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

Mariana Jockenhövel-Barttfeld[a (*)], Stefan Karg[a], Christian Hessler[b] and Herve Bruneliere[c]

[a] Framatome GmbH, Erlangen, Germany
[b] AREVA GmbH, Erlangen, Germany
[c] Framatome SAS, Paris, France

Abstract: Reliability analyses are conducted as part of the verification and validation process during the design phase of the digital instrumentation and control (I&C) systems in modernization and new build projects. The main objective of reliability analyses is to demonstrate that reliability targets (safety goals) imposed on the digital I&C systems by regulators or design authorities within the licensing process are fulfilled. This paper presents a methodology for conducting reliability analyses of digital safety-related I&C systems, which uses fault trees to estimate reliability measures, e.g. failure probability on demand. The approach presented in this paper for conducting reliability analyses goes beyond conventional approaches by considering common cause failures for the hardware modules, software failures for both system and application software and uncertainty analyses for the hardware and software failures. The paper focuses on the interpretation of reliability requirements, the definition of the scope and boundary conditions for the analysis, the level of modelling detail for the fault trees, and on the hardware and software reliability. Finally, the use of reliability results as part of the safety demonstration is addressed.

Keywords: Digital I&C, application software, hardware modules, monitoring, reliability.

1. INTRODUCTION

This paper presents an approach for conducting reliability analyses of digital instrumentation and control (I&C) systems in the frame of nuclear power plants modernization projects, where the existent systems are upgraded, or for new builds. Reliability analyses are conducted as part of the verification and validation (V&V) process of the digital I&C during the system design phase (basic and detailed design).

The main objective of the reliability analysis is to demonstrate that quantitative reliability targets (safety goals) imposed on the digital I&C systems by regulators or by design authorities within the licensing process are fulfilled. Other objectives of reliability analyses are the identification of major contributors which lead to the undesired failure event, the identification of design weaknesses and potential solutions of reliability improvement for the design. For this reason, it is beneficial to conduct reliability analyses at early design phases, considering reliability analyses as a companion tool feeding back into the different design phases.

According to the requirements imposed by the IEC 61226 standard [1], the reliability of the I&C systems that perform category A[*] and B[†] functions shall be assessed and compared to the specifications. Furthermore, [1] states for category A functions that reliability assessments shall consider the effects of common cause failures (CCF), including hardware failures, software failures, and human errors during operation, maintenance, as well as modification and repair activities. The type of analysis chosen shall be

[*] Category A I&C functions are those functions required to reach a non-hazardous stable state, to prevent a design basis event from leading to unacceptable consequences, or to mitigate its consequences (for more details, refer to §5.4.2 of [1]).

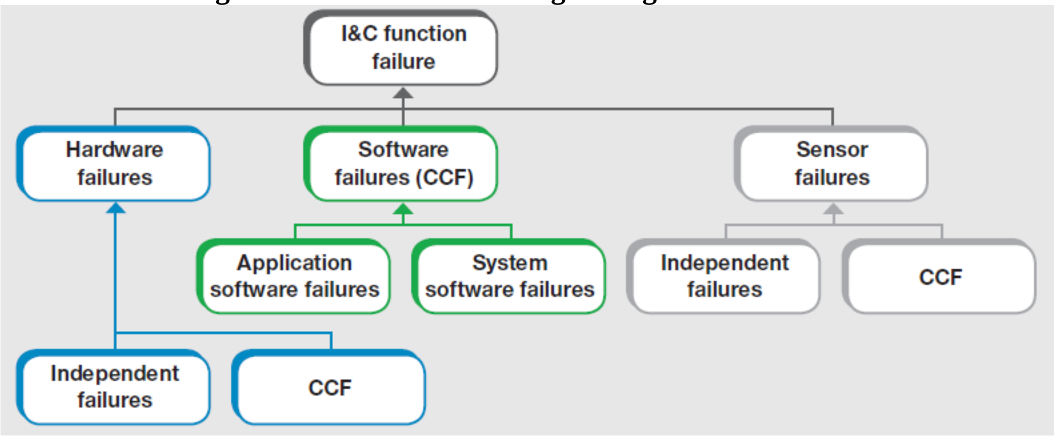
[†] Category B functions are those functions required after a non-hazardous stable state of a design basis event has been reached, to prevent it from leading to unacceptable consequences, or to mitigate the consequences. These are also functions, whose failure during normal operation, would require the operation of a category A function to prevent an accident. For more details, refer to §5.4.3 of [1].

consistent with the reliability requirements; the higher the reliability requirement, the more rigorous the technique.

