QUANTIFICATION OF APPLICATION SOFTWARE FAILURES OF DIGITAL I&C IN PROBABILISTIC SAFETY ANALYSES

Mariana Jockenhövel-Barttfeld, Andre Taurines, Christian Hessler

AREVA GmbH - Henri-Dunant-Strasse 50, 91058 Erlangen, Germany – Email: mariana.jockenhoevel-barttfeld@areva.com

This paper presents a quantification method to predict the reliability of application software of digital instrumentation and control (I&C) systems in a nuclear context. Failure probability on demand and the failure rate (during continuous operation) of application software are estimated using Bayesian networks. The method updates prior beliefs using evidence of the digital system platform TELEPERM® XS into a final assessment. The reliability estimations can be directly considered in probabilistic safety analyses, where application software failures can be modeled using basic events.

I. INTRODUCTION

This paper presents a quantification method developed by AREVA to estimate the reliability of application software of digital instrumentation and control (I&C) systems. The main motivations for developing this method rise from the fact that

- There is no common approach in the nuclear field for quantifying the system[a] and application[b] software reliability of digital I&C systems in probabilistic analyses, and
- There is a clear increasing trend from regulators and authorities requiring the consideration of software failures in probabilistic/reliability analyses.

Previous work in this field done by the authors in collaboration with the DIGREL (DIGital RELiability) project includes a definition of failure modes and effects, and the first quantitative estimations for system and application software failures[1]. Failure rates for the system software have been estimated using the operating experience of TELEPERM® XS (TXS), which is the digital system platform for safety I&C developed by AREVA. The direct use of operating experience to assess system software failures is well justified and accepted by the "nuclear community", mostly based on the fact that safety digital platforms are widely monitored and operate continuously and independently from plant demands.

The use of operating experience for the estimation of the application software reliability is not straightforward and relies on the definition of homogeneous collectives. Digital platforms for safety systems are designed to be very reliable and, as a consequence of this, application software failures are rarely observed during the operation of digital I&C systems. This makes the probabilistic assessment very challenging, given the fact that most of the applications are failure-free but, from the probabilistic point of view, cannot be claimed to be fault-free. Latent faults may still not have been activated given the very infrequent demands associated with I&C safety applications. In addition, conventional approaches used to estimate failure probabilities for systems operating on a demand mode in probabilistic safety analyses (PSA) cannot be applied to the application software. This is mainly because application software faults, which were introduced during the design, cannot be detected during maintenance. The software functionality is no longer tested after the design has been completed. For this reason, the Bayesian approach is considered to estimate the application software reliability. The Bayesian approach takes the view that probabilities are subjective and they represent the strength of belief of an observer about whether certain events will take place. This observer will have some prior beliefs, which will change as a result of seeing the outcome of the "experiment" (i.e. the collection of data). For the purpose of the application software reliability analysis, the major advantage

[a] Operating system, runtime environment and communication software.
[b] Function diagrams of I&C functions using pre-programmed function blocks, which are automatically transformed into a code.

of the Bayesian theory is the idea that prior beliefs can be quantified and incorporated with experimental evidence of the TXS system platform into a final probability assessment.

II. DEFINITION AND CLASSIFICATION OF APPLICATION SOFTWARE FAILURES

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

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

In this paper an AS failure can result either from a faulty definition of the functional requirements or from the faulty implementation of the requirements into the AS and not being detected during testing. Since the AS cannot be proven to be 100% error-free, software design faults are credible sources of software failures. Latent faults may be also related to maintenance or modification activities[2]. AS 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, and there are common triggers which can act upon all divisions of an I&C system, turning latent systematic software faults into coincidental failures. A simplified classification of AS failures based on Ref. 1 is presented in Figure 1. This classification considers the redundant input signals from the process as triggers of AS faults. Other triggers of software faults, such as maintenance, time-dependent effects or hardware failures, have been analyzed in Ref. 1 and are not considered within the scope of this analysis. This classification distinguishes between AS failures which occur independently from a plant demand (triggered by signals during normal operation) and those triggered during a plant demand[c].

