Risk Assessment in Collaborative Robotics

Mehrnoosh Askarpour

DEIB, Politecnico di Milano, Milan, Italy
Mehrnoosh.askarpour@polimi.it

Problems and Objectives. In Human-Robot Collaborative (HRC) applications, operators and robots work in a common space. Their close interactions increase the likelihood of direct physical contacts either due to the intrinsics of the operations (e.g., tools, motions, etc) or behavior of the operator (e.g., mistakes, misuses, instruction misunderstanding, etc). Unintended physical contacts may lead to hazardous situations for the operator. In order to identify and avoid such situations as far as possible, we need to define and develop a comprehensive approach for safety assessment of HRC applications, that includes:

- Complying standards of risk analysis [10] and robotic safety [9].
- Ensuring the absence of unforeseen hazardous situations during design of systems by formal verification.
- Focusing on hazards caused by behavior of operators.
- Identifying hazards automatically, hence none is unconsidered, however keeping a human-in-the-loop attitude to interact with safety analyzers in order to use their operational perspective and experience.
- Estimating the gravity of identified hazard as a quantified value—named risk, which is computed on the basis of detailed analysis of the whole system (e.g., the severity of the injury that it causes).
- Suggesting proper treatments known as Risk Reduction Measures (RRM), which decrease the risk down to a negligible threshold.

Our approach to provide the above mentioned factors exploits formal methods for the specification and verification of system properties and is centered on the following steps: (i) Observe a real HRC case study to achieve a common understanding with robotic community about safety requirements, hazards and their relevant treatments, (ii) Define a coherent methodology to describe applications and verify if they provide a minimum level of safety, (iii) Build a modular model of the case study by use of a decidable fragment of the TRIO metric temporal logic [6] and apply an iterative verification according to the defined methodology, by use of the Zot tool [1], (iv) Generalize the model, so it will be tailored to different case-studies by least possible amount of changes required, (v) Evaluate the methodology and the generality of the model through new case studies, (vi) Create a framework upon them to help safety engineers to design a system, and iteratively equip it with as many RRMs as possible, so to make the design trustworthy, (vii) Extend the use of the framework to runtime, as an assistant to safety engineers to monitor the system, detect hazards and introduce suitable RRM "on the fly".

Literature Review. We performed an extensive literature search on current safety analysis research in robotics, and found that none of them recognizes the operator as a proactive factor. There is no focus on the safety violations that are caused by human activities and interactions with robots. Also there is no compatibility with international standards [9,10,11] and robotic community.

Two main approaches to determine the behavior of human operators in literature are: *cognitive* and *task-analytic* models [3]. The former involves a formal model of human cognition as part of the system model. [5] identifies principles that generate cognitively-plausible human behaviors such as repetition, mis-sequencing and intrusion, and generate formalized templates from cognitive psychology. However cognitive approaches are too hand-crafted and specific models and do not address human fallibility and plausible errors.

Task-analytic approaches, instead, model all the possible combinations of hazardous situations, regardless of their cognitive reasons, by hierarchical structures of tasks to be then decomposed into smaller functional units (atomic actions). Execution and sequence of actions are usually controlled through pre- and post-conditions. Examples of such approaches include the User Action Notation in [7], the ConcurTaskTrees Environment in [15] and the Enhanced Operator Function Model by [4]. These models are limited to normative behavior of human and does not reflect human errors.

In addition, traditional formal approaches such as FMEA (Failure Mode and Effects Analysis) and FTA (Fault Tree Analysis) are not well-suited for HRC applications because they do not capture hazards due to human factors or combinations of hazards. They also produce a large amount of repeated information and false positives and are dependent on the analyzers team which makes them less generic [12]. On the other hand, well-known informal solutions such as STPA (Systems-Theoretic Process Analysis) [13] and HAZOP (Hazard and Operability Study) [8] describe an overall frame for hazard identification and safety design for systems. Although they are used together with (semi-)formal solutions. For example [14] combines UML diagrams and HAZOP to identify hazards.

Other examples of combining formal and informal solutions are [16]. They use assistant robot case studies where operators do not collaborate with robots and are more passive factors in the system, thus lacking a strong model of the operator behavior. The former work addresses the impact of robot errors, while the latter focuses more on application of Brahms language to model a system.