The development of a methodology for the assessment of the digital I&C reliability is very challenging. This is mainly because conventional techniques for probabilistic safety assessment cannot adequately evaluate the features of digital systems, and thus have to be extended. The biggest issue in the evaluation of the digital I&C reliability is taking common cause failures, the reliability of the software and uncertainty analysis into consideration. The approach presented in this paper addresses these challenging issues.

The methodology presented in this paper uses fault trees to estimate the failure probability (unavailability)/frequency (according to the reliability target imposed, see § 2) of one or more functions, which are modeled as top events of the fault tree model (see Figure 1). The immediate causal events leading to the top event (e.g. single or combination of events, CCF events) are identified and connected to the top event through logic gates (e.g. AND, OR). The events on the lowest level are the *basic events* modelling failures of the hardware and software modules processing the function.

Figure 1: Fault tree modelling for digital I&C failures



The fault tree model is analyzed *quantitatively* and *qualitatively*. For a *quantitative analysis* of the fault trees, a reliability model, which is a set of mathematical formulas that specify how to calculate the reliability, has to be defined for each of the basic events in the fault trees. The selection of the reliability model depends mainly on the detectability of the failure modeled by the basic event. The *qualitative analysis* involves the analysis of the minimal cut sets (MCS), defined as the minimum combination of events leading to the top event, i.e. leading to the failure of the I&C function being analyzed.

This paper is structured as follows. In the next chapter, §2, the safety requirements that have to be demonstrated by the reliability analysis are presented. These requirements define the scope and boundary conditions of the reliability analysis (see §3) and the functional analysis (see §4). Insights regarding the level of detail of the fault tree model are presented in §5. Important aspects regarding the reliability of the hardware and software modules are presented in § 6 and §7, respectively. Even though the methodology presented in this paper is general, some reliability aspects of the hardware and software are specific to the digital system platform for safety I&C, TELEPERM® XS, developed at Framatome. However similar approaches can be considered for the reliability analysis of other digital platforms, if operating experience or sufficiently good reliability data is available. The use of reliability results for safety demonstration purposes is presented in §8. Finally the main results and conclusions are highlighted in §9.

2. RELIABILITY REQUIREMENTS

Safety reliability requirements are defined in safety-related digital I&C systems specifications and apply to all I&C safety-related functions implemented within the systems. These are also commonly named “goals” or “targets” because these requirements are quantitative and have to be demonstrated with a reliability analysis. Reliability requirements commonly impose an upper bound for the failure probability or the failure frequency on the functions processed on the system.

The requirement “failure probability $\leq p_{target}$” refers to the non-response of the function as a consequence of the I&C failure, with p_{target} as the probability upper bound. If the I&C function operates in *low-demand mode*[‡] [2] (e.g. protection, limitation functions), this specification can be interpreted as the upper bound for the failure probability (unavailability) on demand of the function. If the function operates in *high-demand mode*[§] [2] (e.g. control functions), the probability p_{target} can be interpreted as the upper bound for the unavailability of the function.

The requirement “frequency of spurious actuation $\leq f_{target}$” refers to the spurious actuation of the function (without it being demanded by the processes) as a consequence of the I&C failure, with f_{target} being the upper bound for the frequency (failures/time). In this case f_{target} is commonly imposed on functions, whose spurious actuation is related to the plant availability, e.g. reactor trip.

The correct interpretation and understanding of the regulatory requirements is a crucial initial step of the reliability analysis. The requirement specification provides the key information needed to define the scope and boundary conditions of the reliability analysis. The specification defines to which parts of the I&C system the requirements apply, e.g. whether failures of the complete signal path have to be included, and whether sensors failures have to be included. The specifications imposed on the system are highly dependent on the specific project type (i.e. modernization, new build), countries and regulators involved in the licensing process. In modernization projects, existing sensors may be replaced and others may be connected to the new I&C system. In this case, the requirements apply to the parts of the systems being upgraded. For this reason, the consideration of sensor failures depends on the scope of the I&C modernization. In new build projects, failures of all I&C-related components which participate in the function processing are usually included in the reliability analysis, also including failures of sensors.

3. DEFINITION OF SCOPE AND BOUNDARY CONDITIONS

The scope of the reliability analysis is mostly defined by the requirement specification imposed on the I&C system. The requirement specifications provide information which helps to define the top gate as a failure probability or as a failure frequency of a function and helps to build the fault tree model.

