ExCuSe – A Method for the Model-Based Safety Assessment of Simulink and Stateflow Models

MATLAB Expo 2018 | 2018-06-26 | München

Julian Rhein

Outline

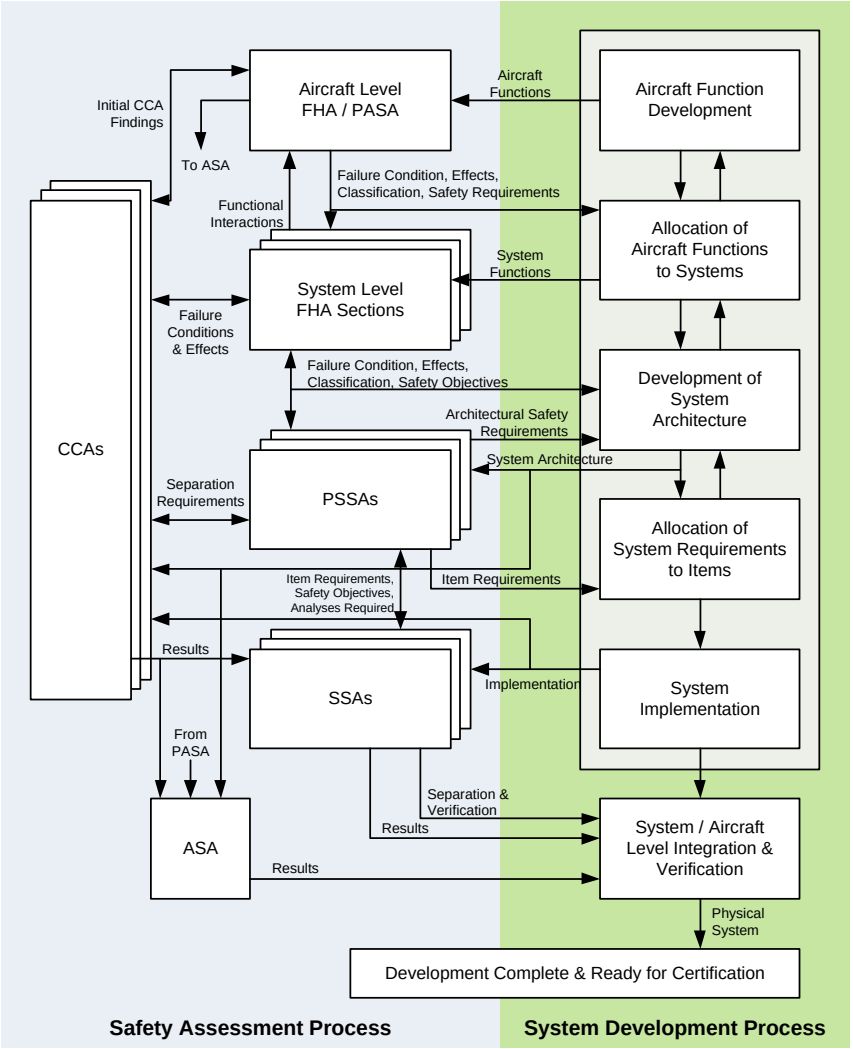
- Introduction
- Property Proving
- Application to Safety Assessment
- Implementation
- Demo
- Summary & Outlook

Introduction

- Functional safety assessment is an integral part of software and systems development according to SAE ARP 4754A
- Task of functional safety assessment
  - Establish relations between component faults and system failure conditions
  - Validation & Verification of safety requirements

Typical methods:

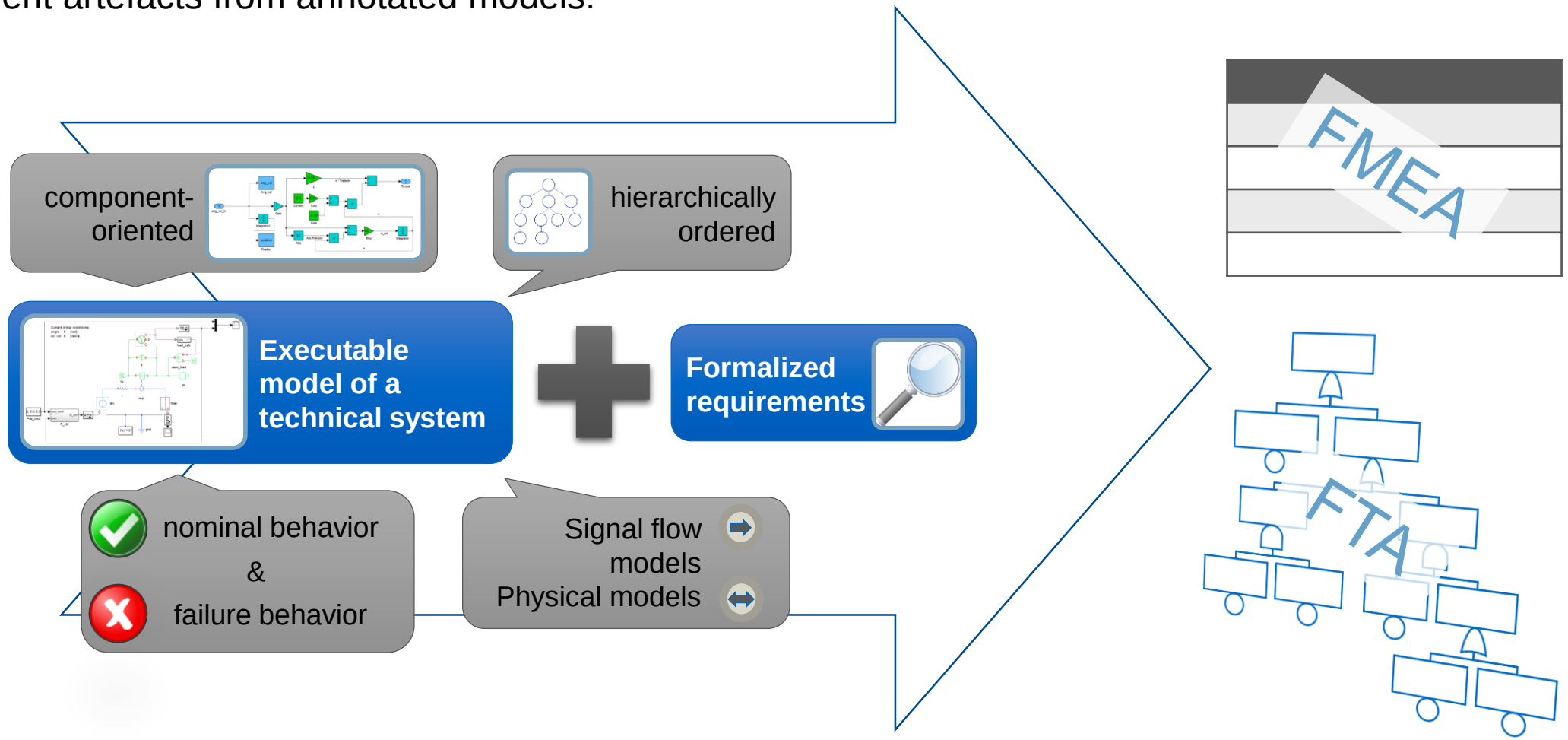
- Failure Modes and Effects Analysis (FMEA)
  - Determine system level effects from (single) component faults
- Fault Tree Analysis (FTA)
  - Find possible causes for system failure conditions



