

NKS-330  
ISBN 978-87-7893-411-6

---

# Guidelines for reliability analysis of digital systems in PSA context - Final report

Stefan Authén<sup>1</sup>

Jan-Erik Holmberg<sup>1</sup>

Tero Tyrväinen<sup>2</sup>

Lisa Zamani<sup>1</sup>

<sup>1</sup>Risk Pilot AB, Sweden

<sup>2</sup>VTT Technical Research Centre of Finland

February 2015

## **Abstract**

The objective of the DIGREL project has been to provide guidelines to analyse and model digital systems in the context of probabilistic safety assessment (PSA). A failure modes taxonomy for digital I&C systems has been developed jointly with OECD/NEA Working Group on Risk Assessment. Reliability modelling has been studied by developing a fictive, simplified PSA model representing a four-redundant distributed protection system. The evaluation of the example PSA has demonstrated the developed taxonomy and verified that it is suitable for PSA purpose. The evaluation shows that the choice of the level of abstraction for the modelling of digital I&C is of high importance for the results. Module level is recommended. Both undetected and detected hardware as well as software failures contribute significantly to the PSA results, indifferently of the assumed fault tolerant design. Similar conclusion can be drawn from the test of using different CCF parameters for undetected and detected failures. Software faults have a non-negligible effect on the results due to their functional impact on all divisions. In order to develop a realistic fault tree model for a digital I&C protection system it is vital that the chosen fault tolerant design is fully understood and correctly described in the model. The treatment of faulty inputs and degraded voting logic sets the foundation of the fault tree analysis.

## **Key words**

Digital I&C system, probabilistic safety assessment, reliability, nuclear power plant safety

NKS-330  
ISBN 978-87-7893-411-6

Electronic report, February 2015  
NKS Secretariat  
P.O. Box 49  
DK - 4000 Roskilde, Denmark  
Phone +45 4677 4041  
[www.nks.org](http://www.nks.org)  
e-mail [nks@nks.org](mailto:nks@nks.org)

# **Guidelines for reliability analysis of digital systems in PSA context — Final report**

**Report from the NKS-R DIGREL activity (Contract: AFT/NKS-R(14)86/3)**

Stefan Authén<sup>1</sup>  
Jan-Erik Holmberg<sup>1</sup>  
Tero Tyrväinen<sup>2</sup>  
Lisa Zamani<sup>1</sup>

<sup>1</sup>Risk Pilot AB, Parmmätargatan 7, SE-11224 Stockholm, Sweden

<sup>2</sup>VTT Technical Research Centre of Finland Ltd., P.O. Box 1000, FI-02044 VTT, Finland

## Table of contents

	Page
Table of contents	2
Abbreviations	4
Summary	6
Acknowledgements	8
1. Introduction	9
2. Scope and objectives	10
3. Definitions	11
4. Safety I&C systems in nuclear power plant	14
5. State-of-the-art of reliability analysis of I&C systems in PSA context	18
5.1 Overview	18
5.2 Modelling digital I&C in PSA	18
5.2.1 Dynamic reliability modelling approaches	19
5.2.2 Software reliability modelling	20
5.3 Reliability data for digital I&C systems	21
5.3.1 Hardware reliability data	21
5.3.2 Software reliability data	22
5.4 Nordic PSA-studies	23
6. Failure modes taxonomy	26
6.1 WGRISK task group DIGREL	26
6.2 General approach for the development of the taxonomy	27
6.3 Requirements	27
6.4 Levels of abstraction	29
6.5 Failure model	31
6.6 Failure mode taxonomy at System and division levels	33
6.7 Failure mode taxonomy at I&C unit and module levels	33

6.7.1	I&C unit level failure modes and effects	34
6.7.2	Module level failure modes and effects	35
6.7.3	Basic component level failure modes and effects	38
6.8	Failure modes and effects analysis (FMEA)	38
7.	PSA modelling	40
7.1	Introduction	40
7.2	Taxonomy for PSA modelling	41
7.2.1	Hardware failure modes	41
7.2.2	Software failure modes	45
7.3	Additional modelling issues	47
7.3.1	Common cause failures	47
7.3.2	Human errors	48
7.4	PSA model structure	49
7.4.1	Introduction	49
7.4.2	RiskSpectrum modelling	49
7.4.3	FinPSA model structure	51
7.4.4	Comparison of RS and FinPSA results	52
7.5	Evaluation of the modelling aspects	52
8.	Conclusions	55
9.	References	58
Appendix A. Description of the example system		64
Appendix B. Evaluation of the model aspects		89

## Abbreviations

A/D	Analog/digital
ACP	AC power system
AIM	Analog input module
ALOCA	Large loss-of-coolant accident
AOM	Analog output module
APU	Acquisition and processing unit
APU-AS	APU application-specific software module
APU-FRS	APU functional requirements specification module
AS	Application software (module)
BBN	Bayesian belief network
BWR	Boiling water reactor
CCF	Common cause failure
CCI	Common cause initiator
CCMT	Cell-to-cell mapping (technique for stochastic simulation)
CCW	Component cooling water system
CD	Core damage
CDF	Core damage frequency
COM	Communication link module
COTS	Commercial off-the-shelf
CPU	Central processing unit
CSNC	Canadian Nuclear Safety Commission
CSNI	Committee on the Safety of Nuclear Installations (OECD/NEA)
CSRM	Context-based software risk model
DFLT	Default value
DF	Detected fault
DFM	Dynamic flowgraph methodology
DIM	Digital input module
DOM	Digital output module
DCS	Data communication software
DCU	Data communication unit
DLC	Data link configuration
ECC	Emergency core cooling system
EDF	Électricité de France
EF	Elementary function
EFW	Emergency feedwater system
ENEL	Ente Nazionale per l'Energia eLettrica, Italy
ESFAS	Engineered safety features actuation system
ET	Event tree
FMEA	Failure mode and effects analysis
FC	Fractional contribution
FPGA	Field-programmable gate array
FRS	Functional requirements specification
FT	Fault tree
FTD	Fault tolerant design
GRS	Gesellschaft für Anlagen- und Reaktorsicherheit, Germany
I&C	Instrumentation and control
I/O	Input/output
IAEA	International Atomic Energy Agency

ICDE	OECD/NEA International Common-cause Failure Data Exchange Project
IEC	International Electrotechnical Commission
IRSN	Institut de Radioprotection et de Sûreté Nucléaire, French Institute for Radiological Protection and Nuclear Safety
JNES	Japan Nuclear Energy Safety Organization
KAERI	Korea Atomic Energy Research Institute
KTH	Kungliga tekniska högskolan, Royal institute of technology in Stockholm
LMFW	Loss of main feedwater
LOCA	Loss-of-coolant accident
LOOP	Loss-of-offsite power
MFW	Main feedwater system
MU	Manual control unit (I&C unit for main control room operations)
NEA	OECD Nuclear Energy Agency
NKS	Nordic nuclear safety research
NPIC-HMIT	Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies conference
NPP	Nuclear power plant
NPSAG	Nordic PSA Group
NRC	U.S. Nuclear Regulatory Commission
NRG	Nuclear Research & consultancy Group, the Netherlands
NRI	Nuclear Research Institute Rez plc
OECD	Organisation for Economic Co-operation and Development
PSA	Probabilistic safety assessment
RDF	Risk decrease factor
RIF	Risk increase factor
RHR	Residual heat removal system
RPS	Reactor protection system
RT	Reactor trip
SAFIR	Finnish Research Programme on Nuclear Power Plant Safety
SCM	Signal conditioning module
SW	Software
SWS	Service water system
SyS	System software
VU	Voting unit
VU-AS	VU application-specific software module
VU-FRS	VU functional requirements specification module
V&V	Verification and validation
VEIKI	Institute for Electric Power Research, Hungary
VTT	Technical Research Centre of Finland
WGRISK	OECD/NEA CSNI Working Group on Risk Assessment

## Summary

Digital protection and control systems appear as upgrades in older plants, and are commonplace in new nuclear power plants. To assess the risk of nuclear power plant operation and to determine the risk impact of digital systems, there is a need to quantitatively assess the reliability of the digital systems in a justifiable manner. The objective of the DIGREL project has been to provide guidelines to analyse and model digital systems in the context of probabilistic safety assessment (PSA). The project has consisted of the following activity areas:

1. Develop a taxonomy of hardware and software failure modes of digital components for common use.
2. Develop guidelines for failure modes analysis and fault tree modelling of digital I&C.
3. Develop an approach for modelling and quantification of software.

The failure modes taxonomy has been developed by a task group of the OECD/NEA CSNI Working Group on Risk Assessment. The taxonomy is based on a hierarchical definition of five levels of abstraction: 1) system level, 2) division level, 3) I&C unit level, 4) I&C unit modules level, 5) basic components level. The main feature of the taxonomy is to describe the failure propagation using a failure model. The failure model and the taxonomy consist of the following elements: fault location, failure mode, uncovering situation, failure effect and the end effect. The purpose of the taxonomy is to support PSA, and therefore it focuses on high level functional aspects rather than low level structural aspects. This focus allows handling of the variability of failure modes and mechanisms of I&C components.

In order to develop guidelines for failure modes analysis and modelling, a fictive, simplified PSA model representing a four-redundant distributed protection system has been developed. The example model has been used to test the effect of different levels of modelling detail, common cause failure (CCF) modelling, fail-safe principle and voting logic. The evaluation of the example PSA demonstrated the developed taxonomy and verified that it is suitable for PSA purpose. The evaluation shows that the choice of the level of abstraction for the modelling of digital I&C is of high importance for the results. The most suitable level of abstraction is found to be the “module level” which concurs with the level of abstraction of the general PSA state of the art. Both undetected and detected hardware and software failures contribute significantly to the PSA results, indifferently of the assumed fault tolerant design. This stresses the importance of not excluding detected failures from the reliability model without thorough investigations. Similar conclusion can be drawn from the test of using different CCF parameters for undetected and detected failures.

Software faults have a non-negligible effect on the results due to their functional impact on all divisions — one or more safety functions can be lost. Therefore attention needs to be paid to the quantification of software faults and the assessment of the degree of diversity between the subsystems of the reactor protection system.

The received results are based on the specific design of the example plant and example I&C system and also the assumed failure data of the digital I&C and assumed CCF parameters. The results of this study should therefore not directly be generalised to other designs. Differences in conclusions may of course be found for different designs and failure data.



In order to develop a realistic fault tree model for a digital I&C protection system it is vital that the chosen fault tolerant design is fully understood and correctly described in the model. The treatment of faulty inputs and degraded voting logic sets the foundation of the fault tree analysis. In general, modelling of digital I&C significantly increases the effort of failure mode analysis, dependency analysis and fault tree modelling. The amount of resource involved in such a task should not be underestimated, neither should the task of quality assurance.

**Acknowledgements**

The work has been financed by NKS (Nordic nuclear safety research), SAFIR2014 (The Finnish Research Programme on Nuclear Power Plant Safety 2011–2014) and the members of the Nordic PSA Group: Forsmark, Oskarshamn Kraftgrupp, Ringhals AB and Swedish Radiation Safety Authority. NKS conveys its gratitude to all organizations and persons who by means of financial support or contributions in kind have made the work presented in this report possible.

**Disclaimer**

The views expressed in this document remain the responsibility of the author(s) and do not necessarily reflect those of NKS. In particular, neither NKS nor any other organization or body supporting NKS activities can be held responsible for the material presented in this report.

## 1. Introduction

Digital protection and control systems appear as upgrades in older plants, and are commonplace in new nuclear power plants. To assess the risk of nuclear power plant operation and to determine the risk impact of digital systems, there is a need to quantitatively assess the reliability of the digital systems in a justifiable manner. Due to many unique attributes of digital systems, a number of modelling and data collection challenges exist, and consensus has not yet been reached.

In 2007, the OECD/NEA CSNI directed the Working Group on Risk Assessment (WGRISK) to set up a task group to coordinate an activity in this field. One of the recommendations was to develop a taxonomy of failure modes of digital components for the purposes of probabilistic safety assessment (PSA) (OECD 2009). This resulted in a follow-up task group. An activity focused on development of a common taxonomy of failure modes was seen as an important step towards standardised digital I&C reliability assessment techniques for PSA. Needs from PSA has guided the work, meaning, e.g., that I&C system and its failures are studied from their functional significance point of view. The taxonomy (OECD 2014) will be the basis of future modelling and quantification efforts. It will also help define a structure for data collection and to review PSA studies.

The Nordic NKS-DIGREL project started with a pre-study where a preliminary comparison of Nordic experiences was performed, and a literature review on main international references was presented (Authén et al. 2010a). The study shows a wide range of approaches and solutions to the challenges given by digital I&C, and also indicates that no state-of-the-art currently exists. The study showed some areas where the different PSA:s agree and gave a basis for development of a common taxonomy for reliability analysis of digital I&C.

DIGREL has taken advantage from ongoing R&D activities, actual PSA applications as well as analyses of operating experience related to digital systems in the OECD/NEA member countries. The scope of the taxonomy includes both protection and control systems of a nuclear power plant, though primary focus is on protection systems. The taxonomy is divided into hardware and software related failure modes, for which purpose example taxonomies have been collected. A representative fictive digital protection system example has been developed to be used as a reference in the application and demonstration of the taxonomy.

This report presents the *final* results from the WGRISK and Nordic activities. Interim results have been presented in reports NKS-230 (Authén et al. 2010a), NKS-261 (Authén et al. 2012), NKS-277 (Authén and Holmberg. 2013), NKS-302 (Authén et al. 2014) and NKS-304 (Bäckström et al. 2014).

## **2. Scope and objectives**

The objective of the project is to provide guidelines to analyse and model digital systems in PSA context, using traditional reliability analysis methods (failure mode and effects analysis, fault tree analysis). Based on the pre-study questionnaire and discussions with the end users in Finland, Sweden and within the WGRISK community, the following focus areas have been identified for the activities:

1. Develop a taxonomy of hardware and software failure modes of digital components for common use (reported in OECD 2014).
2. Develop guidelines regarding level of abstraction in system analysis and screening of components, failure modes and dependencies.
3. Develop an approach for modelling and quantification of common cause failures (CCF) between components.
4. Develop an approach for modelling and quantification of software (reported in Bäckström et al. 2015).

The project covers the whole scope of I&C systems important to safety at nuclear power plants (e.g. protection systems and control systems), both hardware and software aspects as well as different life cycle phases of the systems and plant: design/development, testing, commissioning, operation and maintenance.

The focus of the work has been on protection systems, but many results are applicable to control systems as well. Regarding life cycle phases, design/development phase is considered from the software failure point view. Degree of verification and validation activities can be a factor in the reliability estimate. Role of testing has been discussed both for hardware and software. Commissioning has not been specifically addressed. Operation and maintenance are system modes, which must be accounted in the unavailability analysis of the system.

### 3. Definitions

**Activation condition:** An external event or phenomenon under which a fault becomes a failure. In this report, activation condition is understood broadly. It is not only a transient event triggering the failure but it can also be a long lasting event such environmental conditions.

**Context:** Boundary conditions for the actuation of I&C functions. In this report, context is determined by the plant condition, initiating event and activation conditions.

**Demand:** A plant state or an event that requires an action from I&C. Note: A state of the I&C system requiring an action of an active fault tolerant design feature is not considered a demand.

In this report, “demand” is used in the same meaning as in the reliability metric “probability per demand”, and is a specific uncovering situation, which is distinct from dedicated failure detection mechanisms such as online and offline monitoring. Online and offline monitoring are means to detect a failure before a demand.

**Detected failure:** A failure detected by (quasi-) continuous means, e.g. online detection mechanisms, or by plant behaviour through indications or alarms in the control room.

**Detection mechanism:** The means or methods by which a failure can be discovered by an operator under normal system operation or can be discovered by the maintenance crew by some diagnostic action (US DOD 1984). Note that this includes detection by the system (e.g. continuous detection).

There are two categories of detection mechanisms:

- Online detection mechanisms. Covers various continuous detection mechanisms.
- Offline detection mechanisms. E.g. periodic testing and also other kind of controls (e.g. maintenance).

**Fail safe:** Pertaining to a functional unit that automatically places itself in a safe operating mode in the event of a failure (ISO/IEC/IEEE 2010); “system or component” has been replaced with “functional unit”) Example: a traffic light that reverts to blinking red in all directions when normal operation fails. Note: In general fail safe functional units do not show fail safe behaviour under all possible conditions.

**Failure:** Termination of the ability of a product to perform a required function or its inability to perform within previously specified limits (ISO/IEC 2005). "Failure" is an event, as distinguished from "fault" which is a state.

**Failure effect:** Consequence of a failure mode in terms of the operation, function or status (IEC 2006, “of the system” removed).

**Failure mode:** The physical or functional manifestation of a failure (ISO/IEC/IEEE 2010).

**Failure mechanism:** Relation of a failure to its causes.

**Fatal failure:** The I&C unit or the hardware module stalls. It ceases functioning and does not provide any exterior sign of activity. Fatal failures may be subdivided into:

**Ordered fatal failure:** The outputs of the I&C unit or the hardware module are set to specified, supposedly safe values. The means to force these values are usually exclusively hardware. Equivalent to the definition “Halt/abnormal termination of function with clear message” (Chu et al. 2006).

**Haphazard fatal failure:** The outputs of the I&C unit or the hardware module are in unpredictable states. Equivalent to the definition “Halt/abnormal termination of function without clear message” (Chu et al. 2006).

**Fault:** Defect or abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function (IEC 2010a; “defect” added). Note: “Failure” is an event, as distinguished from “fault” which is a state.

**Fault tolerance:** The ability of a functional unit to continue normal operation despite the presence of failures of one or more of its subunits. Note: Despite the name this definition refers to failures, not faults of subunits. It is therefore distinct from the definition in (ISO/IEC/IEEE 2010). Possible means to achieve fault tolerance include redundancy, diversity, separation and fault detection, isolation and recovery.

**Initiating event:** An initiating event is an event that could lead directly to core damage (e.g. reactor vessel rupture) or that challenges normal operation and which requires successful mitigation using safety or non-safety systems to prevent core damage (IAEA 2010).

**Non-fatal failure:** The I&C unit or the hardware module fails but it continues to generate outputs. Non-fatal failures may be subdivided into:

**Failures with plausible behaviour:** I&C runs with wrong results that are not evident (Chu et al. 2006). An external observer cannot determine whether the I&C unit or the hardware module has failed or not. The unit is still in a state that is compliant to its specifications, or compliant to the context perceived by the observer.

**Failures with implausible behaviour:** I&C runs with evidently wrong results (Chu et al. 2006). An external observer can decide that the I&C unit or the hardware module has failed. The unit is clearly in a state that is not compliant to its specifications, or not compliant to the context perceived by the observer.

**Plant condition:** Given state of the plant, including the configuration of the systems, power level of the reactor and other relevant process parameters.

**Spurious actuation:** A failure where an actuation of an I&C function occurred without a demand. Spurious actuation can be caused by any failure between the process measurement sensors and the actuator, including erroneous operator command or failure of watchdogs.

**Systematic failure:** Failure related in a deterministic way to a certain cause, which can only be eliminated by a modification of the design or of the manufacturing process, operational procedures, documentation or other relevant factors (IEC 2010a).

**Uncovering situation:** The context where the failure becomes visible. The failure may become visible through dedicated “detection mechanisms” (see above), or failures may be discovered by a process event. The latter case includes failures revealed by spurious actuation or revealed (or triggered) by demand.

**Undetected failure:** A failure detected by offline detection mechanisms or by demand. Also called latent failure or hidden failure.

#### 4. Safety I&C systems in nuclear power plant

In the last decades a variety of different safety-related digital I&C systems have been developed and implemented in nuclear installations and facilities around the world. Digital I&C architectures are deployed in several reactors worldwide, not only in turbine automation but also in safety automation, such as Chooz B (France), Sizewell B (United Kingdom), Oskarshamn 1, Ringhals 1 and 2 (Sweden), Temelin-1 and -2 (Czech Republic), and Tianwan (China). Also new designs such as the EPR developed by AREVA, the APWR by Mitsubishi Heavy Industries, Ltd. and the ESBWR by General Electric Hitachi also demonstrate the recent state of digital I&C architectures in NPPs. Descriptions of modern nuclear I&C can be found, e.g., in (IAEA 2011, Kisner et al. 2007, Korsah et al. 2009).

The architecture, the equipment (hardware) and software of the digital safety-related I&C (I&C platform) are designed to meet all safety-related I&C requirements in nuclear power plants. The dissimilarities between different I&C platforms may be significant. Not only the physical design but also the functional design, e.g. fault tolerant features and voting logic, may differ. On the other hand, the stringent safety requirements on design, manufacturing and operating of the safety systems and safety-related systems in the nuclear power plants lead consequently to recognizable similarities of the architecture of several digital safety-related I&C systems and of their functions.

The entire I&C architecture of the nuclear power plant can usually be divided into following levels of the interactions between technological process and process control functions: 1) process interface, 2) system automation and 3) unit supervision and control.

In the continuation of this paper, we will focus on the system automation level. The system automation level of a nuclear power plant usually consists of the reactor protection system (RPS), the safety automation system, the process automation system, and actuation and control equipment. The protection systems and the control systems are the two major parts of the safety automation.

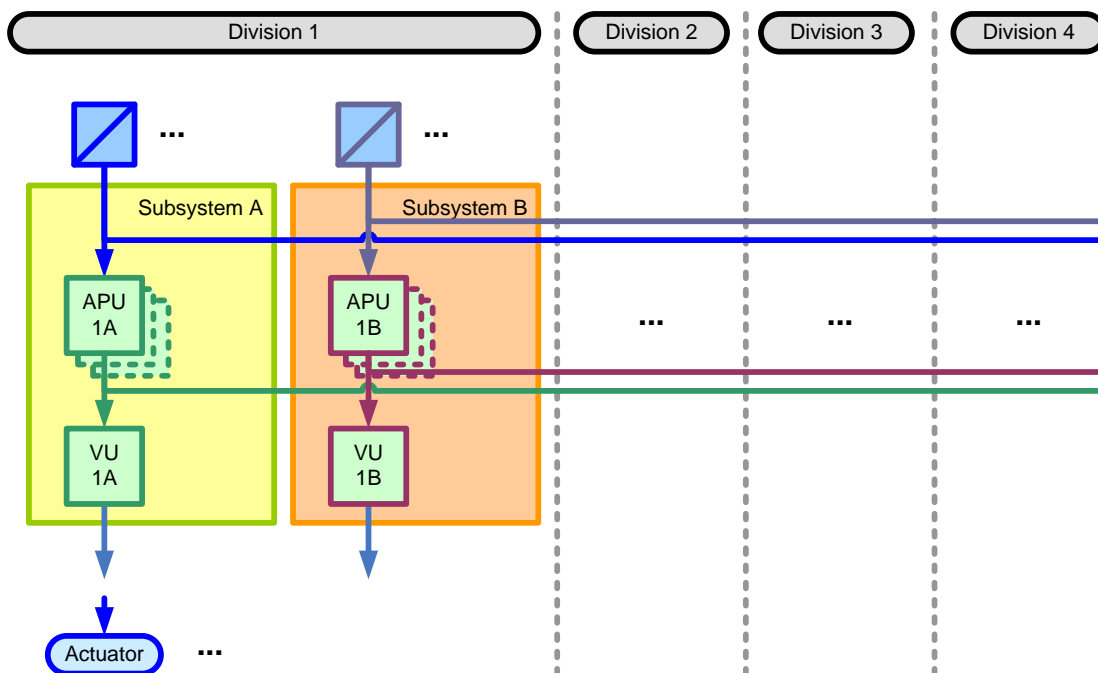
Protection systems, belonging to the highest safety class, which is Cat. A in IEC 61226 (IEC 2009), are responsible for the primary safety functions consisting of reactor trip system and the engineered safety features actuation system (ESFAS). Protection systems (Figure 1) are composed of redundant divisions (or channels) running in parallel microprocessors and they actuate functions on demand (e.g., when process parameter limits are exceeded).

Figure 1 represents a usual modern architecture solution where the reactor protection system is divided into two subsystems (denoted here by RPS-A and RPS-B). The two subsystems are responsible for different I&C functions in order to have diversity in safety functions.

Within a subsystem, the divisions may be of the same or different architectures but in general all perform the same I&C functions. Each division consists of multiple I&C units. As an example, the following I&C units can be found in typical architectures:



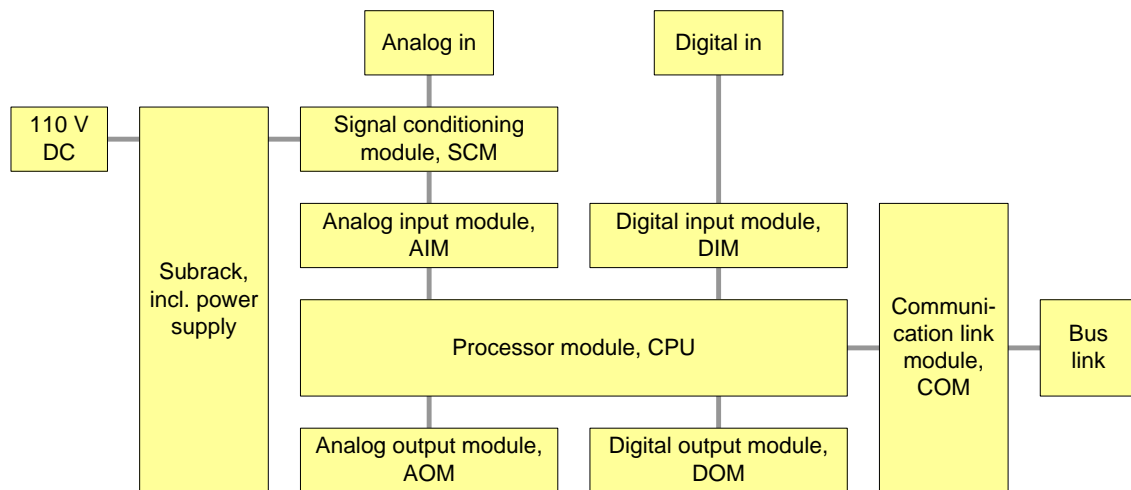
- Acquisition and processing units (or APUs): these units acquire process-related information from sensors, and perform calculations to determine the division outputs. Each subsystem (RPS-A resp. RPS-B) division is composed of one or more APUs implementing different functions. They may also process operator requests related to the functions they implement (such as the modification of a setpoint), but most requests can be performed only one division at a time, when that division is offline.
- Voting units (or VUs): these units receive the results determined by the APUs of their division and subsystem and for which voting is required. They also receive the decisions made by the APUs regarding operational bypasses. They exchange information between themselves across division boundaries (but not subsystem boundaries) in order to perform 2 out-of-4 voting in normal conditions where all four divisions are available. Automatic modification of the voting logic (e.g. from 2oo4 to 1oo2 or 2oo2) are applied in case of detected unavailability of one or more divisions.
- Data Communication Units (or DCUs): these units allow APUs and VUs to communicate with one another. The interface between a DCU and an APU (or a VU) is designed to limit failure propagation in both ways.



