Early Assessment of System-Level Safety Mechanisms through Co-Simulation-based Fault Injection

Tiziano Munaro

fortiss
Research Institute of the Free State of Bavaria
associated with the Technical University of Munich
Munich, Germany

munaro@fortiss.org

Safety mechanisms – technical solutions responsible for maintaining the intended functionality (fail-operational) or transition to a safe state in the presence of hardware and software faults (fail-safe) – ensure the functional safety of cyber-physical systems (cf. ISO 26262, Part 1). An example for a safety mechanism used to meet fail-operational safety goals is run-time task reconfiguration: By deploying copies of a software unit to different processing elements, the loss of one or more of these hardware elements can be tolerated.

Considering their high impact on a system's hardware and software architectures, early validation of safety mechanisms is crucial to reduce engineering and operation costs. However, while the real-time behavior or safety mechanisms is as crucial to their effectiveness as the correctness of their implementation, analytical safety analysis techniques applied to date (e.g., FMEA and STPA) support only coarse time models and do not provide explicit guidance for considering systemic real-time properties. The consequence is that neither technique is able to determine when, for instance, a run-time task reconfiguration is not sufficiently fast to control a fault within the system- and context-specific fault-tolerant time interval. By the time such defects become apparent, the cost of addressing these issues has grown in magnitude.

To address this challenge, we introduce a simulation-based fault injection framework to identify problematic real-time properties in safety concepts. As the simulation replicates the integrated electrical/electronic (E/E) architecture of the system under test, the propagation of faults across the system can be reproduced accurately. Moreover, we leverage the Functional Mock-up Interface (FMI) standard for black-box co-simulation to overcome intellectual property concerns in distributed supply chains and to account for heterogeneous tool landscapes while providing the controllability and observability necessary for the simulation of both loss and erroneous behavior of arbitrary hardware and software components. Finally, using an industry-oriented use case, we exemplify how the simulation's validity for a specific use case can be determined by means of statistical analysis.

Encouraged by the promising results, we are currently taking the next steps towards the continuous engineering of safety-critical cyber-physical systems: As the simulation's configuration and execution can be automated, and the necessary Functional Mock-up Units (FMUs) can be generated from development artefacts, we are working on leveraging the here presented approach in the context of iterative development and maintenance processes with automated integration and deployment mechanisms.