3.1 Identifying potential deviation from required behaviour

Practical guidance – cross-domain

Authors: Faiz Ul Muram, Muhammad Atif Javed and Sasikumar Punnekkat, Mälardalen University, Sweden

The automation, digitalisation and connectivity of autonomous systems with each other or with the infrastructure pose problems for safety assurance. The key to system safety is the identification and elimination/mitigation of potential hazards and documentation of evidences for safety cases. A safety case consists of process-based arguments that can show processes generate trustworthy evidence and product-based arguments that may directly show from the evidence that residual risks for the product are acceptably low. This is generally done during the system design and development phase. However, the parts of safety cases constructed during system design and development phase may turn out to be incorrect, inapplicable or insufficient during the operation. This can be caused by emergent behaviours and changes performed in consequence of market demands, hazardous conditions or system failures. To exploit the full potential of autonomous systems, the risk management and update of safety cases at the operational phase is essential. In this regard, the safety contracts have to be derived for uncertainty sources that provide the means to detect deviations from intended behaviour and perform necessary adaptations at the operational phase.

Overview of approach

The proposed guidelines focus on end-to-end traceability with support of a tool framework that can provide a significant boost for the designers to avoid the culture of 'paper safety' at the expense of actual system safety [2]. The end-to-end tool framework for safety analysis consists of the following steps.

- **Phase 1: Design and Development**
  - Step 1 - Derivation of safety requirements and safety contracts
  - Step 2 - Development of safety cases using safety contracts
- **Phase 2: Operational Phase**
  - Step 3 - Identification of deviations and update of safety cases

Phase 1: Design and development

At the design and development stage, the hazard analysis was carried out for the identification of hazards, their effects, and causal factors [3]. Based on the hazard analysis results, the safety requirements were derived to prevent or mitigate the identified hazards. Safety contracts have been derived for uncertainty sources. We constructed the safety cases and associated safety contracts with them.

Figure 1 shows the integration of tools that can be used for process, system design and safety case modelling and the following subsections provide further details.

