Modeling of Digital I&C and Software Common Cause Failures: Lessons Learned from PSAs of TELEPERM® XS-Based Protection System Applications

Robert S Enzinna[a], Mariana Jockenhoevel-Barttfeld[b], Yousef Abusharkh[b], and Herve Bruneliere[c]

[a]AREVA Inc. Lynchburg, VA, USA
[b]AREVA GmbH, Erlangen, Germany
[c]AREVA SAS, Paris, France

Abstract: The authors have created probabilistic safety assessment (PSA) models of TELEPERM® XS (TXS)-based digital protection systems for a variety of nuclear power plant applications in the USA and around the world. This includes PSA models for digital protection system upgrades, and protection systems for new reactor builds. The PSA models have involved detailed digital instrumentation and control (I&C) fault tree models that have been fully integrated with the full plant PSA model. This paper discusses lessons learned, insights, and modeling recommendations gleaned from this experience.

The paper discusses recommended level of modeling detail, development of failure rate and fault coverage data, treatment of fault tolerant design features and common cause failure (CCF) defenses, fault tree modularization/simplification, and other topics of interest. Practical suggestions for PSA modeling are made based on experience gained from actual digital I&C PSA models built for several internal and external customers.

Modeling of CCF for the TXS hardware modules and for the software is highlighted, especially focusing on the quantification of software common cause failures (SWCCF). The authors describe the methodology used for quantification of SWCCF in the PSA studies, the definition of realistic software CCF modes, and estimation of failure probabilities.

Keywords: PSA, PRA, Digital I&C, Software Common Cause Failure.

1. INTRODUCTION

The guidance provided herein draws upon lessons learned from previous probabilistic safety analysis (PSA) work for digital I&C designs, including the EPR™ advanced nuclear power plant projects in Europe and the U.S., and digital reactor protection system (RPS) and engineered safety features actuation system (ESFAS) upgrades such as for the Oconee nuclear plant. The Oconee RPS/ESFAS upgrade was the first full-scale digital RPS/ESFAS replacement in the US nuclear industry, and the reliability analysis performed for that project was a first-of-a-kind (FOAK) for a U.S. PSA. The purpose of that analysis was to satisfy customer and regulatory requirements for reliability analysis of the safety related I&C system.

The reliability model was created as a stand-alone fault tree model but with the intent that it could be integrated into the customer's plant PSA at a later time. The model was subsequently integrated into the Oconee PSA by the customer after the system was installed.

The lessons learned from the Oconee digital I&C model were then used and developed further in the U.S. EPR™ PSA. For the U.S. as well as for the European EPR™ projects, the digital I&C fault tree models were fully integrated with the overall PSA development from the start. Since previous PSA's for the existing nuclear plant fleet usually contain neither fully digital control rooms, nor fully digital safety-related I&C systems, the EPR™ projects were also unique with respect to the level of regulatory scrutiny and review that the PSA models have received.

Parallel efforts by AREVA teams in the U.S, France and Germany have built digital I&C models for TELEPERM® XS (TXS) customers around the world. These PSA models have reflected the varying needs and dictates of customers and regulatory authorities in several different countries. The lessons learned that are summarized in this paper encompass that broad and diverse experience.

Much of the digital I&C experience at AREVA revolves around the TELEPERM® XS digital I&C, which is the system platform developed at AREVA GmbH to implement safety I&C systems with highest safety responsibility. The first TXS systems were put into operation more than 20 years ago and have an excellent reliably record. TXS I&C systems have been installed in over 60 units at 28 plant sites located in 11 countries and utilizing 10 different reactor designs. The TXS components have clocked billions of hours of operating experience without a common cause failure (CCF) of either software or hardware. Indeed even random failures are extremely rare. This track record is largely due to the efforts of our colleagues at AREVA GmbH who designed the TXS platform from the start to be highly reliable and immune from software common cause failure. The designers of the system studied the failure modes and failure causes of conventional computer-based systems, and then built the TXS platform to exclude those failure modes. Features were built into the platform to reduce latent software defects, eliminate failure triggers, eliminate failure propagation, and minimize failure consequence. One of the challenges of the PSA modeling effort has been to reflect those CCF defenses in a fair and realistic way, while creating a methodology that is sufficiently robust yet practical in its application.

The following sections discuss the nuances and unique issues associated with PSA modeling of modern digital I&C systems, which have surfaced during the collective experience of the authors' work.