**Figure 1.** Example of a four-redundant digital I&C protection system architecture with two subsystems (RPS-A and RPS-B).

I&C units of subsystems A and B can have same or different platforms. In the example analysed later in the report, same platforms are assumed in order to consider possible CCF between subsystems.

Each I&C unit consists of multiple digital modules such as input module, processing module, communication module and output module (Figure 2). Each module comprises basic components such as an analog/digital converter, a multiplexer, a microprocessor and its associated components, a demultiplexer, and an A/D converter.



**Figure 2.** Typical hardware modules in an I&C unit.

Software of safety I&C is decomposed into various software modules which have different functional roles and have different development as well as V&V (verification and validation) histories. The APUs and VUs have typically following kind of modules:

- System software (SyS) includes the operating system and runtime environment (interaction between application and operating system). System software is plant independent.
- Elementary functions (EF). These modules provide readily useable standard (library) functions such as Boolean logic, mathematical functions or delays. They are the same for all units of the example system. However, an important difference with respect to the SyS is that a specific I&C unit will use only a specific subset of all available EFs. Elementary functions can be also called Library Functions or Function(al) Blocks. Elementary functions are plant independent.
- Application software (AS) modules in I&C units are the software modules which are executed by the operating system during an operating cycle of the processing module. These modules implement specific I&C functions in I&C units. Homologous I&C units (APUs resp. VUs) in redundant divisions have the same sets of AS modules. There are usually several AS modules associated with each I&C function. AS modules are plant-specific and are constructed using elementary function modules.

AS modules are generated from Functional requirements specification (FRS) modules, which are virtual software modules. There is typically one such module per I&C function required of an I&C unit, and they exist as function block diagrams, which specify the connections between elementary functions for each I&C function.

Data communication units have the following software modules:

- System software (SyS), which is the same as the SyS of the APUs and VUs.
- Data communication software (DCS) which implements the data communication protocol. It is part of the platform software, and is plant independent.

- Data link configuration (DLC) which specifies the nodes that can be part of a given network (subsystem), and the data messages that can be exchanged between the nodes of the network. DLC is plant-specific.

In addition, there are specific pieces of software present in other hardware modules than processor modules. These are called here as SW in COTS-modules (Commercial off-the-shelf). The implementation in software belongs to a commercial company, and the source code is not freely nor publicly available. It is restricted from use, such as modification or V&V, for the end user.

The differences between different I&C platforms and software may be significant, not only the physical design but also the functional, e.g. fault tolerant features and voting logic. On the other hand, due to the stringent design requirements for protection systems and common functional requirements for safety automation of light water reactors, there are important similarities between design solutions provided by different nuclear safety I&C vendors.

Control systems, e.g., turbine side automation, are versatile having both on demand and continuous functions and they do not necessarily have a redundant structure. Control systems belong to a lower safety class (B or C). A control system is structured in the same manner as protection systems, except that control systems do not often have redundant channels.

## **5. State-of-the-art of reliability analysis of I&C systems in PSA context**

### **5.1 Overview**

Digital I&C systems include unique features, such as complex dynamic interactions and the usage of software, that can be difficult to take into account with traditional PSA methods such as with the event tree-fault tree approach. Generally, dynamic methodologies provide a more accurate representation of probabilistic system evolution in time than the event tree/fault tree (ET/FT) approach. However, the dynamic models are on a trial stage.

A summary of experiences of modelling digital systems in CSNI member countries can be found in (OECD 2009). The report also presents a set of recommendations for method development, data collection and analysis, and international cooperation.

There is a general consensus that protection systems (RPS & ESFAS) shall be included in PSA, while control systems can be treated in a limited manner. The system architecture and the mode of operation of protection systems versus control systems are different, which creates a different basis for the reliability analysis and modelling.

### **5.2 Modelling digital I&C in PSA**

The applicability of traditional PSA methods (event tree-fault tree and Markov modelling) for digital systems has been surveyed in (Chu et al. 2008). Traditional methods are useful in the modelling but also indicates some limitations of the methods. However, the event tree-fault tree approach does not explicitly treat the timing of events in accident sequences and interactions with plant processes are implicitly and approximately considered. A set of desirable characteristics for a probabilistic model of a digital system has been identified. Additionally, a preliminary list of areas where additional research could enhance the state-of-the-art of modelling digital system is identified.

The incorporation of a model of a digital RPS into a PSA is discussed in (Authén et al. 2010b). The work demonstrated that modelling the digital RPS on an adequate level is challenging, and new approaches are required. An overview of the issues regarding the development of a static fault-tree-based risk model is presented in (Kang & Jang 2009). The complicated issues of digital system PSA are categorized into four groups based on their characteristics: hardware module, software, system, and safety function. The key issues related to modelling the PSA of nuclear safety digital I&C systems summarized in (Shi et al. 2010). The quantification techniques are presented to each of the issues.

The utilization of traditional methods to model a digital feedwater control system is discussed in (Chu et al. 2009). In the case study only the Markov method was used as the order of component failures was considered important. The study demonstrated that the proposed approach is feasible for analysing digital system. However, the integration with a PSA based on the ET/FT method may not be a trivial task.

Risk insights associated with digital upgrade is discussed in (Blanchard & Torok 2010). In the development of the digital I&C PSA model a pragmatic approach was taken, as the quantification of software reliability is a challenging problem. The research focused

on important engineering insights that can be reached by understanding the role of the digital system with respect to the plant systems and the plant itself.

For representing the effect of I&C at a PSA level, EDF has since the 90's been developing the compact model (Thuy & Deleuze 2009). The compact model of digital I&C is a functional representation that comprises the main outcomes of digital I&C experts' safety and dependability assessments that can be shared with PSA experts and incorporated in a PSA model. The purpose of the extended compact model is to form a connection between the probabilistic assessment at plant level and the deterministic assessment at I&C level, by a step by step approach. The idea is to "descend" from PSA to critical parameters identification, and to "ascend" from deterministic assessment of factors contributing to I&C safety to its representation in a PSA.

Failure modes and effects analysis (FMEA) is a well-known method for identifying failure modes of a system and their effects or consequences on the system. A few guidance documents for performing a FMEA are available, e.g. (IEEE 1987), but there are no specific guidelines on how to perform FMEA for digital systems. The absence of failure classification is a major issue in the representation of failure modes and mechanisms of digital I&C systems. A preliminary survey on failure modes and failure mechanisms in digital components and systems is presented in (Cetiner et al. 2009).

FMEA by itself may not be a sufficient tool to determine how specific component-level failure modes affect digital systems (Haapanen & Helminen 2002). Therefore, it could be useful to utilize more sophisticated tools, such as simulation tools, to analyse the interactions between the components of a digital system and the effects of one or more failures. A systematic FMEA approach is proposed in (Chu et al. 2010b) for creating reliability models for digital instrumentation and control systems.

### **5.2.1 Dynamic reliability modelling approaches**

There exists several dynamic reliability approaches, for instance, Dynamic Flowgraph Methodology (DFM) (Garrett et al. 1995, Garrett & Apostolakis 2002, Yau et al. 1995), Markov/CCMT (cell-to-cell mapping technique) (Aldemir et al. 2009, Bucci et al. 2008), Petri Nets (Labeau et al. 2000), Bayesian approaches (Pearl 1988, Doguc & Ramirez-Marquez 2009, Kelly & Smith 2009), test-based approaches (Aldemir et al. 2006), Boolean logic Driven Markov Process (Bouissou 2002), and black box approaches (Musa & Okumoto 1984, Schneidewind & Keller 1992). DFM and Markov/CCMT were ranked as the two top dynamic reliability modelling approaches with the most positive features and least negative features (Aldemir et al. 2009).

DFM is based on directed graphs for modelling and analysing the behaviour and interaction of software and hardware within an embedded system (Garrett et al. 1995). Dynamic flowgraphs can predict future failures and integrate hardware and software components. However, extensive technical knowledge is required for the creation of a DFM model. Continuous variables have to be discretized, which is a trade off between model accuracy and complexity and analysis time. The number of time steps that can be analysed in deductive mode is limited by computational constraints.

The Markov/CCMT approach combines the traditional Markov methodology with cell to cell mapping. The approach enables to represent possible couplings between failure events, originated from dynamic interactions between the digital I&C system and the

controlled process, and among the different components of the I&C system (Aldemir et al. 2009). Construction of a full Markov/CCMT model may not be computationally feasible if the analysed system contains a large number of states. It requires a substantially larger amount of technical knowledge compared to that needed for a traditional ET/FT analysis.

A benchmark implementation of a digital feedwater control system modelled with the two methodologies is discussed in (Aldemir et al. 2009). A brief comparison between the results obtained with the two dynamic methodologies and results computed for the same system with traditional PSA methods is discussed in (Chu et al. 2009b). The integration of the results obtained with the dynamic model is fairly straightforward, if the basic events identified by the dynamic models do not also appear as basic events elsewhere in the standard PSA models.

Model checking is a computer aided automatic verification technique for formally verifying the correct functioning of a system design model against its formal specification (Clarke et al. 2000). Model checking is not directly applicable for reliability assessment of digitalized I&C systems. An approach that combines a safety assessment methodology (fault tree analysis) and a formal methodology (model checking) to provide formal, automated and qualitative assistance to informal and quantitative safety assessment is presented in (Koh & Seong 2009). An application of model checking and fault tree analysis for the safety analysis of an embedded system is described in (Ortmeier et al. 2003). The use of model checking for fault coverage analysis has been proposed in (Bozzano & Villafiorita 2007, Bingham & Lach 2009). Also efficient symbolic techniques for probabilistic model checking have been developed, e.g. (Kwiatkowska 2009).

### **5.2.2 Software reliability modelling**

Software failures are in general mainly caused by systematic (i.e. design specification or modification) faults, and not by random errors. Software based systems cannot easily be decomposed into components, and the interdependence of the components cannot easily be identified and modelled. Applying software reliability models in the PSA context is hence not a trivial matter.

Software reliability models usually rely on assumptions and statistical data collected from non-nuclear domain and therefore may not be directly applicable for software products implemented in nuclear power plants. More important than the exact values of failure probabilities are the proper descriptions of the impact that software-based systems have on the dependence between the safety functions and the structure of accident sequences. Conventional FT-approach is, on the other hand, considered sufficient for the modelling of RPS like functions.

In spite of the unsolved issue of addressing software failures there seems to be a consensus regarding some philosophical aspects of software failures and their use in developing a probabilistic model. The basic question: “What is the probability that a safety system or a function fails when demanded” is a fully feasible and well-formed question for all components or systems independently of the technology on which the systems are based (Dahll et al. 2007). A similar conclusion was made in the Workshop on Philosophical Basis for Incorporating Software Failures in a Probabilistic Risk Assessment (Chu et al. 2010c). As part of the open discussion, the panelists unanimously agreed that:

- software fails
- the occurrence of software failures can be treated probabilistically
- it is meaningful to use software failure rates and probabilities
- software failure rates and probabilities can be included in reliability models of digital systems.

For the quantification of software failure rates and probabilities there are several general approaches, e.g., reliability growth methods, Bayesian belief network (BBN) methods, test based methods, rule based methods (Dahll et al. 2007) and software metrics based methods (Smidts & Li 2000, 2004). These methods are reviewed in (Chu et al. 2010a). None of the methods for quantifying digital systems reliability is universally accepted, in particular for highly reliable systems (EPRI 2010).

Reliability growth models are based on the sequence of times between observed and repaired failures (Dahll et al. 2007). The models calculate the reliability and the current failure rate. Additionally, the reliability growth models can predict the time to next failure and required time to remove all faults.

The BBN methodology has been adapted to software safety assessment (Helminen 2001, Helminen & Pulkkinen 2003) and the methodology can be considered as promising. One of the main drawbacks is that a different BBN has to be built for each software development environment. This problem may be solved by using generalized BBN templates which are not restricted to a specific development environment (Eom et al. 2009).

In test based methods a program is executed with selected data and the answer is checked against an ‘oracle’. A reliability measure can be generated, by running a number of tests and measuring the number of failures. Test-based reliability models assume that the input data profile used during the test corresponds to the input profile during real operation. Unfortunately, this correspondence cannot often be guaranteed.

To assess software risk contribution, (Yau & Guarro 2010, Guarro 2010) presents an application of Context-based Software Risk Model (CSRM). CSRM allows assessing the contribution of software and software-intensive digital systems to overall system risk in a way that can be integrated with the PSA format used by NASA described in (Vesely et al. 2002). PSA techniques for modelling digital I&C system software reliability focusing in the modelling of digital system software CCF, and features of I&C systems that minimize potential CCF is described in (Enzinna et al. 2009).

### **5.3 Reliability data for digital I&C systems**

#### **5.3.1 Hardware reliability data**

Usually, hardware failure data is provided by the vendor of the equipment. This is standard requirement in the contract between the utility and the vendor. The data provided by the supplier sets the limit for the detail of the PSA, i.e., it is not feasible to model in more detail due to lack of reliability data.

Two kinds of failure data may be provided by vendors: 1) based on operating experience, 2) based on a part counting method followed by a standard like Siemens SN

29500 (Siemens 2004) or generic data bases such as the reliability prediction database the Military Handbook, MIL-HDBK-217, for "Reliability Prediction of Electronic Equipment" (US DOD 1995). MIL-HDBK-217 contains failure rate models for the various part types used in electronic systems, such as integrated circuits, transistors, diodes, resistors, capacitors, relays, switches, and connectors. These failure rate models are based on mathematical models derived from empirical field failure rates that are gathered for different parts and systems. Those models respect ambient conditions, level of stress, and type of applications.

Failure data is typically provided in terms of failure rate (1/time unit). From the PSA modelling point of view it is necessary to distinguish between detected and latent failures, which depends on the failure detection features of the I&C units. The judgement of the share of detected vs. latent failure rates needs to be provided by the vendor.

A second important reliability parameter needed for PSA is CCF failure rates. CCF parameters are sometimes derived from some generic values, but as an alternative IEC 61508-6 (IEC 2010b) has been used, e.g., in (Authén et al. 2010b).

### **5.3.2 Software reliability data**

Sophisticated software reliability estimation methods presented in the academic literature are not applied in real industrial PSAs. Instead, the numbers are some kind of engineering judgments for which justifications may be hard to find. The engineering judgement approaches can be divided into the following categories depending on the argumentation and evidence they use (Björkman et al. 2012):

- screening out approach
- screening value approach
- expert judgement approach
- operating experience approach.

The reliability model used for software failures is practically always the simple "probability of failure per demand", denoted here by the parameter  $q$ .

Generally, only common cause failures are modelled in PSA. One reason for this is that there has not been a methodology available to correctly describe and incorporate software failures into a fault tree model. The only reliability model which is applied is constant unavailability (probability of failure per demand) and this is used to represent the probability of CCF per demand. Spurious actuations due to software failures are not modelled or no need to consider software failure caused spurious actuations has been concluded.

Software CCF is usually understood as the application software CCF or its meaning has not been specified. Software CCF is generally modelled between processors performing redundant functions, having the same application software and on the same platform. One of the exceptions is the design phase PSA made for the automation renewal of the Loviisa NPP, where four different levels of software failures are considered: 1) single failure, 2) CCF of a single automation system, 3) CCF of programmed systems with



same platforms and or software, and 4) CCF of programmed systems with different platforms and or software (Björkman et al. 2012).

With regard to the reliability numbers used in PSA, it is difficult to trace back where they come from — even in the case of using operating experience. The references indicate the sort of engineering judgement but lacks supporting argumentation. To overcome the shortcomings of the present approaches for software failure rate estimation, an analytical approach is provided in (EPRI 2010).

#### **5.4 Nordic PSA-studies**

A study of existing Nordic PSA:s with digital I&C included has been performed in order to identify similarities and differences, i.e. to identify present Nordic state-of-the art if possible (Authén et al. 2010a). The study identified the types of computerized systems that are included in the PSA models and gives a brief description of the level of details, failure modes considered and data used. Four Nordic PSA:s were included in the study:<sup>1</sup>

- Olkiluoto 1/2 (OL1/2), Siemens and ABB I&C design
- Ringhals 1 (R1), Areva/Siemens I&C design
- Ringhals 2 (R2), Westinghouse I&C design
- Loviisa 1/2 (LO1/2), Areva/Siemens I&C design

Results of the comparison of the modelling approaches are presented in Tables 1 to 4. The comparison shows that among the four PSA:s there are four different approaches on how to describe the system reliability. In general the PSA:s are performed with different prerequisites. Also significant differences in assumptions and simplifications are found when compared, e.g. regarding coverage of I&C design features, level of detail and critical failure modes.

Consensus in all four PSA:s is hard to find, other than that all PSA:s analyses loss of RPS actuation and does not consider single software failures nor dynamic interactions between software and hardware. Only one PSA models spurious RPS actuations (though Loviisa PSA will include it at a later stage) and the same PSA is alone to consistently apply a high level of detail in the analysis.

Most PSA:s model processor failure as a super component and also considers hardware and software CCF:s on a super component level. “Super component” means that the object (i.e., the processor in this case) is treated as a single entity in the model, without breaking it down into a “subcomponent level”. Also, three out of four PSA:s models undetected failures (tested) consistently.

Regarding references on failure data all PSA:s use supplier data for hardware failures, but when it comes to data for hardware and software CCF:s different solutions has been applied.

Parts of the differences in approach can be explained by different designs, status of the design and in some case by different I&C applications. Both the design and the

---

<sup>1</sup> Oskarshamn 1 has also a PSA with digital safety I&C, but it was not included in the study.

application of the I&C of course sets some boundary conditions for the reliability analysis and in the choice of approach and modelling solutions. Chosen approach in the PSA is also dependent on the phase of the implementation process for the I&C system: design phase PSA, detailed design phase PSA or as-built PSA.

**Table 1. Comparison of coverage of digital I&C design aspects in PSA.**

Modelling aspects	OL1/2	R1	R2	LO1/2	Comments
Loss of (RPS) Actuation	●	●	●	●	
Spurious (RPS) Actuation	-	●	-	-	
Engineered Failure Detection	○	●	●	-	
Failure of Eng. Failure Detection	○	-	○	-	
Engineered Fail-Safe Actions	○	●	-	-	
Degraded Voting Logic	s	●	-	-	
Intra Division Communication	○	●	●	-	
Inter Division Communication	○	●	●	-	
Dynamic Interactions	-	-	-	-	

● Modelled as standard, ○ Modelled as exception, special case or qualitatively, s Screened out from the PSA model

**Table 2. Comparison of coverage of failures and failure modes.**

Failures and modes	OL1/2	R1	R2	LO1/2	Comments
Hardware Failure Single Comp.	○	●	○	-	
Hardware Failure Super Comp.	●	-	●	s	
Hardware CCF Single Comp.	○	●	○	-	
Hardware CCF Super Comp.	●	-	●	●	
Software Failure	s	-	○	-	For sensitivity analysis
Software CCF Single Comp.	s	-	-	-	
Software CCF Super Comp.	●	●	-	●	Application software
Undetected Failure	●	●	●	-	
Detected Failure	●	●	○	-	
Spurious Failure	-	○	○	-	Screened out from analysis
Corr. Maint. Single Comp.	○	○	○	-	
Corr. Maint. Super Comp.	○	●	●	-	

● Modelled as standard, ○ Modelled as exception, special case or qualitatively, s Screened out from the PSA model

**Table 3. Comparison of coverage of digital I&C hardware components.**

Hardware Components	OL1/2	R1	R2	LO1/2	Comments
Processor, Super Comp.	●	-	●	●	In OL1/OL2, subcomponents' failure modes analysed (FMEA & FT) in the background documents
Processor	-	●	-	-	
Communication Module	-	●	●	-	
Digital Input/Output Module	-	●	○	-	
Digital Input/Output Channel	-	●	-	-	
Analog Input/Output Module	-	●	-	-	
Analog Input/Output Channel	-	●	-	-	
Signal Conditioning Module	-	●	-	-	
Subrack	-	●	-	●	
Misc. Modules	●	●	●	-	
Watchdog	-	-	○	-	
Controller Module for Continuous Closed-loop Control	-	-	-	-	
Priority unit	-	-	-	●	

● Modelled as standard, ○ Modelled as exception, special case or qualitatively, s Screened out from the PSA model

**Table 4. Comparison of failure data references.**

Failure Data	OL1/2	R1	R2	LO1/2
Hardware failure data	Supplier data	Supplier data	Supplier data	Supplier data
Hardware CCF	Eng. Judge	IEC 61508 / RAB	IEC 61508 / Supplier	Eng. Judge
Software CCF	Supplier data / Eng. Judge	Supplier data	Screened out	Eng. Judge

## **6. Failure modes taxonomy**

### **6.1 WGRISK task group DIGREL**

In 2007, the OECD/NEA CSNI directed the Working Group on Risk Assessment (WGRisk) to set up a task group to coordinate an activity on digital I&C system risk. The focus of this WGRisk activity was on current experiences with reliability modelling and quantification of these systems in the context of PSAs of NPPs. Two workshops were organised to share and discuss experiences with modelling and quantifying digital I&C systems. The participants recognized that several difficult technical challenges remain to be solved. One of the recommendations was to develop a taxonomy of hardware and software failure modes of digital components for the purposes of PSA (OECD 2009).

As a continuation, a new task proposal was made to WGRISK, which was accepted by WGRISK and CSNI in 2010. The objectives of the new task called DIGREL were

- To develop technically sound and feasible failure modes taxonomy (or taxonomies if needed to address variations in modelling methods or data availability) for reliability assessment of digital I&C systems for PSA.
- To provide best practice guidelines on the use of taxonomy in modelling, data collection and quantification of digital I&C reliability.

The activity focused on failure modes taxonomy and its application to modelling, data collection and impacts on quantification. The following items have been considered (among other things):

- Protection systems and control systems,
- Hardware and software,
- Development, operation and maintenance,
- Failure detection and recovery means.

There are many different digital I&C failure mode taxonomies. An activity focused on development of a common taxonomy of failure modes was seen as an important first step towards standardised digital I&C reliability assessment techniques for PSA. Needs from PSA have guided the work, meaning e.g. that the (digital) system and its failures are studied from their functional significance point of view. This was considered a meaningful way to approach the problem.

The taxonomy will be the basis of future modelling and quantification efforts. It will also help to define a structure for data collection. The results of the activity can be directly used in the review of PSA studies.

The activity has taken advantage from recent and ongoing R&D activities carried out in the OECD/NEA member countries in this field. More PSA applications including digital I&C systems have been or are being prepared. Efforts to analyse operating experience from digital systems are in progress. This knowledge will be merged by inviting experts in the field to contribute to the activity. A series of working meetings have been organised and public seminars have been organised annually.

The following organisations participated in the preparation of the taxonomy report (OECD 2014): VTT, Finland; Risk Pilot, Sweden; IRSN, France; EDF, France; AREVA, France; GRS, Germany; KAERI, Korea; NRC, USA; Ohio State University, USA; NRI, Czech; JNES, Japan; VEIKI, Hungary; ENEL, Italy; NRG, the Netherlands; RELKO, Slovakia and CSNC, Canada.

## **6.2 General approach for the development of the taxonomy**

Failure modes taxonomy is a framework of describing, classifying and naming failure modes associated with a system. Main uses of failure modes taxonomies are in the performance of reliability analyses and in the collection of operating experience (failure data) of technological systems. In the DIGREL, the taxonomy is developed jointly by PSA and I&C experts which have slightly different views and needs on defining the failure modes.

The fault tree modelling and systems analysis in PSA is a combination of top down and bottom up approaches. Fault tree modelling is a top down method starting from the top level failure modes defined for the system. In the system level, the two main failure modes are 1) failed function and 2) spurious function. For the failed function more descriptive definitions may be given such as “no function”, “not sufficient output”, “no state transition”, “broken barrier”, “loss of integrity”, etc., depending on the nature of the system. In the fault tree analysis, the system level failure modes are broken down further into sub-system and component level failure modes. The system level failure modes appear thus as fault tree gates in the PSA model, while component level failure modes appear as basic events.

Basically, same failure modes taxonomy can be applied for components as at the system level (failed function, spurious function), but the definitions are usually more characterising, e.g., “sensor freeze of value”, and are closer related to the failure mechanisms or unavailability causes. The component level failure modes are applied in the performance of the FMEA (failure modes and effects analysis) which is a bottom-up analysis approach. The analysis follows the list of components of the system and for each component failure modes, failure causes (mechanisms) and associated effects are identified. FMEA precedes the fault tree modelling but it needs the definitions of the system functions and associated failure modes.

From the PSA point of view, the definitions for the failure modes and the related level of abstraction in the fault tree modelling can be kept in a high level as long as relevant dependencies are captured and reliability data can be found.

## **6.3 Requirements**

The development of a taxonomy is dependent on the overall criteria and prerequisites since they will set boundary conditions e.g. for the needed level of abstraction of hardware resp. software components and for the structure of the failure modes. A different set of criteria may result in a different taxonomy, and the criteria are partly conflicting, in which case some balance needs to be found.

In the context of failure modes taxonomy, the main possible conflict in the requirements is same as with the PSA: the wish to have a realistic and complete taxonomy (or PSA model) and on other hand to have a practical, usable and understandable taxonomy (or

PSA model). There is a pressure both towards perfectionism and towards simplifications between which targets a balance must be decided.