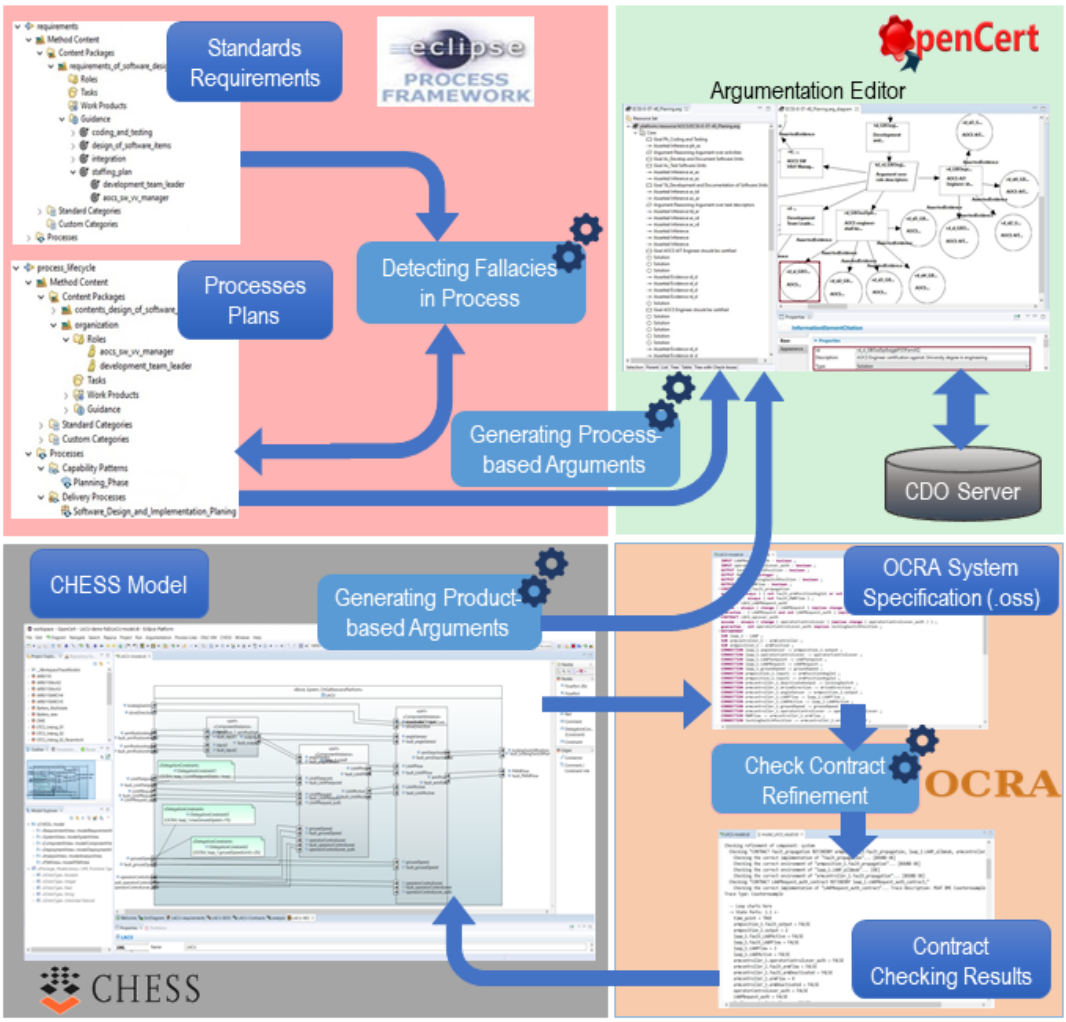

Figure 1. Overview of the process, system design and safety case modelling

Step 1: Derivation of safety requirements and safety contracts

Two hazard analysis techniques, particularly HAZOP and FTA were applied for the identification and mitigation of System-of-Systems (SoS) hazards [4, 5]. The former was applied to identify possible deviations in SoS quarry production, their possible root causes (faults) and consequences. The latter supports in-depth analysis; the fault trees were constructed based on the hazards and their potential effects understood from the HAZOP analysis [6]. The hazard analysis not just focuses on the individual behaviour of the used machines/systems in the production site and their emergent interactions, but also the server or other working equipment were considered. In production sites, autonomous systems follow defined travel paths to move from one zone to another.

The risk reduction and mitigation measures identified through the hazard analysis were translated into the safety requirements. Stringent safety requirements were either derived from the hazard analysis or extracted from the safety standards, such as driverless, automated guided industrial vehicles ANSI/ITSDF B56.5 [12] standard. To address industry-level hazards different safety requirements specified, such as “the autonomous vehicle shall safely travel in the site at the defined speed limit” is considered as a main requirement. To achieve this requirement, several other requirements were specified. After identifying the safety requirements, safety contracts were documented for uncertainty sources. Table 1

shows examples of derived safety contracts, where A and G stand for Assumption and Guarantee, respectively.

	Safety contract to ensure safe transportation
A1:	For an autonomous vehicle no deviation in obstacle detection devices AND no deviation in braking AND no deviation in steering rotation AND no deviation in travel path AND no deviation with defined speed limit AND no obstacle present in tracking range (10m) AND no curves in tracking range of travel path
G1:	The autonomous vehicle safely travels in the site at the defined speed limit of 20km/h
	Safety contract to ensure safe loading and unloading
A2	For an autonomous vehicle no deviation in new speed limit AND vehicle maintained correct position inside the defined boundary AND V2X commands take maximum 50ms AND actuation time of onboard system commands is not greater than 100ms
G2:	The loading and unloading of autonomous vehicles were safely conducted

Table 1. Examples of Derived Safety Contracts

Step 2: Development of safety cases using contracts

Once safety requirements and safety contracts are derived, the safety case can be created or otherwise generated from process or/and system model [7]. To document safety cases, several approaches exist, such as free text, tabular structures and graphical notations. Structured Assurance Case Metamodel (SACM) [8] is the Object Management Group (OMG) standard that integrates and standardizes the broadly used notations for documenting safety cases including Goal Structuring Notation (GSN) and Claims-Arguments-Evidence (CAE). For modelling and visualizing the safety cases OpenCert tool platform can be used. OpenCert is an open source assurance and certification tool; its argumentation editor is based on the GSN graphical notations. The safety cases can be stored in the workspace directory, or in the Connected Data Objects (CDO), which is both a development-time model repository and a run-time persistence framework. The repository sessions provide support for obtaining and modifying them. The derived safety contracts were associated with the safety cases, i.e., Assumption and the Guarantee properties of the contract were specified in the Content field of related claims in argumentation editor. For the undeveloped claim, the contract allows that its guarantees, can be supported by artefacts (e.g. the latter referring some verification results, simulation runs).