It is important to identify the *application extent* of the reliability target, namely: does the reliability target apply to one function being processed in one division or to the complete function (in all divisions)? For the latter, the success criteria of the function have to be considered in the analysis (e.g. two-out-of-four divisions are needed for the successful performance of the function).

As discussed in the previous chapter, the requirements apply to the parts of the systems being updated. For this reason, it is usually assumed that input signals delivered to the I&C system for further processing (e.g. from sensors, if not upgraded, from signals resulting from operator actions) are fault-free in the reliability analysis. Similar assumptions regarding failures of external supply systems (e.g. external power supply for I&C cabinets, cooling of I&C rooms) are made, which are usually not considered within the scope of reliability analyses. Failures of signalization or testing signals do not affect the reliability of the I&C functions and are also excluded.

[‡] According to [2] functions operating in low demand mode have a demand frequency $\leq$ 1 demand/year (e.g. protection and limitation applications).

[§] According to [2] functions operating in high demand mode have a demand frequency > 1 demand/year (e.g. control applications).

Failures of the fault-tolerance design features and their fault-coverage[**] (e.g. signal monitoring) are considered in the scope of reliability analyses. Failures of the *self-monitoring*[††] features are modeled by the hardware failures (see §4.1). Failures of the *engineered monitoring*[‡‡] functions are taken into account in the application software failures (see §7).

4. FUNCTIONAL ANALYSIS

In order to demonstrate the reliability targets imposed on the I&C system, one or more *representative I&C functions* have to be identified for the analysis. A representative function is a function whose processing requirements, complexity and, as a consequence, its reliability are bounding for all other functions or a set of functions implemented on the system.

The selection of a representative function is a challenging task given the large number of I&C functions which are usually implemented on an I&C system. According to our experience it is useful to define selection criteria to classify the functions.

Given the large amount of functions processed in one I&C system, it is usually very challenging to satisfy all criteria defined above with one function, i.e. one function requires the highest number of inputs, issues the highest number of outputs, has the highest dependencies and high relative complexity with respect to other functions. If this is the case, two or more functions have to be selected for the reliability analysis.

5. LEVEL OF DETAIL OF THE PROBABILISTIC MODEL

A digital I&C system has hierarchical levels contained within its architecture, such as

- *Divisions*, which process the signal path from sensor to the actuator (channel). For safety applications a function is processed redundantly by different divisions.
- *I&C units*, which contain several I&C modules and perform a specific task, e.g. an acquisition and processing unit, voter unit.
- *I&C modules*, which carry out a specific part of the function processing, e.g. input/output modules, processor.

As pointed out in [3], reliability analyses of a digital system have to be developed to a level of detail that captures the design features affecting the system reliability, provides the outputs needed for risk evaluation, and for which probabilistic data is available.

The level of detail adopted for the reliability analysis depends on the system design phase in which the analysis is conducted. If the analysis is conducted during an early design phase (e.g. basic design), conservative modelling assumptions have to be done to deal with the missing information related e.g. to the distribution of signals into the hardware, implementation of faulty signals treatment/detection, among others. These assumptions can be refined by updating the reliability model in a later system design phase (e.g. detailed design).

The level of detail adopted in the reliability analysis for modelling failures of the hardware is the *module level*, distinguishing further between failures of the *common board* and the single *channels*, if this is applicable to the module (e.g. for input/output modules). This has been proven to be the correct level of detail, which allows the consideration of dependencies between modules associated to the sharing of hardware (e.g. signals acquired by the same input or issued by the same output module), communication and common cause failures (see §5.2 and §6).

[**] Capability of the I&C system to diagnose failures and to re-configure to reduce or eliminate the impact of the failure. The fault detection coverage is a measure of the system ability to perform the fault detection, isolation and recovery.

[††] Functions implemented on the modules or monitoring mechanisms provided by the processor.

[‡‡] Functions implemented in the application software to detect failures not detected by self-monitoring features.

Failures of the application software (I&C functions in form of code) are modeled at *function level*. This is considered to be the correct level because reliability analyses usually make estimations for single functions (without considering dependencies between them, as done in probabilistic safety analyses, where a software module level can be more convenient). The system software (operating system and communication software) is considered as a whole in order to capture the dependencies of processors operating within the same platform (for more details, refer to §7.2).

The modelling of failures of the *voting logic* involved in the I&C functions can be very laborious, especially if the treatment of faulty signals (e.g. voting degradation) is taken into account. If different redundant types of input signals are available for the function processing, as it is usually the case for safety-related functions (e.g. redundant level, pressure input signals), simplified conservative approaches for modelling the voting logic (e.g. without considering the logic degradation) considerably reduce the modelling effort without leading to overestimated reliability results.