2. LESSONS LEARNED FROM TELEPERM® XS PSA STUDIES

2.1. Lesson 1: Operating Experience shows that SWCCF is Rare in a Well-Designed System

Over the last 20 years, TELEPERM® XS systems have been installed in over 60 nuclear units. These systems contain two versions of the main TXS processing module. These TXS processing modules have clocked 250 million hours of operation without experiencing a SWCCF. Indeed, even processor module hardware failures during this accumulated field experience are very rare (fewer than 20). Some of the reasons for this outstanding reliability will be explained in the paper.

There is a tendency, especially by regulators, to be conservative or “bounding” and ascribe SWCCF probabilities to hypothetical failure modes such as “failure of all computerized I&C.” However, taking diversity requirements into account that are usually included between the different reactor protection subsystems, no relevant/realistic CCF mode of the software can be identified for TXS that would cause the complete failure of the system (e.g., both subsystems).

It is important that the failure modes and effects to be included in the PSA model are credible and are assigned realistic probabilities.

It is cautioned that if the assumed CCF is overly conservative, that it may disguise more meaningful insights. We are not suggesting that SWCCF be omitted from the PSA model, or be overly optimistic. However, if the assumed SWCCF dependency is unrealistic, then its inclusion may mask the importance of other failures modes and the value of corresponding design countermeasures. Therefore, the primary emphasis should be to also include software (SW) failure modes and effects in the PSA model that are realistic relative to the design features of the system.

2.2. Lesson 2: Understand why the CCF Defenses in the Platform Design are Important

As a vendor of safety-related computer systems for nuclear power plants (NPPs), AREVA has studied the failure modes of SW in standard computer systems. This research has resulted in features and defenses in the TXS design to rule out many common SW failure modes and reduce the frequency and consequence of others [1].

An example is the so-called “data-storm” event. This common failure mode, which afflicts standard computer systems, occurs when “special loading” taxes the operating system (OS) capacity. This is a failure mechanism where the communication bus is bogged down by excessive network traffic. It is therefore important for the reliability analyst to understand whether the system uses networks with variable loading (and examine associated loading analysis), or whether the system is the type that uses cyclic processing and invariable bus loading, like TXS. Strictly cyclic operation and constant loading of communication and processing buses involved in TXS prevent this failure mode and ensure that an actual system demand puts no more stress on the OS than any other cycle.

Operating experience in standard digital systems also indicates that interference between application program data (e.g., due to dynamic memory allocation) and faults in releasing system resources (e.g., time dependencies due to internal system clock) are leading causes of failure. In TXS-based systems these failure modes are eliminated by static memory allocation and asynchronous operation. As a general rule, interference from the application SW on the OS and hardware resources is forbidden, and consequences such as process-driven interrupts are not allowed. These features alleviate leading failure causes (such as OS lockup due to memory conflict) that plague standard computer systems. Therefore it is important for the reliability analyst’s understanding to know whether the system in question uses dynamic or static memory allocation.

OS features such as invariant cyclic processing, and invariance of process and communication bus load, are designed to reduce failures due to external influences and ensure that the stress during a demand is the same as during a normal non-demand cycle. These features remove dynamic interaction failure mechanisms from the design. A primary reason for the use of deterministic program execution and cyclic operation in the OS platform is to disconnect the OS from the signal trajectories and establish a pattern of predictable system behavior. Deterministic program execution limits the opportunity for failure due to untested software paths and data sets because there is only one path through the software instructions, which does not change in response to input state changes or plant initiating events.

The platform and OS design have an important role in limiting SW failure triggers and failure consequences. Additional detail on failure modes and defenses is available in the referenced document [1], as well as in industry consensus documents such as IEC-62340 [2].

2.3. Lesson 3: SWCCF Recommendations for Application SW

From a PSA perspective, the authors favor a SWCCF quantification methodology that is realistic, and practical to apply. From the design perspective, the methodology must recognize the value of the defensive features that are built into the platform design and into the system architecture.

In a TXS-based system, defense against SWCCF involves four constituent parts:

- A software lifecycle (SLC) process that reduces latent errors,
- A platform design that reduces failure triggers,
- Platform features that eliminate failure propagation (minimize failure consequence), and
- Functional diversity.

The SWCCF methods that are described below achieve the goal of realistically reflecting the design features that influence SWCCF defense, without requiring excessive PSA resources. Hence the PSA analyst's attention is focused on the design features that most influence safety, and this helps to inform his/her interactions with the design team.