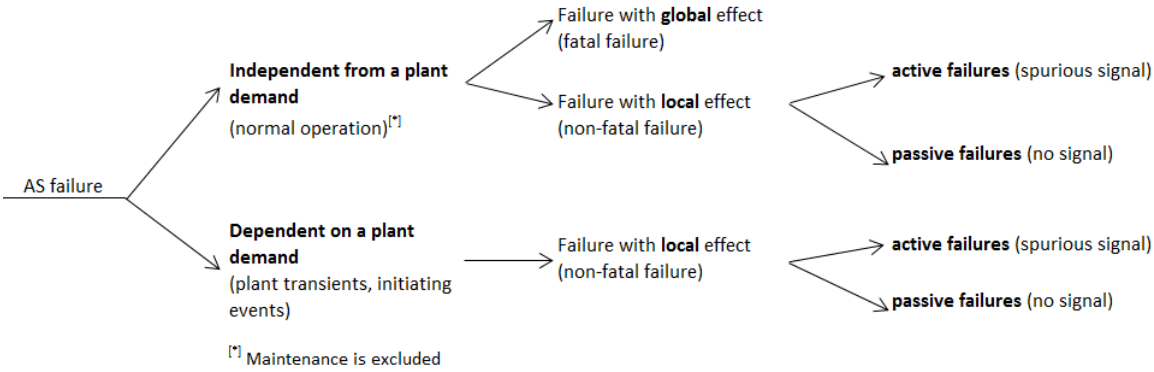


Fig. 1. Classification of application software failures.

Demand-independent AS failures are those failures which are self-reporting, i.e. they are detected by the self-monitoring features of the I&C system via self-monitoring or by an observer (e.g. in case of spurious operation). AS failures which cannot be detected by the self-monitoring (not-self-reporting) and can only be revealed in the case of a demand, are referred as demand-dependent failures. An example of a demand-dependent AS failure is the specification of a faulty threshold value, e.g. too high. The failure becomes evident only after the demand and subsequent unavailability of the application. Note that the state of the plant associated with demand-dependent/independent failures depends on the applications. For limitation and control applications, both demand-dependent and independent failures can occur during the normal operation of the plant. For protection applications, demand-dependent failures take place during the occurrence of an initiating event (e.g. transient) and demand-independent failures can occur during the normal operation of the plant.

AS failures can be further classified according to their impact on the processor operation. AS failures with a global effect are those which affect several applications running on the processor. In the worst case, all applications allocated to the

[c] A demand to an I&C system occurs when a signal value exceeds of falls below a certain threshold, leading to a change in the state of the actuated component.

processor are affected, e.g. shutdown of the processor via an exception handler[d]. Failures with a global effect (so-called fatal failures) have been assessed in Ref. 1 and are outside of the scope of this paper. AS faults with a local effect affect only the faulty application and the processor continues the cyclic processing (non-fatal, all other applications are unaffected). These are the failures assessed in this paper. Note that AS failures with a local effect can result in

- Unavailability of the faulty application (see passive failures in Figure 1) or
- Spurious actuation of the faulty application (see active failures in Figure 1), as a consequence of e.g. a too low specified threshold.

III. USING OPERATING EXPERIENCE AS EVIDENCE FOR RELIABILITY ESTIMATIONS

The challenge of using operating experience as evidence for the assessment of AS failures relies on the definition of homogeneous collective groups for the applications software. The following common key design and functional elements are identified to support the homogeneity of application functions belonging to different I&C systems implemented in different plants using the same digital platform:

- The applications are of the same kind, i.e. implement nuclear application functions of safety-related systems,
- The applications are designed using standard function blocks with a similar parametrization in several plants,
- The application functions are designed under the same or very similar verification and validation (V&V) processes,
- The AS code is automatically generated using the same code generator, compiler/linker tool chain, libraries, and
- The applications are called and controlled by the same runtime environment.

According to the above criteria it is possible to pool together applications independently of the I&C system in which they are implemented (e.g. protection, limitation, control) and independently of the plant to build a cumulative collective for estimation of the application software failure probabilities[1]. For software operating in a demand mode, the operational profile (demand frequency) of the application has to be considered additionally for pooling data. The operational profile is not considered to be a criterion which defines software homogeneity but it affects the number of observed AS failures found during operation. Two cumulative collectives are defined for the assessment of demand-dependent AS using the concept of software operation modes[3]

- Low-demand AS: applications with a demand frequency $\leq$ 1/year (e.g. protection and limitation applications),
- High-demand AS: applications with a demand frequency > 1/year (e.g. control applications).