6. HARDWARE RELIABILITY

This chapter addresses the reliability assessment of independent (see §6.1) and common cause failures (see §6.2) of hardware modules.

6.1. Independent Failures of Hardware Modules

The definition of *functional failure modes* for modelling independent failures of the hardware modules is based on the *effect of the failure* on the module, i.e. on the manifestation of the failure on the module functioning. From the reliability analysis point of view it is desirable to group failure modes with regard to their functional consequence in order to simplify the fault tree modelling process and analysis.

Failures of the hardware modules are modeled using basic events. As mentioned in §1, a reliability model has to be assigned to each basic event for the quantitative assessment. The selection of the reliability model depends on the detectability of the failure. Failures of the hardware modules can be classified into *detected* and *undetected.* Detected failures are those failures which can be identified by *self-monitoring* (self-announcing failures), by *engineered monitoring* or those failures which are obvious (detected after occurrence, e.g. spurious actuation of a function). Undetected failures are those which cannot be identified by the fault-tolerance features of the digital system and remain hidden until the modules are periodically tested.

Detected failures (D) can be modeled using basic events with the reliability model “repairable component”, i.e. the failure can be repaired after detection. The probability of a detected failures p_D is estimated by:

$$p_D = \lambda_D \; MTTR \qquad (1)$$

where λ_D is the failure rate of detected failures and $MTTR$ is the mean time needed to repair the module.

Given the fact that undetected failures can only be identified during periodic testing or in case of demand[§§], undetected failures (U) can be modeled with the “periodically tested component” reliability model. The probability of undetected failures p_U is well approximated by:

$$p_U = \frac{1}{2} \lambda_U \; TI \qquad (2)$$

where λ_U is the failure rate of undetected failures and TI is the test interval for periodic testing.

[§§] This is applicable to functions which operate on low-demand mode.

Failure rates for the hardware modules have to be assigned to each identified failure mode for the module and channels (if applicable). The detection associated to the failure modes are defined based on the fault-detection coverage of each particular module. This information is usually gathered in a failure mode and effect analysis (FMEA), which is used as a basis for modelling the reliability analysis fault trees. This methodology uses *theoretical failure rates* for the TELEPERM XS modules in reliability analyses, which are estimated using part count calculations, according to the Siemens Standard 29500 [5] at a certain reference temperature (air inlet temperature at the subrack). The theoretical failure rate estimates are conservative with respect to operating experience of the TELEPERM XS platform. Failures of TELEPERM XS modules during plant operation are evaluated, and the “field failure rates” are calculated on the basis of the information obtained, and updated four times per year. Experience shows that these field failure rates are usually significantly smaller than the theoretically calculated ones, typically by about one order of magnitude. Theoretical failure rates are bounding and are independent from the modules operating time. This is a great advantage, especially for demonstrating reliability targets using reliability analyses.

Uncertainty parameters for the failure rates of hardware modules are obtained from the operating experience using the Chi-squared distribution. The error factors for the TELEPERM XS components are applied to the theoretical TELEPERM XS failure rates (a lognormal distribution is assumed). This is a conservative treatment because the theoretical estimates used for the base failure rates are significantly more pessimistic than the actual failure rates derived from the operating experience.

6.2. Common Cause Failures of Hardware Modules

As imposed by standards for safety functions (e.g. [1]) the effects of CCF of the hardware have to be considered in reliability analyses of digital I&C. A CCF affecting an I&C system is defined as a result of a triggering event causing coincident failures of two or more separate divisions (channels) in a multiple channel system. A CCF can be caused by the accumulation of undetected faults, which remain hidden until these are triggered, causing the coincidental failure of some or all channels.

Errors can be introduced to the hardware modules during the production or manufacturing. These errors can turn into hardware faults with increased operation time. Ageing effects can also lead to deviations from the specified and tested system behavior. These faults are uncorrelated (random failures), remain hidden (until the modules are periodically tested) and have the potential of accumulating in different divisions of the I&C system. The correlation of these faults is given by a demand (CCF trigger) on the I&C function/s processed by the faulty modules, which manifest the accumulated faults as a common cause failure of the faulty modules.