A probability estimate can be obtained via expert elicitation, given that the experts used have a good understanding of both the features of the SLC development process, and of the digital I&C platform design and its OS defensive measures. The analyst should understand the degree of customization that is allowed in the application software. Features such as the exclusive use of function block libraries (reusable software functional blocks that are simple, fully tested, verified, and rigorously controlled), automated development tools, and automatic code generation help reduce SW errors. It is also important for the PSA analyst get an appreciation for the functional specification process. (Is it a formal process? Is it "user friendly" for both the process and I&C engineers? How is it checked, verified, tested?) The PSA analyst should be familiar with the V&V methodology, the associated tools (e.g., simulation, inverse checking), and how the process conforms to applicable standards of good practice.

For example, TELEPERM® XS uses a functional specification process based upon functional diagrams. These function diagrams are difficult to distinguish from those used for traditional analog designs, and are deliberately designed to be familiar to both the process design engineers and the I&C engineers. The "components" on the functional diagram (bistable, summer, etc.) are software function blocks that mimic their analog counterparts. The executable code is generated automatically from the diagrams, and is checked via simulation, and other tools. TXS has no custom SW development in the traditional sense.

The platform and OS design also have an important role in limiting triggers of application software failure. With deterministic program execution there is just one path through the program instructions, and all of the application code is executed on each cycle. The objective of this type of design is to limit the opportunity for failure due to untested software paths and data sets.

We are wary of SWCCF estimation methods that ignore the platform design. Much of the published research tends to over-emphasize the software development aspect and ignore the profound effect that the platform design characteristics have on reducing SW failure and CCF. Research that is biased towards customized made-from-scratch software development is less valuable for platforms like TXS which restrict use of customized SW and employ a simple predictable operating system. We are suspicious of methodologies that attribute all of the failure probability to the likelihood of a SW defect because they ignore the second aspect of SW failure probability, which is the likelihood of a failure trigger. The defenses built into the I&C platform to reduce triggers have a marked effect on SW failure probability. CCF also requires propagation to redundant trains or diverse functions. And so the defenses built into the platform to reduce failure propagation and consequence are very important as well.

Another importance aspect of the platform design that is often overlooked in the research is the configuration control. When evaluating the software quality and V&V process, the entire SW lifecycle is important, not just the initial development. A good defense involves the whole life-cycle, because approximately 36% of the failures in generic digital I&C operating experience are introduced during maintenance and update activities occurring after product installation [3].

An expert elicitation process can compare the features of the system and process in question with the features typically associated with other high-reliability applications. IEC 61508 [4] and IEC-62340 [2] are suggested as a guideline for this engineering judgment. These are an industry good practices documents and IEC 61508 provides consensus estimates of reliability targets that can be achieved for differing safety integrity levels (SIL). Rigorous guidelines for compliance with each SIL are provided for both hardware and systemic (SW development) aspects of the design process.

Safety Integrity Level Targets From IEC-61508

SIL	**Low demand mode** (Probability of failure on demand)
4	≥ 10–5 to < 10–4
3	≥ 10–4 to < 10–3
2	≥ 10–3 to < 10–2
1	≥ 10–2 to < 10–1

In PSA studies, we have modified the failure probability within the target range, much like a performance shaping factor (PSF) would be used in human reliability analysis (HRA). The value of the PSF used is based on functional complexity.

In early studies we used a simple complexity adjustment that was based on application SW function. A simple one-parameter trip signal would be assigned a failure probability at the low end of the range, for example 1E-5/demand for a simple trip on high pressure in a SIL-4 system. A more complex trip that used two parameters would be assigned the base value for each parameter in the function, for example 2E-5/demand for a trip that combined high pressure and temperature. Redundant channels with the same software are conservatively treated with complete dependence.

Since the safety system designs being analyzed have extensive features to protect against propagation of failure between diverse functions, it was reasonable to assign application SWCCF probabilities to individual software functions, or groups of software functions (characterized by having the same functional requirement specifications, such as plant parameter inputs, algorithms and/or data trajectories), that are common to multiple processors. In the example mentioned above, if the pressure sensor input for the two functions was the same, then this introduced a SWCCF dependency as well (CCF trigger: same signal). Hence functions that were truly diverse (no common input parameters) got more credit in the PSA than functions that shared input.

In more recent work we are exploring the use of a more sophisticated complexity metric to shape the base SWCCF probability. This metric is under development, and analyzes the actual source code that is created by the TXS automatic code generator.

In our example PSA application, we also assigned a beta factor between any diverse SW functions that may appear in the same minimum cut set of the PSA. We used the beta factors when it was necessary to judge the coupling between similar, but not identical, SW functions.