The question which arises is: is it possible to use the operating experience of one application as evidence to estimate reliability parameters for a different application? In other words: which applications can be pooled together for reliability estimations? In Table I and Table II the analysis is presented for demand-dependent failures and demand-independent failures, respectively. For the estimation of a failure probability on demand (PFD) of a protection application, the operating experience of limitation applications can be considered as evidence (see green fields in the first row of Table I).

TABLE I. Use of evidence for the estimation of the PFD of one application

Probability of failure on demand estimation for		Evidence (number of failures/demands)		
		Protection AS	Limitation AS	Control AS
Protection AS	SIL3/low complexity			
Limitation AS	SIL2/medium complexity			
Control AS	SIL2/high complexity			

TABLE II. Use of evidence for the estimation of the failure rate of demand-independent applications

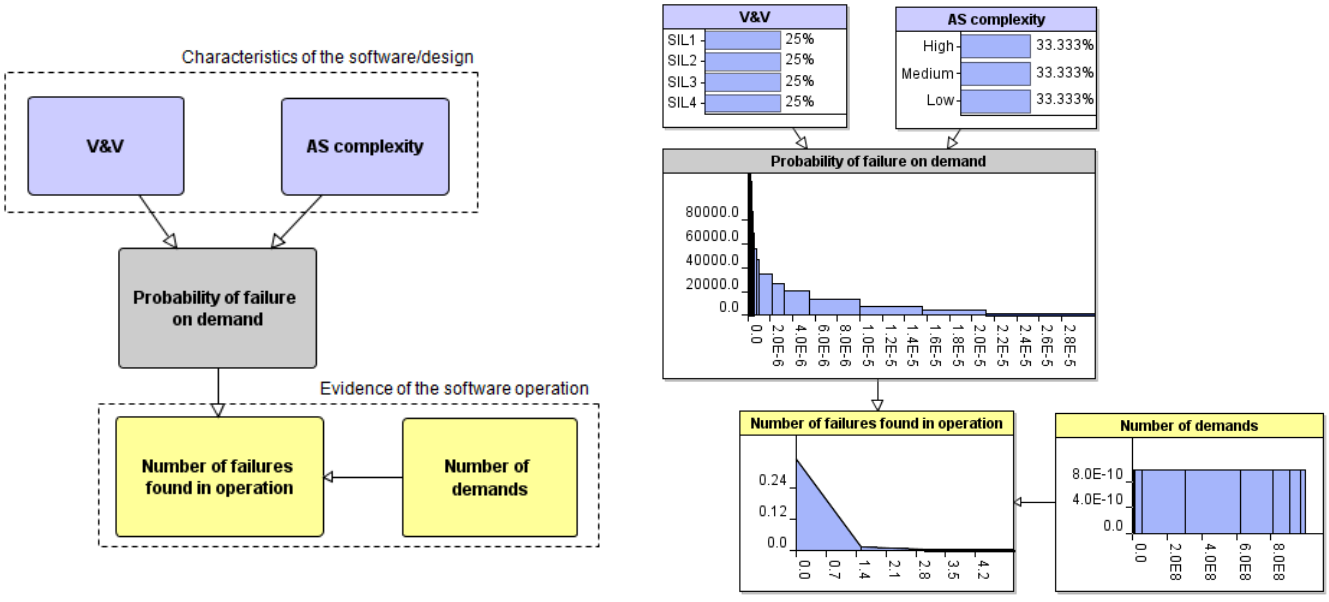
Failure rate estimation for		Evidence (number of failures/operating time)		
		Protection AS	Limitation AS	Control AS
Protection AS	SIL3/low complexity			
Limitation AS	SIL2/medium complexity			
Control AS	SIL2/high complexity			

[d] Demand-dependent failures with global consequences are extremely unlikely and lead, in the worst case, to the shutdown of the processor[1]. These failures are considered to be covered by demand-independent failures with global effects.

This is because limitation applications are lower classified than the protection ones and both operate in a similar demand mode[e] (low-demand operation mode). Protection and control applications should not be used as evidence to estimate the PFD of a limitation application (see red fields in the second row of Table I). This is because protection applications have a lower complexity than limitation ones and control applications operate in a different operation mode from limitation applications (high-demand operation mode). Similarly, the operating experience of protection and limitation applications should not be considered as evidence to estimate the PFD of control applications (see red fields in the third row of Table I). The same reasoning is used to define the use of evidence for the estimation of AS failure rates (see Table II).