Generation of process-based arguments: As mentioned earlier, the process-based arguments can be generated automatically [9]. To generate process-based arguments, the Eclipse Process Framework (EPF) Composer was used for modelling the standards requirements and safety plans, while the OpenCert tool was used for visualizing the generated process-based arguments in the AMASS project [7]. In EPF Composer, the standard requirements and process lifecycle are modelled as plugins by following the guidelines mentioned in [9, 10]. Process-based Argument Generator plugin took the Capability Pattern or Delivery Process as an input and transformed it into arguments (model and diagram). The generated process-based arguments were saved locally in a new project into the current workspace under the name Argumentation. They were also stored in the corresponding destination assurance case in the CDO server under the ARGUMENTATION

folder. To prevent a fallacious generation of arguments, a common type of fallacy (Key-evidence omission) was detected before enabling the generation. The Fallacy Detection plugin took the process and standard requirements as input and validated whether the process contains the sufficient information corresponding to the key evidence for supporting the specific requirements [7, 9].

Generation of product-based arguments: The product-based argument-fragments can be generated from the contract-based architectural specification. For this, Polarys CHESS (Composition with Guarantees for High-integrity Embedded Software Components Assembly) toolset can be used. CHESS provides the support to model all phases of system development and contracts (i.e. the assumption and the guarantee properties). Argument Generator plugin implemented in OpenCert assumes that the analysed model and the contract refinement check results are stored in the refinement analysis context [7, 11]. The generated set of argument-fragments stored in the corresponding destination assurance case in the CDO server stated in the OpenCert preferences.

Contract Refinement Checking: Othello Contracts Refinement Analysis (OCRA) is a command-line tool that provides means for checking the refinement of contracts specified in a linear-time temporal logic and generating the corresponding set of proof obligations. The integration of CHESS with OCRA verification engine allows the validation of component contract assumptions against the specification of other components in the system. The CHESS model together with contracts are transformed into an OCRA System Specification (.oss) file readable by OCRA. The contract refinement checking is done by OCRA and the result is back-propagated to the CHESS model. OCRA runs in background or remotely via OSLC (Open Services for Lifecycle Collaboration), and therefore, the user does not interact with them directly [7].

Phase 2: Operational phase

For systems with enhanced automation and connectivity, there is a need to deal with unknowns and uncertainties during operational phase. As the autonomous systems (things) have limited power, the cloud and fog services can be utilised for storage and processing services [5]. In the context of SUCCESS project, the simulators of real Volvo Construction Equipment (VCE) machines were used. A detailed list of parameters was used to support automated operations of the scenario that can be accessed and changed during operational phase, when necessary. The safety requirements and hazard mitigation recommendations were also implemented in the scenario as code scripts. This not only gives the possibility to detect deficiencies, such as additional hazards and risks but also to identify, monitor, evaluate, and resolve deviations from specified behaviours during the operational phase.

Step 3: Identification of deviations and update of safety case

For the identification and resolution of gaps between the intended behaviour reflected in safety cases and the actual safety of production operations, the operational data was utilised [3,5]. Based on the gathered data, the safety contracts constructed for uncertainty sources were monitored, deviations between the intended and actual behaviour were tracked by comparing the safety contracts with their respective parameters that were gathered from the simulations. This provided the basis for the adaptation of affected parts of safety cases and associated safety contracts. Afterwards, the update command was issued. The safety cases were updated on the CDO server, which was accessed by the

OpenCert argumentation editor connected to the CDO server [5]. In case, the assumption A1 is satisfied then predicted guarantee G1 hold, as mentioned in Table 1. In this way, simulation result related to the satisfied safety contracts was assembled as an evidence to the corresponding undeveloped claim. The safety contract linked to a claim that guarantees "the loading and unloading of autonomous vehicles were safely conducted" (see Table 1), is dependent on the conditions, such as central server commands (e.g., Queue, Pause, Exit), vehicle level actions (e.g., maintaining speed, actuation time), multiple checkpoints and a monitoring system (e.g., correct position, speed limit). The successful and safe loading operation requires the presence of only one autonomous hauler AH in the loading point and AH needs to maintain 0 km/h speed while at the loading point. In case of the transmission delay or actuation delay, the contract assumption A2 did not satisfy and predicted guarantee G2 did not hold. In such case, corresponding control actions were evaluated, particularly, threshold was determined, slow down and stopping time accordingly increased and updated [13]. The adaptations in safety contracts and corresponding safety case elements were triggered in response to the selected control actions. In contrast to the matching of parameter names and their values/ranges for safety contracts, the text-based matching was performed to alter the description of safety case elements.

