framatome

Reliability Analyses of Digital I&C within the Verification and Validation Process

Mariana Jockenhövel-Barttfeld,
Stefan Karg, Christian Hessler, Herve Bruneliere

PSAM 14, 16 - 21 September 2018 – Los Angeles, CA

Overview

- Introduction
- Challenges
- The Framatome methodology
- Hardware reliability
- Software reliability
- Use of reliability results for safety demonstration
- Conclusions

Introduction

- **Digital instrumentation and control (I&C) systems appear as upgrades in existing plants and are fully incorporated in new designs**
- **Reliability analyses are conducted during the verification and validation (V&V) process of the I&C systems**
  - Aim at demonstrating that quantitative safety goals (targets) imposed by regulators / safety authorities are fulfilled
    - Identification of major contributors
      - Failure probability of a function on demand
      - Spurious actuation frequency of a function
    - Identification of design improvements during the design phase
- **Generic approach for digital I&C systems**
  - For illustration purposes: TELEPERM® XS, the system platform for safety I&C developed at Framatome

Challenges

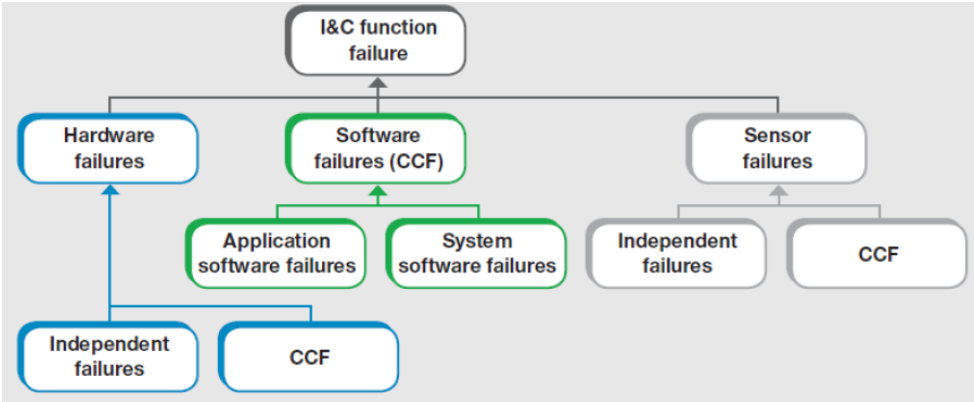
- **Conventional techniques for probabilistic analyses cannot adequately address features of digital systems and have to be extended with regards to**
  - Fault-tolerance / coverage features, voting logic degradation
  - Reliability of the hardware including common cause failures (CCF)
  - Reliability of the software including CCF
  - Uncertainty considerations (e.g. estimation of error factors)

> **These topics are required to be included for the safety demonstration during licensing of I&C systems processing cat. A safety functions**

- **Interpretation of regulatory safety requirements**
  - Apply to the parts of the system being updated
  - Provide key information to define the scope and boundary conditions for the reliability analysis
  - Impose an upper bound for the failure probability / frequency on the functions
    - The higher the requirement, the more rigorous (detailed) the modelling

Our Methodology (1/2)

- **Reliability analyses use fault trees**
  - Model failure contributions of the complete digital signal path (from sensors to actuators)
    - Independent failures and CCF for hardware and software



  - Determine the failure probability on demand (unavailability) and/or the frequency of spurious actuation of a function

- **Functional analysis**
  - Select representative functions for the reliability analysis
    - Based on pre-defined selection criteria (e.g. processing requirements, complexity)
    - The reliability of representative functions is bounding for other functions

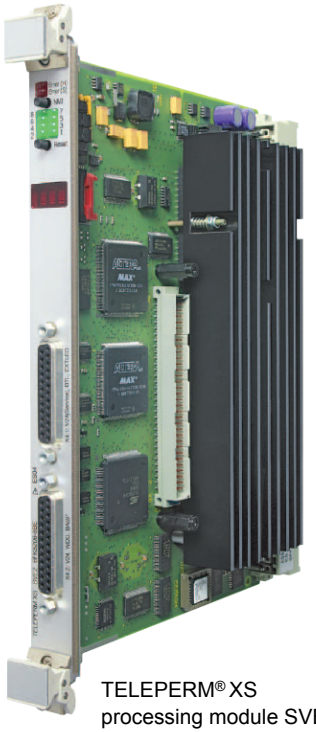
framatome

Our Methodology (2/2)

- **Level of detail for modelling the fault trees**
  - Capture design features and dependencies affecting the system reliability
    - Hardware modules: *common board* and *single channels*
    - Software modules: *system* and *application software*
  - Trade-off between modelling effort and improvement of reliability results
    - Treatment of faulty signals, voting logic degradation
    - Simplified conservative approaches can considerably reduce the modelling effort without overestimating the reliability results significantly
- **Systematic definition of failure modes for hardware / software using system failure mode and effect analyses (FMEA)**
  - *Functional failure modes* based on the effect of the failure on the module
  - Reliability models for basic events depends on the failure detectability
    - **Detected failures** (e.g. self-reporting failures, failures detected by engineered monitoring features) ⇨ "repairable component"
    - **Undetected failures** ⇨ "tested component"