Current Progress. Our proposed methodology called SAFER-HRC (Safety Assessment through Formal vERification in HRC applications) uses concepts of temporal logic and satisfiability checking to automate the classic risk assessment approach as much as possible [10]. SAFER-HRC starts from designing a formal model for a robotic application and incrementally refines it until required safety properties according to ISO 10218 are satisfied. The refinements are done via interacting with a safety engineer. The method can also be adopted at runtime by using added measures at design time and also pre-defined urgent reactions. We have introduced a preliminary version of it in [2]. Here its principles and some further developments are explained.

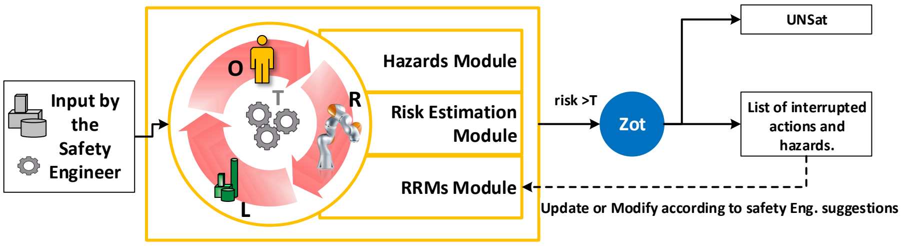

Input by the Safety Engineer
O
T
R
L
Hazards Module
Risk Estimation Module
RRMs Module
risk >T
Zot
UNSat
List of interrupted actions and hazards.
Update or Modify according to safety Eng. suggestions

Fig. 1: Overview of SAFER-HRC methodology.

In order to make it easier for safety engineers to build formal models, we introduced a general model, depicted in Fig. 1, which can be tailored to different scenarios with a minimum amount of required input. It consists of four main modules written in TRIO. The *ORL-module* includes formal descriptions for operator O, robot R and layout L. The descriptions of O and R are generic descriptions and there can be multiple instances of each of them, according to the case study (e.g., defining predicates that describe parts of body and robot). The rule set related to L divides the layout into fine-grained regions so that positioning of O and R and their movements are expressible.

The model also includes a set of rules T concerning the definition of tasks, what O and R are supposed to execute in L. The definition of tasks varies for different applications,thus the safety analyzer provide them as input. Tasks are broken down into a set of elementary actions that are the smallest possible functional units and are executed either by the operator or the robot. Each action is defined by its pre and post-conditions and can be in one of the following states at any given time: “not started” (ns), “waiting” (wt), “executing” (sfex), “executing together with at least one RRM ” (exe), “hold” (hd), “exit” (exit), and “done” (dn). The wt phase is defined only for operator actions that asserts *preC* of the action are true but operator hesitates to start the execution.The total state of the model is the Cartesian product of the state of actions and the value of predicates (e.g., positions, risk values).

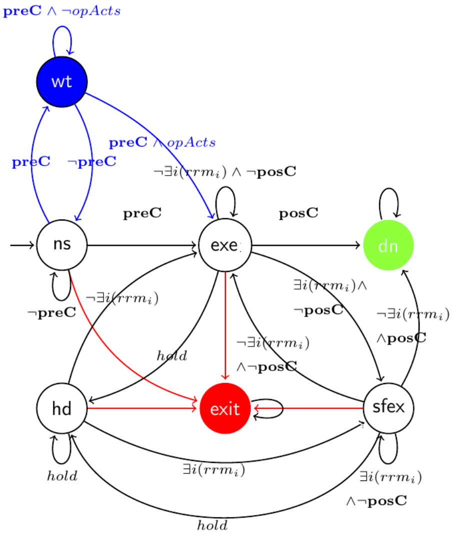


Fig. 2: In addition to their guards, black and blue edges also have $\exists i(\mathsf{risk}_i \leq T)$, while red ones have $\exists i(\mathsf{risk}_i > T)$.

Figure.2 displays a state diagram representing the behavior of actions and shows how their state changes.