Reliability analyses take into consideration that faults of the hardware modules of the same type processing redundant signals can accumulate and lead to a CCF. This is applicable to hardware faults which remain undetected (the faults can accumulate between two periodical tests or demands). The CCF of detected failures is usually not considered in reliability analyses. This is because detected failures can be identified and repaired in a very short period of time (e.g. eight hours), resulting in very low probabilities of simultaneous failures with respect to the undetected faults, which can accumulate over a much longer period of time.

CCF failures of the hardware modules can be modeled using traditional CCF models, such as the Alpha Factor Model and the Beta Factor Model (see [6]). The disadvantage of traditional methods is that the CCF parameters available are generic and rather conservative for I&C hardware modules, leading to overly conservative reliability results. In the IEC 61508 standard [2] the Beta Factor Model was adapted for I&C systems. The method involves the estimation of Beta parameters depending on the level of redundancy and on the results obtained during a check list evaluation. The estimation of the parameters involves the use of pre-defined (pre-established) values, for which no justification is found in the method. According to our experience, the CCF parameters obtained with the I&C-extended Beta Factor Model [2] for safety-relevant systems, do not differ considerably from the generic Alpha Factor Model parameters, but the level of effort needed to obtain these parameters is considerably higher. For this reason, the CCF modelling of hardware in I&C reliability analyses is still a challenging issue and in order to resolve it, the development of I&C-specific methods and parameters is needed.

7. SOFTWARE RELIABILITY

The consideration of the reliability of software in reliability analyses is a very challenging task. The IAEA standard [4] points out that the main difficulties that arise in the licensing of digital I&C are related to the software. The main challenges associated with software reliability are the definition of covering failure modes (capturing all possible software failure effects on the I&C system) and their quantitative assessment.

This chapter presents a qualitative analysis on software faults, failures and triggers (see §7.1) and the definition of failure modes and their quantitative assessment for reliability analyses (see §7.2).

7.1 Software Faults, Failures and Triggers

Software failures result in a combination of a latent fault with a trigger and are caused by systematic faults (i.e. due to errors made when writing the design specification or implementing the design, or when performing modification). The randomness associated with software failures arises from the way the operational environment changes. In the case of digital systems, the software works incorrectly i.e. it does not perform its intended function, if:

- Its specification was inadequate, incomplete or incorrect,
- Its specification was interpreted incorrectly during implementation, and
- Testing did not include the specific signal trajectory that reveals the fault.

Since software cannot be proven to be 100% error free, software design faults are a credible source of software failures. As pointed out in [9], latent faults may also be related to maintenance or modification activities. Software failures have a *common cause nature* given the fact that:

- The same single piece (module) of software is processed in all divisions of a redundant I&C system,
- There are common triggers which can act upon all divisions of an I&C system, turning latent systematic faults of the software into coincidental failures.

The activation of a software fault might lead either to a *fatal* or a *non-fatal failure* of the processor. A *fatal failure* is characterized by the ceasing of the processor activity, i.e. the generation of outputs ceases and an exception handler sets the output values into defined fail-safe (predefined) values. A *non-fatal failure* is characterized by the correct execution of the code, but the code contains, for example, an algorithm which is inappropriate for certain values of the signals. Generally, the effect is that the processing unit continues to operate cyclically (non-fatal consequences for the processor), but the requested function is not executed (unavailable) or a different response than expected is obtained (spurious actuation of the function).

Based on [9], the relevant triggers that can activate latent software faults causing coincidental failures relevant for reliability analyses[***] (according to the scope defined in §3), are:

- Temporal effects
- Events associated with faulty communication between processors (e.g. faulty telegrams)
- Signal trajectory/internal states
- Human actions (e.g. inadequate/faulty maintenance).

The *"temporal effects" trigger* encompasses common cause failures which may be initiated by time-dependent effects (internal trigger mechanisms), such as the depletion of resources by time (e.g. leakages in the memory allocation), or by accumulated time of operation. All processors with the same operation time can be affected by this trigger.

The *"faulty telegrams" trigger* encompasses common cause failures which may be triggered by the transmission of information via serial data links. The failure mechanism is given by the existence of an undetected random failure in a sending processor, which causes the transmission of invalid data. If the system software of the receiver processor contains an undetected fault in the validation of the received data (e.g. wrong implementation of message checking), the corrupt data remain undetected. If these corrupt data are processed, an exception (interruption) is activated, leading to the shutdown of the processor. All processors with direct communication can be affected by this trigger.

The *"same signal trajectory/internal state" trigger* encompasses common cause failures which can be triggered by a sequence of input data from the process[†††]. All processors with the same operating system, the same application software and which process exactly the same signal trajectories, can be affected by this trigger.