IV. BAYESIAN NETWORK FOR THE ESTIMATION OF THE PROBABILITY OF FAILURE ON DEMAND OF ONE APPLICATION FUNCTION

A BN is a direct graph together with a set of probability distributions where each node of the network represents uncertainty variables and the arcs represent the existence of a causal/influential relationship between two variables. In the BN model shown in Figure 2 (a) the probability of failure on demand of one application function (in all divisions in which it is implemented) is of interest to predict. The PFD depends on the software design process quality (verification and validation - V&V- process) and on the complexity of the application software (see blue nodes in Figure 2 (a)). The software V&V is considered as a measure of the quality of the design process, including the experience of the design team and the quality and effort of testing activities during the software validation phase. Note that the higher the system classification, the higher the V&V requirements and consequently the higher the likelihood to discover latent faults during the system design phase. Regarding software complexity, the more complex the application, the higher the likelihood of having remaining latent faults in the software (not discovered by tests during the software design validation). In addition, the PFD of an application depends on the operation profile (i.e. demand frequency) of the application and on the number of failures found in operation (see yellow nodes in Figure 2 (a)). Note that the demands are the triggers of AS faults, i.e. the more frequently the application is demanded, the more explored the spectrum of input signals combinations and the higher the likelihood to activate a latent fault into a failure. For the quantitative part of the BN there is a conditional probability distribution associated with each child node. In Figure 2 (a), the nodes "V&V" and "AS complexity" are the parent nodes of the child node "probability of failure on demand". The nodes without parents, "V&V"[f] and "AS complexity", are ranked nodes to model qualitative judgments in Bayesian networks and are quantified through their marginal probability distributions (uniform prior distributions are assumed).



(a) (b)

Fig. 2. (a) BN model for predicting the failure probability on demand of an application (b) Marginal distribution for variables (high-demand application software operation mode).

[e] Protection and limitation applications have a similar number of accumulated demands[1].
[f] Note that for the V&V node, the Safety Integrity Level (SIL) classification[3] is used only as a qualitative measure to indicate different categories of the design process quality.

For the prior distribution of the probability of failure on demand a Beta distribution, *Beta(α, β, 0, 1)*, is assumed, where values for (*α, β*) are chosen to reflect our experience . The prior distribution is specified as a node probability table, which defines the PFD beta prior distributions for each combination of the states of the parents (V&V and AS complexity) as a partitioned expression. Two prior Beta distributions are considered, one for the low-demand and another one for high-demand operation modes of the AS. For the node "number of failures found in operation" it is assumed that the number of failures has a Binomial distribution, *B(x, PFD)* where x = 1, 2, …, n_D is the number of trials (demands) and $PFD \in [0,1]$:

$$P(X = x) = \frac{n_D!}{(n_D - x)!x!} PFD^x (1 - PFD)^{n_D - x} \qquad (1)$$

Figure 2 (b) shows the marginal distributions of the high-demand AS BN models before any evidence has been entered. This represents the uncertainty before any specific information (evidence) is entered about the variables. In Section VI the result of entering observations (evidence) into the model is shown.

V. BAYESIAN NETWORK FOR ESTIMATION OF THE FAILURE RATE OF ONE APPLICATION FUNCTION

The BN model for predicting the failure rate or the mean time between failures (MTBF, with MTBF = 1/failure rate) of demand-independent application failures is presented in Figure 3 (a). The structure of the model is similar to the BN used for the prediction of the PFD (see Figure 2). The failure rate (or MTBF) of an application function depends on the software design process quality (V&V) and on the complexity of the application (see blue nodes in Figure 3 (a)). In addition the failure rate (or MTBF) of an application function depends on the software operation (execution) time and on the number of failures observed during the operation (see yellow nodes in Figure 3 (a)). As described in the previous section there is a conditional probability distribution associated to each child node (e.g. MTBF and the failure rate). For the prior estimate of the MTBF a Gamma distribution, Gamma (α, β), is assumed:

$$P(X) = x^{\alpha - 1} \frac{\beta^{\alpha} \exp(-\beta x)}{\Gamma(\alpha)} \qquad (2)$$