A related question is to what extent the plausibility of a failure mode is a criterion for defining the taxonomy. On one hand, we may define all theoretically possible failure modes regardless of their likelihood, and let the user of the taxonomy to decide (e.g. based on available data) which are relevant for the application. This approach is however problematic since our imagination may produce a large set of failure modes which is impractical basis for the use of the taxonomy. The plausible failure modes approach could be thus preferred, but it may be difficult to generally define which failure modes are relevant for certain components.

As a conclusion, the used approach to develop a taxonomy compromises between the simplicity and completeness targets.

Following the general principles of taxonomy construction and the particular requirements set by the domain of study, i.e. failure modes for digital instrumentation and control systems for application to PSA practice, the following set of criteria has been defined:

- **Criterion 1: Defined unambiguously and distinctly**  
There should be a clear definition of each failure mode with distinct characteristics which allow the analyst to clearly distinguish one failure mode from another. This criterion will ensure repeatable classification and hence help to ensure the quality of the information (e.g. failure data) collected.
- **Criterion 2: Form a complete/exhaustive set**  
This criterion stems from the need to cover all possible types of failures of software-based digital instrumentation and control systems so as to not leave potential risk contributors unidentified.
- **Criterion 3: Be organized hierarchically**  
This criterion allows easy organization of the taxonomic information and retrieval of the information. It also allows access to multiple levels of modelling.
- **Criterion 4: Be mutually exclusive**  
This criterion ensures that each failure mode will belong to one and only one taxonomic class at each taxonomic level. This is important for the failure data classification and consistent estimation of failure rates.
- **Criterion 5: Data to support the taxonomy should be available now or in the future**  
This criterion stems from the planned usage of the taxonomy and data collected on failure modes for PSA quantification. This criterion states that, if such a system does not yet exist, one should be able to put in place a data collection system that would allow accurate reporting of occurrence of such failure modes as well as number of opportunities for such occurrence. Presently data collection is seen problematic especially with regard to software faults. This taxonomy aims to support better data collection in future.

- Criterion 6: There should be analogy between failure modes of different components

For many components there is a natural decomposition of the failure modes. However, there is benefit for using a consistent failure mode taxonomy for components that accomplish comparable functions and have similar failure modes. While it is recognized that model fidelity and realism may require the introduction of component specific failure modes, this criterion should provide a guiding principle for consistent taxonomy development.

- Criterion 7: At the very least, the lowest level of the taxonomy should be sufficient to pinpoint existing dependencies of importance to PSA modelling

Dependencies between components may lead to dependent failures that are potentially high impact risk contributors. The taxonomic levels should be such that one or multiple levels of the taxonomy allow accurate representation of such dependencies. This criterion is challenging in the sense that the number of potential faults in digital I&C is very high and we have a limiting ability to identify all dependencies and event propagation paths.

- Criterion 8: Should support PSA practice, and fulfil PSA requirements and conditions

This criterion comprises of a wide range of aspects, which vary between PSA projects, e.g.

1. Form a feasible basis for PSA experts to perform FMEA and fault tree analysis
  2. Possible to implement into existing tools
  3. Possible to review by a PSA-expert
  4. Allow living PSA, e.g. possible to maintain and update with reasonable resources
  5. Available and maintainable failure data, i.e., allows collection and evaluation of operational events
  6. Support PSA applications.
- Criterion 9: Should capture defensive measures against fault propagation (detection, isolation and correction) and other essential design features of digital I&C

The larger part of the failures within a digital I&C RPS will be detected by monitoring features such as self-surveillance, open circuit monitoring, cross channel comparison etc., while a small part only will be detected by periodic tests or actual need of the equipment. There are many fault tolerant features implemented at different levels of detail that may be platform and application specific. The failure parameters (i.e., failure rates and coverage) need to accurately capture the fault tolerant features.

## 6.4 Levels of abstraction

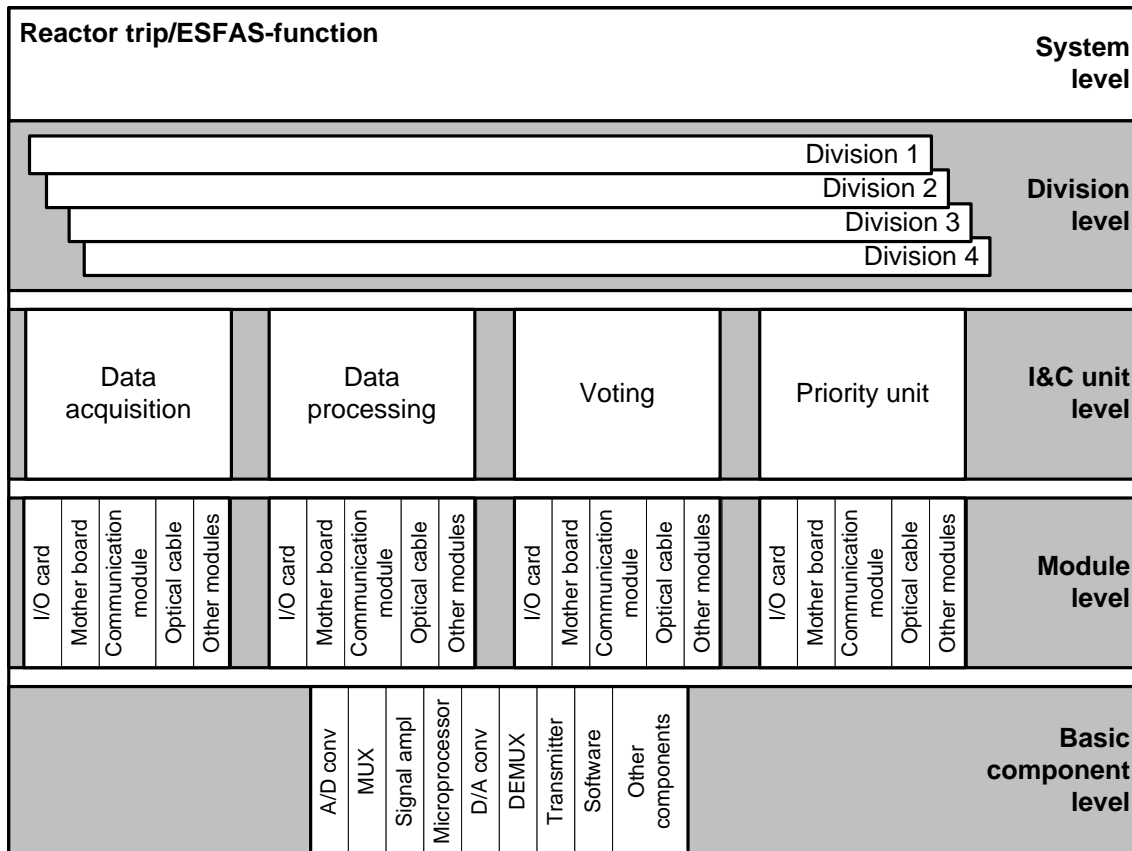
A failure modes taxonomy is based on an architecture structure that provides a hierarchical view on the system and its parts. Different levels of abstraction may be

defined and failure modes can be defined from a function point of view or from a component point of view.

With regard to the analysis and modelling of protection systems, the following levels of details are distinguished (Figure 3):

- System level: a collection of equipment or platforms (subsystems) that is configured and operated to serve some specific plant function as defined by terminology of each utility. For a digital protection system, at the system level, the software consists of the collection of software running on various microprocessors of the system and failure modes can be defined at this highest level.
- Division level: the system can be carried out in redundant or diverse divisions. In this case, a division may consist of the pathway(s) from sensor(s) to generation of an actuation signal. One such pathway is designated as a channel. The actuation signal can be sent to multiple actuators. A division can be decomposed further in I&C units. For the redundant or diverse divisions of a digital protection system, the collection of software running on the microprocessors of a single division may also fail and cause the failure of that division.
- I&C unit level: a division consists of one or more I&C units that perform specific tasks or functions that are essential for a system in rendering its intended services. I&C units consist of one or more modules. There is a limited number of I&C unit categories in a protection system.
- Module level: an I&C unit can be decomposed into modules that carry out a specific part of the process. For example, input/output-cards, motherboard, and communication cards, etc. An I&C unit may contain only a subset of these modules. The software program running on a particular microprocessor is also decomposed into modules (see Table 5).
- Basic component level: a module is composed of a set of basic components bounded together on a circuit board in order to interact. Consequently, the states of a module are the set of the combined (external) states of its basic components. Failure modes defined at the basic component level should be independent of design or vendor. Basic component level decomposition is only considered for hardware modules. For software it is not considered meaningful to go beyond the module level indicated in Table 5.





**Figure 3.** Principal structuring of safety I&C into different levels of details.

**Table 5.** Software modules in I&C units.

Unit	Software modules
I&C unit <ul style="list-style-type: none"> <li>Acquisition and processing unit (APU)</li> <li>Voting unit (VU)</li> </ul>	<ul style="list-style-type: none"> <li>System software</li> <li>Application specific software modules</li> <li>Elementary functions</li> <li>Functional requirements specification (virtual software)</li> </ul>
Data communication unit (DCU)	<ul style="list-style-type: none"> <li>System software</li> <li>Data communication software</li> <li>Data link configuration</li> <li>Functional requirements specification (virtual software)</li> </ul>
Potentially any kind of I&C unit	<ul style="list-style-type: none"> <li>Proprietary SW modules (specific pieces of software present in hardware modules), also called COTS modules</li> </ul>

## 6.5 Failure model

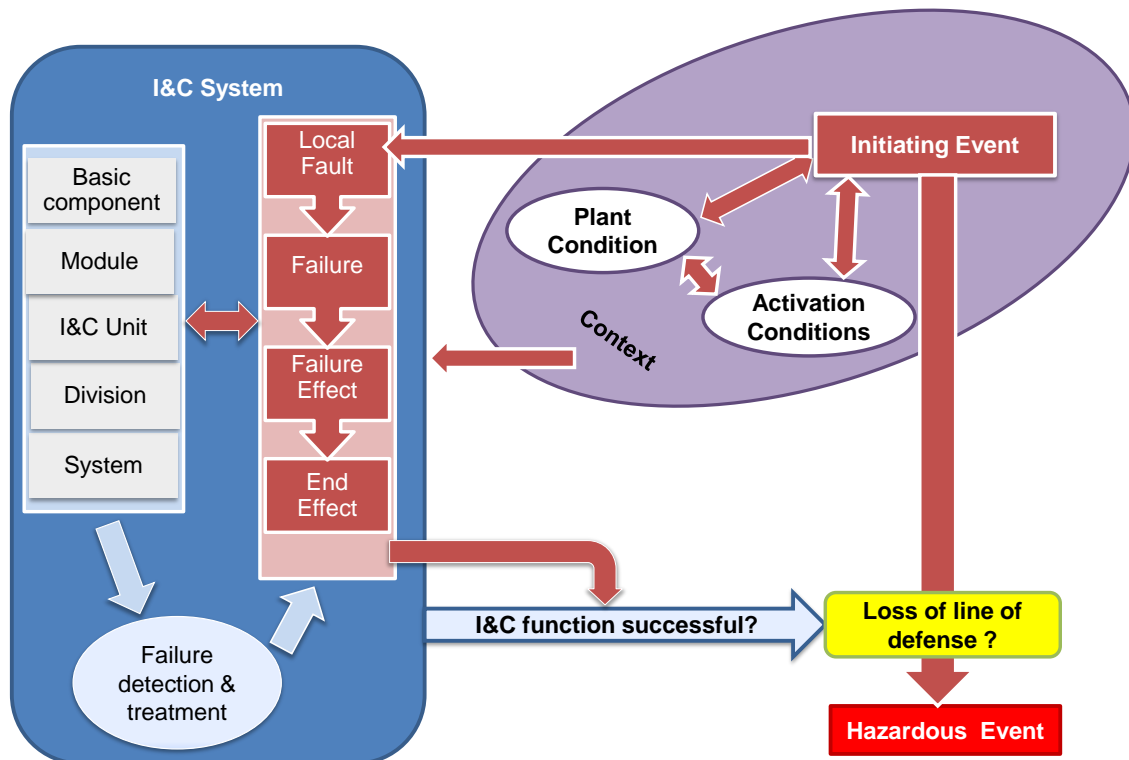
The taxonomy is developed using a specific conceptual model of failure and failure propagation. The important elements of the failure model are:

- fault location,
- failure mode,
- uncovering situation,
- failure effect,
- end effect.

These concepts are applied, in particular, to define the relationship between a *fault* in hardware or software modules (module level *failure modes*) and the *end effect* on I&C units (I&C unit level failure modes). In the analysis, a fault is postulated in a hardware or software module (*fault location*). For hardware modules, different *failure modes* are explicitly defined. Software module *failure modes* are directly associated with the *failure effect*. *Uncovering situation* describes when, where and how the module failure is significant at the I&C unit level. A taxonomy of generic *failure effects* is defined to provide a simple but exhaustive way to categorise the effect of wrong output in a module.

The *end effect* describes the final propagation of the failure, taking into consideration all these elements of the failure model. In this consideration, a distinction can be made between the “maximum possible end effect”, when fault tolerance design (FTD) is not effective or does not exist, and the “most likely end effect”, assumes that FTD features are present and effective. FTD is effective only when the fault is detected by online monitoring, which is one of the uncovering situation categories.

A comprehensive description of the failure model can be found in (OECD 2014) and is illustrated in Figure 2. Failure propagation is the path from a “locally” postulated fault to a system or plant level end effect, and it is dependent on the “context”, which defined by the “plant condition”, “initiating event” and “activation conditions”. The propagation can be considered at different levels of abstraction following the I&C architecture. The most interesting part for PSA modelling is though the propagation between module and I&C unit levels.



**Figure 4.** Failure model (OECD 2014).

This general approach has been developed in the course of the DIGREL project. Its applicability and usefulness need to be assessed in further research efforts.

## 6.6 Failure mode taxonomy at System and division levels

Practically, the safety-related function of the system is defined as the generation of safety-related actuation signal in a predefined time interval only when required. An output of the system is a set of outputs of the divisions. Thus, the failure modes in the division level are similar with those of the system level, which are

- failure to actuate the function (including late actuation),
- spurious actuation.

## 6.7 Failure mode taxonomy at I&C unit and module levels

The key part of the digital I&C failure modes taxonomy is in the I&C unit and module levels where the fundamental functionality of the system can be discussed, e.g., the defensive measures against faults. It is practical to keep these two levels together in the taxonomy since the meaning is to define the relation between failure modes of an I&C unit and the modules.

In the analysis, the existence of faults is postulated in the modules (hardware or software), and the question is to determine 1) how the I&C unit is affected and 2) how other units that communicate with the defected unit are affected. In order to answer to these questions, the following issues need to be defined:

- The fault location: In which hardware or software module the fault is located?
- Failure effect:
  - *Fatal, ordered failure*: generation of outputs ceases, outputs are set to specified, supposedly safe values. Halt/abnormal termination of function with clear message.
  - *Fatal, haphazard failure*: generation of outputs ceases, outputs are in unpredictable states. Halt/abnormal termination of function without clear message.
  - *Non-fatal, plausible behaviour*: I&C runs with wrong results that are not evident. An external observer cannot determine whether the I&C unit or the hardware module has failed or not.
  - *Non-fatal, non-plausible behaviour*: I&C runs with evidently wrong results. An external observer can decide that the I&C unit or the hardware module has failed.
- Uncovering situation:
  - *Online detection*. Covers various continuous detection mechanisms.
  - *Offline detection*. E.g. periodic testing, and also other kind of periodic controls which can be credited in PSA.
  - *Revealed by demand*.
    - Latent failure, revealed by demand. A failure is present that is not detectable by online or offline mechanisms (test independent failure).
    - Failure triggered by demand. A specification error causes a failure on demand in an unexpected context.
  - *Revealed by spurious actuation*. An event in which the occurrence of the failure immediately triggers spurious actuation. Spurious actuation may happen before a demand or it may cause a demand (common cause initiator). Spurious actuation may be caused by the fail-safe behaviour of I&C initiated

by online monitoring or the activation of the fault triggers spurious actuation before any FTD has time to take place. This situation covers two variants:

- Spurious actuation due to functional failure, incl. voting logic
- Spurious actuation due to failure of detection mechanism.

The combination of fault location, failure effect, uncovering situation together with the fault tolerant design (FTD) of the system are usually sufficient to determine the functional end effect in the I&C unit (APU or VU). Determination must be done case by case and is the essential part of the failure analysis.

An important issue is that it is neither necessary nor reasonable to assume all possible combinations, which considerably reduces the number of relevant failure modes (see Table 6). Fatal haphazard failures are not considered in this analysis, because here it is assumed that modules of the reactor protection system do not fail in an unknown state (OECD 2014). Fatal failures are ordered and are detected by online detection or by spurious effect.

Non-fatal failures are more dangerous since any uncovering situation may be possible. In case of non-plausible behaviour, failure is detected by online detection or by spurious effect. “Plausible behaviour” refers to the case where the failure is not detected by online detection.

**Table 6.** Relevance of the combinations of local effects and detection situations.

<b>Failure effect</b>	<b>Uncovering situation</b>				
	Online detection	Offline detection	Revealed by spurious action	Latent revealed by demand	Triggered by demand
Fatal, ordered	R		R		R
Fatal, haphazard		R	R	R	R
Non-fatal, plausible behaviour		R	R	R	R
Non-fatal, non-plausible behaviour	R		R		R

**R:** Combination relevant for further analysis of end effects

### 6.7.1 I&C unit level failure modes and effects

In the analysis of functional impacts on I&C units, we distinguish between the impact on a single I&C unit and impact on multiple I&C units. The latter is especially important when analysing the impacts of software faults (systematic fault in the design).

From a single I&C unit point of view, the following functional failure modes can be considered

- Loss of all functions (outputs) of the I&C unit,
- Loss of a specific function,
- Spurious output (one function),
- Spurious output (all functions).

The above list is not exhaustive, and, e.g., for voting units the functional end effect may be more complex (e.g. degraded voting logic). Diesel load sequencer is also an example of a rather complex I&C function, for which a large number of failure modes may be assumed (but it can be sufficient to model only few of them in PSA).

In the example I&C architecture (Figure 1), the following end effects of a failure can be assumed:

- FF-1SS: Failure of one Function (or more) in one subsystem. This case refers to non-fatal software failures that result in the misbehaviour of one or more I&C functions in one subsystem. The I&C functions that are dependent on the failed functions could also fail. Those dependent functions are necessarily in the same subsystem.
- FF-1D-1SS: Failure of one Function (or more) in only one division in one subsystem. This case refers to non-common cause, non-fatal software failures of I&C functions without vote.
- FF-AllSS: Failure of one Function (or more) in all subsystems
- 1APU/1VU: Failure of one set of redundant APUs/VUs. This case refers to fatal software failures affecting only one set of redundant APUs/VUs (necessarily in the same subsystem).
- MAPU-1SS/MVU-1SS: Failure of multiple sets of redundant APUs/VUs in only one subsystem
- 1SS: Loss of one subsystem.
- MAPU-AllSS/MVU-AllSS: Failure of multiple sets of redundant APUs/VUs in both subsystems
- 1SS-APU/1SS-VU: Loss of one Subsystem and of one or more sets of redundant APUs/VUs in the other subsystem.
- SYSTEM: Loss of both subsystems.

### **6.7.2 Module level failure modes and effects**

At the module level, a distinction is made between the treatment of hardware and software related failure modes. The failure effects classification defined at modules level is the same at the I&C unit level.

Table 7 and Table 8 give typical examples of failure modes for hardware resp. software modules, which have been collected from the taxonomy working group members (see appendix of the taxonomy report (OECD 2014)). In chapter 7.2 it will be shown how these failure modes are developed further to match the needs of PSA modelling.

**Table 7.** Failure modes and failure effects of hardware modules collected from different taxonomies.

<b>I&amp;C module output</b>	<b>Module types</b>	<b>Failure modes</b>	<b>Failure effect</b>
I&C modules with digital outputs	Digital input modules, digital output modules	Hang, Crash (no output)	Fatal failure
		Output* fails to 1 Output fails to 0 Output stuck to current value Output fails to the opposite state Delayed output Random output	Non-fatal failure
	Processing module	Hang, crash (no output)	Fatal failure
		Wrong output Delayed output Random output Other failure modes depending on the platform	Non-fatal failure
	Digital communication modules	Failure modes are protocol dependent	Protocol dependent
I&C modules with analog outputs	Analog input modules, analog output modules	Hang, crash (no output)	Fatal failure
		Output fails to MAX	Non-fatal failure
		Output fails to MIN/0	Non-fatal failure
		Output fails to an erroneous value (out of range) Delayed output Random output (output fluctuates, in range, between minimal and maximal value)	Non-fatal failure
		Drifted output (output is x% more than actual value)	Non-significant or non-functional effect; with plausible or implausible behaviour

\*Output can be a single output, several outputs or all outputs of the module, which needs to be specified in the failure analysis.

**Table 8.** Failure modes and failure effects of software modules collected from different taxonomies.

Module types	Failure modes	Failure effect
System software (SyS)	Hang, crash (no output). <ul style="list-style-type: none"><li>For example: Software stuck in an infinite loop, divisions by zero or illegal access to memory (e.g., writes to ROM or read/writes to inexistent memory addresses), attempt to use illegal instruction, access to invalid data or code, attempt of operation not allowed in the current CPU mode</li><li>These failures are trapped by the microprocessor exception features</li></ul>	Fatal failure
Elementary functions (EF), application specific software (AS), functional requirements specification (FRS)	Hang, crash (no output).	Fatal failure
	Output* fails to 1 Output fails to 0 Output stuck to current value Output fails to the opposite state Delayed output Random output	Non-fatal failure
Digital communication modules (DCS, DLC)	Failure modes are protocol dependent	Protocol dependent
Proprietary modules (COTS-SW)	Failure modes are function dependent	Function dependent

To link taxonomy and PSA, and to assess failure propagation, the effects of module failures at I&C units level have to be analysed, especially for I&C units that share similar software or hardware modules. The effect of the I&C module failure at I&C unit level is dependent of the function of the module. For example, a signal stuck to current value in an APU output module may lead to a failure with plausible behaviour of the unit that is not the case in a DCU, etc. Also, in some cases, the failure of one module in an I&C unit may affect only some functions processed by the unit. The other functions may remain unaffected and behave correctly, unless they are functionally dependent on the failed function.

The approach for software modules is to successively postulate a single software fault in each software module regardless of the likelihood of such faults, and to determine the maximum possible extent of the failure, regardless of the measures taken by design or operation to limit that extent. Table 9 includes a number of possible failure effects for different software faults.

**Table 9. Effects of software module faults (OECD 2014).**

Effect*	SW fault location									
	SyS	EF (in APU)	APU-FRS	APU-AS	COTS-SW	VU-FRS	VU-AS	EF (in VU)	DCS	DLC
<b>FF-1SS</b>	R	R	R	R		R	R	R		
<b>FF-1D-1SS</b>	R	R	R	R						
<b>FF-AI1SS</b>	R	R								
<b>1APU</b>	R	R	R	R	R					
<b>1VU</b>	R				R	R	R	R		
<b>MAPU-1SS</b>	R	R			R					
<b>1SS</b>	R	R	R		R	R	R	R	R	R
<b>MAPU-AI1SS</b>	R	R			R					
<b>1SS-APU</b>	R	R			R					
<b>SYSTEM</b>	R	R			R	R	R	R	R	

\* Effects are explained in chapter 6.7.1

R = Relevant combination of SW fault and failure effect

### 6.7.3 Basic component level failure modes and effects

Basic components are individual standard hardware or software elements. The term “standard” means that identical basic components are present in various locations of the system.

With regard to hardware of basic components, the assessment of failure modes and effects is similar to the module level assessment. In most cases, module level assessment is sufficient. Basic component level may be needed if failure data is available at that level and if basic components form CCF groups which are not covered by module level CCF groups.

With regard to software, the module level as defined in the previous chapter is the most detailed level of abstraction considered. The next level would be the line of codes, which are both far too detailed elements for reliability any analysis and also practically non-accessible for the analysts. Chapter 7.2.2 discusses further the definition of software modules and associated software fault cases, which are reasonable to model in PSA.

### 6.8 Failure modes and effects analysis (FMEA)

An important part of a system analysis in developing a reliability model is performance of an FMEA. The results of the FMEA can provide a basis of the associated reliability model, such as a (system) fault tree model to be part of the plant-specific PSA. The FMEA would provide the relationships between the system level failure modes and more detailed level failure modes, fault tolerance design features, and dependencies (including possibly plant processes and operator actions).

FMEA for reactor protection system can be developed e.g. in the following levels:

1. the actuators (pumps, valves, diesel generators, etc.)
2. I&C units and communication links
3. I&C functions
4. hardware and software modules of I&C units



FMEA for the actuators is carried out in standard manner as part of the process system FMEAs. In this analysis the critical actuation signals and associated DC power supply dependencies need to be identified. The analysis shall provide link to the I&C units and I&C functions controlling the actuator.

In the FMEA for I&C units (e.g. VUs and APUs), power supply, the I&C functions, modules and communication links are identified.

Analysis of I&C functions shall identify associated I&C units and software modules for each function. Fail-safe principles can be identified in this context.

FMEA for hardware and software modules can be performed in a generic manner. Failure modes and effects are module type specific but otherwise generic. This is demonstrated in chapter 7.2.

With regard to the input and output modules, allocation of I&C functions between the modules and even the channels of the modules should be identified. This is needed for the determination of the test interval of the input and output modules.

## 7. PSA modelling

### 7.1 Introduction

The main purpose of the developed failure mode taxonomy is to serve as basis for the modelling of digital I&C reliability in PSA:s. The intent of this chapter is to demonstrate the usage of the developed taxonomy for PSA modelling. Another purpose of this chapter is to address the different challenges in performing a reliability model of a digital reactor protection system (RPS), and to give guidance in aspects vital for achieving a sound PSA.