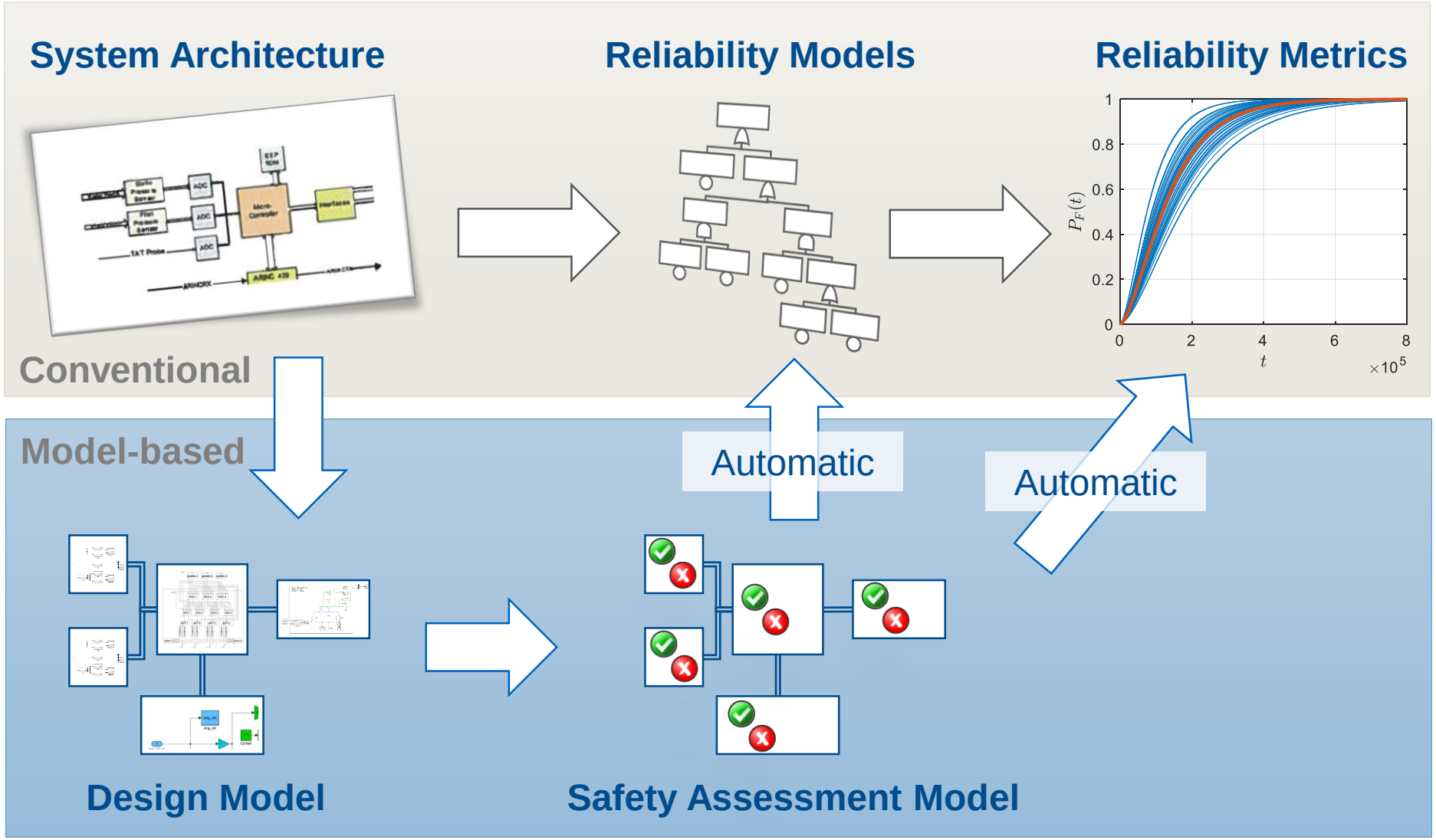
[SAE ARP 4754A]

The Big Picture

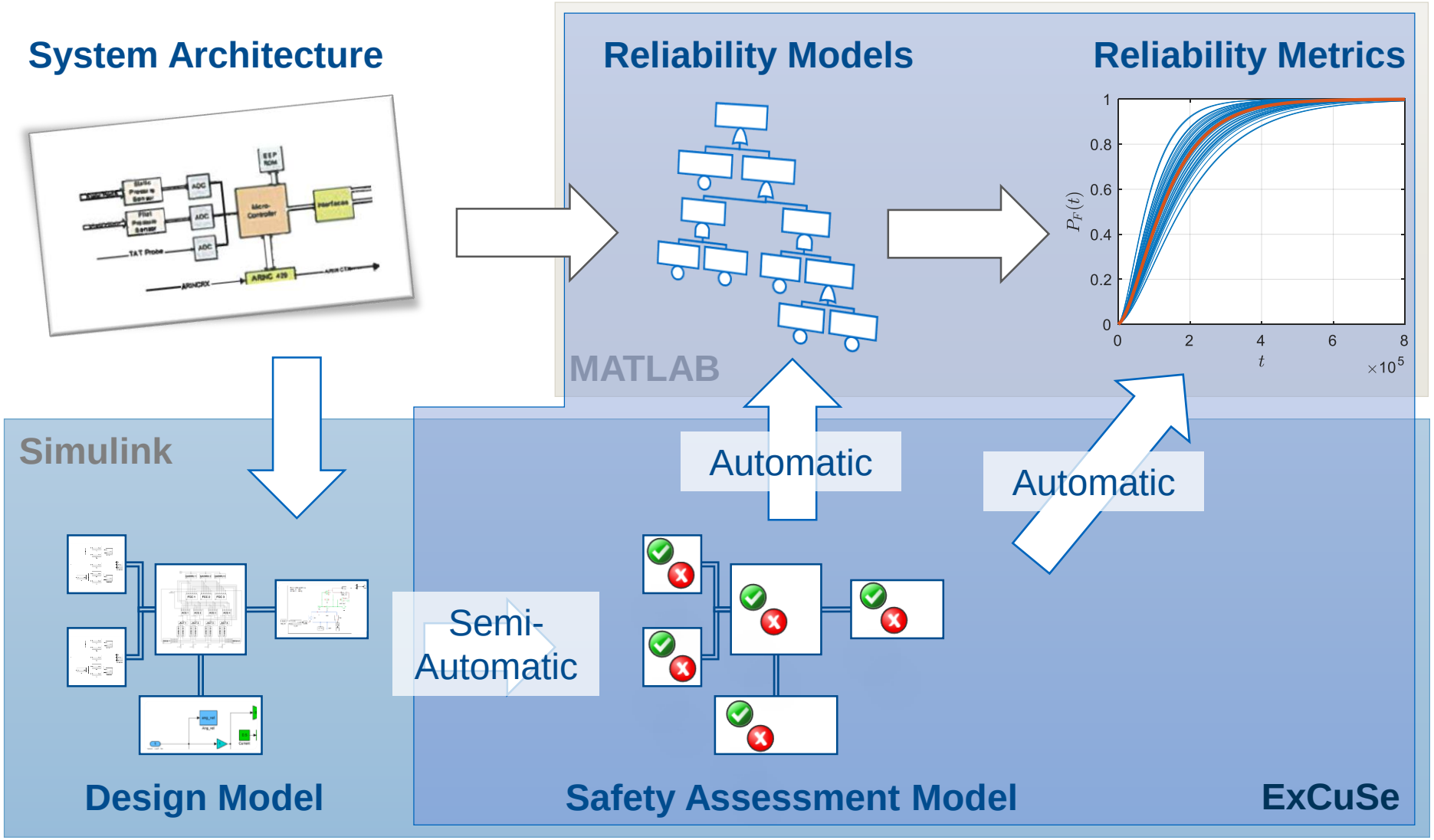
Our goal: Partially automated generation of safety assessment artefacts from annotated models:



