2.7 – Using simulation

Practical guidance – cobots (collaborative robots)

Author: CSI: Cobot demonstrator project

Simulations capture a model of a system and its interactions with the real world over time, a model which can be experimented upon and queried [1]. Simulations offer the opportunity to gain insight into the behaviour of complex systems, before their implementation, deployment, or without disturbances to the real system. For collaborative systems, where human operators work alongside robots, various safety aspects can be explored without endangering the operator or machines through simulation.

The variety and versatility of modelling techniques allowed simulations to pervade multiple domains, from healthcare to autonomous vehicles. In particular, this led to an increasing use of simulation in manufacturing to support a wide range of activities, from factory layout optimisation to product design [2]. From the perspective of an assurance process, simulation allows for the collection of evidence to assess the safety of a system.

Simulations as they abstract a system and its internal and external interactions still need to provide a sufficient representation of said system. The notion of sufficiency depends on what needs to be modelled, and why it needs to be modelled. The simulation should capture the interactions relevant to the questions asked (the what), to a sufficient level of detail to faithfully answer the question (the why). In other words, the model should be a complete and representative one for the analysis at hand.

The following sections discuss the concerns related to the use of simulations as part of the safety assurance process. The guidance considers how the safety analysis can inform the simulation of situations to monitor, assess the coverage of configurations used during testing, and provision for out of scope situations. Without loss of generality, this guidance follows the process defined by the SASSI method for collaborative robots [3]. We further consider a simple use case, analysed in Section 1.2.1 of the Body of Knowledge, where a cobot is used to isolate operators from dangerous machinery. The operator hands over a component for processing performed by the cobot, that is welding through an automated welder.

SASSI method for cobots

The SASSI method, outlined in Figure 1, relies at its core on the notion of situation coverage [4]. It focuses on situations and situation components, identified during the safety analysis of a system, to guide both monitoring requirements on a simulation environment, and coverage criteria to assess the completeness of a test suite. The simulation is monitored for the occurrence of accidents, hazards, or undesirable situations, and guided towards the coverage of its configuration domain or towards safety occurrences.

The design of the system under analysis is the root of the process. It defines the simulated environment, acceptable operating conditions, and safety requirements. A safety analysis helps understand the nature of hazards in the system, and which safety situations should be prevented. Safety situations capture an undesirable state of the system. Hence, they inform the simulation on events and variables to expose such occurrences. Coverage of the situation space further guides the user in selecting interesting system configurations, to ensure a reasonable spectrum of operating conditions has been considered.

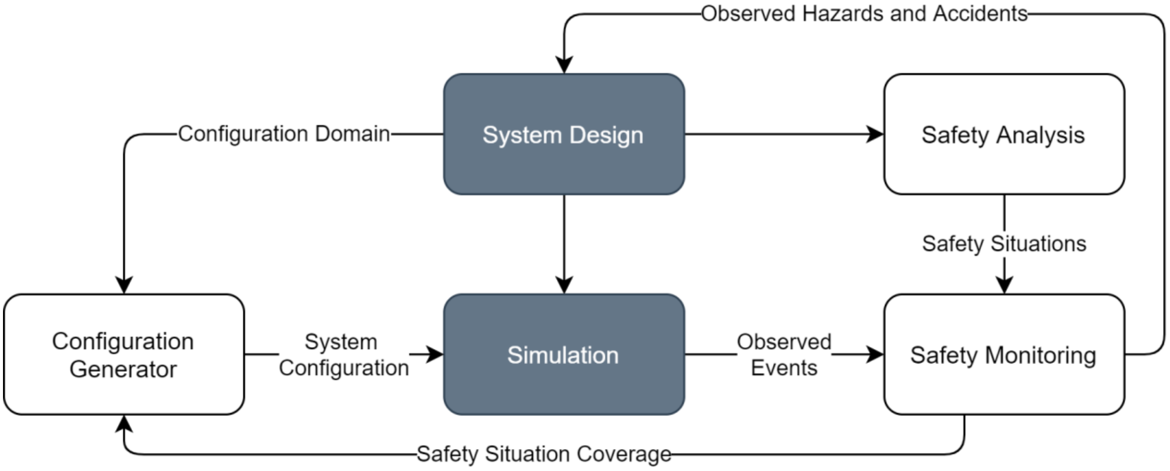

Figure 1. Overview of the SASSI technique positioning simulations in the safety assessment

Agent-based simulation is well suited to observing interactions as they occur on collaborative systems where human operators and robots, modelled as independent agents, interact with each other. The evaluation of the SASSI technique relied on such an agent-based simulator. Other types of simulations, such as discrete-event simulations, may be more suited to monitoring different properties but the general principles outlined in the following still apply.