with an expected value $E[X] = \alpha \beta$ and parameters $\alpha > 0$ and $\beta > 0$. The prior distribution is chosen to reflect our experience with the MTBF for each combination of states of the parents "V&V" and "AS complexity" (using a partitioned expression). The node "number of failures found in operation" is assumed to be Poisson distributed so that the number of failures is conditioned to the failure probability λ = 1/MTBF. This node is a deterministic node which is modeled as $\lambda * T_{AF}$, with T_{AF} as the operating time observed for the application. Figure 3 (b) shows the marginal distributions of the BN failure rate model before any evidence has been entered. In Section VI the result of entering evidence into the model is shown.

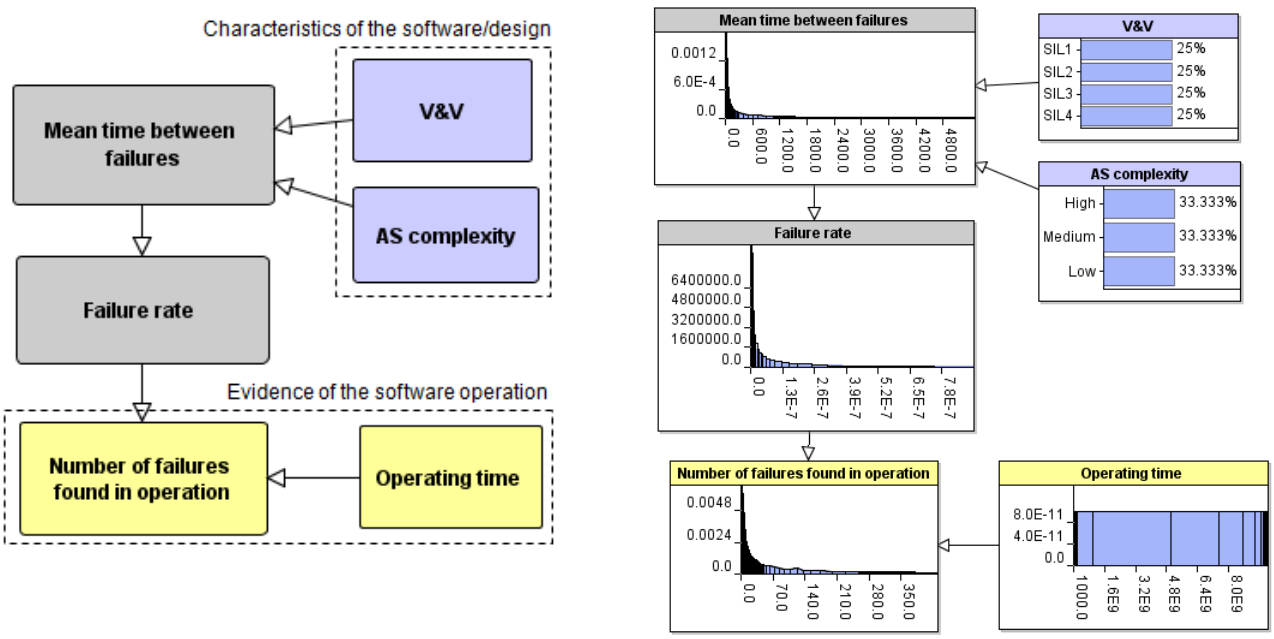
VI. RELIABILITY ESTIMATIONS FOR PROBABILISTIC ANALYSES

The BN models are implemented in the software AgenaRisk 7.0 Lite[4]. Three examples are presented next to illustrate the estimations of the PFD and application software failure rates for probabilistic/reliability analyses. The operating experience of the TXS system platform is considered as evidence for the BN models. The total accumulated operating time of the TXS applications functions until end of 2013 amounts to 1.7E+8 hours[1]. The accumulated operating times for the protection, limitation and control applications are 1.1E+8 hours, 2.0E+7 hours and 3.6E+7 hours, respectively.

The first example considers a control application function and the following evidence:

- The control system is claimed to be SIL2 and runs in a high-demand operation mode,
- High complexity is assumed for this control application function,
- The accumulated operation time observed for TXS control applications amounts to 3.6E+7 hours with no evidence of demand-independent failures and
- This application function is demanded approx. four times a day[g]. The cumulative demands during the operating experience of control applications amounts to approx. 7E+6 with no evidence of demand-dependent failures.

