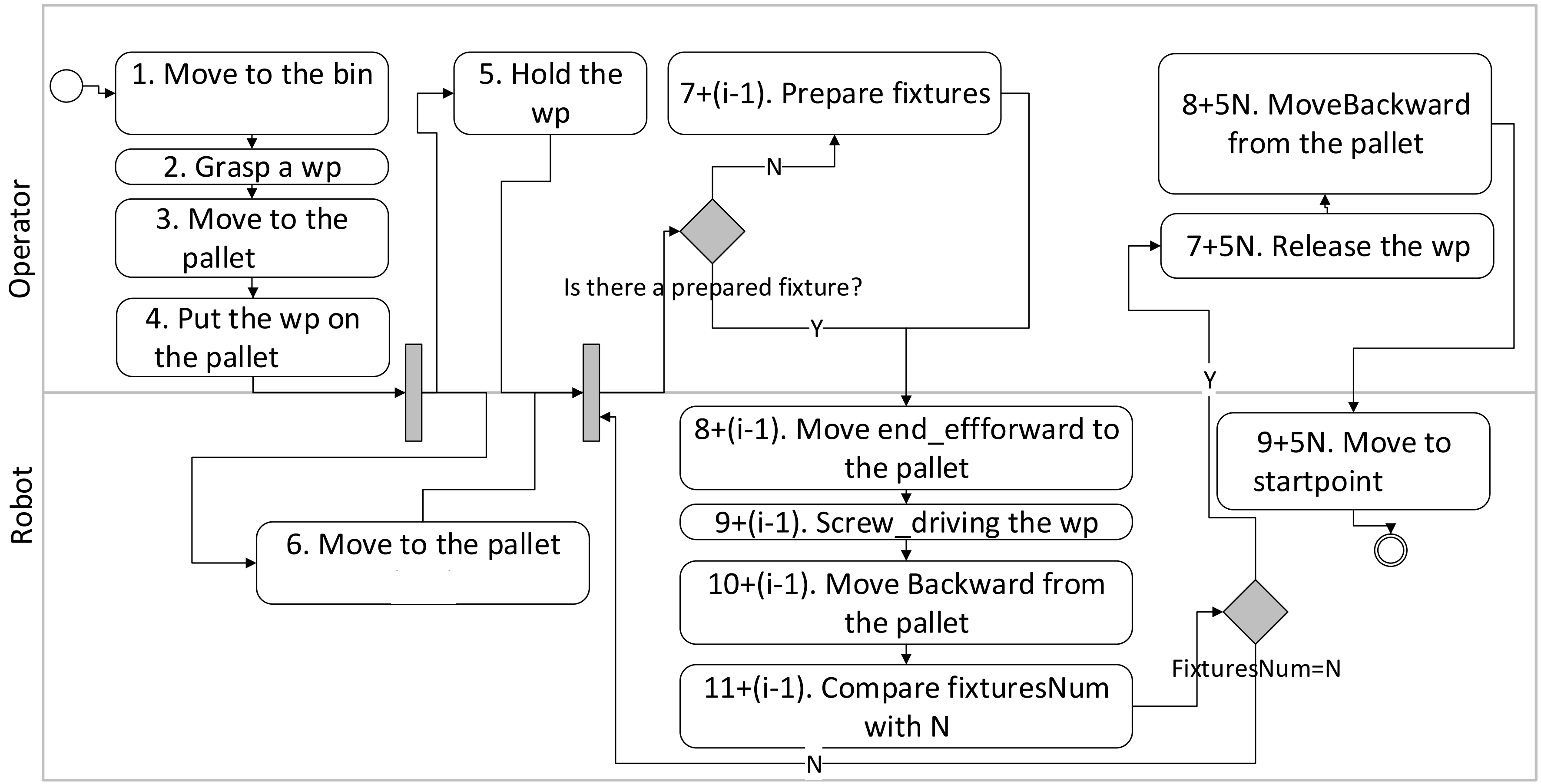
$$\mathrm{Som}\,(\exists i\,(a_{i,sts} = \mathrm{ps}\ \wedge\ (\forall x(\neg hazard_x)\ \vee\ \exists x(hazard_x \wedge \neg RRM_x))))$$

Case study

- Layout of the cell:



semi-formal task definition



Some possible hazards

- “Head hit by link R1 of robot”

$$hzd_{hhr1} \Leftrightarrow (\mathrm{R1}_{reg} = head_{reg} \vee \mathrm{R1}_{reg} = neck_{reg} \vee \mathrm{R1}_{reg} = shoulders_{reg}) \wedge \\ (\mathrm{R1}_{reg} = L_{3_a} \vee \mathrm{R1}_{reg} = L_{3_b} \vee \mathrm{R1}_{reg} = L_{3_c})$$

- “Head entrapped by R1”

$$hzd_{her1} \Leftrightarrow (\mathrm{R1}_{reg} = head_{reg} \vee \mathrm{R1}_{reg} = neck_{reg} \vee \mathrm{R1}_{reg} = shoulders_{reg}) \wedge \\ (\mathrm{R1}_{reg} = L_{5_a} \vee \mathrm{R1}_{reg} = L_{5_b} \vee \mathrm{R1}_{reg} = L_1)$$

Detected hazards

- operator mistakenly sends the activation signal to the robot before settling the part on the fixtures
- operator bends and brings head close to tomb while workpiece is being screwdriven, just as screwdriving is finishing and end-effector is about to move backwards from the tomb
- operator stays on the right side of the tomb while holding the workpiece to be screwdriven
  - this can lead to the operator getting entangled between the tombstone and a robot link or to getting hit by a sweeping robot arm

A look ahead

- Increase automated support to safety engineers
  - in creating formal models
  - in identifying problems
  - in suggesting countermeasures

- Improve and refine the formal model
  - incremental addition of details to remove possible false positives