Human actions can trigger latent software faults mainly in maintenance-related activities. Faulty maintenance can be a trigger of latent software faults and can also introduce latent faults into the software. For most of the safety I&C architectures, failures of maintenance can, at worst, lead to the failure of all processors which communicate with each other. Even though maintenance is usually done in one division at a time, the failures caused by faulty maintenance can potentially spread in all divisions due to the communication between processors. For this reason, the effect of software failures triggered or induced by maintenance, are the same as the effects caused by failure mechanisms triggered by faulty telegrams mentioned above.

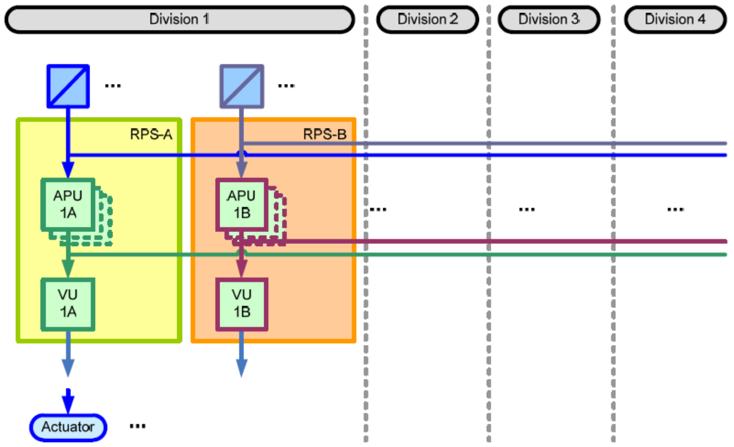
7.2 Software Failure Modes and Effects

The generic safety I&C architecture of a Reactor Protection System (RPS) defined in [7] (see Figure 2) is considered with the aim of defining failure modes for software failures and defining their effects. The RPS consists of two diverse subsystems, called RPS-A and RPS-B, both divided into four physically separated divisions. The platforms of both subsystems are assumed to be identical, namely TELEPERM XS. The extent of diversity between RPS-A and RPS-B may vary, but it is assumed that both subsystems perform different functions. It is assumed that the system has more than one acquisition and processing units (APU) and voting units (VU) per subsystem and division.

[***] The "external events" trigger (such as seismic event, flooding, and extreme ambient conditions) is not relevant to the reliability analyses scope (see §3) as they do not interact with the software. They may only act indirectly as triggers (through the plant response), which is only relevant if the probabilistic analysis includes the influence of such events.

[†††] This CCF mechanism presumes that a very rare (not tested) signal trajectory can be combined with a latent software fault.

Figure 2: Example I&C architecture of a protection system [7]



The TELEPERM XS *system software* consists of the operating system, communication software and runtime environment. The *application software* is a representation of the application functions in the form of code, which is executed and controlled by the runtime environment during the operating cycle. The operation of the TELEPERM XS platform is independent from plant demands. Detected failures always lead to a ceasing of the processor activity (shutdown, fatal failures) with outputs set to pre-defined fail-safe values. In addition to the design measures aimed at such fail-safe behavior, various design features, minimize the fault propagation, e.g. separation between system and application software, separation between application functions, and communication independence between processing units.

Table 2 presents the relevant failure modes for the TELEPERM XS system and application software for reliability analyses of safety systems (for more details, refer to [7] and [8], respectively). These failure modes are exhaustive because they consider all possible potential CCF triggering mechanisms, and effects which can result from the triggers analyzed in §7.1.

Table 2: Generic failure modes for modelling software reliability in reliability analyses

Basic event	**Description (failure effect)**	**Reliability parameter**
SYSTEM_OFF	Unavailability of the complete system	Failure rate
1SS_OFF	Unavailability of one subsystem	Failure rate
PROC-COM_OFF	Unavailability of all processors which communicate with each other	Failure rate
PROC-AF_OFF	Unavailability of all processors in which the faulty function is processed	Failure rate
1AF_OFF	Unavailability of one function in all divisions	Failure probability / rate[‡‡‡]
1AF_SPR	Spurious actuation of one function in all divisions	Failure probability / rate[‡‡]

The unavailability of the complete system can be modeled using the basic event “SYSTEM_OFF“ (e.g. RPS-A <u>and</u> RPS-B for the architecture of Figure 2). Failure mechanisms affecting processors within both subsystems can be triggered by the same internal states (latent fault in the system software) or by the same signal trajectories (latent fault in the application software). In both cases, failure of the complete system results from an insufficient diversity of application software in both subsystems. The system failure is caused by an impermissible interference of the application software with the system software, which triggers an exception, and thus shuts down the processor (fatal failure, output signals set into “fail-safe” values). If sufficient diversity is implemented in the design of the application software, the failure

[‡‡‡] Failure probability on demand has to be estimated for a function operating in low-demand mode, whereas a failure rate has to be estimated for a function operating in high-demand mode.

rate associated to such an event is extremely low given the weak correlation between both RPS subsystems.