Design standards such as IEC 62340 [2] provide strong endorsement of functional diversity as a defense against SWCCF. When coupled with the other defenses (reducing defects, reducing triggers, preventing propagation), function diversity provides an effective defense against specification errors, and reduces the probability of a common failure trigger by employing different signal trajectories. The functional diversity may be achieved within the digital system by using different input parameters, algorithms, and data trajectories, as well as by using diversity inherent in the plant process systems. The OS defensive features, discussed above, provide assurance that software failures do not propagate to diverse functions. The functional diversity is even more effective if it is implemented on independent computers.

Since the functional diversity may involve using some common SW elements, the PSA analyst may desire to model a dependency between the two digital functions using the familiar beta-factor approach. Quantification of these beta factors using hard data or analytical methods is difficult. Therefore, assignment of beta factor values will require the use of expert judgment, based on a qualitative assessment of the similarities between the functions. The recommended beta factor is between 0.1 and 0.001 depending on the similarity of the software functions (input parameter, algorithm, and data trajectory). Some suggestions are available in the referenced papers [5, 6].

2.4. Lesson 4: SWCCF Recommendations for Operating System SW - It is Helpful if the Platform has a Proven Track Record

If the OS used in the NPP I&C platform is supported by a mature operating history, then this may allow statistical inference methods to be used to assess this part of the software failure probability. For example, the AREVA TXS platform has performed for more than twenty years in dozens of plants worldwide. The computer processor modules have 250 million cumulative operating hours of service, without an OS failure. The authors have used this experience to generate upper bound failure probabilities using 95%-chi-squared statistics. This treatment is possible because the OS has features to ensure its independence from the application SW and from the plant process, and therefore the OS failure rate is not influenced by variations in the application SW or by interference from transient loading.

The fundamental contributor to OS reliability is the features that limit the propagation and the severity of application SW failures. Specific features of the OS platform such as strictly cyclic operation, constant bus loading, static memory allocation, and prohibition of process-driven interrupts are used to ensure predictable OS performance and behavior that is free of interference from the application program. These features are designed to ensure that application SW failures caused by special loading, unanticipated input signal trajectories, or other application program design errors will not affect the OS, and hence propagate a failure to other functions.

In safety-related NPP applications, there are also requirements for independence between redundant channels. Simultaneous OS failure in independent safety-related computers is rare, and not observed in the field data. Therefore, bounding statistical treatment (given sufficient failure-free operating experience) and/or expert judgment will be necessary to quantify the probability of CCF of the OS in redundant channels. However, even a very small probability assumed for system-wide CCF failure of the OS may dominate the PSA results. Therefore, it is cautioned not to be overly conservative with these estimates, as that may mask the sensitivity of the PSA to more realistic failure modes, and the design features (hardware and software) that influence them.

2.5. Lesson 5: There is no Substitute for Vendor Failure Rate Data for I&C Modules

AREVA has accumulated an extensive failure rate data base for the various modules that are used in the TXS platform. When modules are new, theoretical estimates are generated using the part-stress analysis methodology of the Siemens SN29500 standard. Once modules are in service, field data is collected and updated on a quarterly basis. Since TELEPERM® XS installations have been operating for 20 years, the operating experience is extensive. The most common TXS modules, such as processing modules and input/output (I/O) modules, have accumulated hundreds of millions of operating hours. The field data for these modules is processed to produce best estimate as well as 95%-chi-squared upper bounds. As the field data accumulates, the theoretical estimates from the part stress analysis invariably prove to be conservative, even compared to the upper bound field data. Since the design and reliability of digital I&C modules is very much vendor specific, the authors cannot recommend a generic data base that is a good substitute.

2.6. Lesson 6: Adjust Level of Modeling Detail to Availability of Data and Supporting Analysis

For digital I&C, the failure data (field data or theoretical estimates) is usually generated at the module or board level of detail. This provides a convenient level of detail for the PSA model.

However, the I&C design team will usually produce other useful analysis that is at a much finer level of detail. This may include failure modes and effects analysis (FMEA) at the piece part level, fault coverage analysis aligned to the failure modes of specific circuits, and failure mode taxonomy for specific types of triggering events.

This is contrasted with the PSA model which is typically aligned with functional failure of the associated process system. The I&C failure modes are reflected at the functional level of the actuated component or system, regardless of how the digital system itself may fail. However, understanding of the failure mode taxonomy is important for other reasons, namely allocating parameters for failure likelihood and detectability (fault tolerance), identifying common dependencies between functions, and for guiding the PSA analyst in identification of which components (hardware or software) can contribute to loss of safety function.