The *Hazards module* includes formal definition of situations generating a *significant hazard*. It contains a set of formulae that use modal operators such as $\mathsf{Contact}_{,}(\xi,\omega)$ $\mathsf{Approach}_{,}(\xi,\omega)$ $\mathsf{Depart}_{,}(\xi,\omega)\cdots$, where ξ,ω are parts of R and O, to express the hazardous situations. Consequences of different hazards and measuring their severity are explained exhaustively in [9]. It also includes reasonably foreseeable operator errors which lead to hazardous situations. Hazard sources need to be updated as new situations and errors are detected.

The *Risk Estimation module* includes the information regarding the computation of risk of hazards according to the hybrid method reported in [10].

Finally, the *RRM module* defines all the RRMs that should additionally be present in the model when a hazard occurs in a trace of the model. At Each verification iteration, we check the full formal model explained above, against the property $\mathsf{Alw}(\forall x(\mathsf{risk}_x \leq T))$. The property actually asserts that for all the occurred hazards x, the value of risk should always be less than en acceptable threshold T. The Zot explores the state-space of the model and looks for states that conform with at least one hazard definition. If such state is detected and its risk value is non-negligible, then verification is failed and Zot outputs the problematic state so that we can refine the model.

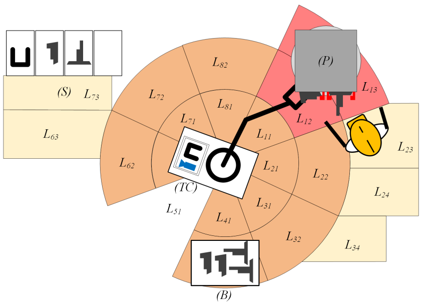
Case Study. We studied an assembly task in which workpieces should be machined into fixtures attached to a pallet. A small collaborative robot, Kuka lightweight manipulator, carrying an interchangeable screwdriver/gripper end-effector is installed on a cart in close position to both the pallet and the operator. The work-cell is equipped with cameras and sensors that detect the position of robot and operator. The work-cell of the scenario and its division is shown in Figure. 3(a).

In the task, one of the executors (operator/robot) is supposed to set a workpiece (wp) in place from a bin into the tombstone fixtures and holds the wp until the other executor (robot/operator) completes the screwdriving. The operator chooses the executor of each part of the task and sends relevant commands to the robot through an interface.

The user-provided description of the task is broken into several actions, each formalized through a set of pre-/post-conditions. This is done through an intermediate stage of translating the textual description to UML notation, in particular activity diagrams (see Figure.3(b))[1].

We found out about some hazardous situations that have been overlooked in earlier versions of the model, through the verification process. For example different timing behavior of the operator and robot (considering the timeout of the operator actions) can lead to different circumstances. Also replacing actions of a task with actions from elsewhere (e.g., with actions of different tasks, that may happen due to misunderstanding of the task instructions) by the operator creates situations that have not been anticipated initially.

[1] The complete formal model and the experiments can be found at repository.



(a)

$\mathbf{a}_1$ move to the bin (B); $a_{1,acr}$ = op;
$\mathbf{a}_2$ grasp a wp; $a_{1,acr}$ = op;
$\mathbf{a}_3$ bring the wp to the pallet; $a_{1,acr}$ = op;
$\mathbf{a}_4$ put the wp on the pallet; $a_{1,acr}$ = op;
$\mathbf{a}_5$ Hold the part until screwdriving is done; $a_{1,acr}$ = op;
$\mathbf{a}_6$ send activation signal to the robot; $a_{1,acr}$ = op;
$\mathbf{a}_7$ move from home to tombstone after receiving the operator's activation signal;
$\mathbf{a}_8$ prepare the jigs; $a_{1,acr}$ = op;
$\mathbf{a}_9$ move the end-effector vertically towards the pallet,
$\mathbf{a}_{10}$ screw the held wp;
$\mathbf{a}_{11}$ move the end-effector away from the tombstone;
$\mathbf{a}_{12}$ check the number of screwdriven jigs;
$\mathbf{a}_{13}$ if there are still some non-screwed jigs, go back to 8;
$\mathbf{a}_{14}$ Release the part after termination of 13; $a_{1,acr}$ = op;
$\mathbf{a}_{15}$ move to home.