The unavailability of one subsystem can be modeled using the basic event “1SS_OFF” (e.g. RPS-A or RPS-B). The unavailability of processors with communication (within one subsystem) can be modeled using the basic event “PROC-COM_OFF, see Table 2).

The TELEPERM XS platform failures of the system software (modeled with the basic events “SYSTEM_OFF”, “1SS_OFF” and “PROC-COM_OFF”, see Table 2) can be estimated using the TELEPERM XS operating experience (for details, refer to [7]). Uncertainty parameters can also be estimated using the operating experience in a similar way as was done for the hardware modules (see §6.1).

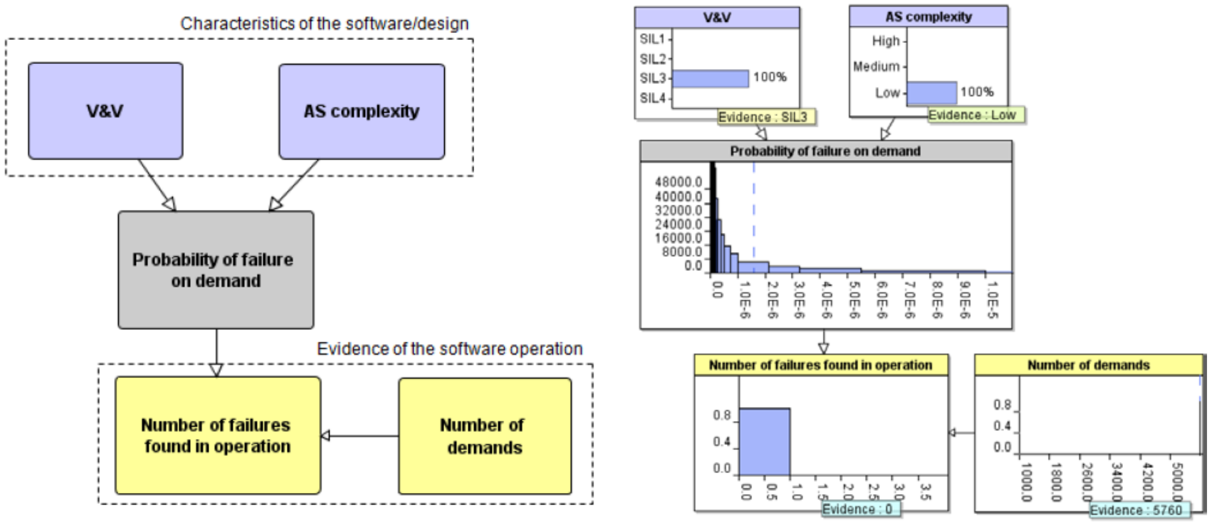
Failures of the application software can lead to the shutdown of the processor (“PROC-AF_OFF”), and to the unavailability (“1AF_OFF”) or to the spurious actuation of the faulty function (“1AF_SPR”) in all divisions in which the function is processed. The reliability and uncertainty parameters for the failure of the processor (shutdown), caused by fatal failures of the application software (basic event: “PROC-AF_OFF”), can be estimated using the TELEPERM XS operating experience (for more details, refer to [7]). The reliability and uncertainty parameters for the failure of one function (basic events: “1AF_OFF”, “1AF_SPR”) can be estimated using the Bayesian Network shown in Figure 3, which combines the use of statistical methods with TELEPERM XS operating experience (for more details, refer to [8]).

8. USE OF RELIABILITY RESULTS FOR SAFETY DEMONSTRATION

The quantitative analysis of the fault tree model calculates a failure probability (unavailability) or a frequency of spurious actuation of a function, according to the reliability requirements imposed on the system (see §2). A list of minimal cut sets is generated, showing the most important events or combination of events leading to the failure of the I&C function.

The result of the reliability analysis is compared with the reliability requirements specified in the requirement specification in order to verify that the digital system fulfils the reliability requirements. If reliability requirements are fulfilled (reliability of the system $\leq$ target), the analysis of the MCS and their contribution to the function failure can be used to demonstrate that the I&C system has a well-balanced design, by showing that no failure combination completely dominates the reliability results.