[g] This is an average demand rate for the borating pumps estimated with operating experience from two plants over approx. 24 years of operation[1].



(a) (b)

Fig. 3. (a) BN model for predicting the failure rate of an application – (b) Marginal distribution for variables.

The BN model estimates a failure probability on demand (expected value, see blue dotted line in PFD node of Figure 4 (a)) of 1.4E-6 for one control application function in all divisions (with prior distribution for high-demand operation mode). The failure rate of this control application function during normal plant operation (demand-independent) is approx. 1.2E-7/h (see Figure 4 (b)). If a mission time of 24 hours is considered (in line with PSA considerations), the probability of demand-independent failures is 3.0E-6. The total failure probability of the control application function is approx. 4.4E-6 (sum of demand-dependent and independent failures).

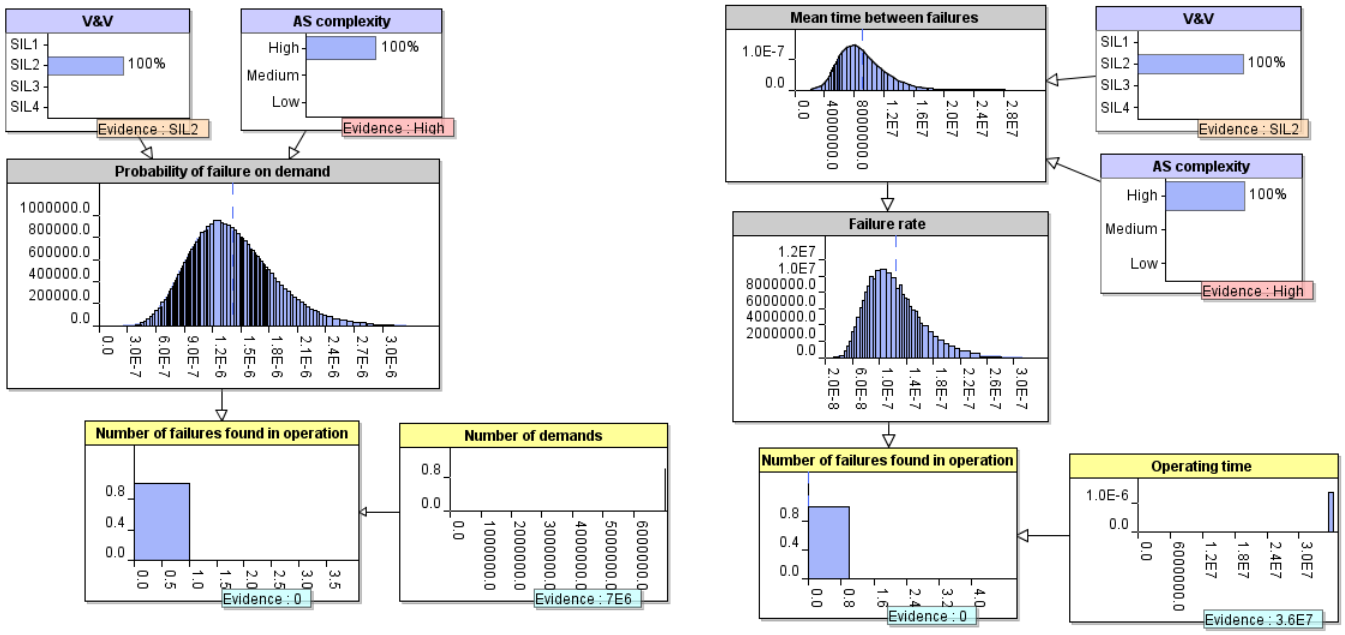
The second example considers a limitation application function and the following evidence:

- The limitation system is claimed to be SIL2 and runs in a low-demand operation mode.
- Medium complexity is assumed for this limitation application.
- In this example, the PFD is considered as evidence, e.g. 1E-5, as a requirement to be fulfilled by the limitation application function.

In this case the BN model estimates 100110 failure-free demands, which have to be observed to achieve the required PFD of 1E-5 for the limitation application function (see Figure 5 (a)). If one failure is observed in the future (postulated case for illustration purposes), then 200070 demands on the limitation applications have to be observed in order to meet the PFD of 1E-5 (see Figure 5 (b).