Hardware Reliability

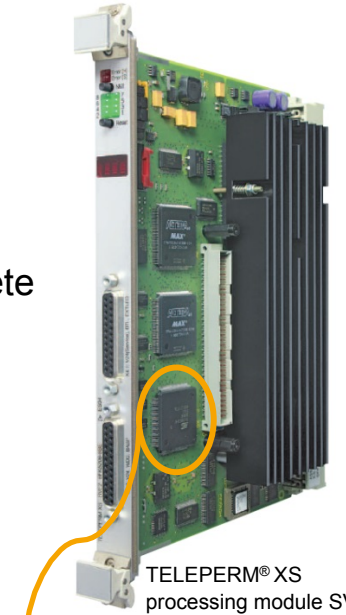
- **Theoretical failure rates for independent failures**
  - Uncertainty parameters estimated from the operating experience
- **CCF of hardware modules**
  - The operating experience of TELEPERM XS shows that hardware failures occur as random, independent failures (one module at a time)
    - **Coincident failures of hardware modules have not been observed**
  - Effects of CCF of the hardware have to be included in reliability analyses (imposed by standards)
  - Modelling in reliability analyses
    - Assumption: two or more identical modules processing redundant signals have deficiencies that would result in component failures if the components were requested to operate
    - CCF probabilities for undetected failures are usually larger than CCF probabilities for detected failures
      - Detected failures can be identified and repaired within a short amount of time
    - There is a lack of I&C-specific CCF parameters for existent CCF models
      - Generic parameters lead to very conservative reliability results



TELEPERM® XS processing module SVE2

Software Reliability (1/2)

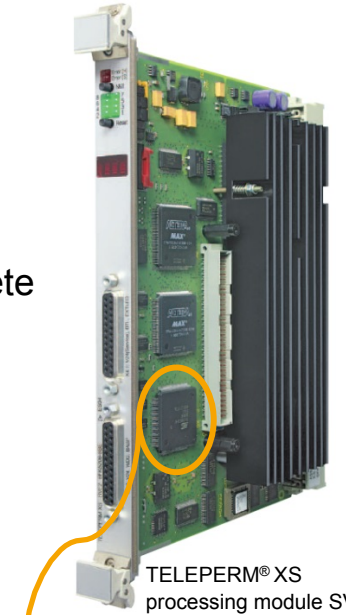
- **Software failures result as a combination of a latent systematic fault with a trigger**
  - **Latent faults**
    - Software specification/implementation was inadequate, incorrect, incomplete
    - Testing did not include the specific signal trajectory that reveals the fault
  - **Relevant triggers**
    - Time-dependent effects, faulty telegrams, same signal trajectories/states, faulty maintenance (can also introduce faults into the software)
  - **Software failures have a *common cause nature***
    - Same faulty software in different divisions affected by common triggers



TELEPERM® XS
processing module SVE2

Software Reliability (1/2)

- **Software failures result as a combination of a latent systematic fault with a trigger**
  - **Latent faults**
    - Software specification/implementation was inadequate, incorrect, incomplete
    - Testing did not included the specific signal trajectory that reveals the fault
  - **Relevant triggers**
    - Time-dependent effects, faulty telegrams, same signal trajectories/states, faulty maintenance (can also introduce faults into the software)
  - **Software failures have a *common cause nature***
    - Same faulty software in different divisions affected by common triggers

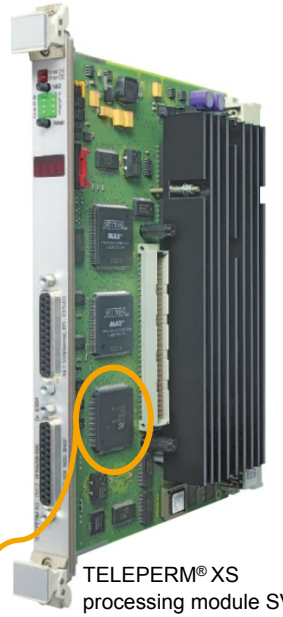


TELEPERM® XS
processing module SVE2

- **Software failure effects**
  - **Non-fatal failure: processor remains in operation but the faulty function is either unavailable or spuriously actuated**