The task of incorporating a reliability model of a digital I&C based RPS into a traditional PSA model meets a number of challenges due to the specific features of digital I&C, e.g. features such as functional dependencies, signal exchange and communication, fail-safe design and treatment of degraded voting logic. This requires both new modelling approaches and new fault tree structures, which are to be incorporated within the existing PSA model structure. Another challenge due to the complexity and number of components within a digital I&C RPS is to keep the PSA model comprehensive at a reasonable size, e.g., number of FT:s and basic events, and to meet requirements regarding realism, quality assurance, maintainability, etc.

In order to demonstrate the taxonomy and to present and support modelling recommendations, a number of test cases have been performed by using the example PSA model presented in Appendix A.

The example PSA model was first developed as a Master's Thesis at Royal Institute of Technology (KTH) in cooperation with the NKS/DIGREL project (Gustafsson 2012). The example was based on RiskSpectrum example model (EXPSA). The model has been further developed in order to better describe a generic BWR. The improvements cover among other things diversity of safety functions, four-redundant front line safety systems and a diversified reactor protection system. The digital I&C reliability model has been updated with new ESFAS and scram functions, and adapted to the hardware taxonomy.

The main objectives of the test cases are:

- Demonstrate the developed taxonomy and verify the usability for PSA purpose
- Produce and verify recommendations regarding
  - Level of detail of the reliability model
  - System, division, I&C unit and module level
  - Fault tolerant design, e.g. modelling of default values at detected failures and different voting logics
  - Hardware failure modes
  - Critical equipment, risk contribution of detected and undetected failures, etc.
  - Modelling of CCF (between hardware modules).
  - Software failure modes. Software failures are modelled as CCF, with different impact depending on the fault location.

Since the dominating method for performing state-of-the-art PSA is fault tree/event tree analysis, it will be the focus of this chapter. It is however recognised that other, more advanced, methods can be considered and that they in certain situations may be better suited for reliability analysis of digital I&C than traditional fault tree/event tree analysis. The taxonomy does not exclude the use of other methods than fault tree/event tree analysis.

## **7.2 Taxonomy for PSA modelling**

Chapter 6 presented generic failure mode taxonomies at different levels of abstraction. The required level of abstraction to apply in the PSA depends as earlier discussed on several factors such as complexity of the digital I&C design and the RPS architecture, purpose of the PSA, diversity of the reactor protection system and safety systems in general.

The purpose here is to demonstrate the taxonomy and to evaluate different modelling aspects, among others the required level of abstraction, why a detailed level of abstraction is required in the example PSA. Hence, the failure mode taxonomy for the module level will be applied for the example PSA. The detailed level of abstraction is necessary initially to classify the basic failure modes of each digital I&C module into one of the defined generic failure modes, in order to decide the effect of the failure on a functional level.

### **7.2.1 Hardware failure modes**

From the PSA modelling perspective, it is beneficial to define the failure modes by the functional effect to keep down the number of events and the model size. It will simplify the modelling efforts and make the fault tree structure and the dependencies more comprehensible to the PSA user. Therefore it is preferable to perform the grouping at as a high functional level as possible, taking into account failure characteristics vital for the functional effect. Such characteristics that must be considered for a digital RPS are in general means of failure detection since this decides whether or not the failure will be covered by the fault tolerant design and also the actions taken accordingly. Other characteristics that may need to be considered when defining the failure mode groups are differences in test intervals, CCF categorization and failure mode timing issues.

The described approach has been used for the example PSA to further categorize and group failures of the different digital I&C modules to achieve a more simple and PSA adapted failure modes taxonomy.

The main steps in developing the taxonomy for the example PSA are:

1. Failure effect according to the failure modes taxonomy at the module level (Table 10) is assigned to the failure modes of the digital RPS example system hardware modules presented in Appendix A, see Table A-8. Then the uncovering situation and functional impact on I&C units can be defined for the example system.
2. Compressed failure modes describing the functional impact on I&C unit level are defined based on the functional impact on I&C units and uncovering situation for the failure modes. The compressed failure modes distinguish between failures detected by the fault tolerant design (detected failures) and failures that

are not detected by the fault tolerant design (undetected/latent failures). The categories for failure detection are also further developed in order to provide information on the location of detection, and also adapted to Nordic PSA terminology, by defining generic failure detection means. See Table 11.

3. Based on the knowledge of functional impact on I&C unit level, whether detected failure will be covered by the fault tolerant design or not and the location of the detection, it is possible to define the failure end effect, i.e. the impact on RT/ESFAS actuation signals for a given module failure, see Table 11.
4. In the last step information in Table 11 which is not necessary for the fault tree modelling is removed, and the PSA adapted taxonomy presented in Table 12 then covers all the vital aspects needed for implementation in the fault trees.

**Table 10.** Demonstration of the taxonomy for the example PSA, step 1.

Hardware module	Failure mode examples	Failure effect	Uncovering situation	Functional impact on I&C units
Processor module	Hang	Fatal, ordered	Online Detection	Loss of APU or VU functions (all)
	Communication dropout	Non-fatal, implausible	Online Detection	Loss of APU or VU functions (all)
	Delayed signal	Non-fatal, plausible	Latent revealed by demand	Loss of APU or VU functions (all)
	Random behaviour	Non-fatal, plausible	Latent revealed by demand	Loss of APU or VU functions (all)
		Non-fatal, implausible	Online Detection	Loss of APU or VU functions (all)
			Spurious effect	Spurious APU/VU function(s)
Analog input module	Signal fails high/low	Non-fatal, implausible	Online Detection	Loss of all Module Application Functions
	Signal drifts	Non-fatal, implausible	Online Detection	Loss of all Module Application Functions
	Signal hangs/freezes	Non-fatal, plausible	Latent revealed by demand	Loss of all Module Application Functions
		Non-fatal, implausible	Online Detection	Loss of all Module Application Functions
Digital input module	Signal stuck to current value	Non-fatal, plausible	Latent revealed by demand	Loss of specific Module Application Functions
		Non-fatal, implausible	Online Detection	Loss of specific Module Application Functions
	Signal fails to opposite state	Non-fatal, implausible	Spurious effect	One spurious Module Application Function
Digital output module	Signal stuck to current value	Non-fatal, implausible	Online Detection	Loss of specific Module Application Functions
		Non-fatal, plausible	Latent revealed by demand	Loss of specific Module Application Functions
	Signal fails to opposite state	Non-fatal, implausible	Spurious effect	One spurious Module Application Function
Communication Module	Interruption	Non-fatal, implausible	Online Detection	Loss of specific application functions
Backplane	Loss of backplane	Fatal, ordered	Online Detection	Loss of APU or VU functions (all)
Power Supply	Interruption	Fatal, ordered	Online Detection	Loss of APU or VU functions (all)
	Short circuit	Fatal, ordered	Online Detection	Loss of APU or VU functions (all)
	Ground contact	Fatal, ordered	Online Detection	Loss of APU or VU functions (all)
Measurement	Fails high	Non-fatal, implausible	Online Detection	Loss of specific Module Application Functions
	Fails low	Non-fatal, implausible	Online Detection	Loss of specific Module Application Functions
	Drift of value	Non-fatal, implausible	Online Detection	Loss of specific Module Application Functions
	Freeze of value	Non-fatal, plausible	Latent revealed by demand	Loss of specific Module Application Functions

**Table 11.** Demonstration of the taxonomy for the example PSA, steps 2 and 3.

Hardware module	Uncovering situation	Functional impact on I&C units	Compressed failure mode	Failure detection	Failure end effect (RT or ESFAS)
Processor module	Online detection	Loss of APU or VU functions (all)	Loss of function	Monitoring <sup>1</sup>	All outputs of APU or VU acc. to FTD
	Latent revealed by demand	Loss of APU or VU functions (all)	Latent loss of function	Periodic test <sup>2</sup>	Loss of all APU/VU outputs
	Spurious effect	Spurious APU/VU function(s)	Spurious function	Self-revealing	Spurious APU/VU output(s)
Analog input module	Online detection	Loss of all module application functions	Loss of function	Self-monitoring <sup>3</sup>	1oo4 conditions of specific <sup>4</sup> APU/VU outputs acc. to FTD
	Latent revealed by demand	Loss of all module application functions	Latent loss of function	Periodic test	Loss of 1oo4 conditions of specific APU/VU outputs
Digital input module	Latent revealed by demand	Loss of all module application functions	Latent loss of function	Periodic test	Loss of 1oo4 conditions of specific APU/VU outputs
	Online detection	Loss of all module application functions	Latent loss of function	Self-monitoring	1oo4 conditions of specific APU/VU outputs acc. to FTD
	Spurious effect	One spurious module application function	Spurious function	Self-revealing	Spurious 1oo4 conditions of specific APU/VU output
Digital output module	Online detection	Loss of all module application functions	Loss of function	Self-monitoring	Specific APU/VU outputs acc. to FTD
	Latent revealed by demand	Loss of all module application functions	Latent loss of function	Periodic test	Loss of specific APU/VU outputs
	Spurious effect	One spurious module application function	Spurious function	Self-revealing	Spurious APU/VU output
Communication module	Online detection	Loss of specific application functions	Latent loss of function	Self-monitoring	1oo4 conditions of specific APU/VU outputs acc. to FTD
Backplane	Online detection	Loss of APU or VU functions (all)	Loss of function	Monitoring	All outputs of APU or VU acc. to FTD
Power supply	Online detection	Loss of APU or VU functions (all)	Loss of function	Monitoring	All outputs of APU or VU acc. to FTD
Measurement	Online detection	Loss of specific application functions	Loss of function	Monitoring	1oo4 conditions of specific APU/VU outputs acc. to FTD
	Latent revealed by demand	Loss of specific application functions	Latent loss of function	Periodic test	Loss of specific APU/VU output
<sup>1</sup> Detected by monitoring functions in the next level of I&C units, i.e. units communicating with the faulty unit. <sup>2</sup> Periodic tests according to Technical Specifications <sup>3</sup> Detected by the self-monitoring functions implemented in the module, or by monitoring mechanisms, provided by controlling modules <sup>4</sup> The end effect of the failure is restricted to outputs dependent on the failed module Offline detection is not considered here since it is only relevant with regard to unavailability due to corrective maintenance					

**Table 12.** Demonstration of the PSA adapted taxonomy for the example PSA, step 4.

Hardware module	Compressed failure modes	Failure detection	Failure end effect (RT or ESFAS)
Processor module	Loss of function	Monitoring <sup>1</sup>	All outputs of APU or VU acc. to FTD
	Latent loss of function	Periodic test <sup>2</sup>	Loss of all APU/VU outputs
	Spurious function	Self-revealing	Spurious APU/VU output(s)
Analog input module	Loss of function	Self-monitoring <sup>3</sup>	1oo4 conditions of specific <sup>4</sup> APU/VU outputs acc. to FTD
	Latent loss of function	Periodic test	Loss of 1oo4 conditions of specific APU/VU outputs
Digital input module	Latent loss of function	Periodic test	Loss of 1oo4 conditions of specific APU/VU outputs
	Loss of function	Self-monitoring	1oo4 conditions of specific APU/VU outputs acc. to FTD
Digital output module	Loss of function	Self-monitoring	Specific APU/VU outputs acc. to FTD
	Latent loss of function	Periodic test	Loss of specific APU/VU outputs
Communication module	Loss of function	Monitoring <sup>1</sup>	1oo4 conditions of specific APU/VU outputs acc. to FTD
Backplane	Loss of function	Monitoring	All outputs of APU or VU acc. to FTD
Power supply	Loss of function	Monitoring <sup>1</sup>	All outputs of APU or VU acc. to FTD
Measurement	Loss of function	Monitoring <sup>3</sup>	1oo4 conditions of specific APU/VU outputs acc. to FTD
	Latent loss of function	Periodic test	Loss of 1oo4 conditions of specific APU/VU outputs
<sup>1</sup> Detected by monitoring functions in the next level of I&C units, i.e. units communicating with the faulty unit. <sup>2</sup> Periodic tests according to Technical Specifications <sup>3</sup> Detected by the self- monitoring functions implemented in the module, or by monitoring mechanisms, provided by controlling modules <sup>4</sup> The end effect of the failure is restricted in outputs dependent on the failed module Offline detection is not considered here since it is only relevant with regard to unavailability due to corrective maintenance			

## 7.2.2 Software failure modes

Table 13 summarises software faults which ideally could be considered in PSA. In PSA it is reasonable to consider a limited number of end effects. The selection should, however, be large enough to cover all relevant cases (i.e. end effects).

Firstly, the selection of postulated software faults is dependent on the system architecture why not all end effects are of interest to take into account. A natural simplification is to assume large end effect and ignore smaller end effects since they are covered by the larger case. Large end effects include complete CCF of the system (SYSTEM), and CCF of one subsystem (ISS).

Secondly, the selection of postulated software faults is dependent on the SW quantification method. In most cases, SW fault probability is based on a simple engineering judgement and pooling of available data. As long as it is impossible to

refine the probability judgement, it is meaningless to refine the set of modelled software faults. In DIGREL, the software faults and effects proposed in Table 13 are considered.

**Table 13.** Screening of relevant software fault cases for PSA modelling.

Effect	SW fault location									
	SyS	EF (in APU)	APU-FRS	APU-AS	COTS-SW	VU-FRS	VU-AS	EF (in VU)	DCS	DLC
<b>FF-1SS</b>			4a	4a		4b	4b			
<b>FF-1D-1SS</b>			4c	4c						
<b>FF-allSS</b>										
<b>1APU/1VU</b>			3a	3a		3b	3b			
<b>MAPU-1SS</b>										
<b>1SS</b>	2a	2a	2a		2a	2a	2a	2a	2b	2b
<b>MAPU-allSS</b>										
<b>1SS-APU</b>										
<b>SYSTEM</b>	1	1			1			1	1	

1. Software fault causing loss of both subsystems (SYSTEM). This is a complete CCF covering all subsystems that have the same SyS. The probability of such an event is naturally extremely low, but the basic event can be used to evaluate the level of hardware diversity in the actuation of safety functions. It is only reasonable to consider a fatal failure consisting in a crash of the processing units, i.e., transition of the computers to a shut-down state. This maximal end effect covers all the other principally possible end effects. Software fault can be located in SyS, EFs, proprietary SW-modules in APUs/VUs, DCS, but it can be represented in a model by a single basic event.

For this event, a single generic probability needs to be estimated, denoted here  $P(\text{SYSTEM-SyS fatal CCF})$ .

2. Software fault causing loss of one subsystem (1SS). This is a complete CCF causing a fatal failure which crashes the processing units in one subsystem, i.e., transition of the computers to a shut-down state. The software fault can be located in a) the SyS, EF (APU/VU), APU-FRS, proprietary SW-modules in APUs/VUs, VU-FRS or VU-AS, or b) DCS or DLC.

Difference is that in case of fatal failure in DCS or DLC (b), VUs run and can take safe fail states. In case (a), the whole subsystem stops running and also takes a safe state.

For each case, a generic probability needs to be estimated, denoted here  $P(1\text{SS-SyS fatal CCF})$  resp.  $P(1\text{SS-DCU fatal CCF})$ .

3. Software fault causing failure of redundant set of APUs (3a, see Table 13) or VUs (3b) in one subsystem (1APU, 1VU, respectively). This is a fatal fault causing loss of all functions. The fault can be in APU/VU-FRS or APU/VU-AS.

There is a variant, where the software fault could cause the failure of multiple sets of APUs in one subsystem (MAPU-1SS). It remains to be analysed case-specifically whether there is a need to consider such CCF.



For this event, a single generic probability needs to be estimated, denoted here as  $P(\text{AS fatal fault})$ . In the example model, one basic event per set of redundant APUs resp. VUs is modelled.

4. Software fault causing a failure of one or more application functions. This is a non-fatal failure and can be failure to actuate the function or spurious actuation. The fault can be in the APUs (4a), VUs (4b) or have effect only in one division (4c). For instance, there can be safety functions which are actuated on 2-o-o-4 basis or are not implemented in all divisions.

Cases 4a–4c are modelled by application function module and failure mode specific basic events. It should be noted that the “application software module” can be defined in various manner. The largest meaningful entity is an I&C function. However an I&C function usually consists of several sub-modules, and several I&C function can utilize common sub-module. Sub-module level of abstraction can be found in the functional requirements specification. Modelling each sub-module as a basic event may however lead to a very large number of basic events, and therefore the analyst may need to group them into larger modules for practical reasons. It should be also noted definition of the AS module must be incompatible with the way of estimation of the failure probability.

For more discussion on software faults, see also (Bäckström et al. 2015).

### **7.3 Additional modelling issues**

#### **7.3.1 Common cause failures**

Analysis of common cause failures is different for hardware and software modules. For hardware modules, the challenges are same as for any other mechanical components, though with a higher degree of complexity, and there is a problem of getting justifiable CCF. The binding parameter to define the groups of hardware module failures will in most cases show to be the test procedures, rather than redundant functions since these often are difficult to define unambiguously at module level.

Another question in case of hardware modules is to what extent distinction should be made between detected and undetected failure modes. Intuitively, it can be questioned if it is reasonable to use the same CCF parameter values for both detected and undetected failure modes, e.g. with regard to time factors. It is often argued that the likelihood of CCF for detected failures should be smaller than for undetected failures. If calculated CCF parameters for conventional equipment are studied, e.g. (Marshall et al. 1988), no evidence for this can be found. The comparison is however not completely accurate and it could still be the case for digital I&C.

For software modules, it is more or less obvious that for identical modules a complete CCF should be assumed (if there is a fault in a module, it is in all identical modules and is triggered at the same time).

One debatable case is the possibility of common cause failure between diverse subsystems (case 1 of previous chapter). It is sometimes argued that common cause failure is practically eliminated due to different input trajectories of diverse subsystems. This argument is however difficult to justify.

Another critical issue for PSA results is whether CCF between related AS modules is considered or not. This case concerns AS modules which are not identical, but there is a potential to CCF due to

- use of same elementary functions
- common functional requirements specifications.

With regard to the fault coupling by elementary functions, it is in principle possible to assume fault in an elementary function module which would be then a common fault for more than one AS module. It is, however, more likely, that an AS fault is caused by a wrong usage of complex elementary function. Thus, the risk of CCF is more related to the use of complex elementary functions, and the fault coupling can be associated with the coupling via common functional requirements specifications. Bäckström et al. (2015) discusses this topic further.

### **7.3.2 Human errors**

HRA for digital systems has not been addressed in the DIGREL project. This chapter gives a short introduction to the relevant questions.

The human interactions can be divided into three categories, corresponding to the three human error categories used in PSA:

- type A: pre-initiating event human errors — human interactions related to testing, maintenance, installation, calibration of digital I&C system, in which context erroneous conditions may be introduced. The condition remains latent.
- type B: human errors causing an initiating event — human interactions related to any situation, in which context an erroneous action triggers an initiating event. Difference to type A error is that there is an immediate process consequence of the error.
- type C: post-initiating event human errors — human interactions after an initiating event. This category comprises mostly control room operator actions

The approach and challenges of HRA is different for type A, B, and C errors. In case of type A, testing and maintenance procedures need to be analysed and it is relevant to identify the strength of V&V methods associated to modifications and the fault tolerant features of the systems. Analysis of operating experience may provide insight on plausible failure mechanisms.

In case of type B, the situation is similar to “pre-digitalization” phase. The focus is on low power and shutdown plant operating state. It is matter of a task analysis to identify how the control room design and operator interfaces can affect to the likelihood of errors.

In case of type C errors, two issues are important. Firstly, it is important to identify in which situations operator back-up is technically possible to credit. Depending on the location of the I&C system failure, the failure may or may not eliminate operator actions from the control room or locally. Secondly, the features of the digitalized control room

design including procedures need to be addressed, e.g., in terms of performance shaping factors.

## **7.4 PSA model structure**

### **7.4.1 Introduction**

The complex design with failure detection, default values and degraded voting significantly increases the effort of fault tree modelling, the complexity and the size of the model, compared to a model of an old relay-based RPS. These issues can to some extent be managed by the use of *modelling blocks* and *standardized fault tree structures*.

The purpose of the modelling blocks is to group HW/SW module failure modes and FT-structures that have the same impact on the system behaviour, can be modelled in the same positions in the fault tree structure, and makes it possible to model effects of the fault tolerant design. This procedure will keep down the number of fault trees and minimize the number of event occurrences in the fault trees. It will also lead to a harmonisation of the fault trees and the fault tree structures, and hence increase the model clarity.

### **7.4.2 RiskSpectrum modelling**

In order to achieve the goal stated in subsection 7.1, a number of new standardized fault tree types have been created. Table 14 describes the applied fault tree structures and modelling blocks. The fault tree structure allows the model to describe a voting that combines failures in I&C hardware with failures of measurements, compared to the more commonly used and simplified approach where votings of these failures are modelled separately. The importance of this difference in the PSA quantification have not yet been evaluated, though it will likely have impact when considering area events and common cause initiators (CCI) in power supply.

Appendix A contains an example of fault tree modelling. Fault tree pages related to one safety function are shown, following the structure explained in Table 14.

**Table 14.** RPS digital I&C fault tree structure.

<b>Fault tree type</b>	<b>Fault tree description</b>
Safety function	The FT models failure of a Safety Function by transfer to one or several System Function FT:s.
System function	The FT models System Function success criteria and transfers to FT:s of System Divisions.
System division	The FT models System Division failures by transfers to FT:s of critical components.
Component (actuator)	The FT models basic events for mechanical component failures and functional dependencies by transfers to FT:s of e.g. Actuator Signal and power supply
Actuator signal	The FT models signal dependencies for specific component failure mode by transfers to FT:s of voltage supply, Output Module failure and RPS Actuation Signal.
Output module <sup>1</sup>	The FT models Actuator Signal failure due to failure in transfer of RPS Actuation Signal from Voting Unit via an Output Module. Output Module failure is modelled by basic events and failure of Voting Unit by transfer to VU fault tree page.
RPS actuation signal <sup>2</sup>	The FT models failure in the processing and voting of RPS Actuation Signals, and failures in signal exchange of RPS Protection Function status between VU and APU. SW failure modes type 1, 2a and 4b are modelled by basic events. Transfers are made to FT:s of RPS Protection Functions and to FT:s of failures in communication between VU:s and APU:s.
RPS protection function <sup>2</sup>	The FT models failure in the acquisition and processing of process measurements into RPS Protection Functions, and signal exchange of these values between APU:s. SW failure modes type 4a are modelled by basic events. Transfers are made to FT:s of Process Measurement and APU to APU communication failures. Transfer may also be modelled to FT:s of sub-functions of an RPS Protection Function.
Communication VU-APU <sup>1</sup>	The FT models failure in the signal exchange of RPS Protection Functions from APU:s to VU:s, by modelling failure of the communication module and SW failure modes type 2b by basic events and failure of sending APU by transfer to specific APU FT.
Communication APU-APU <sup>1</sup>	The FT models failure in the signal exchange of Process Measurement values between specific APU:s, by modelling failure of the communication module and SW failure modes type 2b by basic events and failure of sending APU by transfer to specific APU FT.
Process measurement <sup>1</sup>	The FT models failure in the Process Measurements and the acquisition of these signals via Input Modules. Failure of sensors is modelled by basic events and failure of Input Module by transfer to specific FT.
Acquisition & processing unit, APU <sup>1</sup>	The FT models failure of APU processor and subrack by basic events, SW failure modes type 3a and voltage supply failure by a FT transfer.
Voting unit, VU <sup>1</sup>	The FT models failure of VU processor and subrack by basic events, SW failure modes type 3b and voltage supply failure by a FT transfer.
Input module <sup>1</sup>	The FT models failure of Input Module by basic events

<sup>1</sup> Separate FT:s for latent and detected failures in order to account for effects of default values.

<sup>2</sup> One FT per division and RPS Actuation Signal or Protection Function.

Based on the taxonomy developed in section 6.1 and the safety I&C protection functions and fault tolerant design defined in Appendix A, the fault tree model of the example PSA with digital I&C has been developed by applying the fault tree structure of Table 14. The main tasks of the procedure (in a bottom-up perspective) are:

- Grouping of module failures into modelling blocks taking into account:
  - Possible failure modes
  - Possible default values at detected failure.
- Allocation of modelling blocks for each specific RPS safety protection functions (Table A-3) with regard to

- Failure mode of the function
- The consequence of applied default values at detected failure
- Type of voting logic.
- Allocation of modelling blocks for each specific RPS actuation signal (Table A-2) with regard to
  - Failure mode of the actuation signal
  - The consequence of applied default values at detected failure
  - Type of voting logic.
- Allocation of modelling blocks for each actuator with regard to
  - Failure mode of the actuator
  - Fail-safe state of the actuator.

The reliability model has been developed with a somewhat expanded fault tree structure in order to increase the flexibility and to make it possible to evaluate different modelling aspects. The model of the digital I&C currently consists of 485 fault trees pages, 329 basic events and 82 hardware CCF groups. Software faults are modelled with a total of 43 CCF basic events. The developed I&C model follows a generic coding system for fault trees and events. See appendix B for an example of structuring the fault tree into pages.

### 7.4.3 FinPSA model structure

FinPSA model is otherwise similar to RiskSpectrum model except that I&C systems are modelled using I&C modelling feature of FinPSA (Niemelä 2012). In FinPSA, I&C model is built using success logic instead of failure logic. The system is described as a communication network so that each line of the model code represents a simple dependency structure: the element of the left hand side of the equation needs the elements of the right hand side of the equation to function. The model is written in a text file using operands ‘\*’, ‘+’ and ‘K/N’, which are presented in Table 15, to define the dependencies.

**Table 15.** Operands of the I&C model.

Operand	Example	Possible interpretation
*	$S1 = C1 * C2 * C3$	Signal S1 is TRUE if components C1, C2 and C3 work.
+	$S2 = C1 + C2$	Signal S2 is TRUE if component C1 or C2 works.
K/N	$S3 = <2 C1 + C2 + C3>$	Signal S3 is TRUE if two of components C1, C2 and C3 work.

The I&C model is fully integrated to other PSA model parts. Fault trees contain links to I&C model and I&C model includes links to fault trees. I&C model also uses the same data base as fault trees. When minimal cut sets are generated for the PSA model, the I&C model is automatically transformed into fault trees which are linked to LIC gates in fault trees. This transformation does not increase the calculation time much.