Modelling and monitoring safety situations

The safety situations identified by the safety analysis of a system define requirements on the scope of simulations used as part of the safety assurance process. Hazards or accidents capture undesirable configurations of the environment. Those are micro-interactions, small-scale safety situations, occurring while the system is running. Simulations should thus provide for monitoring such occurrences, directly or indirectly capturing the required events and variables.

Informal or high-level situations provide little guidance on simulation requirements. The formalisation of a situation into a runtime monitoring specification [5] can help in identifying its core components and, in turn, the properties to be modelled and exposed by the simulation. As an example, consider the hazard H2 identified in Section 1.2.1 of the Body of Knowledge "*Individual or Objects enter dangerous areas*". This situation can be refined to highlight the case where "*The operator is close to an active welder*". The simulation should thus model two situation components i) the "*proximity of the operator to the welder*", and ii) the "*welder active status*".

The situation components help define a minimum set of events and variables to be modelled by a simulation, thus supporting an argument for the sufficiency of said simulation. However, simulations come with an inherent level of abstraction. Such abstractions should be considered as part of the assurance process to frame the scope of the simulation. An abstract safety monitor ideally subsumes its situation definition, over-estimating the occurrence of the situation. Consider the previously identified component "*proximity of the operator to the welder*". It can be modelled simply through the closest distance between the operator and the welder. The distance metrics abstracts existing barriers around the welder which might reduce the dangerous area. Assorted subsuming monitors might thus flag an issue even if the operator is close to the welder but protected by such a barrier.

A safety situation can be modelled by its components composed through a monitoring formalism, e.g. "*(distance(operator, welder)* ***less than*** *1m)* ***and*** *(welder status* ***is*** *active)*". This formalisation is an artefact for the safety assurance case, and an argument needs to ensure that the formalisation of the safety situation is valid. Observations from the simulation might result in refinements to the safety situations' definition, and vice versa. All changes to the simulation or monitored situations should be reflected where appropriate in the system design or the safety analysis results. Any processing from simulation outputs for monitoring purposes similarly implies a potential loss of precision, and it should similarly be justified in the safety argument.

The configuration of the simulation model should also not prevent situations of interest from being observed. This configuration impacts the granularity of the simulation. Consider as an example the time-granularity of the simulation. Some simulation models proceed at fixed timesteps, interpolating the events that occur between two steps. A large timestep, a low granularity simulation, is faster but might not account for short-lived events, such as collisions, and prevent the evaluation of tight timing constraints. A higher granularity will result in more precision at the cost of additional data and slower runtimes.

Situation coverage through simulations

The simulation configuration sets up the operating conditions for the system. These macro-level components define the initial situation and the scenario in the simulation. Each configuration parameter can potentially trigger different micro-level situations and responses in the system. Considered as a whole, the set of configurations explored across multiple tests can support the claim that a system is safe. However, this requires ensuring the system has been exercised in a variety of scenarios, i.e. testing achieves good coverage of micro- and macro-level situations.

The configuration domain limits acceptable variations in the system configuration. The domain identifies the parameters that can vary between deployments or executions, and the acceptable range of values for each parameter. As an example, this can cover the type of intersections and road users encountered by an autonomous vehicle, or the time of day and lighting conditions in a work cell. The configuration domain definition should be considered a component of the safety assurance case; variations outside the domain scope might result in unsafe behaviours, they should be documented and monitored.

Testing can initially focus on exploring the configuration domain to ensure that the system is safe under nominal conditions. The identified parameters can be modified and their impact explored through simulation-based testing. Full domain coverage is however intractable in the general case due to the sheer number of combinations and configurations. Design of Experiment techniques [6] scope the configuration domain to generate a set of partial covering experiments. Further sensitivity testing using surrogate safety metrics can help identify the key parameters for safety that deserve further consideration and variations [7].