Conventional vs. Model-Based Safety Assessment



The ExCuSe Tool



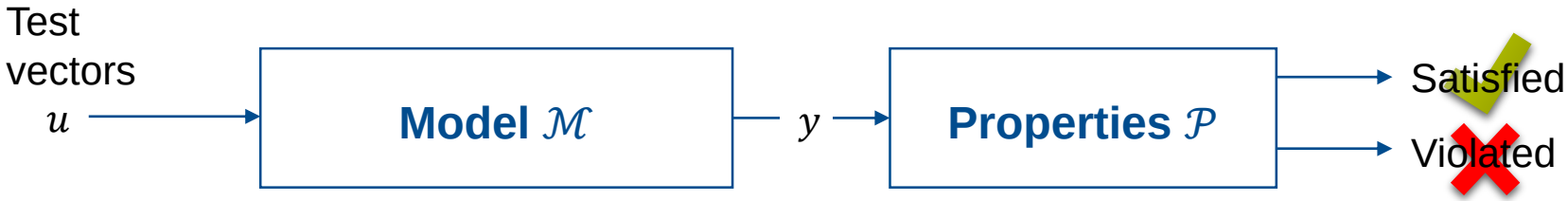
Advantages of MBSA

- ✓ Support of analysts during repetitive, error-prone tasks
- ✓ Provides methods to system designers and software engineers to evaluate their designs prior to formal safety assessment
- ✓ Enables modularity of safety assessment and reusability of artefacts
- ✓ Additional validation of manual assessment results

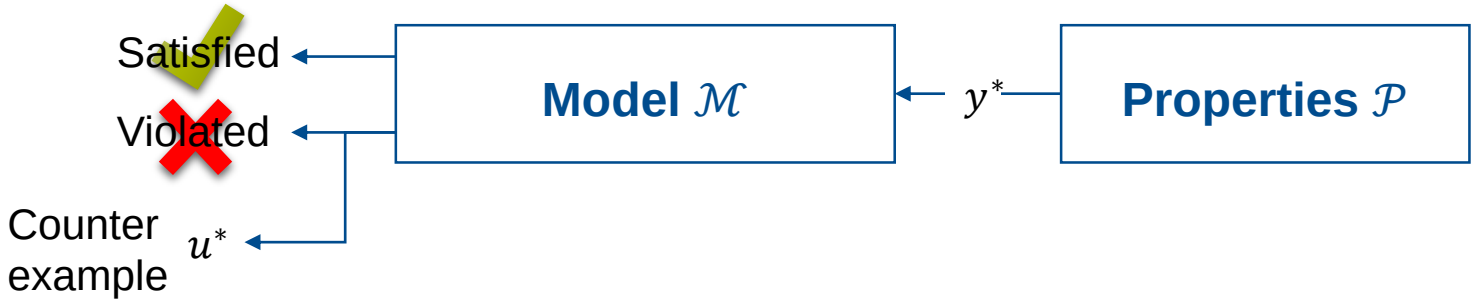
Introduction to Property Proving

- Traditionally used in software verification
- Formal approach to prove that a property is satisfied or violated by a system
- Properties are disproved by counterexamples
- Infinite state and continuous systems are treated by inductive proving and SMT satisfiability
- Powerful free/open-source solvers available

Testing:



Property proving:



Safety Assessment as Property Proving Problem

Formal concept

- Fault injection

$$M(u) \Rightarrow M^*(u, f), f \in \mathcal{F}$$

- Cut-sets computation

$$CS \coloneqq \left\{cs_i \in \mathcal{F} \mid M^*(u^*, f^*) \not\models AG\ \mathcal{P} \wedge \exists j\ s.t.\ f_{i,j}\right\}$$

- Minimal cut-sets

$$MCS \coloneqq \{cs \in CS \mid \nexists cs' \in CS \wedge cs' \subset cs\}$$

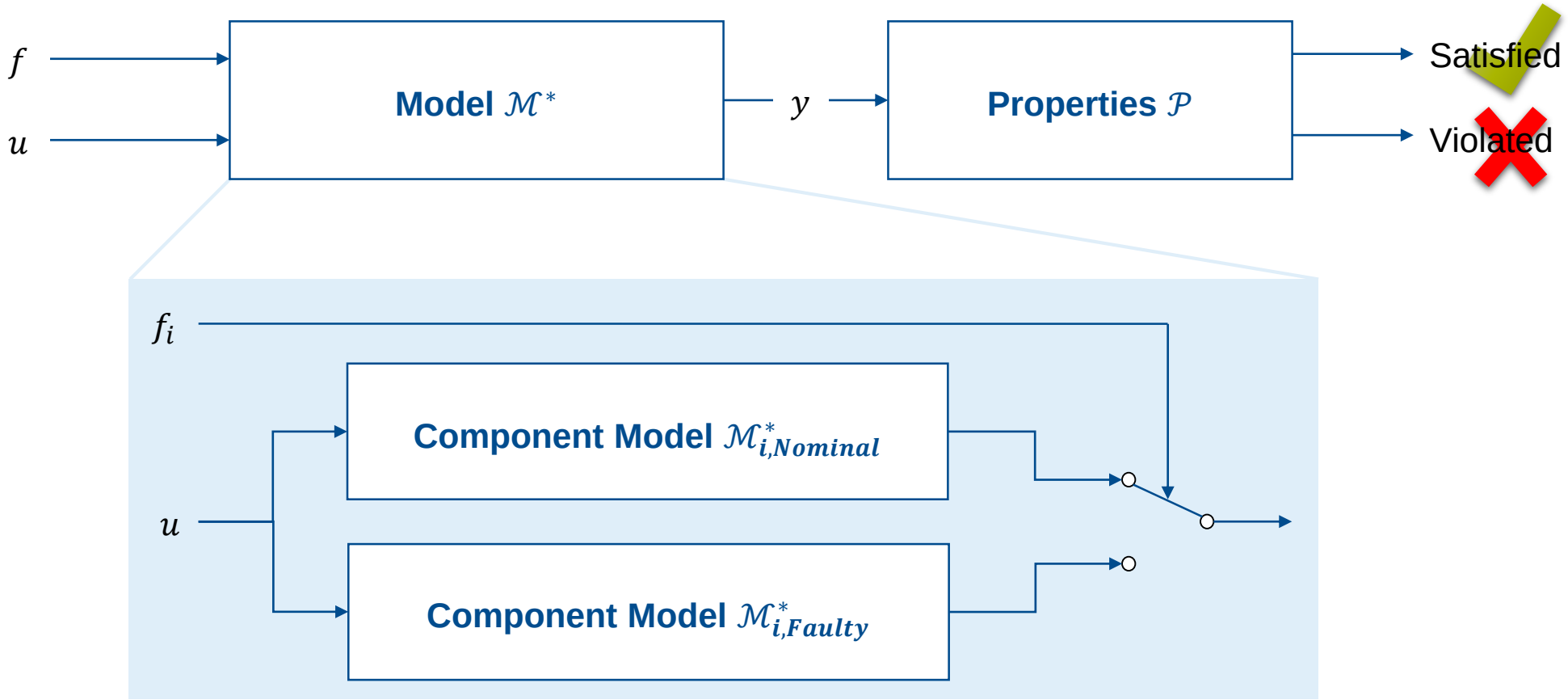
- Verification: $|\mathcal{F}| = 0\ , |\mathcal{P}| > 0$
- FMEA: $|\mathcal{F}| = 1\ , |\mathcal{P}| > 0$
- FTA: $|\mathcal{F}| > 0\ , |\mathcal{P}| = 1$