The I&C modelling feature is an alternative and complementary to fault tree modelling. Benefits of I&C modelling are the compact and simple representation and ease of making modifications. Model can also be imported using simple copy and paste. This makes, for example, changing of voting logic simple and efficient.

The I&C model replaces Actuator Signal, RPS actuation signal and RPS protection function fault trees of the RiskSpectrum model (Table 9). The general structure of the I&C model text file is from top to bottom:

1. Inputs to APU voting
2. APU votings
3. Dependencies between signals
4. Inputs to VU voting
5. VU votings
6. Actuation signals.

The I&C model mainly consists of links to fault trees, I&C model elements which are defined in their own I&C model equations and basic events representing software failures. Other basic events appear only in fault trees (with two exceptions).

#### **7.4.4 Comparison of RS and FinPSA results**

The most significant difference between RiskSpectrum and FinPSA is in CCF related probability calculations as programs interpret the basic event probabilities given in the data bases differently. In RiskSpectrum, the basic event probability that is defined in the data base is the total probability that includes both the probabilities of CCFs and the single failure. The single failure probability is obtained when the portion of CCFs is multiplied out. In FinPSA, the basic event probability that is defined in the data base is the probability of the single failure. The total probability is calculated from it by adding the CCF portion. Because of this difference, exactly same probability calculations cannot be performed using these two programs.

The minimal cut set results of RiskSpectrum and FinPSA are very similar. There are only some differences in CCF calculations and truncation of minimal cut sets with small probabilities. Common cause failure groups with over four basic events cannot be modelled in FinPSA except with the beta-factor model. Because of this, FinPSA results are missing some minimal cut sets. In FinPSA, there can be a CCF between a group of basic events only if they all appear in a fault tree. In the analysed case, one of the main feedwater pumps is not in operation, and hence, FinPSA results include only a CCF of the two pumps that are in operation, while RiskSpectrum results include also a CCF of all three pumps.

#### **7.5 Evaluation of the modelling aspects**

The example PSA model has been designed in a dynamic manner to allow major changes of the modelling of different digital I&C aspects. The model changes are mainly performed by the use of boundary condition sets in the consequence analysis cases.

Since the model and the data are fictive, it is not meaningful to draw conclusions from numerical results. The evaluation have instead been made by comparing importance measures such as risk increase factor (RIF), risk decrease factor (RDF) and sensitivity factors, and by qualitative analysis of minimal cut sets (number, rank, why a minimal cut set, which are missing, etc.), for different configurations of design and modelling aspects.

All initiating events as presented in Appendix A (Table A-1) have been analysed, but conclusions are mainly made based on the analysis of the initiating event “Transient” since this event will give the most unbiased results. The other initiating events all have impact on one or more core damage barriers, which will affect the importance of the digital I&C equipment.

The modelling aspects that have been addressed in this project are:

- Hardware failure modes. Relative importance of digital I&C modules and hardware failure modes (detected vs. undetected failures).
- Level of detail. System level vs. I&C unit level vs. module level.
- Default values. Importance of default value modelling.
- Intelligent voting logic. Importance of handling detected faulty input signals in voting logic.
- CCF parameter importance.
- Software failures modes. Relative importance of digital I&C units and software failure modes (detected vs. undetected failures).

The results from the evaluation of these aspects are presented in Appendix B. The evaluation of the example PSA shows that both undetected and detected failures of hardware and software contribute significantly to the PSA result, indifferently of the assumed fault tolerant design.

The results show that the choice of level of abstraction for the modelling of digital I&C is of high importance for the result. Modelling at the I&C unit level can result in large conservatism that may produce misleading results e.g. regarding dominating core damage sequences and significance of I&C failure modes with regard to the plant risk, which in turn may lead to erroneous risk informed decisions.

In the case where spurious signals can occur due to that default values of 1 are applied at detected failures, detected failures can even dominate the contribution from digital I&C to the plant risk. This stresses the importance of not excluding detected failures from the reliability model without thorough investigations.

The microprocessor technology enables flexible treatment of detected faulty input signals. This “intelligent voting” can be cumbersome to model exactly in PSA, and therefore simplifying assumptions are usually made. In the basic case of the example PSA, it is simply assumed that detected faulty inputs are treated either as 1 or 0 depending on the default value. Gustavsson (2012) made a sensitivity study with the earlier version of the example PSA, in which the first faulty input signal (of 2-o-o-4 voting) is ignored. In that example, it had some impact in the result.

For the current example, large number of different possible intelligent voting logics were evaluated with the FinPSA version of the example model (see details in appendix B) and with slightly different software reliability data. The core damage frequency with a simple default 1 logic was 11–13% larger than with intelligent logics. Hence, if a protection system uses an intelligent voting logic, somewhat conservative results can be obtained by performing the modelling according to default 1 logic instead. However, a modeller must be aware that this causes some differences in risk rankings of

components and the differences are especially large with regard to detected CCFs of some I&C components.

Default 0 logic gave only a little larger core damage frequency than the intelligent logics (1–4%). Hence, using it instead of an intelligent logic would not even be very conservative. However, this can be only a property of this example model. In another design, default 0 logic might even give larger core damage frequency than default 1 logic. Evaluation of both default values is therefore recommended to choose which one is better conservative substitute for an intelligent logic. The differences between different intelligent voting logics were very small, less than 3%. Thus, in this example, it does not make much difference which intelligent logic is chosen.

The evaluation results show that the contribution from hardware failures is almost exclusively given by CCF events both for detected and undetected failures which are expected due to the design and redundancy of the digital I&C systems. In the example model the same CCF parameters have been assigned for both detected and undetected failures, which can be questioned. It is often argued that the likelihood of CCF for detected failures should be smaller than for undetected failures. The results from the sensitivity analysis show that lowering the CCF parameters of the detected failures have a significant impact on the CDF. Detected failures still have a significantly higher fractional contribution compared to undetected failures, i.e. by a factor 2.5. Once again, this stresses the importance of not excluding detected failures from the reliability model. Also relevant CCF parameters are of interest in order to achieve a relevant result.

SW faults have a non-negligible effect on the results due to their functional impact on all divisions — one or more safety functions can be lost. Therefore attention needs to be paid to the quantification of software faults and the assessment of the degree of diversity between the subsystems of the reactor protection system.

The received results are based on the specific design of the example plant and example I&C system and also the assumed failure data of the digital I&C and assumed CCF parameters. The results of this study should therefore not directly be generalised to other designs.



## 8. Conclusions

Due to the many unique attributes of digital systems, a number of modelling and data collection challenges exist, and consensus has not yet been reached. Currently in PSA computer-based systems are usually analysed using simple approaches with a primary goal to model dependencies. There is a general consensus that protection systems shall be included in PSA, while control systems can be treated in a limited manner.

The objective of OECD/NEA WGRISK DIGREL task was to develop a failure mode taxonomy for reliability assessment of digital I&C systems for use in PSA. The proposed failure modes taxonomy was developed by first collecting examples of taxonomies provided by the task group organisations. This material showed some variety in the handling of I&C hardware failure modes, depending on the context where the failure modes have been defined. Regarding the software part of I&C, failure modes defined in NPP PSAs have been simple — typically a software CCF failing identical processing units.

The failure modes taxonomy is based on a failure propagation model and the hierarchical definition of different levels of abstraction. To handle complexity, at the level of system, division and I&C units, failure modes are considered as much as possible only from the functional point of view. No significant distinction is made between hardware or software aspects at these levels. At the module and basic component levels, the taxonomy differentiates between hardware and software related failure modes.

The failure propagation is described using a failure model. Five important elements of the failure model stand out, on which the taxonomy focuses: fault location, failure mode, uncovering situation, failure effect and end effect. These concepts are applied in particular to define the relationship between fault in hardware or software modules (module level failure modes) and the effect on I&C units (I&C unit level failure modes).

The purpose of the taxonomy is to support PSA, and therefore focus was placed on high level functional aspects rather than low level structural aspects. This focus allows handling of the variability of failure modes and mechanisms of I&C components. It reduces the difficulties associated with the complex structural aspects of software in redundant distributed systems. This taxonomy report can be seen as a step of towards more harmonised approach to analyse and model digital I&C in PSA.

The evaluation of the example PSA demonstrated the developed taxonomy and verified that it is suitable for PSA purpose. The evaluation shows that the choice of the level of abstraction for the modelling of digital I&C is of high importance for the results. The most suitable level of abstraction is found to be the “module level” which concurs with the level of abstraction of the general PSA state of the art. The module level makes it feasible to perform, maintain and review a PSA of digital I&C with reasonable resources while capturing critical dependencies. It is also possible to capture fault tolerant features of the digital system and the safety functions’ impact on the reliability. Modelling on the I&C unit level of abstraction can result in large conservatisms that may produce misleading results, e.g., regarding dominating core damage sequences and

significance of I&C failure modes with regard to the plant risk, which in turn may lead to erroneous risk informed decisions.

The evaluation of the example PSA also shows that both undetected and detected hardware and software failures contribute significantly to the PSA results, indifferently of the assumed fault tolerant design. This stresses the importance of not excluding detected failures from the reliability model without thorough investigations. Similar conclusion can be drawn from the test of using different CCF parameters for undetected and detected failures.

Software faults have a non-negligible effect on the results due to their functional impact on all divisions — one or more safety functions can be lost. Therefore attention needs to be paid to the quantification of software faults and the assessment of the degree of diversity between the subsystems of the reactor protection system.

The received results are based on the specific design of the example plant and example I&C system and also the assumed failure data of the digital I&C and assumed CCF parameters. The results of this study should therefore not directly be generalised to other designs. Differences in conclusions may of course be found for different designs and failure data.

In order to develop a realistic fault tree model for a digital I&C protection system it is vital that the chosen fault tolerant design is fully understood and correctly described in the model. The treatment of faulty inputs and degraded voting logic sets the foundation of the fault tree analysis. In general, modelling of digital I&C significantly increases the effort of failure mode analysis, dependency analysis and fault tree modelling. The amount of resource involved in such a task should not be underestimated, neither should the task of quality assurance.

As a result of DIGREL, there is a good understanding of sufficient level of details for PSA modelling and a proposal for the treatment of software failures. DIGREL's scope is, however, limited to a simple reactor protection system architecture, and, also human factor aspects have been out of the scope. A number of relevant issues has been identified for future research and development:

- consensus guidelines for failure mode and effects analysis (FMEA) for digital I&C (hardware and software)
- collaborative efforts to collect failure data, including hardware, software and common cause failure data
- validation of the method for the software reliability quantification
- development of an approach to analyse systematically and comprehensively spurious actuations
- development of an approach to analyse common cause initiators
- handling of control systems
- analysis of human errors (pre-initiator, initiator, post-initiator) related to digital I&C
- assessment of defence-in-depth and diversity and complexity
- integration of risk-informed assessments by PSA in the safety justification framework for digital I&C
- analysis of area events (fire and flooding) for digital I&C systems

- analysis and modelling of impact from preventive maintenance in assessment of defence-in-depth and diversity and complexity
- handling of field-programmable gate array (FPGA) technology
- comparisons of design alternatives (impact of different diversities, intelligent voting logics, etc.)
- tools (templates) for analysis and review of PSA with digital I&C systems
- pilot studies with a plant-specific PSA.

## 9. References

- Aldemir, T., Guarro, S., Kirschenbaum, J., Mandelli, D., Mangan, L.A., Bucci, P., Yau, M., Johnson, B., Elks, C., Ekici, E., Stovsky, M.P. Miller, D.W., Sun, X., Arndt, S.A., Nguyen, T. & Dion, J. 2009. A Benchmark Implementation of Two Dynamic Methodologies for the Reliability Modeling of Digital Instrumentation and Control Systems. NUREG/CR-6985, United States Nuclear Regulatory Commission, Washington D.C.
- Authén, S., Björkman, K., Holmberg, J.-E. & Larsson, J. 2010a. Guidelines for reliability analysis of digital systems in PSA context — Phase 1 Status Report. NKS-230 Nordic nuclear safety research (NKS), Roskilde.
- Authén, S., Wallgren, E. & Eriksson, S. 2010b. Development of the Ringhals 1 PSA with Regard to the Implementation of a Digital Reactor Protection System. In: Proc. 10th International Probabilistic Safety Assessment & Management Conference, PSAM 10, Seattle, Washington, June 7–11, 2010, paper 213.
- Authén, S., Gustafsson, J. & Holmberg, J.-E. 2012. Guidelines for reliability analysis of digital systems in PSA context — Phase 2 Status Report. NKS-261 Nordic nuclear safety research (NKS), Roskilde.
- Authén, S. & Holmberg, J.-E. 2013. Guidelines for reliability analysis of digital systems in PSA context — Phase 3 Status Report. NKS-277, Nordic nuclear safety research (NKS), Roskilde.
- Authén, S., Holmberg, J.-E., Lanner, L. & Tyrväinen, T. 2014. Guidelines for reliability analysis of digital systems in PSA context — Phase 4 Status Report. NKS-302, Nordic nuclear safety research (NKS), Roskilde.
- Bingham, S. & Lach, J. 2009. Exhaustive Integrated Circuit Fault Coverage Analysis Using Formal Methods. In Proc. of Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5–9, 2009.
- Björkman, K., Bäckström, O. & Holmberg, J.-E. 2012. Use of IEC 61508 in Nuclear Applications Regarding Software Reliability — Pre-study. VTT-R-09293-11, VTT, Espoo.
- Blanchard, D. & Torok, R. 2010. Risk Insights Associated with Digital Upgrades, Proc. 10th International Probabilistic Safety Assessment & Management Conference, PSAM 10, Seattle, Washington, June 7–11, 2010, paper 453.
- Bouissou, M. 2002. Boolean logic driven Markov processes: A powerful new formalism for specifying and solving very large Markov models. In: Proceedings of the 6th International Conference on Probabilistic Safety Assessment & Management, 23–28 June 2002, San Juan, Puerto Rico, USA.

Bozzano, M. & Villafiorita, A. 2007. The FSAP/NuSMV-SA Safety Analysis Platform, *International Journal on Software Tools for Technology Transfer* 9(2007)5–24.

Bucci, P., Kirschenbaum, J., Mangan, L. A., Aldemir, T., Smith, C. & Wood, T. 2008. Construction of event-tree/fault-tree models from a Markov approach to dynamic system reliability, *Reliability Engineering & System Safety* 93(2008)1616–1627.

Bäckström, O., Holmberg, J.-E., Jockenhövel-Barttfeld, M., Porthin, M. & Taurines, A. 2014. Software reliability analysis for PSA. NKS-304, Nordic nuclear safety research (NKS), Roskilde.

Bäckström, O., Holmberg, J.-E., Jockenhövel-Barttfeld, M., Porthin, M. & Taurines, A. 2015. Software reliability analysis for PSA — Final report. NKS-xxx, Nordic nuclear safety research (NKS), Roskilde.

Cetiner, S.M., Korsah, K. & Muhlheim, M.D. 2009. Survey on Failure Modes and Failure Mechanisms in Digital Components and Systems, *Proc.6th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009*, Knoxville, Tennessee, April 5–9, 2009.

Chu, T.L., Martinez-Guridi, G., Yue, M. & Lehner, J. 2006. A review of software induced failure experience. 5th NPIC HMIT meeting, November 2006, BNL-NUREG-77124-2006-CP.

Chu, T.L., Martinez-Guridi, G., Yue, M., Lehner, J. & Samanta, P. 2008. Traditional Probabilistic Risk Assessment Methods for Digital Systems, NUREG/CR-6962, United States Nuclear Regulatory Commission, Washington D.C.

Chu, T.L., Yue, M., Martinez-Guridi, G., Mernick, K., Lehner, J. & Kuritzky, A. 2009. Modeling a Digital Feedwater Control System Using Traditional Probabilistic Risk Assessment Methods, NUREG/CR-6997 BNL-NUREG-90315-2009, United States Nuclear Regulatory Commission, Washington D.C.

Chu, T.-L., Yue, M., Martinez-Guridi, G. & Lehner, J. 2010a. Review of Quantitative Software Reliability Methods, Brookhaven National Laboratory Letter Report, Digital System Software PRA JCN N-6725.

Chu, T.-L., Yue, M., Martinez-Guridi, G. & Lehner, J. 2010b. A Generic Failure Modes and Effects Analysis (FMEA) Approach for Reliability Modeling of Digital Instrumentation and Control (I&C) Systems. In: *Proc. of 10th International Probabilistic Safety Assessment & Management Conference, PSAM 10*, Seattle, Washington, June 7–11, 2010, paper 82.

Chu, T.-L., Martinez-Guridi, G., Yue, M., Samanta, P., Vinod, G. & Lehner, J. 2010c. Establishing a Philosophical Basis for Probabilistic Modeling of Software Failures, In *Proc. of 10th International Probabilistic Safety Assessment & Management Conference, PSAM 10*, Seattle, Washington, June 7–11, 2010, paper 84.

Clarke Jr., E.M., Grumberg, O. & Peled, D.A. 1999. Model Checking, The MIT Press, Cambridge, Massachusetts.

Dahll, G., Liwång, B. & Pulkkinen, U. 2007. Software-Based System Reliability, Technical Note, NEA/SEN/SIN/WGRISK(2007)1, Working Group on Risk Assessment (WGRISK) of the Nuclear Energy Agency, Paris.

Doguc, O. & Ramirez-Marquez, J. E. 2009. A generic method for estimating system reliability using Bayesian networks, Reliability Engineering & System Safety, 94(2009)542–550.

Enzinna, B., Shi, L. & Yang, S. 2009. Software Common-Cause Failure Probability Assessment. In: Proc. of Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5–9, 2009.

Eom, H.-S., Park, G.-Y., Kang, H.-G. & Jang, S.-C. 2009. Reliability Assessment Of A Safety-Critical Software By Using Generalized Bayesian Nets. In: Proc. of Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5–9, 2009

EPRI. 2010. Estimating Failure Rates in Highly Reliable Digital Systems. EPRI TR-1021077, Electric Power Research Institute, Inc., Palo Alto, CA. Limited distribution.

Garrett, C.J, Guarro, S.B. & Apostolakis, G.E. 1995. The Dynamic Flowgraph Methodology for Assessing the Dependability of Embedded Software Systems, IEEE Trans. on Systems, Man and Cybernetics 25(1995)824–840.

Garrett, C. J. & Apostolakis, G.E. 2002. Automated hazard analysis of digital control systems, Reliability Engineering & System Safety 77(2002)1–17.

Guarro, S. 2010. Risk-Informed Safety Assurance and Probabilistic Assessment of Mission-Critical Software-Intensive Systems, NASA Technical Paper AR 07-01, JSC-CN-19704.

Gustafsson, J. 2012. Reliability analysis of digital protection system of a nuclear power plant, Master's Thesis. KTH, Stockholm.

Haapanen, P. & Helminen, A. 2002. Failure mode and effects analysis of software-based automation systems. STUK-YTO-TR 190. STUK, Helsinki.

IAEA. 2010. Development and application of level 1 probabilistic safety assessment for nuclear power plants for protecting people and the environment, IAEA Specific Safety Guide No. SSG-3, International Atomic Energy Agency. Vienna.

IAEA. 2011. Core Knowledge on Instrumentation and Control Systems in Nuclear Power Plants, IAEA Nuclear Energy Series No. NP-T-3.12, International Atomic Energy Agency, Vienna.

IEC. 2006. Analysis techniques for system reliability, Procedure for failure mode and effects analysis (FMEA). IEC 60812, ed. 2.0. International Electrotechnical Commission (IEC), Geneva.

IEC. 2009. Nuclear power plants. Instrumentation and control important to safety. Classification of instrumentation and control functions. IEC 61226, ed. 3.0. International Electrotechnical Commission, Geneva.

IEC. 2010a. Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 4: Definitions and abbreviations. IEC 61508-4, ed. 2.0. International Electrotechnical Commission, Geneva.

IEC. 2010b. Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 6: Guidelines on the application of IEC 61508-2 and IEC 61508-3, IEC 61508-6, ed. 2.0. International Electrotechnical Commission, Geneva.

IEEE. 1987. IEEE Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems, IEEE Std. 352, Institute of Electrical and Electronics Engineers, Inc., New York.

ISO/IEC. 2005. Software engineering — Software product quality requirements and evaluation (SQuaRE) — Guide to SQuaRE, ISO/IEC 25000:2005. International Electrotechnical Commission, Geneva.

ISO/IEC/IEEE. 2010. Systems and software engineering – Vocabulary, ISO/IEC/IEEE 24765:2010. International Electrotechnical Commission, Geneva.

Kang, H.G. & Jang, S.-C. 2009. Issues And Research Status For Static Risk Modeling Of Digitalized Nuclear Power Plants, Proc.6th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5–9, 2009.

Kelly, D. L. & Smith, C. L. 2009. Bayesian inference in probabilistic risk assessment — The current state of the art, Reliability Engineering & System Safety, 94(2009)628–643.

Kisner, R., Mullens, J., Wilson, T., Wood, R., Korsah, K., Qualls, A., Muhlheim, M., Holcomb, D. & Loebel, A. 2007. Safety and Non-Safety Communications and Interactions in International Nuclear Power Plants, Guidelines for the Design of Highly Integrated Control Rooms, ORNL/NRC/LTR-07/05, Oak Ridge Laboratory, Oak Ridge.

Koh, K. Y. & Seong, P. H. 2009. SACS2. A Dynamic and Formal Approach to Safety Analysis for Complex Safety Critical Systems. In: Sixth American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5–9, 2009.

Korsah, K., Holcomb, D.E., Muhlheim, M.D., Mullens, J.A., Loebel, A., Bobrek, M., Howlader, M.K., Killougha, S.M., Moore, M.R., Ewing, P.D., Sharpe, M., Shourbaji, A.A., Cetiner, S.M., Wilson, Jr., T.L. & Kisner, R.A. 2009. Instrumentation and

Controls in Nuclear Power Plants: An Emerging Technologies Update, NUREG/CR-6992, United States Nuclear Regulatory Commission, Washington D.C.

Kwiatkowska, M., Norman, G. & Parker, D. 2009. PRISM: Probabilistic Model Checking for Performance and Reliability Analysis. ACM SIGMETRICS Performance Evaluation Review, 36(4), pages 40–45.

Labeau, P.E., Smidts, C., Swaminathan, S. 2000. Dynamic reliability: towards an integrated platform for probabilistic risk assessment, Reliability Engineering & System Safety 68(2000)219–254.

Marshall, F.M., Rasmuson, D.M., Mosleh, A. Common-Cause Failure Parameter Estimation, NUREG/CR-5497, U.S. Nuclear Regulatory Commission, Washington D.C., 1988.

Musa, J.D. & Okumoto, K. 1984. A Logarithmic Poisson Execution Time Model for Software Reliability Measurement, Proceedings of Seventh International Conference on Software Engineering, 230-238, Orlando, FL, 1984.

Niemelä, I. 2012. Isolation of I&C model from PRA fault tree model. In: Proc. of 11th International Probabilistic Safety Assessment and Management Conference & The Annual European Safety and Reliability Conference, Helsinki, 25–29.6.2012, 10-We3-4.

Ortmeier, F., Schellhorn, G., Thums, A., Reif, W., Hering, B. & Trappschuh, H. 2003. Safety analysis of the height control system for the Elbtunnel, Reliability Engineering & System Safety, 81(2003)259–268.

OECD. 2009. Recommendations on assessing digital system reliability in probabilistic risk assessments of nuclear power plants, NEA/CSNI/R(2009)18, OECD/NEA/CSNI, Paris.

OECD. 2014. Failure modes taxonomy for reliability assessment of digital I&C systems for PRA, report prepared by a task group of OECD/NEA Working Group RISK, NEA/CSNI/R(2014)16, OECD/NEA/CSNI, Paris.

Pearl, J. 1988. Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann Publishers, San Mateo, CA.

Porthin, M. & Holmberg, J-E. 2013. Modelling software failures using Bayesian nets, VTT Research Report VTT-R-08279-12, VTT, Espoo.

Schneidewind, N.F. & Keller, T.W. 1992. Applying Reliability Models to the Space Shuttle, IEEE Software, Vol. 9(4) 28–33.

Sedlak, J. 2009. Software critical for safety in reliability models, In: Proc. of European Safety and Reliability (ESREL) Conference, ESREL 2009, Prague, September 7–10, 2009.



Shi, L., Enzinna, R., Yang, S. and Blodgett, S. 2010. Probabilistic Risk Assessments of Digital I&C in Nuclear Power Plant, In: Proc. 10th International Probabilistic Safety Assessment & Management Conference, PSAM 10, Seattle, Washington, June 7–11, 2010, paper 173.

Siemens. 2004. Failure Rates of Components. SN 29500. Siemens AG, Munich, Germany.

Smidts, C. & Li, M. 2000. Software Engineering Measures for Predicting Software Reliability in Safety Critical Digital Systems, NUREG/GR-0019, U.S. Regulatory Commission, Washington D.C.

Smidts, C. & Li, M. 2004. Preliminary Validation of a Methodology for Assessing Software Quality, NUREG/CR-6848, U.S. Regulatory Commission, Washington D.C.

Smith, D.J. & Simpson, K.G.L. 2010. Safety Critical Systems Handbook Safety Critical Systems Handbook. A Straightforward Guide to Functional Safety: IEC 61508 and Related Standards Including: Process IEC 61511, Machinery IEC 62061 and ISO 13849, 3rd edition.

SSM. 2010. Licensing of safety critical software for nuclear reactors — Common position of seven European nuclear regulators and authorized technical support organisations, SSM Report 2010:01, SSM, Stockholm.

Thuy, N. & Deleuze, G. 2009. A Mixed Approach to Assess the Impact of I&C in PSA. In: Proc.6th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC&HMIT 2009, Knoxville, Tennessee, April 5-9, 2009.

US DOD. 1984. Procedures for performing a failure mode, effects and criticality analysis, MIL STD 1629A, US Department of Defense, Washington D.C.

US DOD. 1995. Reliability Prediction of Electronic Equipment, Notice 2. MIL-HDBK-217F(2), US Department of Defense, Washington D.C.

Vesely, W., Stamatelatos, M., Dugan, J., Fragola, J., Minarick III, J. & Railsback, J. 2002. Fault Tree Handbook with Aerospace Applications, NASA Headquarters, Washington, D.C.