The PSA model should also be a tool that drives the design to improve. IEC standard 62340 [2], provides useful insights on the leading causes of latent defects (e.g., specification errors), and failure triggers (e.g., environmental stress, input signal trajectory). The standard provides recommendations for reducing latent defects, reducing failure triggers, and for reducing consequences to other channels and functions. Paramount in these recommendations is the use of functional diversity in the design. This includes functional diversity within the digital system design, as well as potential diversity that exists in the plant process systems. Functional diversity provides double protection because it safeguards against functional specification errors as well as triggers in the data trajectory. The focus of the standard suggests that an effective level of detail for the SW failure contribution in the PSA model is one that recognizes and encourages functional diversity in the design.

Diverse functions are helpful to some degree whether they reside on the same processors, on separate processors, or on an entirely different system. In any case, the PSA should make an assessment of the effectiveness of the OS design features that are supposed to prevent software failures from propagating to other functions and assign a reasonable contribution for OS CCF (appropriate to the degree of separation) to capture the probability that this objective is not achieved. If the assigned CCF probability is too conservative, then it may have the undesired effect of discouraging functional diversity.

2.7. Lesson 7: Fault Coverage Data is as Important as the Failure Rate Data

Each component or module has a parameter called “fault coverage.” Fault coverage is an estimate of the percentage of the failure rate for each module that represents self-monitored (SM) versus non-self-monitored (NSM) failure modes. Failure modes that are self-monitored, or “covered,” are those faults that can be detected and compensated for by the components downstream. To the PSA analyst, the coverage represents an estimate of the effectiveness of the fault-tolerant features and fault-propagation barriers in the integrated hardware/software design.

Fault coverage has an important role in the PSA model because it drives which mathematical unavailability model (repair-time model, test-interval model, or both) is used for each component. It determines if the reliability is modeled with a short or long mean-time-to-repair (MTTR). In a digital system, known failures can typically be repaired quickly via replacement of a rack-mounted module.

Undetected failures on the other hand may stay in the system for a relatively long time, for example until a scheduled surveillance test (periodicity is generally from few months to 2 years).

Because of the fault tolerant design, the system may compensate instantly for a “covered” failure. For example, in a protection system application of TXS, a certain module may perform a 2-out-of-4 coincidence logic. If the module senses that an input is faulted, then it is programmed to change the coincidence. As bad inputs are recognized, the coincidence can be programmed to transition from 2-out-of-4 to 2-out-of-3, then to 1-out-of-2 or even 1-out-of-1 if necessary (degradation). It can also be programmed to go to a pre-defined safe state, if desired.

Consequently, the postulated failures involving NSM failure modes will almost always dominate over the SM failure modes. This is true even if the NSM percentage of the failure rate is very small relative to the SM percentage. Therefore, the results are sensitive to the fault coverage parameter. Because of the importance of fault coverage, detailed FMEA of the TXS modules is performed to determine fault coverage.

2.8. Lesson 8: Failure Mode Taxonomy

The definition of realistic failure modes for SWCCF is an important input to the PSA study. In highly-redundant NPP safety systems, the specific values assigned to the SW reliability are less important to the overall system PSA model than the choice of which SW failure modes and effects (e.g., fails a single function, or fails multiple functions) to include in the model. In a multi-channel safety system, the PSA results are easily dominated by any SWCCF that is assumed to affect the function of redundant trains or diverse functions. It is therefore critically important that the SWCCFs that are included in the model are realistic, credible and representative.

2.9. Lesson 9: Use Fault Tree Modularization to Simplify the Analysis

Fault tree modularization is an effective means for simplifying the analysis. This may not be easy to accomplish given the integrated nature of digital I&C design applications. Modularization tends to increase model conservatism, but is a trade off with simplification of fault tree displays and presentation of minimum cut set results.

The referenced paper [7] has described the methodology developed by AREVA for the modeling of probabilities of failure per demand of I&C functions in the PSAs of Nuclear Power Plants for which the I&C detailed design (allocation of functions in the units) is clearly defined. The principle of the method (except the need for modeling software failures) remains applicable for analog platforms.