Intuitive explanation

- Extension of the model by additional inputs to trigger fault events
- Computation of sets (combinations) of failures, which lead to a requirements violation from the counterexamples
- Minimal combinations of failures, i.e. failure configurations that are necessary for the occurrence of system failures
- Verify that the system fulfills all requirements in the failure free case
- Determine all possible effects (i.e. violation of requirements) of single failures
- Determine all possible causes of single a single effect (i.e. its MCSs)

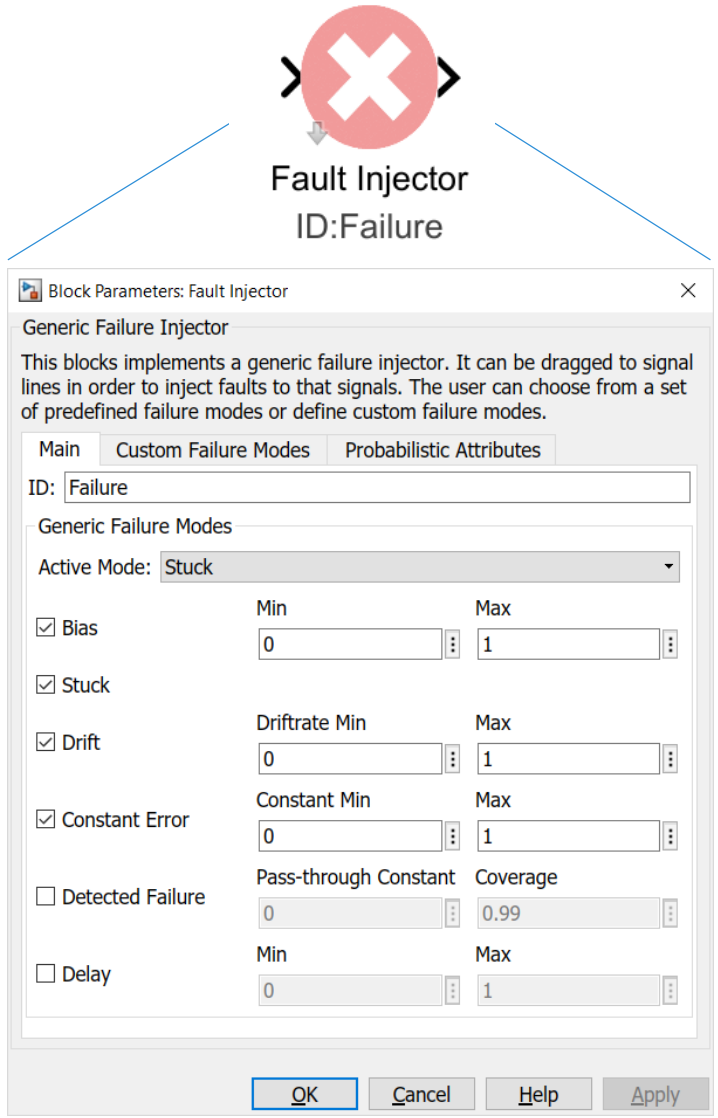
Fault Injection

- Extension of the model by additional component failure models and additional inputs to trigger the activation and deactivation of component faults

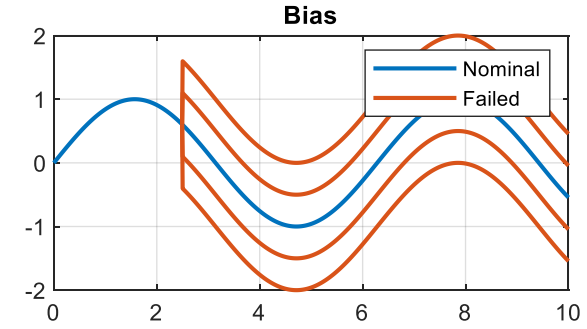


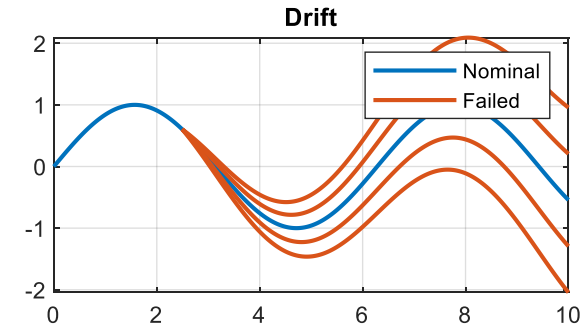
Implementation

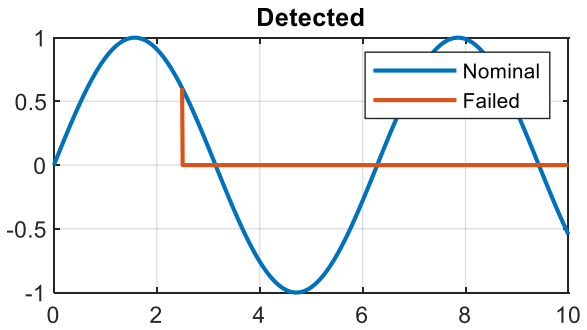
- Implementation in Simulink/Stateflow by fault injection interface
- **Idea:**
  - Extend the model by a nondeterministic layer
  - Provide modelling facilities for failure logic modeling or failure injection
  - Provide interface for automatic analysis
- Share failure flow information between components without requiring additional signals
- Allows common cause modeling
- Predefined and custom fault models
- Automatic cut-set analysis based on the Simulink Design Verifier property proving function