Yau, M., Guarro, S. & Apostolakis, G. 1995. Demonstration of the Dynamic Flowgraph Methodology using the Titan II Space Launch Vehicle Digital Flight Control System, Reliability Engineering & System Safety 49(1995)335–353.

Yau, M. & Guarro, S. 2010. Application of Context-based Software Risk Model (CSRM) to Assess Software Risk Contribution in Constellation Project PRAs. In: 10th International Probabilistic Safety Assessment & Management Conference, PSAM 10, Seattle, Washington, June 7–11, 2010, paper 186

## **Appendix A. Description of the example system**

### **Overview of the front-line safety systems**

The example PSA-model represents a fictive boiling water reactor (BWR), which has four-redundant safety systems. The example model includes the following systems:

- ACP – AC power system
- ADS – Automatic depressurisation system
- CCW – Component cooling water system
- ECC – Emergency core cooling system
- EFW – Emergency feedwater system
- FCV – Filtered containment venting system
- HVA – Heating, venting and air conditioning system
- MFW – Main feedwater system
- RHR – Residual heat removal system
- RSS – Reactor scram system
- SWS – Service water system.

Figure A-1 and A-2 show a simplified flow diagram and line diagram related to the safety systems relevant to the example. It should be noted that this example must not be interpreted as a representative boiling water reactor, but rather as an example for demonstrating the reliability analysis of representative nuclear safety I&C.

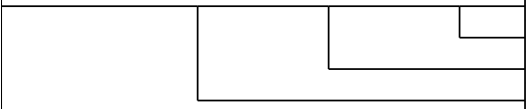


Four initiating events are considered, see Table A-1. Depending on the initiating event there are different success criteria for the front line safety systems.

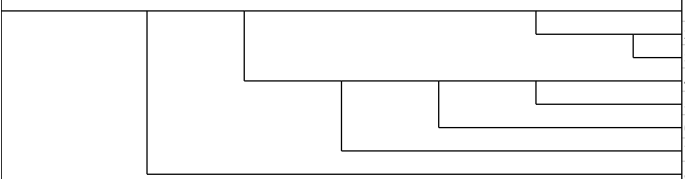
**Table A-1.** Front line safety system success criteria.

Initiating event	MFW	EFW	ADS	ECC	RHR
ALOCA – Large Loca	No credit	No credit	Not required	1oo4	1oo4
LMFW – Loss of main feedwater	No credit	1oo4	4oo8	1oo4	1oo4
LOOP – Loss of offsite power	2oo3	1oo4	4oo8	1oo4	1oo4
TRAN – General transient	2oo3	1oo4	4oo8	1oo4	1oo4

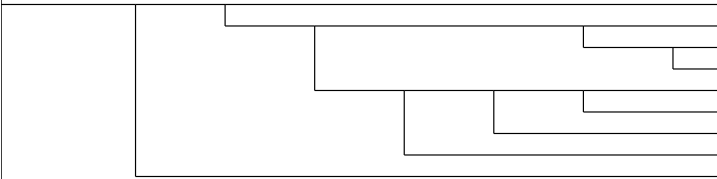
Event trees are shown in Figures A-3 to A-6. Consequence “CD” refer to core damage, “CD1” to core damage due to failed reactivity control, “CD2” to core damage due to failed core cooling”, and “CD3” to core damage due to failed residual heat removal.

Large Loss Of Coolant Accident	Reactor Scram	Emergency Core Cooling	Residual Heat Removal				
ALOCA	C	V	W1	No.	Freq.	Conseq.	Code
				1		OK	
				2		CD,CD3	W1
				3		CD,CD2	V
				4		CD,CD1	C

**Figure A-3.** Event tree for large LOCA.

Loss of Main Feed Water	Reactor Scram	Emergency Feed Water	Depressurization	Emergency Core Cooling	Residual Heat Removal	Filtered Containment Venting				
LMFW	C	U	X	V	W1	W3	No.	Freq.	Conseq.	Code
							1		OK	
							2		OK	W1
							3		CD,CD3	W1-W3
							4		OK	U
							5		CD,CD3	U-W1
							6		CD,CD2	U-V
							7		CD,CD2	U-X
							8		CD,CD1	C

**Figure A-4.** Event tree for Loss of main feedwater.

Loss Of Offsite Power	Reactor Scram	Main Feedwater	Emergency Feed Water	Depressurization	Emergency Core Cooling	Residual Heat Removal	Filtered Containment Venting				
LOOP	C	Q	U	X	V	W1	W3	No.	Freq.	Conseq.	Code
								1		OK	
								2		OK	Q
								3		OK	Q-W1
								4		CD,CD3	Q-W1-W3
								5		OK	Q-U
								6		CD,CD3	Q-U-W1
								7		CD,CD2	Q-U-V
								8		CD,CD2	Q-U-X
								9		CD,CD1	C

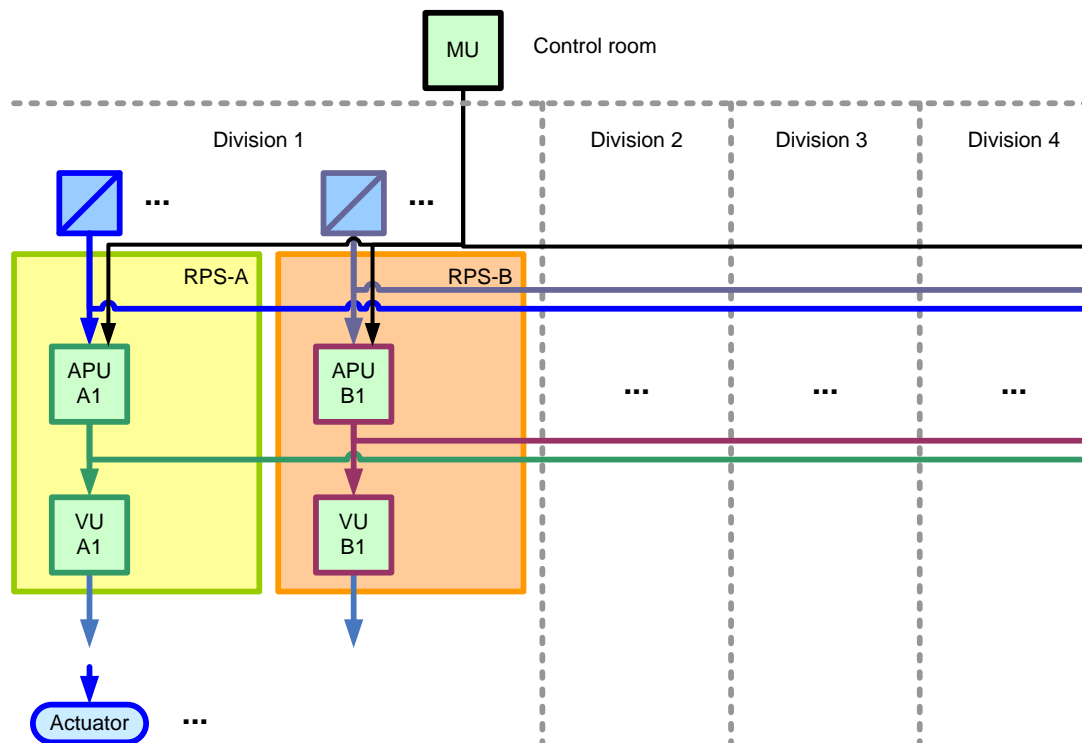
**Figure A-5.** Event tree for Loss of offsite power.

Transient	Reactor Scram	Main Feedwater	Emergency Feed Water	Depressurization	Emergency Core Cooling	Residual Heat Removal	Filtered Containment Venting				
T	C	Q	U	X	V	W1	W3	No.	Freq.	Conseq.	Code
								1		OK	
								2		OK	Q
								3		OK	Q-W1
								4		CD,CD3	Q-W1-W3
								5		OK	Q-U
								6		CD,CD3	Q-U-W1
								7		CD,CD2	Q-U-V
								8		CD,CD2	Q-U-X
								9		CD,CD1	C

**Figure A-6.** Event tree for General transient.

### Safety I&C architecture and fault tolerant design

The architecture of the safety I&C is presented in Figure A-3. The protection system is divided into two subsystems, called RPS-A and RPS-B. In addition to the APU:s and VU:s, the I&C architecture includes an I&C unit for operator actions, abbreviated by MU. This I&C unit is relevant for the manual actuation of the primary circuit depressurization and manual actuation signal of main feedwater pumps.



**Figure A-7.** I&C architecture.

The example PSA digital I&C protection system is designed with fault tolerant features (fault tolerant design), which provides means to detect failures and mark faulty signals, e.g. self-surveillance, dynamic self-test, open circuit monitoring, cross channel comparison etc. Fault processing is implemented in the design of the hardware circuits and the software logic, and it can be defined on a case-by-case basis how the logic shall react if invalid input signals are present, and how output signals shall be set in case of faulty logic signals. In general, the following applies for detected failures of the example I&C protection system:

- Detected failure in input signals, in intra I&C unit signal processing or in inter I&C unit signal exchange will cause corresponding signals to be replaced by a default value of 0 or 1.
- Complete, or fatal, failure of an I&C unit, e.g. processor failure or power supply failure, will cause all output channels of the I&C unit to 0 and controlled actuators will go to the predefined fail-safe state.

There are different solutions for voting applied in the safety I&C system for actuation signals to the actuators:

- Hardwired 2/4 voting by relays or pilot valves (e.g. scram).
- Software 2/4 voting performed in VU:s and APUs with possible treatment of degraded voting logic as considered in Section Modelling intelligent voting logics in Appendix B.

The fail-safe actions are separately defined for each I&C function and for each actuation signal. I&C functions using the same inputs, may apply different default values and different types of voting logic.

### **Safety I&C protection functions**

The general principle is that the EFW is controlled by the RPS-B and the ECC and ADS are controlled by the RPS-A. Pumps and valves in the respective system have same actuation signals. Also the support systems needed for cooling of the systems have same actuation signals.

In case of loss of feedwater transient, the normal consequence is the reactor scram actuated e.g. by the protection signal on low level in reactor pressure vessel (signal ID SS04), which is actuated both by the RPS-A (ASS04) and the RPS-B (BSS04). BSS04 will also actuate the EFW by starting the pump and opening the valve for the emergency feedwater injection.

If the emergency feedwater injection fails, the extreme low level protection signal will actuate (signal ID I002), also both by the RPS-A (AI002) and the RPS-B (BI002). I002 will in turn actuate the containment isolation protection signal I000, which is the start signal of the ECC (AI000). On the other hand BI000 is a secondary start signal for the EFW, if BSS04 has failed.

ECC will not be able to inject water to the RPV without depressurization of the primary circuit. The pressure relief valves of the ADS are actuated by the protection signal ATB00. ATB00 requires two sub-conditions to be actuated ATB01 and ATB02. The relief valves are actuated by solenoid valves which receive actuation signals from APU:s. Each APU controls two ADS valve lines.

**Table A-2.** Actuators and their actuation signals ( $i$  = division 1, 2, 3 or 4).

System	Actuator	Control	Condition for control type	VU Signal ID <sup>1</sup>	APU Signal ID <sup>1</sup>	DFLT <sup>2</sup>
ACP	Diesel generator	Start	Reactor scram due to containment isolation or low voltage in respective bus bar	AACP1 + BACP1	ASS12 + AZ00 <i>i</i> + BZ00 <i>i</i>	0
		Stop	Manual stop and not active start signal	AACP2 + BACP2	NOT(ASS12 + AZ00 <i>i</i> + BZ00 <i>i</i> ) * MAN-0iDG01 <sup>3</sup>	1
ADS	Pressure relief valve	Open	Depressurisation signal	–	AADS1 {ATB0}	0
		Close	Manual close and not active depressurisation signal	–	AADS2 {NOT(ATB00) * MAN-ADS <i>j</i> , $j = 1-8$ }	1
CCW	Pump	Start	Reactor scram or high temperature in the condensation pool	ACCW1	ASS00 + AX003	0
		Stop	Manual stop and not active start signal	ACCW2	NOT(ASS00 + AX003) * MAN-CCW0iPM01 <sup>3</sup>	1
ECC	Pump	Start	Containment isolation and no water leakage in the respective pump room	AECC1	NOT(AH00 <i>i</i> ) * AI000	0
		Stop	Water leakage in the respective pump room	AECC2	AH00 <i>i</i>	0
ECC	Motor-operated valve	Open	Containment isolation and no water leakage in the respective pump room	AECC1	NOT(AH00 <i>i</i> ) * AI000	0
		Close	Water leakage in the respective pump room	AECC2	AH00 <i>i</i>	0
EFW	Pump	Start	Feedwater system isolation, reactor scram due to low water level in reactor or containment isolation and no water leakage in the respective pump room	BEFW1	NOT(BH00 <i>i</i> ) * (BSS04 + BI000)	0
		Stop	Water leakage in the respective pump room	BEFW2	BH00 <i>i</i>	1
EFW	Motor-operated valve	Open	Reactor scram due to low water level in reactor, diverse low water level condition or very low water level condition and no water leakage in the respective pump room	BEFW3	NOT(BH00 <i>i</i> ) * (BSS04 + BX001 + BI002)	0
		Close	Water leakage in the respective pump room or very high water level in reactor	BEFW4	BH00 <i>i</i> + BSS05	1
HVA	AC cooler	Start	Start EFW	BEFW1	NOT(BH00 <i>i</i> ) * (BSS04 + BI000)	0
		Stop	Manually	BHVA1	BH00 <i>i</i> + MAN-HVA0iAC01 <sup>3</sup>	1
MFW	Pump	Start	Manual start and not active stop signal	AMFW1	NOT(AM000 + ASS05) * MAN-MFW <i>i</i> , $i = 1, 2, 3$	0
		Stop	Feedwater system isolation or very high water level in reactor	AMFW2	AM005 + ASS05	1
RHR	Pump	Start	Reactor scram or high temperature in the condensation pool and no water leakage in the respective pump room	ARHR1	ASS00 + AX003	0
		Stop	Manual stop and not active start signal	ARHR2	NOT(ASS00 + AX003) * MAN-RHR0iPM01	0
RHR	Motor-operated valve	Open	Reactor scram or high temperature in the condensation pool and no water leakage in the respective pump room	ARHR1	ASS00 + AX003	0
		Close	Manual stop and not active start signal	ARHR2	NOT(ASS00 + AX003) * MAN-RHR0iVM02	0
SWS	Pump	Start	Reactor scram or high temperature in the condensation pool	ARHR1	ASS00 + AX003	0
		Stop	Manual stop and not active start signal	ARHR2	NOT(ASS00 + AX003) * MAN-RHR0iVM02	0

System	Actuator	Control	Condition for control type	VU Signal ID <sup>1</sup>	APU Signal ID <sup>1</sup>	DFLT <sup>2</sup>
RSS	Control rods (solenoid valves)	Open	Reactor scram	–	ASS {ASS00} + BSS {BSS00}	1

<sup>1</sup> Fictive IDs used as identifiers in the coding of elements in the PSA model

<sup>2</sup> Default value applied at the loss of VU

<sup>3</sup> DFLT 0 is applied for manual signal. Manual signal is only modelled for ADS closure



**Table A-3.** RPS-A and -B safety functions (*i* = division 1, 2, 3 or 4).

Signal	Description	Condition <sup>1</sup>	DFLT <sup>2</sup>
<b>RPS-A</b>			
AH00 <i>i</i>	Isolation of the ECC pump room <i>i</i>	ECC <i>i</i> 0CL001-H1 + ECC <i>i</i> 0CL002-H1	1
AI000	Containment isolation	2/4*(AI002- <i>i</i> + AI005- <i>i</i> )	1
AI002	Containment isolation due to extremely low level in RPV	2/4*(RPV <i>i</i> 0CL002-L4)	1
AI005	Isolation due to high pressure in containment	2/4*(RCO <i>i</i> 0CP001-H1)	1
AM000	Feedwater isolation	2/4*(AM005- <i>i</i> )	1
AM005	Feedwater isolation due to high temperature in feedwater system compartment	2/4*(MFW <i>i</i> 0CT001-H1)	1
ASS00	Reactor scram	2/4*(ASS04- <i>i</i> + ASS05- <i>i</i> + ASS12- <i>i</i> + ASS13- <i>i</i> )	1
ASS04	Reactor scram due to low water level in RPV	2/4*(RPV <i>i</i> 0CL001-L2)	1
ASS05	Reactor scram due to high water level in RPV	2/4*(RPV <i>i</i> 0CL001-H2)	1
ASS12	Reactor scram due to containment isolation (I- or M-isolation)	2/4*(AI000- <i>i</i> + AM005- <i>i</i> )	1
ASS13	Low pressure before feedwater pump	2/4*(MFW <i>i</i> 0CP001-L1)	1
ATB00	Depressurisation of the primary circuit	ATB01 * ATB02	0
ATB01	Depressurisation of the primary circuit condition 1: extreme low level in reactor (same as I002)	2/4*(RPV <i>i</i> 0CL002-L4)	0
ATB02	Depressurisation of the primary circuit condition 2: high pressure in containment (same as I005) or manual actuation	ATB03 + 2/4*(RCO <i>i</i> 0CP001-H1)	0
ATB03	Manual TB	MAN-TB	0
AX003	High temperature in condensation pool	2/4*(RCO <i>i</i> 0CT001-H1)	1
AZ00 <i>i</i>	Low voltage in AC bus bar <i>i</i>	ACP <i>i</i> 0CE001-L1	1
<b>RPS-B</b>			
BH00 <i>i</i>	Isolation of the EFW pump room <i>i</i>	EFW <i>i</i> 0CL001-H1 + EFW <i>i</i> 0CL002-H1	1
BI000	Containment isolation	2/4*(BI002- <i>i</i> + BI005- <i>i</i> )	1
BI002	Containment isolation due to extremely low level in RPV	2/4*(RPV <i>i</i> 0CL002-L4)	1
BI005	Isolation due to high pressure in containment	2/4*(RCO <i>i</i> 0CP001-H1)	1
BSS00	Reactor scram	2/4*(BSS04- <i>i</i> + BSS05- <i>i</i> + BSS12- <i>i</i> + BSS13- <i>i</i> )	1
BSS04	Reactor scram due to low water level in RPV	2/4*(RPV <i>i</i> 0CL001-L2)	1
BSS05	Reactor scram due to high water level in RPV	2/4*(RPV <i>i</i> 0CL001-H2)	1
BSS12	Reactor scram due to containment isolation (I- or M-isolation)	2/4*(BI000- <i>i</i> + BM000- <i>i</i> )	1
BX001	Extra low level in RPV	2/4*(RPV <i>i</i> 0CL002-L3)	1
BZ00 <i>i</i>	Low voltage in AC bus bar <i>i</i>	ACP <i>i</i> 0CE001-L1	1

<sup>1</sup> “+” = OR, “\*” = AND, “2/4” = 2-o-o-4

<sup>2</sup> Default value applied by APU at loss of input signal from measurement or other APU:s

RPS-A and RPS-B have partly different input signals but they also share several measurements, see Table A-4.

**Table A-4.** Measurements (*i* = division 1, 2, 3 or 4).

Measurement	Component ID	Limit		Purpose	RPS-A	RPS-B
RPV water level, fine level	RPVi1CL001	L2	Low level	Core cooling protection	ASS04	
	RPVi2CL001	H2	Extra high level	RPV overfilling protection		BSS05
	RPVi2CL001	L2	Low level	Core cooling protection		BSS04
RPV water level, coarse level	RPVi1CL002	L4	Extremely low level	Core cooling protection	AI002 ATB01	
	RPVi2CL002	L3	Extra low level	Core cooling protection		BX001
	RPVi2CL002	L4	Extremely low level	Core cooling protection		BI002
Feedwater system pump suction pressure	MFWi0CP001	L1	Low pressure before feedwater pump	Loss of feedwater supervision		BSS13
Feedwater system room temperature	MFWi0CT001	H1	High room temperature	Leakage supervision		BM005
Containment pressure	RCOi1CP001	H1	High pressure in containment	Leakage supervision	AI005 ATB02	
	RCOi2CP001	H1	High pressure in containment	Leakage supervision		BI005
Condensation pool temperature	RCOi0CT001	H1	High temperature in condensation pool	Residual heat removal	AX003	
Water level in the ECC pump room	ECCi0CL001	H1	Water on the floor	Leakage supervision	AH00 <i>i</i>	
Water level in the EFW pump room	EFWi0CL001	H1	Water on the floor	Leakage supervision		BH00 <i>i</i>
AC power voltage bus bar ACP- <i>i</i>	ACPi1CE001	L1	Low voltage on bus bar ACP- <i>i</i>	Loss of offsite power supervision	AZ00 <i>i</i>	
	ACPi2CE001	L1	Low voltage on bus bar ACP- <i>i</i>	Loss of offsite power supervision		BZ00 <i>i</i>

### Front line safety system failure modes

Table A-5 describes failure modes of the systems EFW, ECC and ADS related to the initiating event LOFW. Support system failure modes are not included in the table. Since EFW and ECC are similar from the failure modes and effects analysis point of view, they are shown in the same lines in this table. I&C failures are further in the next chapter.

**Table A-5.** Failure modes and effects analysis of EFW, ECC and ADS.

System/component ( <i>i</i> = division)	Failure modes	Failure cause	Failure effect
EFW (ECC)	Failure to provide coolant injection		No water to RPV
EFW division <i>i</i> (ECC division <i>i</i> )	Failure to provide coolant injection		EFW (ECC) train <i>i</i> unavailable for coolant injection
EFW <i>i</i> 0PM01 (ECC <i>i</i> 0PM01)	Failure to start Spurious stop	Mechanical failure Power supply I&C failure Component cooling failure Maintenance Alignment error	EFW (ECC) train <i>i</i> unavailable for coolant injection
EFW <i>i</i> 0VM02 (ECC <i>i</i> 0VM02)	Failure to open Spurious closure	Mechanical failure Power supply I&C failure Maintenance Alignment error	Train <i>i</i> unavailable for coolant injection
EFW <i>i</i> 0VC01 (ECC <i>i</i> 0VC01)	Failure to open Spurious closure	Mechanical failure	Train <i>i</i> unavailable for coolant injection
ADS	Failure to depressurize the primary circuit		ECC cannot inject water to RPV
ADS valve line <i>j</i> (8 valve lines)	Failure to open		Valve line unavailable for depressurization
ADS <i>i</i> 0VS01, VS02	Failure to open	Mechanical failure Power supply I&C failure Operator error	Valve line unavailable for depressurization

### I&C system failure modes

The relevant failure modes of I&C can be analysed from the actuator failure modes point of view (see Table A-10). Therefore in practice, the failure modes of RPS-A and RPS-B are either failure on demand or spurious actuation of critical signals for the actuators. For instance, the relevant I&C failure modes related to the pump EFW*i*0PM01 are

- failure to start on BEFW1 signal
  - failure-on-demand to actuate BSS04-signal
  - failure-on-demand to actuate BI000-signal
- spurious stop on BEFW2 signal
  - spurious actuation of BH00*i*-signal.

The next step is to analyse which I&C units can contribute to these failure modes, in other words a failure analysis in the I&C unit level.

### I&C unit failure modes

As an example, the failure modes related to the pump EFW*i*0PM01 are analysed.

Voting units are assumed to fail to provide EFW1 and EFW2 signal if power supply fails or if there is an internal I&C unit failure (i.e. the default value is 0). At detected failure of communication between VU and APU, default values according to Table A-3 will be applied for the EFW1 and EFW2 inputs and cause an activation in an 2-o-o-4

condition. Undetected failure of APU units will fail EFW1 activation in a 3-o-o-4 condition. In case of APU safety functions, detected failures of BI000 and BSS04 input signals from measurements or from other APU:s cause an actuation (i.e. the default value is 1) in an 2-o-o-4 condition. Internal I&C unit failures are analysed in the module level.

**Table A-6.** Failure modes and causes of the I&C units.

Unit	Failure modes	Failure causes
VU	Failure to actuate EFW1 to EFWi0PM001	VU internal failure <ul style="list-style-type: none"> <li>- undetected failure</li> <li>- detected failure</li> </ul> Power supply failure No EFW actuation signal from APU:s (3-o-o-4)
	Spurious stop signal EFW2 to EFWi0PM001	VU failure causing spurious signal <ul style="list-style-type: none"> <li>- detected failure</li> </ul> VU-APU communication link failure <ul style="list-style-type: none"> <li>- detected failure</li> </ul> Spurious stop signal from APU:s (2-o-o-4)
APU	No EFW1 actuation signals from APU	APU internal failure <ul style="list-style-type: none"> <li>- undetected failure</li> </ul> Failure of BI000 and BSS04
	Failure to actuate BI000	Failure of BI002
	Failure to actuate BI002	Failure of BI002 actuation from APU:s (3-o-o-4) <ul style="list-style-type: none"> <li>- undetected failure</li> </ul> Failure of measurements for I002 <ul style="list-style-type: none"> <li>- undetected failure</li> </ul>
	Failure to actuate BSS04	Failure of BSS04 actuation from APU:s (3-o-o-4) <ul style="list-style-type: none"> <li>- undetected failure</li> </ul> Failure of measurements for BSS04 <ul style="list-style-type: none"> <li>- undetected failure</li> </ul>
	Spurious BH00i	APU internal failure <ul style="list-style-type: none"> <li>- detected failure</li> </ul> APU-APU communication link failure Failure of BH00i actuation from APU:s (3-o-o-4) <ul style="list-style-type: none"> <li>- detected failure</li> </ul> Failure of measurements for BH00i <ul style="list-style-type: none"> <li>- undetected failure</li> <li>- detected failure</li> </ul>
MU	No relevant failure modes with respect to EFW functions	

Single I&C unit failure is typically not critical but a CCF is required to have an effect on safety functions. This is analysed in Table A-7.