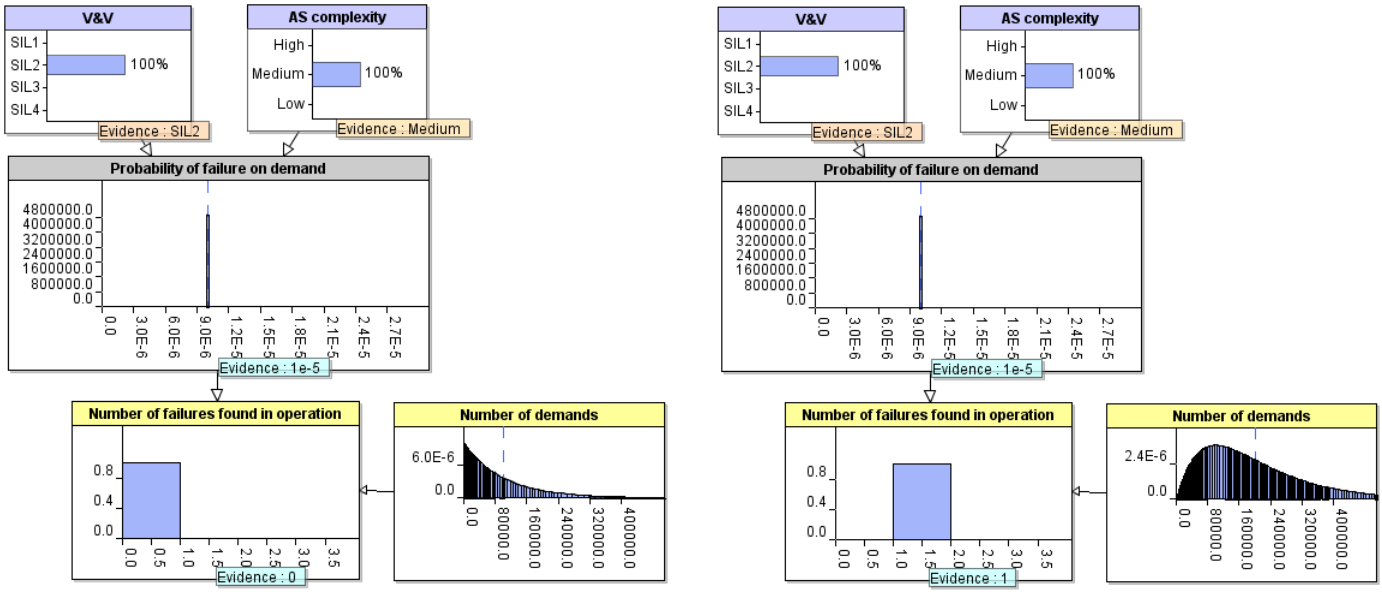
The third illustration example involves a protection application function, for which the following evidence is considered:

- The protection system is claimed to be SIL3.
- The application runs in a low-demand operation mode and has low complexity.
- The accumulated operation time of protection applications is 1.1E+8 hours with no evidence of demand-independent failures. The operating time of TXS protection, limitation and control applications are considered as evidence (see Section III), which results in a total operating time of 1.7E+8 hours.



(a) (b)