(b)

Fig. 3: (a) The layout representation of the work-cell, where P is the pallet and B is the bin of workpieces. Different colors for each region on the layout, represents the type of hazards which the region is exposed to. For example entanglement of the operator between two layers is probable in red regions due to their physical structure. (b) The list of actions in case the operators chooses to do grasping and sends the commands to the robot to do the actions regarding to screwdriving completion. $a_{i,acr}$ is the executor of each action.

Further Work. SAFER-HRC takes into account also the concurrency of multiple actions, in particular actions that must be executed by the operator and by the robot in full coordination. It explores all the different order of execution of actions. For example, traces a_i, a_j and a_j, a_i achieve the same goal, but executing a_i after a_j causes a hazard that does not exist in the first trace (e.g., , a change in positioning of the operator w.r.t the robot). In the remaining year of my phD, we plan to exploit this feature of the work using larger scale and more complex case studies. Next we will develop a stand alone framework for interactive risk assessment of HRC applications for system designer and safety engineers. In a longer-term perspective, we aim to empower the framework with libraries that define most of the tasks that a robot performs (e.g., Kuka lightweight manipulator). Therefore, we believe our framework will suit for real HRC scenarios.

References

1. Zot: a bounded satisfiability checker. `github.com/fm-polimi/zot` (2012)
2. SAFER-HRC: Safety Analysis Through Formal vERification in Human-Robot Collaboration (2016)
3. Bolton, M.L., Bass, E.J., Siminiceanu, R.I.: Generating phenotypical erroneous human behavior to evaluate human-automation interaction using model checking. Int. J. Hum.-Comput. Stud. 70(11), 888–906 (Nov 2012), `http://dx.doi.org/10.1016/j.ijhcs.2012.05.010`
4. Bolton, M.L., Siminiceanu, R.I., Bass, E.J.: A systematic approach to model checking human-automation interaction using task analytic models. IEEE Trans.

Systems, Man, and Cybernetics, Part A 41(5), 961–976 (2011), http://dblp.uni-trier.de/db/journals/tsmc/tsmca41.html#BoltonSB11

5. Curzon, P., Rukšėnas, R., Blandford, A.: An approach to formal verification of human–computer interaction. Formal Aspects of Computing 19(4), 513–550 (2007), http://dx.doi.org/10.1007/s00165-007-0035-6
6. Furia, C.A., Mandrioli, D., Morzenti, A., Rossi, M.: Modeling Time in Computing. Monographs in Theoretical Computer Science. An EATCS Series, Springer (2012)
7. Hartson, H.R., Siochi, A.C., Hix, D.: The uan: A user-oriented representation for direct manipulation interface designs. ACM Trans. Inf. Syst. 8(3), 181–203 (Jul 1990), http://doi.acm.org/10.1145/98188.98191
8. International Electrotechnical Commission: IEC 61882, Hazard and operability studies (HAZOP studies)-Application guide (2001)
9. ISO: ISO 10218-2:2011: Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration. International Organization for Standardization, Geneva, Switzerland (2011)
10. ISO: ISO/TR 14121-2:2012: Safety of machinery – Risk assessment – Part 2: Practical guidance and examples of methods. International Organization for Standardization, Geneva, Switzerland (2012)
11. ISO: ISO/TS 15066:2016: Robots and robotic devices – Collaborative robots. International Organization for Standardization, Geneva, Switzerland (2016)
12. Joshi, G., Joshi, H.: FMEA and alternatives v/s enhanced risk assessment mechanism. International Journal of Computer Applications (2014)
13. Leveson, N.: Engineering a safer world: Systems thinking applied to safety. MIT Press (2011)
14. Machin, M., Dufossé, F., Guiochet, J., Powell, D., Roy, M., Waeselynck, H.: Model-checking and game theory for synthesis of safety rules. In: Proc. of HASE (2015)
15. Paterno', F.: Formal reasoning about dialogue properties with automatic support. Interacting with Computers 9(2), 173 – 196 (1997), http://www.sciencedirect.com/science/article/pii/S0953543897000155
16. Webster, M., Dixon, C., Fisher, M., Salem, M., Saunders, J., Koay, K.L., Dautenhahn, K., Saez-Pons, J.: Toward reliable autonomous robotic assistants through formal verification: A case study. IEEE Trans. Human-Machine Systems pp. 186–196 (2016)