**Table A-7.** Failure effects of I&C units on front line safety systems.

I&C unit failure (RPS-A/RPS-B)	Safety system failure effect		
	EFW (RPS-B)	ADS (RPS-A)	ECC (RPS-A)
VU failure detected or undetected	no start	-	no start
CCF between communication links APU-VU 2/4 detected	spurious close of valves	-	-
3/4 detected	spurious close of valves	-	-
CCF between APU:s 1/4 detected	-	no open of 2 valves	-
1/4 undetected	-	no open of 2 valves	-
2/4 detected	spurious close of valves	no open of 4 valves	-
2/4 undetected	-	no open of 4 valves	-
3/4 detected	spurious close of valves	no open of 6 valves	-
3/4 undetected	no start	no open of 6 valves	no start
4/4 detected	spurious close of valves	no open of 8 valves	-
4/4 undetected	no start	no open of 8 valves	no start
CCF between communication links APU-APU 12/12 detected	-	no open of 8 valves	-
MU failure Detected or undetected	-	no manual open	-
CCF between communication links MU-APU Detected or undetected	-	no manual open	-

### Hardware modules and failure modes

The hardware modules and corresponding basic failure modes that are included in the example PSA model are presented in Table A-8.

I&C units are designated as follows

$RPS_{ij}PU00k$ , where

$i$  = division

$j$  = subsystem,  $j = 1$  for RPS-A,  $j = 2$  for RPS-B

$k = 1$  for APU,  $k = 2$  for VU

Modules are designated by the ID of the I&C unit and the module component ID (see Table A-8). Measurement sensors are designated with corresponding process measurement system, e.g., system RPV for reactor pressure vessel measurements.

**Table A-8.** Hardware modules and basic failure modes.

Hardware component Component ID	Failure mode
Processor module PM01	Hang
	Communication dropout
	Delayed signal
	Random behaviour
Analog input module AI0 $x$ , $x = 1, 2, \dots$	Signal fails high/low
	Signal drifts
	Signal hangs/freezes
Digital input module DI0 $x$ , $x = 1, 2, \dots$	Signals stuck to current value
Digital output module DO0 $x$ , $x = 1, 2, \dots$	Signals stuck to current value
Communication module LLPU $ij\_kl$ signal is received by the I&C unit (VU/APU) $i$ of division $k$ and is sent by the I&C unit (VU/APU) $j$ of division $l$	Interruption
Backplane (subrack)	Loss of backplane
Power supply (subrack) SR01	Interruption
	Short circuit
	Ground contact
Measurement Cx $x = E$ for voltage $x = L$ for level $x = P$ for pressure $x = T$ for temperature	Fails high
	Fails low
	Drift of value
	Freeze of value

### Software failure modes

Assumed software basic events for the example PSA are presented in table A-9. Basic events leading to the same end effect have been merged together. Common cause failure is assumed between application software basic events based on (practically) identical software modules in RPS-A and RPS-B.

**Table A-9.** Assumed software fault basic events (*i* = division 1, 2, 3 or 4).

CCF software group	Failed func. or I&C unit	Failure modes description	Fault cases	Uncovering situation	FTD APU	FTD VU
CCF_SW1	SyS	Complete CCF covering faults in SyS of both subsystem. Fatal failure.	1	Online Detection	Outputs to 0	Outputs to 0
CCF_SW2A_A CCF_SW2A_B	APU & VU	All APU and VU fails due to a failure of a common SW module. Fatal failure.	2a	Online Detection	Outputs to 0	Outputs to 0
CCF_SW2B_A CCF_SW2B_B	DCS or DLC	All communication fails due to a failure of a common SW module. Fatal failure.	2b	Online Detection	DFLT-values	DFLT-values
CCF_SW3A_A CCF_SW3A_B	APU	Loss of all functions in APU due to failure of SW module. Fatal failure.	3a	Online Detection	DFLT-values	-
CCF_SW3B_A CCF_SW3B_B	VU	Loss of all functions in VU due to failure of SW module. Fatal failure.	3b	Online Detection	-	Outputs to 0
CCF_SW4A_AADS1	AADS1	Failure to actuate AADS1 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_AH00 <i>i</i> CCF_SW4A_BH00 <i>i</i>	AH00 <i>i</i> BH00 <i>i</i>	Spurious actuation xH00 <i>i</i> due to failure of SW module. Non-fatal failure.	4a	Spurious effect	No	-
CCF_SW4A_AI002 CCF_SW4A_BI002	AI002 BI002	Failure to actuate xI002 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_AI005 CCF_SW4A_BI005	AI005 BI005	Failure to actuate xI005 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_AM005	AM005	Spurious actuation AM005 due to failure of SW module. Non-fatal failure.	4a	Spurious effect	No	-
CCF_SW4A_ASS04 CCF_SW4A_BSS04	ASS04 BSS04	Failure to actuate xSS04 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_ASS05 CCF_SW4A_BSS05	ASS05 BSS05	Spurious actuation xSS05 due to failure of SW module. Non-fatal failure.	4a	Spurious effect	No	-
CCF_SW4A_ASS12 CCF_SW4A_BSS12	ASS12 BSS12	Failure to actuate xSS12 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_ASS13	ASS13	Failure to actuate ASS13 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_ATB01	ATB01	Failure to actuate ATB01 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-

CCF software group	Failed func. or I&C unit	Failure modes description	Fault cases	Uncovering situation	FTD APU	FTD VU
CCF_SW4A_ATB02	ATB02	Failure to actuate ATB02 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_ATB03	ATB03	Failure to actuate ATB03 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_BX001	BX001	Failure to actuate BX001 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_AX003	AX003	Failure to actuate AX003 due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4A_AZS00i CCF_SW4A_BZS00i	AZ00i BZ00i	Failure to actuate xZ00i due to failure of SW module. Non-fatal failure.	4a	Latent revealed by demand	No	-
CCF_SW4B_AACP1 CCF_SW4B_BACP1	AACP1 BACP1	Failure to actuate xACP1 due to failure of SW module. Non-fatal failure.	4b	Latent revealed by demand	-	No
CCF_SW4B_ACCW1	ACCW1	Failure to actuate ACCW1 due to failure of SW module. Non-fatal failure.	4b	Latent revealed by demand	-	No
CCF_SW4B_AECC1	AECC1	Failure to actuate AECC1 due to failure of SW module. Non-fatal failure.	4b	Latent revealed by demand	-	No
CCF_SW4B_AECC2	AECC2	Spurious actuation AECC2 due to failure of SW module. Non-fatal failure.	4b	Spurious effect	-	No
CCF_SW4B_AMFW1	AMFW1	Failure to actuate AMFW1 due to failure of SW module. Non-fatal failure.	4b	Latent revealed by demand	-	No
CCF_SW4B_AMFW2	AMFW2	Spurious actuation AMFW2 due to failure of SW module. Non-fatal failure.	4b	Spurious effect	-	No
CCF_SW4B_ARHR1	ARHR1	Failure to actuate ARHR1 due to failure of SW module. Non-fatal failure.	4b	Latent revealed by demand	-	No
CCF_SW4B_BEFW1	BEFW1	Failure to actuate BEFW1 due to failure of SW module. Non-fatal failure.	4b	Latent revealed by demand	-	No
CCF_SW4B_BEFW2	BEFW2	Spurious actuation BEFW2 due to failure of SW module. Non-fatal failure.	4b	Spurious effect	-	No
CCF_SW4B_BEFW3	BEFW3	Failure to actuate BEFW3 due to failure of SW module. Non-fatal failure.	4b	Latent revealed by demand	-	No
CCF_SW4B_BEFW4	BEFW4	Spurious actuation BEFW4 due to failure of SW module. Non-fatal failure.	4b	Spurious effect	-	No



**Failure data**

- Safety system equipment: Generic data (T-book)
- IE frequencies : Assumed based on Nordic operating experience
- Digital I&C hardware: Fictive data, engineering judgement, see Table A-10
- Digital I&C hardware CCF: Generic data (NUREG/CR-5497) , see Table A-11
- Digital I&C software: Assumed based on engineering judgement (Bäckström et al. 2015), see Table A-12

**Table A-10.** Assumed hardware failure rates for digital I&C units.

I&C modules		Failure rate Total	Detection coverage	Rate undetected failures	Rate detected failures
ID	Description	[/h]	[%]	[/h]	[/h]
PM <sup>1</sup>	Processor module	2,0E-6	99%	2,0E-8	2,0E-6
LL	Communication link module	7,5E-6	100%	0,0E+0	7,5E-6
DI	Digital input module	1,7E-6	75%	4,2E-7	1,3E-6
DO	Digital output module	4,4E-6	91%	4,0E-7	4,0E-6
AI	Analog input module	2,3E-6	65%	7,9E-7	1,5E-6
AO	Analog output module	4,0E-6	87%	5,3E-7	3,5E-6
SR	Subrack incl. power supply	1,0E-5	100%	0,0E+0	1,0E-5

I&C units <sup>2</sup>		Failure rate Total	Detection coverage	Rate undetected failures	Rate detected failures
ID	Description	[/h]	[%]	[/h]	[/h]
APU	Acquisition and processing unit	2,6E-5	95%	1,2E-6	2,5E-5
VU	Voting unit	2,4E-5	98%	4,2E-7	2,3E-5
MU	Manual control unit	2,1E-5	98%	4,4E-7	2,1E-5

I&C modules <sup>3</sup>		#Items in I&C Unit		
ID	Description	APU	VU	MU
PM	Processor module	1	1	1
LL	Communication link module	8	4	4
DI	Digital input module	0	0	1
DO	Digital output module	3	4	0
AI	Analog input module	6	0	0
AO	Analog output module	0	0	0
SC	Signal conditioning module	0	0	0
SR	Subrack incl. power supply	1	1	1

<sup>1</sup> Includes two processors for data processing and communication  
<sup>2</sup> Failure rates includes 1 of each relevant module  
<sup>3</sup> Number of items equals the number modelled items at the module level

**Table A-11.** Assumed CCF parameters for hardware modules (alpha-factor model).

Failure Mode	$\alpha_{2/3}$	$\alpha_{2/4}$	$\alpha_{3/3}$	$\alpha_{3/4}$	$\alpha_{4/4}$
Detected Failure	5E-2	5E-2	1E-2	1E-2	1E-3
Undetected Failure	5E-2	5E-2	1E-2	1E-2	1E-3

**Table A-12.** Assumed CCF failure data for software modules (Bäckström et al. 2015).

SW failure end effect	SW CCF case	Prob.*
SYSTEM: Loss of both subsystems, fatal CCF	1	1E-7
1SS: Loss of one subsystem, fatal CCF	2a	1E-6
1SS: Loss of DCUs in one subsystem, fatal CCF	2b	1E-5
APU-1SS/VU-1SS: Loss of redundant APUs/VUs in one subsystem, fatal CCF	3	1E-7
FF-1SS: Failure of one (or more) application function (Spurious actuation or Failure to actuate), fault in AS module causing a non-fatal CCF	4a, 4b	1E-7

\* These are rounded numbers compared to (Bäckström et al. 2015)

## Fault tree model structure

As an example the fault tree structure for the actuation of EFW pumps is explained in this chapter (see also ch. 7.4.2). The first figure is the fault tree page for the pump failure. Not all transfers are shown and explained but only one “path” from the actuator to sensors. The path follows the physical structure of the architecture starting from VU, communication links, APUs down to sensors. Power supply fault trees are not presented.

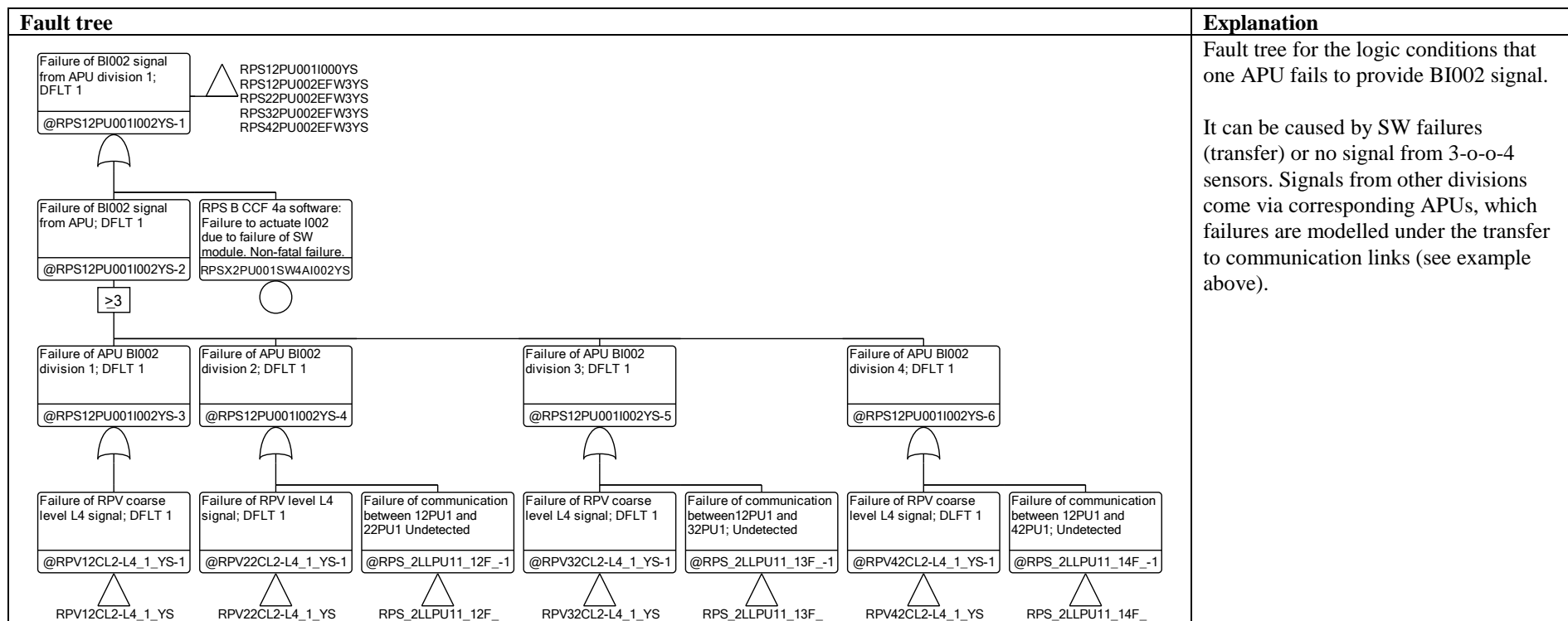
Fault tree	Explanation
	<p>EFW pump 1 failure, which can be caused by pump failure, power supply failure, cooling failure or signal failure.</p> <p>Modelling of "Failure of start signal" will be explained in the next fault trees</p>
	<p>Failure of receiving start signal by the pump. Loss of DC power may fail it or no signal from the corresponding VU, which is caused by the failure modes of the VU (transfer RPS12PU002DO4_Y_) and the failure modes of the signal (transfer RPS12PU002EFW1YS).</p> <p>Fault trees for DC power failure are not shown this example. They follow the ordinary structuring of fault trees for auxiliary power system.</p>

Fault tree	Explanation
	<p>Failure modes of the VU include the failure of the digital output module and the failure modes of the processor (transfer).</p>
	<p>Both detected and undetected failure modes are relevant for VU with respect to providing the EFW start signal when demanded.</p>

Fault tree	Explanation
	<p>Fault tree for detected failure modes of a processor (VU)</p>
	<p>Fault tree for the logic conditions that VU fails to provide EFW start signal. It can be caused by SW failures (transfer) or no signal from 3-o-o-4 APUs (transfers). APU failures are modelled under the transfer to communication links.</p> <p>Signal failure from each APU is modelled under other transfers. Only the signal path BI000 will be explained below.</p>

Fault tree	Explanation
	<p>General SW failures modes causing loss of all functions of RPS B</p>
	<p>Failure of communication between APU and VU. Detected failure mode is not relevant due to fail-safe principle. Practically this is a transfer to APU's undetected failure modes.</p> <p>Detected failure modes appear for sensitivity study purposes.</p>

Fault tree	Explanation
	<p>APU processor failure (undetected).</p> <p>Detected failure modes appear for sensitivity study purposes.</p>
	<p>Fault tree for the logic conditions that one APU fails to provide BI000 signal. Subconditions depend on the initiating event, which is taken into account by house events.</p> <p>Signal path BI002 is explained below</p>





Fault tree	Explanation
	<p>Sensor failure (needed for BI002 condition). Only undetected failure mode is relevant.</p> <p>Detected failure modes appear for sensitivity study purposes.</p>
	<p>Undetected failure of receiving signal from a sensor can be due to sensor failure or APU input module failure.</p>

Fault tree	Explanation
<pre> graph TD     A["Failure of APU analog input module no. 2 RPS B division 1; Undetected @RPS12PU001AI2_F_-1"]     B["Failure of analog input; Undetected RPS12PU001AI002___F"]     A --- C{ }     C --- B     A --- D[RPV12CL002___FS]     B --- E[ ]     style D fill:none,stroke:none     style E fill:none,stroke:none </pre>	<p>APU input module failure (undetected).</p>

## Appendix B. Evaluation of the model aspects

### Reference model

In the evaluation of the model aspects, the comparisons are done with respect to minimal cut sets containing I&C related basic events. The absolute numbers are less interesting since the example is fictive, and many aspects of the model are unrealistic. Above all, the model includes only a small number of initiating events, systems, functions and components.

Core damage frequencies for different initiating events in the reference case are given in Table B-1. The contribution of I&C is 23%. Initiating events Loss of main feedwater and Loss of offsite power are dominating since they have relatively large frequency and the main feedwater system is lost in these scenarios. It should also be noted that in the overall model, the emergency power supply has a significant contribution to the core damage frequency, since it is not diversified (a common system for the emergency feedwater system and emergency core cooling system).

**Table B-1. Core damage frequencies for initiating events in the reference case.**

<b>Initiating event</b>	<b>Total CDF [1/year]</b>	<b>Fractional contribution of I&amp;C</b>	<b>CDF sensitivity to I&amp;C* [1/year]</b>
Loss of main feedwater system	1,5E-6	45%	1,1E-5
Loss of coolant accident	2,3E-8	6%	3,7E-8
Loss of offsite power	1,6E-6	1%	1,7E-6
Transient	4,5E-8	11%	4,4E-7
<b>SUM</b>	<b>3,2E-6</b>	<b>23%</b>	<b>1,3E-5</b>

\*CDF when failure probability of I&C equipment is increased by a factor 10

Dominating minimal cut sets and basic events for initiating event “Transient”, are related to sequences with loss of offsite power as a post-transient event in combination with failure of backup power, resulting in a station blackout. The dominating events causing these sequences are unrelated to digital I&C. Cut sets containing digital I&C have a low individual contribution to the top frequency and the highest contribution is given by cut sets with CCF of EFW pumps and loss of RPS A system due to software failure of communication modules (fault case 2b, Table 13), and from cut sets containing failures of RPS B subracks that affect EFW system and software failure of RPS A communication modules (fault case 2b).

Analysis case for initiating event “Loss of offsite power” is also dominated by station blackout sequences without I&C failure events. The highest contribution from cut sets involving I&C is given by sequences with failure of main feedwater due to failed gas turbine in combination with failure of EFW and ECC systems due to software CCF events (fault case 1 and 2b).

The “Loss of coolant accident” analysis case is dominated by CCF events within the ECC or the RHR systems. Contribution from I&C events is low and is dominated by hardware and software CCFs causing complete failure of APUs and/or VUs.

The one case where failure of I&C equipment is within the dominating sequences is “Loss of main feedwater system”, since the example model is fully dependant on digital I&C in sequences where the main feedwater is unavailable. The dominating events are software CCF

causing failure of both RPS sub systems (fault case 1), or either software failure of RPS B communication modules (fault case 2B), or CCF of RPS B APU subracks, in combination with failure of manual depressurisation. All sequences causes failure of both EFW and ECC systems.

Below evaluations of the different digital I&C modelling aspects is presented. The evaluations are based on the results from initiating event “Transient”.

### **Hardware failure modes**

The fault tree model has been developed at module level of abstraction with modules and failure modes according to Table 7. Importance measures have been calculated for each module type and combined failure mode.

The results show that both undetected and detected failures contribute significantly to the result, in fact detected failures have almost 11 times higher fractional contribution than undetected failures. The contribution is almost exclusively given by CCF events both for detected and undetected failures.

The reason to the high contribution from the detected failures is found in the fault tolerant design of the RPS, where several safety functions (mainly isolation signals) apply a default value of 1 (i.e. 1-o-o-4 conditions tripped) at a detected failure in the APU:s, see Appendix A (Tables A-6, A-7). With failure in more than one division, e.g., by a CCF, this will lead to a spurious VU activation of one or several actuation signals, which in turn may cause stop of one or several safety systems. The main contributor to the detected failures is the subrack module which affects the complete I&C unit and also has a relatively high failure probability compared to the other I&C modules. The contribution to detected failures from digital output modules is small since these only can affect a single system function.

The contribution from undetected failures was found to be of the same magnitude for the different modules. No module or failure mode was found to have insignificant contribution to the plant risk.

The results stress the importance of *not* excluding detected failures from the reliability model.

### **Level of detail**

In order to evaluate the effect on plant risk measures of performing the digital I&C reliability model at different levels of detail, the example model has been developed with the possibility to evaluate the reliability of the digital protection system at I&C unit level.

This is performed by applying the hardware taxonomy of section 6.7 for the I&C unit level and modelling corresponding failure modes as exchange events for the basic events of processor failure modelled at module level. All other basic events at the module level receive a failure probability of 0. SW failure modes are however unchanged since they already represent I&C unit level and are modelled according to section 7.2.2.

One important task for the I&C unit level modelling is to calculate realistic failure rates and probabilities with regard to the number of sub-components (i.e. modules) critical for the I&C units function and the test interval of the I&C unit. In this project, the impact with regard to simplifications in modelling of dependencies rather than conservatism in reliability data is

the objective. Thus, for the purpose of evaluating modelling aspects, the failure rate is calculated as the sum of failure rates and failure probabilities of the modules in the I&C unit. This gives the lowest possible failure rate for the I&C unit and the differences in results compared to the module level reliability model will to a larger extent be the result of simplifications in functional dependencies. The test interval for undetected failures is assumed to be the same as for the processor module, i.e., one year.

When results from the general transient event tree analysis case in the example model at the I&C unit level are compared to the results from the module level model, a CDF increase of a factor 2,5 is observed for the I&C unit level case.

The importance of the I&C increases with almost a factor 6. These systems gain the highest fractional contributions among the modelled safety systems. The largest increase in importance is found for the undetected failures where the fractional contribution increases with a factor 67 while the increase factor for detected failures is less than 2. At the I&C unit level undetected failures also have a higher risk contribution than detected failures by a factor 4, whereas in the module level of abstraction the detected failures had a 11 times higher risk contribution than the undetected failures. This shows that the modelling at a higher level of abstraction (less details) may produce misleading results which in turn may lead to erroneous risk informed decisions.

One reason for the large increase in the importance of undetected failures is that a test interval of 1 year is applied to the I&C unit, while in the module level of abstraction the test interval for digital outputs is assumed to be 4 weeks, i.e. the failure probability of a single digital output is increased with a factor of 13 (all other modules have in the module level a test interval of 1 year). The results show however also that a large increase can be found due to the simplifications of dependencies to input and output modules, and also communication modules, that are applied when modelling at I&C unit level.

The rather low increase in the importance of detected failures is due to that the subrack is by far the largest contributor to detected failures. The failure probability of the complete I&C unit is a factor 2 compared to that of the subrack, which implies that the impact of modelling detected failures on a higher level of abstraction is negligible, i.e. the increase found is solely due to increase in the failure probability. The reason for this result is that failure of the subrack has the same impact as a failure of a complete I&C unit in combination with the subrack dominating the contribution from detected failures. In a case with lower failure probability of the subrack a larger relative increase in importance of detected failures when modelling at I&C unit level should be expected.

By comparing the cut set lists of the I&C unit and module level major differences can be observed. The list at module level is dominated by sequences with loss of offsite power as a post transient event in combination with failure of backup power resulting in a station blackout. The dominating events causing these sequences are unrelated to digital I&C. Cut sets containing digital I&C have a low individual contribution to the top frequency and the highest contribution is given by cut sets with CCF of EFW pumps and loss of RPS A system due to software failure of communication modules (fault case 2b, Table 13), and from cut sets containing failures of RPS B subracks that affect EFW system and software failure of RPS A communication modules (fault case 2b).

The cut set list at the I&C unit level is not dominated by the station blackout sequences as in the module level, though these sequences are still high ranked. In addition the I&C unit level cut set list contains a large number of cut sets containing threefold CCF for undetected failure of APU:s respectively. The sequence leads to the failure of reactor scram, which in comparison is a core damage sequence with quite low importance in the module level PSA. There are two major reasons for the increase in importance. The first is that dependencies for individual scram conditions to different input and output modules are not considered when modelling on the I&C unit level, i.e. they all fail at the same time. The second reason is that correct test intervals of the digital outputs for the reactor scram cannot be applied at I&C unit level modelling, which incorrectly results in a high risk contribution from reactor scram sequences.

It should be noted that the chosen approach for the I&C unit failure rate estimation produces lowest possible failure rate and that a more realistic approach should be expected to produce much higher results than presented here. A more realistic treatment of test intervals by calculating a mean value would decrease the results, but the differences described above would still be evident, only somewhat smaller.

### **Impact of default values**

As described in Appendix A and discussed in previous sections, the assumed fault tolerant design of the example digital I&C systems apply default values of 1 in case of detected failures for some safety functions and actuator signals. This has the effect that spurious signals can occur and affect the safety systems availability, which is also reflected in the results of the evaluation of the modelling aspects performed on the reference model. It is hence relevant to also evaluate the impact of the digital I&C for a fault tolerant design with a minimum of spurious signals.

For this purpose, the example PSA has been evaluated under the assumption that a default value of 1 is applied to detected failures only for the reactor scram safety function. For all the other safety functions a default value of 0 is applied to detected failures, which means that no spurious signals can be caused by the digital I&C and detected failures instead contribute to loss of actuator signals.

The evaluation shows a small decrease in the core damage frequency at the module level of abstraction, which means that the decrease of the probability of spurious signals has bigger effect than the increase of the probability for failure to actuate caused by detected failures. The importance of detected failures decreases significantly compared to the reference model. The fractional contribution (FC) is of the same size for detected failures as for undetected failures.