Tools and applicability

The tools used in this guidance are integrated in the OpenCert bundle that can be downloaded from the following link www.eclipse.org/opencert/downloads/. For more detailed information, a user manual of the particular tools and a developers' guide to set up the workspaces is provided in [7].

The end-to-end traceability and a tool framework for dynamic safety assurance is generally applicable to various autonomous systems. The initial effort is although required to derive the safety contracts for uncertainty sources, but it is worthwhile to detect deviations from intended behaviour and perform necessary adaptations at the operational phase.

Industrial case study - electric site

The electric site research project [1] was used as a use case for applying this guidance, which falls under the construction equipment domain. The quarry operation is carried out using different kinds of machines, such as an excavator, a movable primary crusher, a wheel loader, a stationary secondary crusher and autonomous haulers. In particular, they collaborate to realize the targeted production goals. The quarry production site is divided into a set of zones, such as parking, loading, charging, transporting, and unloading (dumping). The autonomous haulers (systems) are used to transport material in the quarry site. In particular, they follow defined travel paths to move from one zone to another. Their operations are similar to the Automated Guided Vehicles (AGVs). The site management system is responsible for commanding the autonomous haulers or systems. The values regarding timing, location, path points, load capacity and speed limits are provided to the user interface of the site management.

References

- [1] Volvo Construction Equipment, "Emission-free quarry," [Online] https://www.volvoce.com/global/en/news-and-events/press-releases/2018/testing-begins-at-worlds-first-emission-free-quarry/.

- [2] C. Haddon-Cave, The Nimrod Review: An Independent Review into the Broader Issues surrounding the Loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/229037/1025.pdf
- [3] Transportation risks in production sites", in: 16th European Dependable Computing Conference, EDCC 2020 Companion Proceedings, Munich, Germany, September 7–10, 2020.
- [4] F. U. Muram, M. A. Javed, and S. Punnekkat, "System of systems hazard analysis using HAZOP and FTA for advanced quarry production," in 2019 4th International Conference on System Reliability and Safety (ICSRS), Rome, Italy, November 20-22, 2019, pp. 394–401.
- [5] M. A. Javed, F. U. Muram, H. Hansson, S. Punnekkat and H. Thane, "Towards Dynamic Safety Assurance for Industry 4.0", Journal of Systems Architecture (JSA), 2020.
- [6] C. A. Ericson, Hazard Analysis Techniques for System Safety, 2 edition, John Wiley & Sons, 2015.
- [7] AMASS User guidance and Methodological framework D2.5. https://www.amass-ecsel.eu/sites/amass.drupal.pulsartecnalia.com/files/D2.5_User-guidance-and-methodological-framework_AMASS_Final.pdf (2018).
- [8] Object Management Group, 2018. Structured Assurance Case Metamodel (SACM), Version 2.0. https://www.omg.org/spec/SACM/2.0.
- [9] F.U. Muram, B. Gallina, L. G. Rodriguez, "Preventing Omission of Key Evidence Fallacy in Process-based Argumentations". In: 11th International Conference on the Quality of Information and Communications Technology (QUATIC), Coimbra, Portugal, September 4-7, 2018. pp. 65–73
- [10] McIsaac, B.: IBM Rational Method Composer: Standards Mapping. Tech. rep., IBM Developer Works (2015).
- [11] I. Šljivo, B. Gallina, J. Carlson, and H. Hansson: Strong and Weak Contract Formalism for Third-Party Component Reuse. 3rd International Workshop on Software Certification, pages 359—364. November 2013.
- [12] American National Standards Institute/Industrial Truck Safety Development Foundation, Safety Standard for Driverless, Automatic Guided Industrial Vehicles and Automated Functions of Manned Industrial Vehicles, December 2019, [Online] http://www.itsdf.org
- [13] F. U. Muram, M. A. Javed, S. Punnekkat and H. Hansson, "Dynamic Reconfiguration of Safety-Critical Production Systems", in: 25th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC '20), Perth, Australia