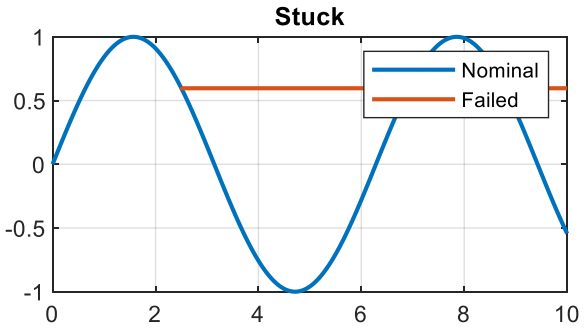


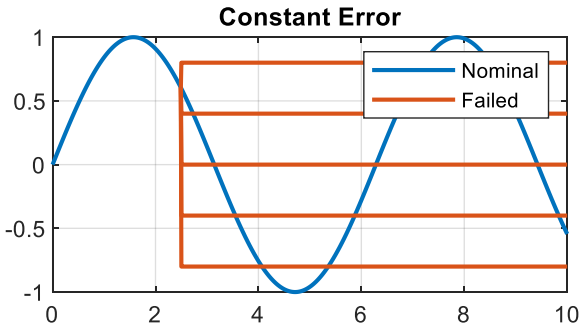
Generic Failure Models

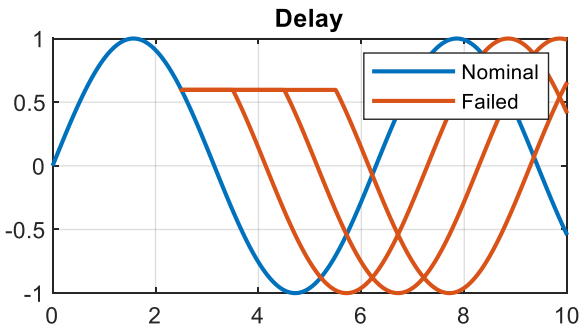


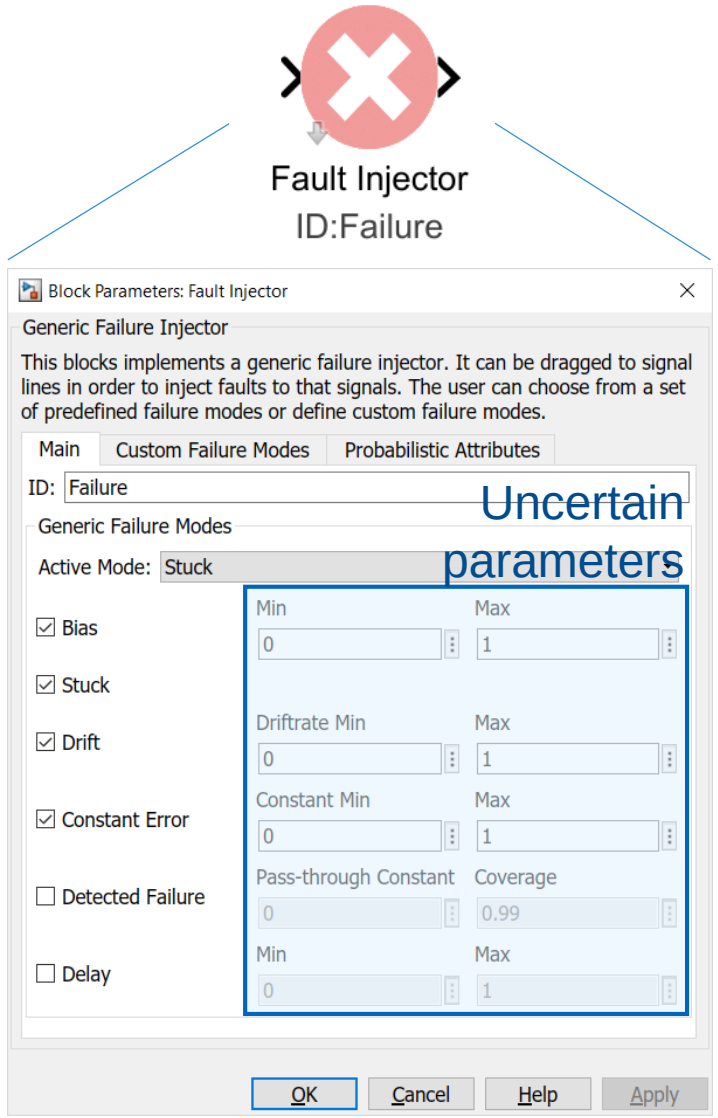






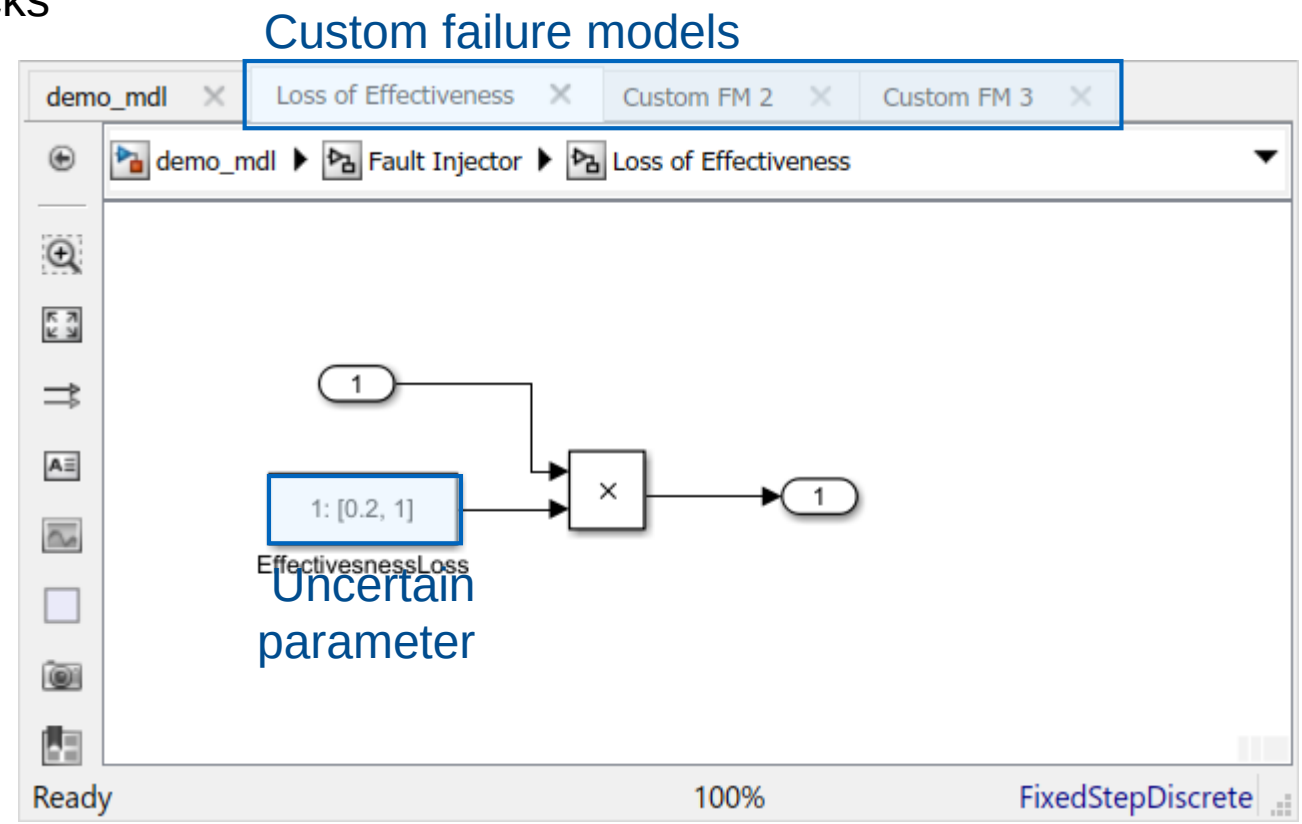