Figure 3: (a) BN model for predicting the failure probability on demand of an application function – (b) BN with evidence for a protection function [8]



(a) (b)

If reliability requirements are not fulfilled (reliability of the system > target), the largest contributors to the reliability results have to be identified. The reasons for these large contributions have to be critically analysed:

1. *Do the failure combinations in the MCS lead to the failure of the function?*
   It can be the case that too conservative modelling assumptions were considered to simplify the fault tree modelling. These modelling assumptions may involve: dependencies assumed between different software/hardware modules, simplifying modelling assumptions regarding voting logic degradation, and faulty signals treatment.

2. *If the failure combinations in the MCS are realistic, are the inputs for the basic events realistic?*
   It can be the case that too conservative inputs (e.g. failure rates, test intervals) were used for the reliability analysis, which do not reflect the real behaviour of the system and give rise to unrealistic reliability results (e.g. poor fault-coverage considered, resulting in large failure rates for undetected failures, conservative CCF factors).

3. *If the modelling assumptions and the reliability model inputs are realistic, is it possible to reach the reliability requirements by increasing the frequency of periodic testing?*
   The results of the reliability analysis are usually dominated by a combination of undetected failures (CCF and independent undetected failures). For this reason, a measure to decrease the reliability results is to increase the periodic testing frequency. This means that, based on the reliability analysis results, a more frequent periodic testing (generally affecting input/output modules) can be suggested. This is in-line with standard [1], which states that for category A functions, the intervals for the proof test shall be determined by using the likelihood of occurrence of the undetected failure and the required reliability of the function. Sensitivity analyses are very helpful when exploring how changes of the input parameter influence the reliability results. For example, the reliability analysis can provide a range of periodic test frequencies for which the reliability targets are still fulfilled.

9. CONCLUSION

This paper presented an approach for conducting reliability analyses of digital I&C systems, aiming to demonstrate the reliability requirements imposed during licensing. These analyses are commonly conducted as part of the V&V activities during the system design.

The approach presented in this paper is generic and it was developed based on our experience with different digital platforms, which are commonly involved in the safety I&C design of new builds. The approach involves the development of a fault tree model to estimate the function failure probability/frequency (depending on the requirements) of one or more representative functions. The paper discusses the level of detail for modelling the fault trees in a proper way, so that the design features affecting the system reliability can be captured. Reliability aspects of the hardware and software are presented. These include the definitions of functional failure modes for independent and common cause failures, the estimation of failure rates and uncertainty parameters. Furthermore, the use of reliability results for safety demonstration purposes is discussed together with the importance of sensitivity analyses, which estimate a variation range of the input parameters (e.g. test intervals) for which the system reliability target can still be fulfilled.

Future work on reliability analyses for digital I&C involves the estimation of I&C-specific common cause failure parameters, combining the use of the TELEPERM XS operating experience with statistical methods.

References

[1] IEC 61226, International Standards – Nuclear power plants – *Instrumentation and control important to safety – Classification of instrumentation and control functions*

[2] IEC 61508, International Standards – Nuclear power plants – *Functional safety of electrical/electronic/programmable electronic safety-related systems*

[3] NUREG/CR-6962 – *Traditional Probabilistic Risk Assessment Methods for Digital Systems*

[4] IAEA Nuclear Energy Series No NP-T-1.4 – *Implementing Digital Instrumentation and Control Systems in the Modernization of Nuclear Power Plants*

[5] Siemens Standard 29500 - *Failure rates of electronic components, part 1: General, expected values*

[6] NUREG/CR-5485 – *Guidelines on Modelling Common-Cause Failures in Probabilistic Risk Assessment*

[7] O. Bäckstrom, J.-E. Holmberg, M. Jockenhövel-Barttfeld, M. Porthin, A. Taurines and T. Tyrväinen, *"Software reliability analysis for PSA: Failure Mode and Data"*, Nordic nuclear safety research (NKS) Report, NKS-341 (2015)

[8] M. Jockenhövel-Barttfeld, A. Taurines and C. Hessler, *"Quantification of application software Failures of digital I&C in probabilistic safety analyses"*, Proceedings of the 13th International Conference on Probabilistic Safety Assessment and Management (PSAM 13) (2016)

[9] IAEA Nuclear Energy Series No NP-T-1.5 – Protecting against common cause failures in digital I&C systems of nuclear power plants