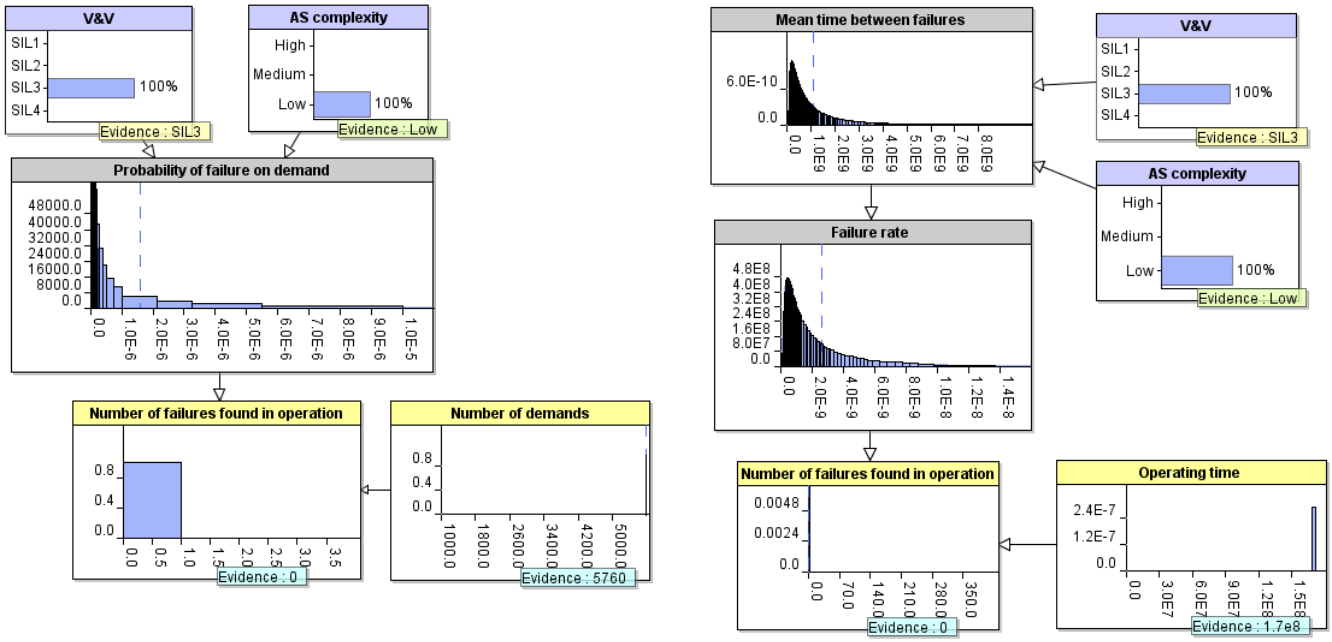
Fig. 4. BN models for reliability prediction of a control application function
(a) Prediction of PFD – (b) Prediction of failure rate [1/h].



(a) (b)

Fig. 5. BN models for reliability prediction of a limitation application function
(a) Demands estimation for a PFD of 1E-5 (no failures observed) – (b) Demands estimation for a pdf of 1E-5 (one postulated failure observed).

- The demand frequency of the protection application is assumed to be 0.25/year (calculated as the sum of all PSA initiating event frequencies), which results in approx. 3360 cumulative demands with no evidence of demand-dependent failures. The demands on the limitation applications are also considered as evidence (see Section III), resulting in 5750 total demands accumulated by TXS limitation and protection applications.
- The BN model estimates a failure probability on demand (expected value, see blue dotted line in PFD node of Figure 6 (a)) of 1.6E-6 for the protection application function in all divisions (considering the prior distribution for low-demand operation mode).



(a) (b)

Fig. 6. BN models for reliability prediction of a protection application function
(a) Prediction of PFD – (b) Prediction of failure rate [1/h].

The failure rate of the protection application function during normal plant operation is approx. 2.6E-9/h (see Figure 6 (b)), resulting in a probability of approx. 6.2E-8 (mission time = 24 h). The total failure probability of the protection application function in all divisions is approx. 1.7E-6 (sum of demand-dependent and independent failures).

VII. CONCLUSIONS

This paper presented a quantification method to estimate failure probabilities/rates for the simultaneous failure of identical application functions in all divisions of a digital I&C system in which the function is implemented. The method combines the use of Bayesian networks with evidence from the digital platform for safety I&C TELEPERM® XS. The application software reliability results can be used as input for probabilistic safety analyses. According to the analysis, applications operating in a low-demand mode (such as protection or limitation applications) have a probability of demand-dependent failures which is higher than the probability of failures occurring during the normal operation of the plant (maintenance excluded). However, for control applications operating in a high-demand mode, the probabilities estimated for demand-dependent and independent failures are of similar orders of magnitude.

The quantitative method is general and can be applied to other digital platforms for safety I&C if operating experience is available. Future work includes the consideration of simultaneous failures of different application functions which may share similar function blocks or functional requirement specifications.

REFERENCES

1. 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)
2. IAEA - Protecting against common cause failures in digital I&C systems of nuclear power plants - Nuclear Energy Series No. NP-T-1.5 (2009)
3. IEC 61508, International Standards – Edition 2.0, 2010-04
4. Agena Ltd. AgenaRisk (2007), http://www.agenarisk.com