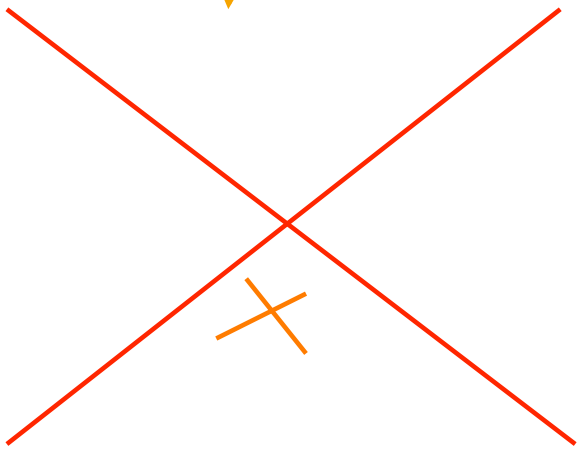
Software Reliability (1/2)

- **Software failures result as a combination of a latent systematic fault with a trigger**
  - **Latent faults**
    - Software specification/implementation was inadequate, incorrect, incomplete
    - Testing did not included the specific signal trajectory that reveals the fault
  - **Relevant triggers**
    - Time-dependent effects, faulty telegrams, same signal trajectories/states, faulty maintenance (can also introduce faults into the software)
  - **Software failures have a *common cause nature***
    - Same faulty software in different divisions affected by common triggers



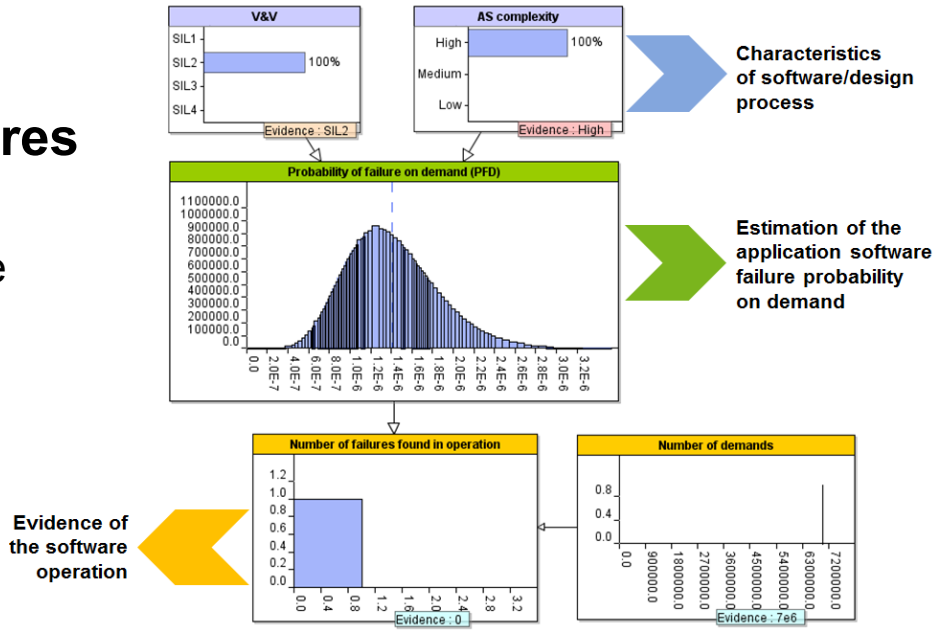
TELEPERM® XS processing module SVE2

- **Software failure effects**
  - **Non-fatal failure:** processor remains in operation but the faulty function is either unavailable or spuriously actuated
  - **Fatal failure:** processor shuts down (unavailable), output signals are set into pre-defined values



Software Reliability (2/2)

- **Relevant failure modes defined systematically on a software FMEA:**
  - Unavailability of complete system (postulated)
  - Unavailability of one subsystem
  - Unavailability of processors which communicate with each other
  - Unavailability of processors in which the same faulty function is processed
  - Unavailability/spurious actuation of one function in all divisions
- **Quantification of software failures**
  - **System software**: using the TELEPERM XS operating experience
  - **Application software**: using Bayesian Networks to combine the characteristics of the software/design with the evidence of the software operation



Use of Reliability Results for Safety Demonstration

- **Results of the reliability analysis are compared with the safety requirements**
  - If requirements are fulfilled → analysis of minimal cut sets (MCS) to demonstrate a well-balanced design
  - If requirements are not fulfilled → analysis of largest contributors
    - Do the failure combinations lead to the failure of the function? → Too conservative modelling assumptions?
    - Are the failure rates/test intervals/testing strategy realistic? → Too conservative input data?
    - Increase the frequency of periodic testing?
- **Sensitivity analyses are important to determine the variation range for the input parameters, in which the targets are still fulfilled**

Conclusions

- **Robust and flexible approach for the verification of safety requirements of digital I&C systems during V&V**

  New approach based on our experience licensing different digital platforms for system upgrade projects and new power plants in different countries

  - Strongly supports the fulfilment of regulatory requirements, minimizing licensing risks
  - Includes failure combinations from the complete signal path (independent failures and CCF considerations for hardware and software)
    - Complete scope of failure modes for hardware and software failures supported by systematic and detailed FMEAs
  - **Credible, realistic and justifiable failure probabilities for software failures**
    - Based on the combination of Bayesian networks with TELEPERM XS operating experience developed within an R&D program at Framatome