The absence of observed hazards or accidents during testing is not sufficient to demonstrate the safety of a system. Good coverage of the configuration domain, macro-level components, indicates the system has been observed in a wide variety of initial configurations. Coverage of micro-level situations could further indicate the system successfully encountered specific events. Safety situations, while undesirable themselves, can help guide the definitions of such situations to be observed during testing. As an example, from hazard H2, "*Individual or Objects enter dangerous areas*", we can infer testing should cover the case where the operator approaches an active welder to assess the response of the system.

Emergent hazards and fault tolerance

Emergent hazards and faults result from complex interactions in the system and as such they might implicitly or explicitly be omitted from a simple simulation model. In the considered use case, an oily assembly might slip from the cobot gripper resulting in damage to the assembly itself, or other components it hits. Accurate simulations to capture such occurrences would require a complex physics model. Overall, costly, purpose-built models might be hard to integrate in general-purpose simulation frameworks.

It is important to understand the scope of the simulation to understand if it is suited to the monitored hazards. The safety assessment might not need to consider how such complex interactions occur, but whether the system's reaction is appropriate. In the previous example, it would be ensuring that the cobot speed and direction never results in an assembly turning into a high velocity projectile if it is unexpectedly released. Such a fault, as a result of slippage or gripper failure, can be injected into the simulation through a simple model to assess the system's response, i.e. randomly releasing the assembly during operation.

Faults are another example of factors raised by the safety analysis that need to be explicitly included in the simulation, in addition to the monitoring requirements. Fault injection during testing helps understand how the system tolerates faults. The inclusion of fault models in the simulation however increases the configuration space to be considered during testing. Given a limited testing effort, fault-centered testing should be proportional to the likelihood of faults or the severity of related hazards.

Summary of results

During the development and evaluation of the SASSI method, we identified a number of considerations for the use of simulations to support safety assurance cases in collaborative systems:

- Safety situations identified by safety and risk analyses help define requirements on the simulation environment, and break down the argument for the representativeness of the simulation into smaller situation components. Formalised situations can furthermore monitor the occurrence of hazards in the simulated system.
- Each model and simulation type comes with inherent limitations and requirements. Those need to be understood and argued as part of the safety assurance case to

ensure the simulation is suited to the evaluation. Similarly, intermediate data processing or formalisation steps should be documented and understood.

- The configuration domain bounds the acceptable, initial situations for the environment under evaluation. While full domain coverage is highly unlikely, it supports coverage arguments and rationalises the configuration used during simulation-based testing.
- Provision should be made for faults and complex situations out of the scope of the general-purpose simulation models. Fault models are a suitable proxy to evaluate the resilience of a system.

References

- [1] Chung, C.A. (Ed.). (2003). Simulation Modeling Handbook: A Practical Approach (1st ed.). CRC Press. https://doi.org/10.1201/9780203496466
- [2] D. Mourtzis, M. Doukas, D. Bernidaki, Simulation in Manufacturing: Review and Challenges, Procedia CIRP, Volume 25, 2014, Pages 213-229 [pdf copy]
- [3] B. Lesage, R. Alexander, Rob, SASSI: Safety Analysis using Simulation-based Situation Coverage for Cobot Systems. Proceedings of SafeComp 2021. [pdf copy]
- [4] R. Alexander, H. Hawkins, A. Rae, *Situation coverage – a coverage criterion for testing autonomous robots.* Report, 2015, Department of Computer Science, University of York. [pdf copy]
- [5] I. Cassar, A. Francalanza, L. Aceto, A. Ingólfsdóttir, A Survey of Runtime Monitoring Instrumentation Techniques, Electronic Proceedings in Theoretical Computer Science, Volume 254, 2017, Pages 15-28. [pdf copy]
- [6] M. Grindal, J. Offutt, S.F. Andler, Combination testing strategies: a survey. Software Testing, Verification and Reliability 15(3), 2005, Pages 167-199. [pdf copy]
- [7] W. Young, A. Sobhani, M. Lenné, M.Sarvi, Simulation of safety: A review of the state of the art in road safety simulation modelling, Accident Analysis & Prevention, Volume 66, 2014, Pages 89-103.