Sensors required for the elaboration of each I&C signal are individually modeled as well as their related conditioning components. Common cause failures are applied for sensors as well as for conditioning modules on a case by case basis. The elementary components used for the modeling of I&C processing parts are the single processing units. A processing unit typically consists of a sub rack with one or more processing modules, I/O modules and communication modules. A detailed model of the unit is developed separately and inserted into the fault tree as a modular basic event. CCF for the processing parts (or automation parts) are modeled at the functional level as well as the platform level.

This methodology has been successfully implemented in PSAs for new builds and existing plants.

The following advantages have been identified:

- The links between I&C and support systems are easy to implement in the model,
- The hazards analyses integrates I&C,
- A detailed modeling of units allows the detection of asymmetries or imbalances in the I&C design (inadequate allocation of signals in the processing units),
- This modeling is easily understandable with respect to the PSA cut sets analysis,
- The I&C architecture is accurately represented in the PSA.

3. CONCLUSION

Final Lesson: Always Remember that the Objective is to Improve the Design

The most important feature of a PSA methodology for digital I&C is that it represents credible failure modes and realistic (but not necessarily precise) likelihood estimates. SWCCF probabilities that are too conservative, or which represent hypothetical failure modes that are not credible, will drive the design function in directions that may not be productive. SWCCF estimates that are subjective, but well founded will better serve the design.

Since SW failures require both a latent defect and a trigger, the desirable PSA methodology must account not only for the characteristics of the software life cycle development process, but also for the characteristics of the platform and OS design that work to reduce the failure triggers. Also, since the NPP PSA study is primarily sensitive to CCF as opposed to failure of individual functions or channels, the desirable SW reliability method also addresses the likelihood that the SW failure propagates to redundant channels and/or diverse functions. Therefore, it is desirable that the methodology also consider the platform and OS design features that are intended to limit failure consequence. The best quantitative methodology is one that accounts for the features of both the design and of the SLC development process. This is the most important characteristic of a useful methodology, even if it results in a methodology that involves a high degree of engineering judgment and qualitative insight.

To represent the design fairly, it is important for the PSA analyst to understand the behavior of the system being modeled. This is accomplished through a close working relationship with the design activity, including the hardware and SW design as well as the SLC process. A FMEA from the design activity is especially helpful. Other useful material from design engineering may include functional diagrams, architecture diagrams, function block library definitions, fault coverage analysis, operating history, description of platform CCF defenses, and other information. It is also important for the PSA analyst to investigate the quality of the SLC process, to get an appreciation of where errors may be introduced (e.g., functional specification, SW maintenance and update), how they are avoided (e.g., formal specification methodology, reusable SW/function blocks, automatic code generation), where errors may be caught (e.g., V&V, testing, simulation), and how the process conforms to applicable standards of good practice. It is through these activities that the PSA engineer gets an appreciation for the effectiveness of the design and process defenses against defects, failure triggers, and failure propagation.

Acknowledgements

The authors acknowledge Dr. Arnold Graf, Dr. Christian Hessler, and others at AREVA GmbH who have devoted their careers to the study of digital I&C failure modes, and whose research has resulted in the

defenses of the TELEPERM® XS design that eliminate many software failure modes, and reduce the likelihood and consequence of postulated common cause failures.

References

[1] EMF-2110(NP)(A) Revision 1, "*TELEPERM XS: A Digital Reactor Protection System,*" May 2000, AREVA NP/Siemens Power Corporation. (ADAMS Accession Number ML003732662)

[2] IEC-62340, "*Nuclear Power Plants – Instrumentation and Control Systems Important to Safety – Requirements to Cope with Common Cause Failure,"* International Electrotechnical Commission.

[3] "*IEC 61508 Overview Report*," version 2.0, exida, January 2006.

[4] IEC 61508, "*Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Devices,"* International Electrotechnical Commission.

[5] Bob Enzinna, Li Shi, & Steve Yang (AREVA NP Inc.), "*Software Common-Cause Failure Probability Assessment*," 6th ANS International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technology, Knoxville Tennessee, April 2009.

[6] "*Modeling of Digital I&C in Nuclear Power Plant Probabilistic Risk Assessments*," white paper by Nuclear Energy Institute Digital I&C Working Group, July 2007.

[7] Hervé Brunelière, Caroline Leroy, Laurent Michaud (AREVA NP SAS), Nissia Sabri (AREVA NP Inc.), Peter Otto (AREVA NP GmbH), "*Finding the best approach for I&C modeling in the PSA in the different design phases*" 11th International Probabilistic Safety Assessment and Management Conference (PSAM11) & The Annual European Safety and Reliability Conference (ESREL12), Helsinki Finland, June 2012.