When evaluating this case at the I&C unit level of abstraction one major difference is observed compared to the module level. The importance of undetected failures is still very high while the importance of detected failures decreases significantly. The FC of undetected failures is a factor 30 higher than the FC for detected failures. The reason for this is the increased importance of the event sequences related to failure of the scram system. Since the scram safety function in this case still applies a default value of 1 at detected failures, the conservatism applied for undetected failures when modelling on the I&C unit level comes even more evident in this case. Compared to the FC of undetected failures at the module level of abstraction, the I&C unit level of abstraction FC is a factor 70 higher.

### Sensitivity of CCF parameters of detected faults

The evaluation results show that the contribution from hardware failures is almost exclusively given by CCF events both for detected and undetected failures which are expected due to the design and redundancy of the digital I&C systems.

The assigned CCF parameters will have large impact on the importance of the digital I&C for the plant safety, and in the example model the same CCF parameters have been assigned for both detected and undetected failures. CCF parameter values of digital I&C units and hardware modules are found in Appendix A, Table A-11. For the example model generic values have been used due to lack of I&C specific values.

It can be questioned if it is reasonable to use the same CCF parameter values for both detected and undetected failure modes, e.g., with regard to time factors. It is often argued that the likelihood of CCF for detected failures should be smaller than for undetected failures. If calculated CCF parameters for conventional equipment are studied, e.g., NUREG/CR-5496, no evidence for this can be found. The comparison is however not completely accurate and it could still be the case for digital I&C.

Since the results show that detected failures have a significantly higher fractional contribution than undetected failures (11 times higher), it is reasonable to perform a sensitivity analysis where the values of the CCF parameters for detected failures are significantly lowered. The sensitivity analysis therefore applies values up to ten times lower than the parameters of the undetected failures. Assumed CCF parameters are presented in Table B-2.

**Table B-2** Assumed CCF parameters for digital I&C units and hardware modules (alpha-factor model).

<b>Failure mode</b>	<b><math>\alpha_{2/3}</math></b>	<b><math>\alpha_{2/4}</math></b>	<b><math>\alpha_{3/3}</math></b>	<b><math>\alpha_{3/4}</math></b>	<b><math>\alpha_{4/4}</math></b>
Detected failure in sensitivity analysis	1E-2	1E-2	1E-3	1E-3	1E-4
Detected failure in original analysis	5E-2	5E-2	1E-2	1E-2	1E-3
Undetected failure in original and sensitivity analyses	5E-2	5E-2	1E-2	1E-2	1E-3

The results from the sensitivity analysis show that lowering the CCF parameters of the detected failures have a significant impact on the results. The total core damage frequency decreases with 8%. The results show that detected failures still have a significantly higher fractional contribution compared to undetected failures, by a factor 2,5. Once again, this stresses the importance of *not* excluding detected failures from the reliability model. Also relevant CCF parameters are of interest in order to achieve a relevant result.

### Software failure modes

The results show that software faults have a significant impact on the overall result. Software faults in total have a fractional contribution of about 5%.

A comparison of the different software fault cases shows that software fault case 2 has the highest fractional contribution of 4%, followed by fault case 1, fault case 3 and finally software fault case 4, in that order (see Table 8 for fault cases).

Fault case 2b has a significant impact on the core damage frequency. This is due to that the VU:s applies default values to the outputs at failure of the communication software. In the example model it causes the main feedwater pumps to stop due to spurious actuation. The fault case also causes malfunction of the open signal to the automatic depressurisation system relief valves.

It should be remembered that the failure probability differs between different fault cases, see Table A-12. Fault case 1 that has the worst end effect has a low fractional contribution of approx. 0,3%, and also the lowest failure probability. If however the failure probability is increased by a factor 100 to 1E-5, the core damage frequency increases with 30%. The impact of the different software fault cases is hence, and naturally, highly dependent on the assigned failure probabilities.

The impact of spurious signals is large which is shown when default values of 0 instead of 1 are applied in accordance with section 6.4.1.2. In this case the fractional contribution of software faults in total is 1%, e.g. the same magnitude as for hardware failures, whereof fault case 2b contributes with approx. 97% of this.

It is worth noting that type 4 failures could, depending on design, cause critical spurious failure due to actuation of signals that normally is seen as non-critical in the PSA, e.g., different types of manual actuation signals, indications and selector signals, which typically is screened out from the analysis at an early stage.

Based on the above, it can be concluded that software faults in general have a non-negligible effect on the results and should be considered in a digital I&C PSA. Quantification of software faults and the assessment of the degree of diversity between subsystems can therefore be significant from the overall PSA results point of view.



## FinPSA modelling

Different I&C model parts are described in the following subsections.

### Inputs to APU voting

An APU voting has always four inputs. An input is TRUE if the measurement sensor works and the communication link between this APU and the APU from the sensor's division works. If the measurement comes from the same division, only the measurement sensor needs to function. Example:

$$\text{RPS31PU001I0002\_2\_I} = \text{RPS21PU001VL002\_F\_S} * \text{RPS\_1LLPU11\_32\_F\_S},$$

where  $\text{RPS31PU001I0002\_2\_I}$  is an input to APU voting,  $\text{RPS21PU001VL002\_F\_S}$  is a measurement sensor and  $\text{RPS\_1LLPU11\_32\_F\_S}$  is a communication link. Failures of measurement sensors and communication links are modelled in fault trees. Hence,  $\text{RPS21PU001VL002\_F\_S}$  and  $\text{RPS\_1LLPU11\_32\_F\_S}$  are names of fault trees. Failures of APU analog input modules are modelled in the fault trees of measurement sensors as in the RiskSpectrum model (Table 9).

### APU votings

APU votings in the example model are 2 out of 4 votings as in the following example:

$$\begin{aligned} &\text{RPS31PU001I0002\_V\_I} \\ &= <2 \text{ RPS31PU001I0002\_1\_I} + \text{RPS31PU001I0002\_2\_I} + \text{RPS31PU001I0002\_3\_I} + \text{RPS31PU001I0002\_4\_I}> \end{aligned}$$

All the inputs are I&C model elements that are defined in 'Inputs to APU voting' section.

### Dependencies between signals

Some signals are combined after the APU votings. For example, signals I002 and I005 are combined to form I000. Example equation:

$$\begin{aligned} &\text{RPS31PU001I0000\_V\_I} \\ &= (\text{RPS31PU001I0002\_V\_I} * \text{RPSX1PU001SW4AI002YS}) + (\text{RPS31PU001I0005\_V\_I} * \text{RPSX1PU001SW4AI005YS}), \end{aligned}$$

where  $\text{RPS31PU001I0002\_V\_I}$  and  $\text{RPS31PU001I0005\_V\_I}$  are defined in APU votings, and  $\text{RPSX1PU001SW4AI002YS}$  and  $\text{RPSX1PU001SW4AI005YS}$  are software modules. The equation means that I000 signal is TRUE if I002 or I005 is TRUE and the corresponding software module works.

### Inputs to VU voting

A VU voting has always four inputs. An input is TRUE if signal from APU (e.g. I000) is TRUE and the communication between this VU and the APU works. Example:

$$\text{RPS41PU002EC001\_3\_I} = \text{RPS31PU001I0000\_V\_I} * \text{RPS\_1LLPU12\_34\_F\_S},$$

where  $\text{RPS41PU002EC001\_3\_I}$  is an input to VU voting of ECC pump start signal,  $\text{RPS31PU001I0000\_V\_I}$  is I000 signal defined in 'Dependencies between signals' section and  $\text{RPS\_1LLPU12\_34\_F\_S}$  is a communication link between APU 3 and VU 4. The equation means that the input to VU voting is TRUE if I000 is TRUE and the communication link works. The failure of the communication link is modelled in a fault tree named  $\text{RPS\_1LLPU12\_34\_F\_S}$ .

### VU votings

VU votings in the example model are 2 out of 4 votings. Example:

```
RPS41PU002EC001_S_I  
= <2 RPS41PU002EC001_1_I + RPS41PU002EC001_2_I + RPS41PU002EC001_3_I + RPS41PU002EC001_4_I>
```

All the inputs are I&C model elements that are defined in ‘Inputs to VU voting’ section.

### Actuation signals

An actuation signal is typically TRUE if the voting result in VU is positive, the digital output module of VU works, a DC power system bus works, and software failures do not cause failure on demand. Example:

```
ECC40PM001AS001_S_I + ECC40VM002AS001_S_I  
= RPS41PU002EC001_V_I * RPSX1SW001GE001_A_S * RPSX1PU002SW4BECC1YS *  
RPS41PU002DO003_A_S * DCP41BT001DG001_G_S,
```

where ECC40PM001AS001\_S\_I is an ECC pump start signal, ECC40VM002AS001\_S\_I is an ECC valve open signal, RPS41PU002EC001\_S\_I is a VU voting result, RPS41PU002DO003\_A\_S is a digital output module, DCP41BT001DG001\_G\_S is a DC power system bus, RPSX1PU002SW4BECC1YS is a software module and RPSX1SW001GE001\_A\_S is TRUE if all of functions of RPS A have not been lost by software failure. The actuation signals for the pump and the valve are same. They could also be defined in separate equations but the model is more compact this way. RPS41PU002EC001\_S\_I is defined in VU votings and the failures of the digital output module and the DC power system bus are modelled in fault trees named RPS41PU002DO003\_A\_S and DCP41BT001DG001\_G\_S. Loss of all RPS A functions is modelled in fault tree RPSX1SW001GE001\_A\_S. Fault trees of ECC pump and valve include LIC gates with names ECC40PM001AS001\_S\_I and ECC40VM002AS001\_S\_I. When minimal cut sets are generated, fault trees of the actuation signals are created and linked to LIC gates.

### **Modelling intelligent voting logics**

The example model includes many 2/4 votings performed in APUs and VUs. In intelligent voting, different detected failures are treated differently: some of them are ignored but not all. For example, 2/4 logic can degrade to 1/3 at first faulty input, and to 1/2 at second and to 1/1 at third faulty input. In total, there are ten different logics for 2/4 voting, two of which are not intelligent. They are presented in Table B-3. “DF” refers to “detected failure”.

**Table B-3: Intelligent logics for 2/4 voting (DF = detected fault).**

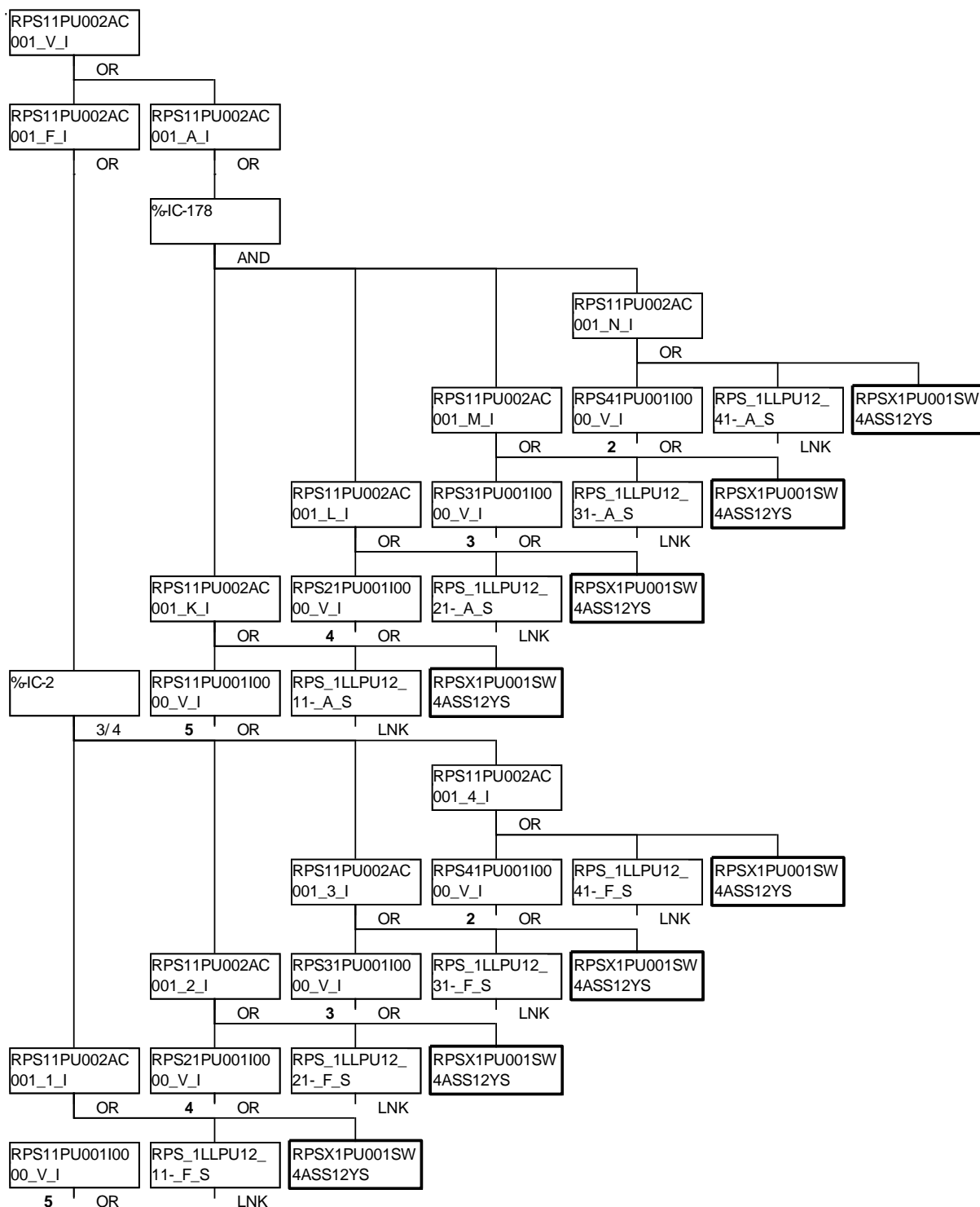
<b>Voting logic</b>	<b>First DF</b>	<b>Second DF</b>	<b>Third DF</b>	<b>Fourth DF</b>
DFLT 1	1-o-o-3	trip		
DFLT 0	2-o-o-3	2-o-o-2	no trip	
(1,0,0,0)	1-o-o-3	1-o-o-2	1-o-o-1	no trip
(0,1,0,0)	2-o-o-3	1-o-o-2	1-o-o-1	no trip
(0,0,1,0)	2-o-o-3	2-o-o-2	1-o-o-1	no trip
(0,0,1,1)	2-o-o-3	2-o-o-2	1-o-o-1	trip
(1,0,0,1)	1-o-o-3	1-o-o-2	1-o-o-1	trip
(0,1,0,1)	2-o-o-3	1-o-o-2	1-o-o-1	trip
(1,0,1,-)	1-o-o-3	1-o-o-2	trip	
(0,1,1,-)	2-o-o-3	1-o-o-2	trip	

For each voting logic, a FinPSA I&C model version was written. In each case, the chosen voting logic was applied to all 2/4 votings in VUs and all 2/4 votings in APUs except those that are related to ADS system. The voting logic was taken into account both in the modelling of actuation signals and spurious stop signals. In addition, in the isolation of an ECC/EFW pump room, the strategy for treating the first detected failure determined if the default value was 0 or 1 (e.g. 0 for logic (0,0,1,1) and 1 for (1,0,0,1)).

Writing a different I&C model for each voting logic is not a small task, but it can be done. In this case, it took few working days. The model contains many similar parts that differ only in names. Hence, a lot of work can be done by applying copy and paste functions. Text based editing is faster than fault tree editing. The modelling process could be automated partly because a voting logic is always modelled using similar equations which could be written automatically by computer code. It could also be useful to implement standard intelligent voting fault tree gates or I&C model functions in PSA software to simplify the modelling. The verification of the correctness of the model is mostly similar to the verification of the correctness of a fault tree model. However, the interpretation of intelligent voting modelling is easier in fault tree format. Hence, transforming the I&C model in fault trees can help in the verification.

To model an intelligent voting logic, the simplest way is to consider which input failure combinations can fail the output and build a fault tree according to that. For example, with logic (1,0,0,0), the output can fail due to three undetected input failures or four input failures which can be detected or undetected. The first detected failure changes the voting logic to 1/3, but after that detected failures affect in the same way as undetected failures. Hence, if the other three inputs fail after the first detected failure, the output fails. In other words, four input failures cause the failure of the output. Figure B-1 presents a fault tree that is built according to this reasoning.

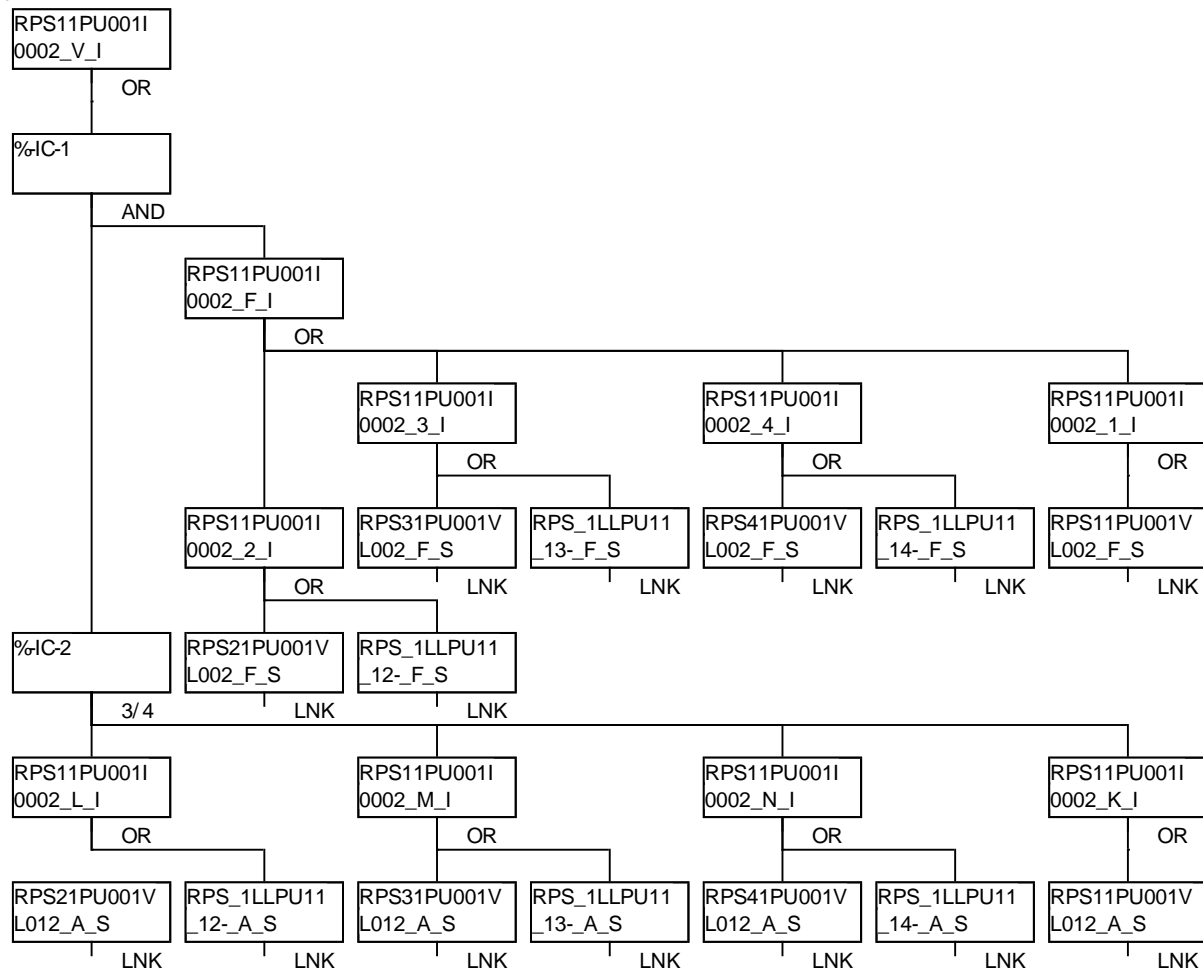
In the example of Figure B-2, an input fails undetectably if I000 signal fails, the communication link fails undetectably or software module fails to actuate SS12 signal (RPSX1PU001SW4ASS12YS). For example, in name RPS\_1LLPU12\_11-\_F\_S, 'F' means that only undetected failures are modelled in the linked fault tree. An input fails detectably if the communication link fails detectably. In name RPS\_1LLPU12\_11-\_A\_S, 'A' means that both detected and undetected failures are modelled in the linked fault tree. With this logic, spurious stop signals are impossible, except for ECC and EFW pumps.



**Figure B-1: The fault tree of ECC actuation signal with voting logic (1,0,0,0) in FinPSA.**

As an example of modelling voting logic (0,0,1,1), the modelling of I002 signal failure is presented in Figure B-2 in fault tree logic. With this logic, the actuation signal fails if three inputs fail so that at least one of the failures is undetected. Only two first detected failures fail the inputs, and hence, at least the third failure has to be undetected. In the fault tree, the part under RPS11PU001I0002\_F\_I gate represents the condition that at least one undetected failure is required. The condition that three failures are required in total is represented by the part that is under 3/4 gate. The AND gate ensures that both conditions must be satisfied.

In the fault tree of Figure B-2, inputs can fail due to undetected or detected failures of communication links or analog input modules. Their failures are modelled in other fault trees. For example, in name RPS21PU001VL012\_A\_S, 'A' means that both detected and undetected failures are modelled in the linked fault tree. In name RPS21PU001VL012\_F\_S, 'F' means that only undetected failures are modelled in the linked fault tree. Spurious stop signals are also possible with this logic, except for ECC and EFW pumps. For example, four detected failures can cause a spurious stop signal of a MFW pump.



**Figure B-2: The fault tree of I002 signal with voting logic (0,0,1,1) in FinPSA.**

The results were calculated using each voting logic. The used reliability data was slightly different than in the RiskSpectrum model. Table B-4 presents the core damage frequency calculated from I&C related minimal cut sets in each case. The results show that there are differences but they are relatively small. Default 1 logic is however significantly inferior to other logics. Logics (0,0,1,1) and (0,1,0,1) are the best. Default 0 logic compares quite well against the intelligent logics but is still slightly inferior. The reasons why the differences between voting logics are small are that non-fatal software failures have a dominant role in I&C related CDF. If the probabilities of non-fatal failures appeared to be overestimated, the differences would be larger.

**Table B-4: Results with different voting logics calculated using FinPSA.**

<b>Voting logic</b>	<b>I&amp;C related CDF [1/year]</b>
(0,0,1,1)	7,31E-7
(0,1,0,1)	7,31E-7
(1,0,0,1)	7,35E-7
(0,1,0,0)	7,98E-7
(0,0,1,0)	7,98E-7
(1,0,0,0)	8,03E-7
(0,1,1,-)	8,09E-7
(1,0,1,-)	8,09E-7
DFLT 0	8,62E-7
DFLT 1	1,18E-6

Voting logics have also some effects on risk importance measure results. The largest difference is naturally in the importance values of detected failures. For example, risk increase factor for detected CCF of three APU processor modules of RPS B is over 1000 for voting logics DFLT 0, (1,0,1,-), (0,1,1,-) and DFLT 1 but less than 6 for others. These results correspond to how three detected failures impact in the voting logic. In the cases where the risk increase factor is high, three detected failures alone result with failed output (either failure to actuate or spurious actuation), but in other cases not. Risk increase factor for detected CCF of two APU processor modules of RPS B is over 1000 for default 1 logic but less than 5 for others. With regard to Fussell-Vesely, the differences are similar. For other detected failures, the results are mostly similar. However, risk increase factor for detected CCF of all communication links between APUs of RPS B is over 1000 for voting logics DFLT 0, (1,0,1,-), (0,1,1,-) and DFLT 1 but less than 3 for others. This is because detected CCF of all communication links between APUs causes three detectably failed inputs for each APU voting. Again, Fussell-Vesely results are similar.

The differences in the contributions of detected failures also affect risk importance measure values of basic events that appear in same minimal cut sets as detected failures.

The rankings are slightly different for different voting logics. Even the order of initiating events loss of offsite power and loss of main feedwater varies with regard to Fussell-Vesely.

Title	Guidelines for reliability analysis of digital systems in PSA context — Final report
Author(s)	Stefan Authén <sup>1</sup> , Jan-Erik Holmberg <sup>1</sup> , Tero Tyrväinen <sup>2</sup> , Lisa Zamani <sup>1</sup>
Affiliation(s)	<sup>1</sup> Risk Pilot AB, <sup>2</sup> VTT Technical Research Centre of Finland Ltd.
ISBN	978-87-7893-411-6
Date	February 2015
Project	DIGREL
No. of pages	101
No. of tables	31
No. of illustrations	13
No. of references	65
Abstract max. 2000 characters	The objective of the DIGREL project has been to provide guidelines to analyse and model digital systems in the context of probabilistic safety assessment (PSA). A failure modes taxonomy for digital I&C systems has been developed jointly with OECD/NEA Working Group on Risk Assessment. Reliability modelling has been studied by developing a fictive, simplified PSA model representing a four-redundant distributed protection system. The evaluation of the example PSA has demonstrated the developed taxonomy and verified that it is suitable for PSA purpose. The evaluation shows that the choice of the level of abstraction for the modelling of digital I&C is of high importance for the results. Module level is recommended. Both undetected and detected hardware as well as software failures contribute significantly to the PSA results, indifferently of the assumed fault tolerant design. Similar conclusion can be drawn from the test of using different CCF parameters for undetected and detected failures. Software faults have a non-negligible effect on the results due to their functional impact on all divisions. In order to develop a realistic fault tree model for a digital I&C protection system it is vital that the chosen fault tolerant design is fully understood and correctly described in the model. The treatment of faulty inputs and degraded voting logic sets the foundation of the fault tree analysis.
Key words	Digital I&C system, probabilistic safety assessment, reliability, nuclear power plant safety

---

Available on request from the NKS Secretariat, P.O.Box 49, DK-4000 Roskilde, Denmark.  
Phone (+45) 4677 4041, e-mail [nks@nks.org](mailto:nks@nks.org), [www.nks.org](http://www.nks.org)