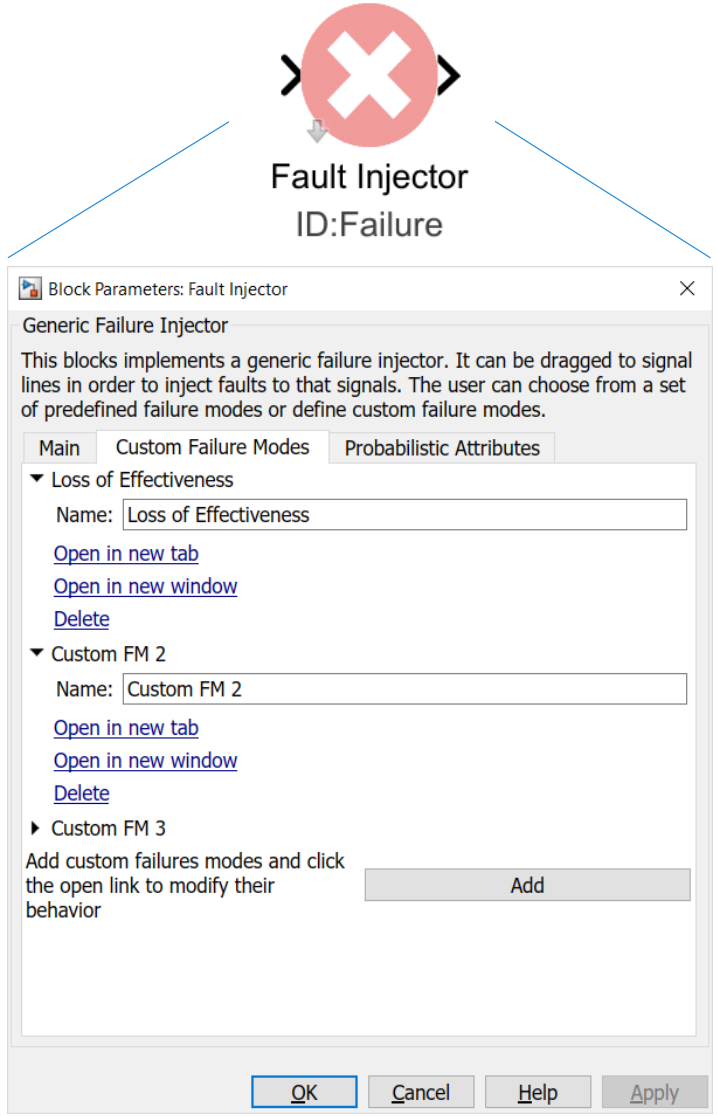




Custom Failure Models

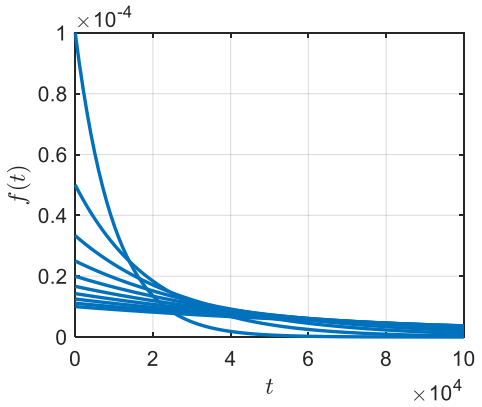
- Enable definition of arbitrary, user-defined fault models
- Uncertain parameters can be modelled using special source blocks

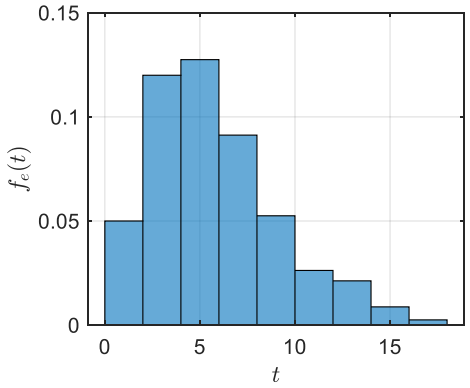


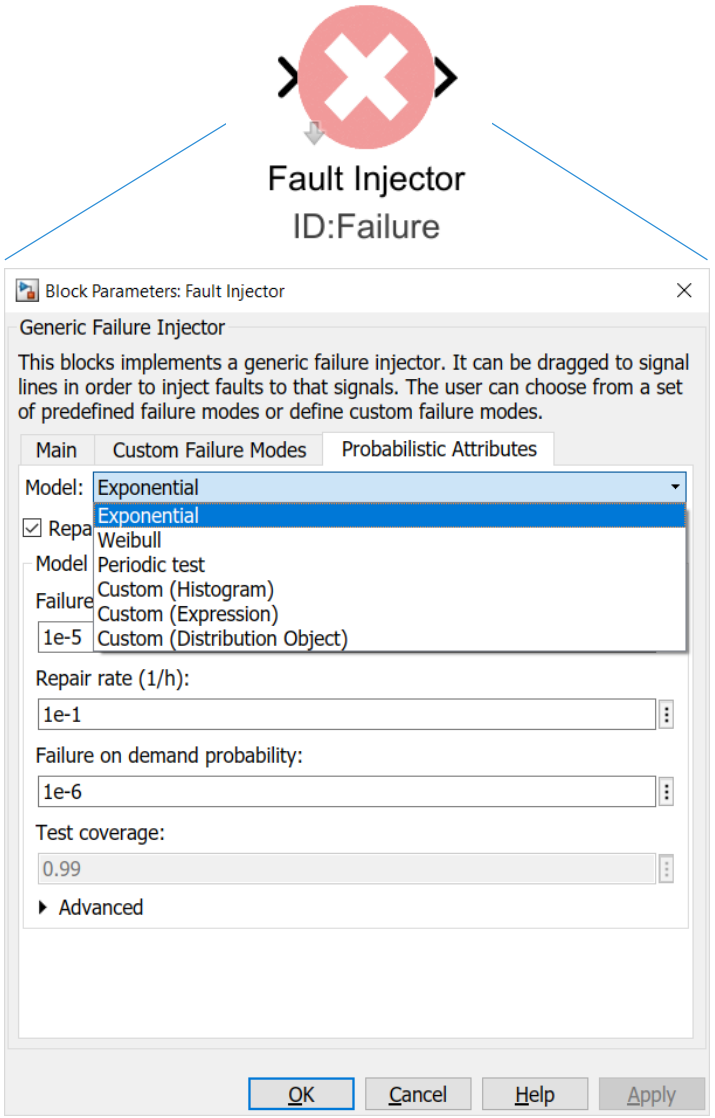


Probabilistic Attributes

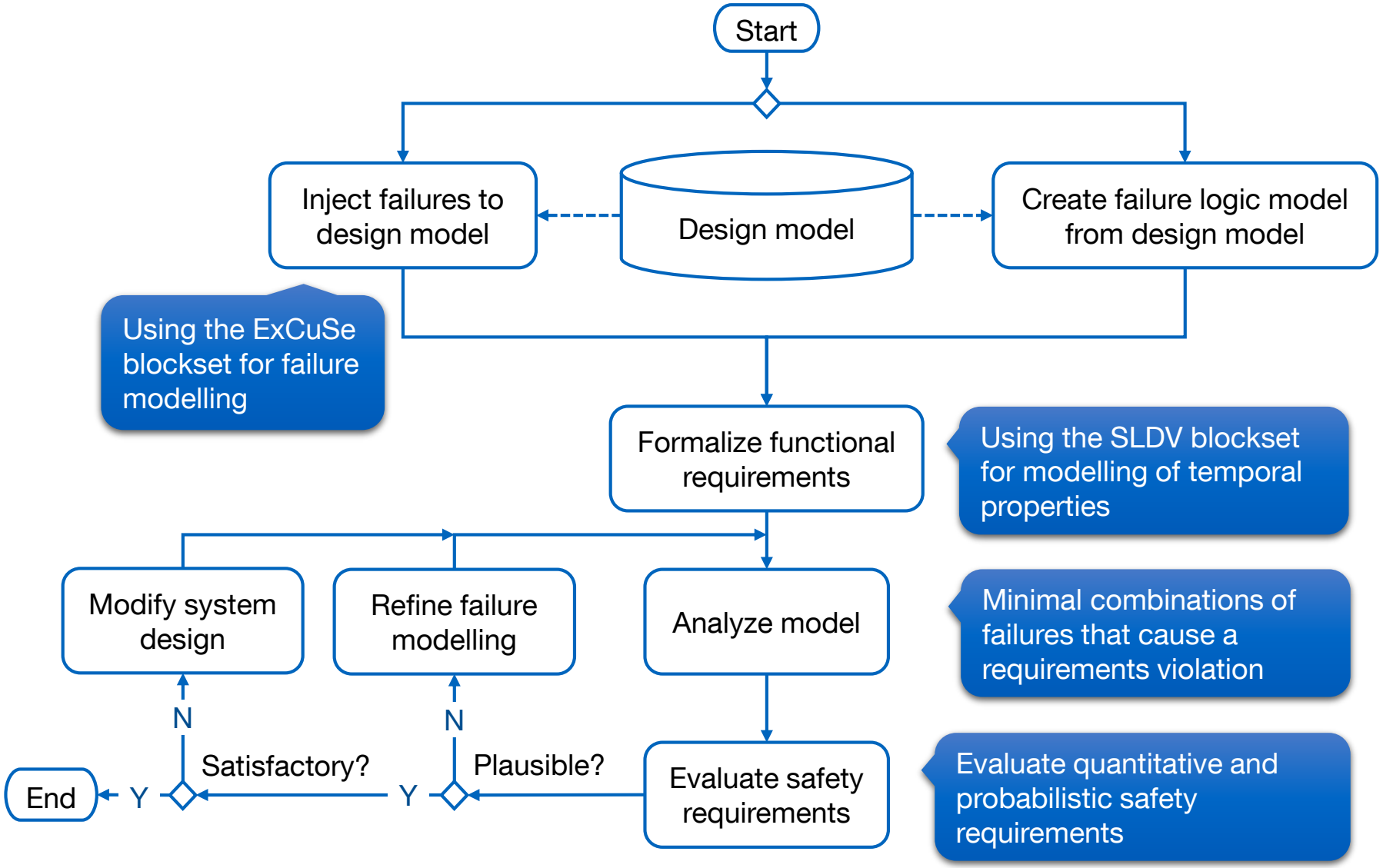
- Distributions of the component lifetime and repair time
- Similar to basic event models in fault tree analysis
- Common models are built-in:
  - Exponential distribution
  - Weibull distribution
  - Periodic test
- Custom models can be specified as custom expressions or histograms





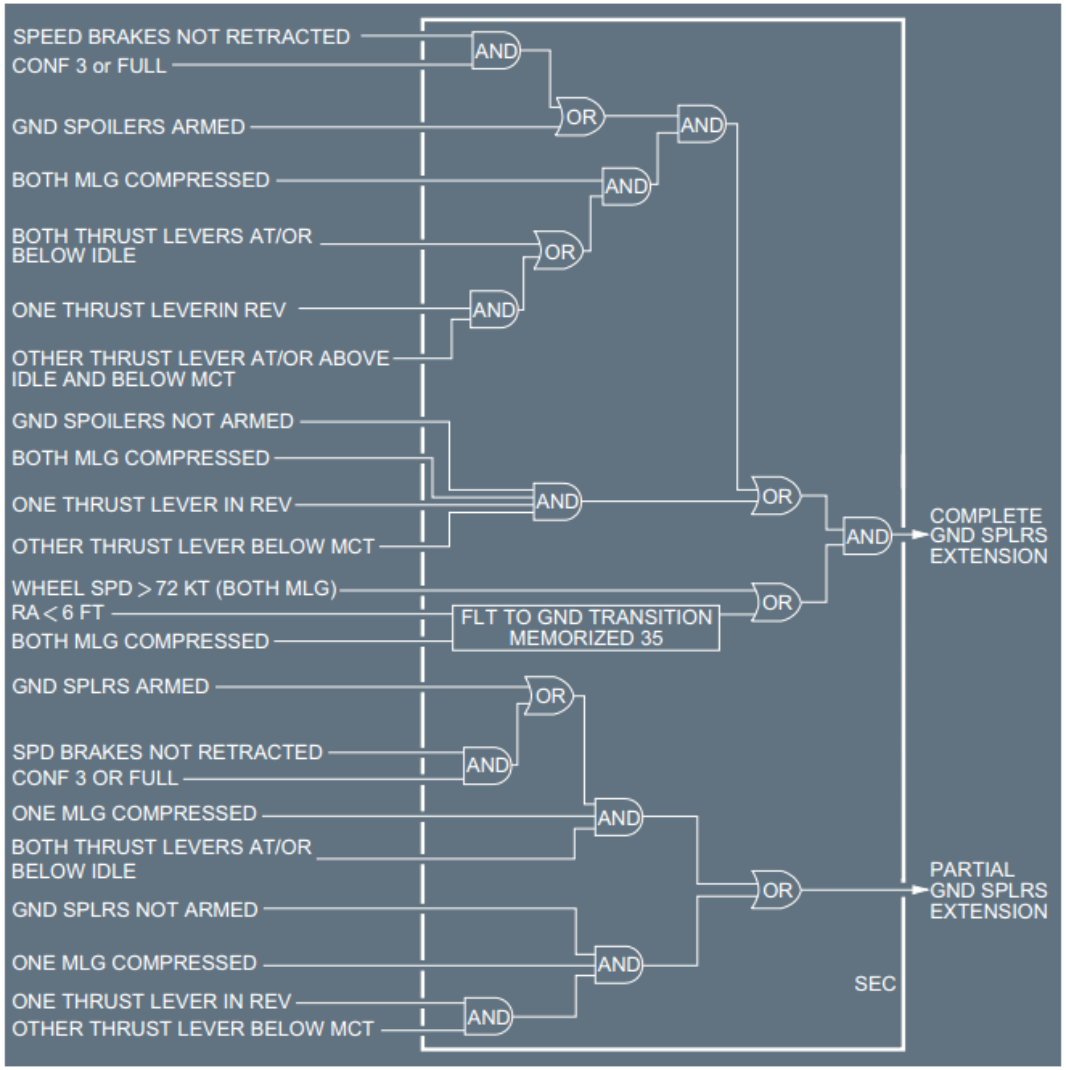


Model-Based Safety Assessment with ExCuSe



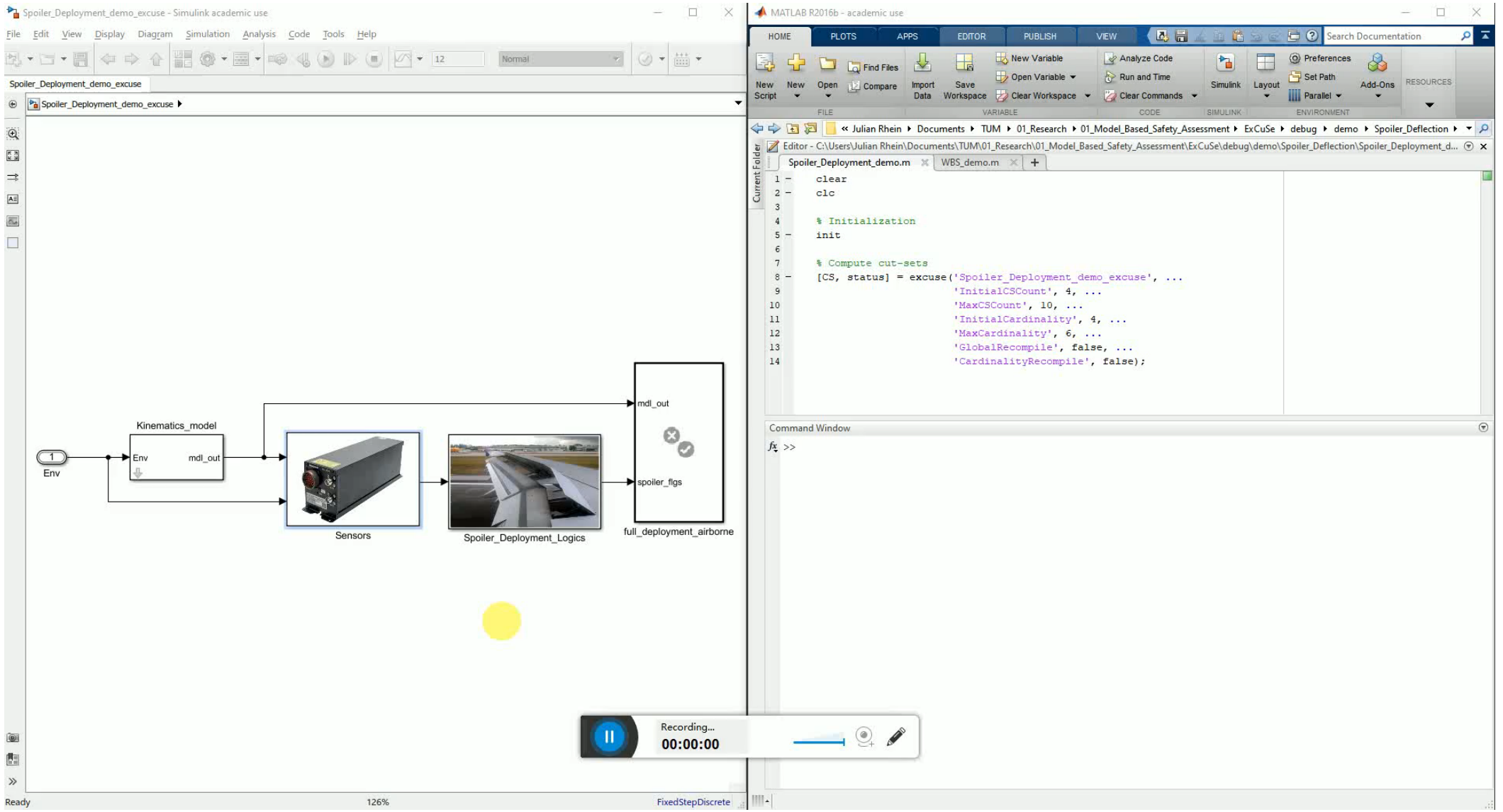
Demo: Ground Spoiler Deployment Logic

- Example problem: A320 SEC decision logic for ground spoiler deployment
- Hazards:
  - Inadvertent full/partial spoiler deflection inflight
  - Missing full/partial spoiler deflection on ground
- Question: Which combinations of sensor failures can potentially cause the hazards



[Airbus A320 AMM]

Demo: Ground Spoiler Deployment Logic



Summary & Outlook

Summary

- Model checking provides a powerful method for model-based safety assessment
- Cut-set analysis can be expressed as property proving problem
  - Performance enhancement by incremental search
  - Anytime approximation of probability boundaries
- Successful integration in Simulink/Stateflow

Outlook

- Scalability considerations
- Creation of structured fault trees from the minimal cut-sets
- Using structural analysis to obtain initial guess of the minimal cut-sets
- Extension to undirected models

Thank you for your attention!

Questions?

Contact

Julian Rhein
Research Associate

Lehrstuhl für Flugsystemdynamik /
Institute of Flight System Dynamics
Technische Universität München
Boltzmannstraße 15
D-85748 Garching
Germany

Phone: +49 (89) 289-16051
Fax: +49 (89) 289-16058
Email: julian.rhein@tum.de