Nuclear Safety
NEA/CSNI/R(2014)16
February 2015
www.oecd-nea.org

Failure Modes Taxonomy for Reliability Assessment of Digital Instrumentation and Control Systems for Probabilistic Risk Analysis

OECD
BETTER POLICIES FOR BETTER LIVES

NEA
NUCLEAR ENERGY AGENCY

Unclassified **NEA/CSNI/R(2014)16**

Organisation de Coopération et de Développement Économiques
Organisation for Economic Co-operation and Development

16-Feb-2015

English text only

NUCLEAR ENERGY AGENCY
COMMITTEE ON THE SAFETY OF NUCLEAR INSTALLATIONS

NEA/CSNI/R(2014)16
Unclassified

FAILURE MODES TAXONOMY FOR RELIABILITY ASSESSMENT OF DIGITAL I&C SYSTEMS FOR PRA

JT03370601

Complete document available on OLIS in its original format

This document and any map included herein are without prejudice to the status of or sovereignty over any territory, to the delimitation of international frontiers and boundaries and to the name of any territory, city or area.

English text only

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT

The OECD is a unique forum where the governments of 34 democracies work together to address the economic, social and environmental challenges of globalisation. The OECD is also at the forefront of efforts to understand and to help governments respond to new developments and concerns, such as corporate governance, the information economy and the challenges of an ageing population. The Organisation provides a setting where governments can compare policy experiences, seek answers to common problems, identify good practice and work to co-ordinate domestic and international policies.

The OECD member countries are: Australia, Austria, Belgium, Canada, Chile, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Israel, Italy, Japan, Luxembourg, Mexico, the Netherlands, New Zealand, Norway, Poland, Portugal, the Republic of Korea, the Slovak Republic, Slovenia, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States. The European Commission takes part in the work of the OECD.

OECD Publishing disseminates widely the results of the Organisation’s statistics gathering and research on economic, social and environmental issues, as well as the conventions, guidelines and standards agreed by its members.

NUCLEAR ENERGY AGENCY

The OECD Nuclear Energy Agency (NEA) was established on 1 February 1958. Current NEA membership consists of 31 countries: Australia, Austria, Belgium, Canada, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Japan, Luxembourg, Mexico, the Netherlands, Norway, Poland, Portugal, the Republic of Korea, the Russian Federation, the Slovak Republic, Slovenia, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States. The European Commission also takes part in the work of the Agency.

The mission of the NEA is:

– to assist its member countries in maintaining and further developing, through international co-operation, the scientific, technological and legal bases required for a safe, environmentally friendly and economical use of nuclear energy for peaceful purposes;

– to provide authoritative assessments and to forge common understandings on key issues, as input to government decisions on nuclear energy policy and to broader OECD policy analyses in areas such as energy and sustainable development.

Specific areas of competence of the NEA include the safety and regulation of nuclear activities, radioactive waste management, radiological protection, nuclear science, economic and technical analyses of the nuclear fuel cycle, nuclear law and liability, and public information.

The NEA Data Bank provides nuclear data and computer program services for participating countries. In these and related tasks, the NEA works in close collaboration with the International Atomic Energy Agency in Vienna, with which it has a Co-operation Agreement, as well as with other international organisations in the nuclear field.

This document and any map included herein are without prejudice to the status of or sovereignty over any territory, to the delimitation of international frontiers and boundaries and to the name of any territory, city or area.

Corrigenda to OECD publications may be found online at: *www.oecd.org/publishing/corrigenda*.

COMMITTEE ON THE SAFETY OF NUCLEAR INSTALLATIONS

Within the OECD framework, the NEA Committee on the Safety of Nuclear Installations (CSNI) is an international committee made of senior scientists and engineers, with broad responsibilities for safety technology and research programmes, as well as representatives from regulatory authorities. It was set up in 1973 to develop and co-ordinate the activities of the NEA concerning the technical aspects of the design, construction and operation of nuclear installations insofar as they affect the safety of such installations.

The committee's purpose is to foster international co-operation in nuclear safety amongst the NEA member countries. The CSNI's main tasks are to exchange technical information and to promote collaboration between research, development, engineering and regulatory organisations; to review operating experience and the state of knowledge on selected topics of nuclear safety technology and safety assessment; to initiate and conduct programmes to overcome discrepancies, develop improvements and research consensus on technical issues; and to promote the co-ordination of work that serves to maintain competence in nuclear safety matters, including the establishment of joint undertakings.

The clear priority of the committee is on the safety of nuclear installations and the design and construction of new reactors and installations. For advanced reactor designs the committee provides a forum for improving safety related knowledge and a vehicle for joint research.

In implementing its programme, the CSNI establishes co-operate mechanisms with the NEA's Committee on Nuclear Regulatory Activities (CNRA) which is responsible for the programme of the Agency concerning the regulation, licensing and inspection of nuclear installations with regard to safety. It also co-operates with the other NEA's Standing Committees as well as with key international organisations (e.g., the IAEA) on matters of common interest.

TABLE OF CONTENTS

LIST OF ACRONYMS

A/D	Analog/digital
AIM	Analog input module
AOM	Analog output module
ALU	Actuation logic unit
APU	Acquistion and processing unit
AS	Application software
ASIC	Application-specific integrated circuit
BIOS	Basic Input Output System (firmware of the microprocessor computer)
CCF	Common cause failure
CDF	Core damage frequency
CL, C.L.	Current loop
COM	Communicatino link module
COMPSIS	OECD/NEA Computer-based Systems Important to Safety Project
COTS	Commercial off-the-shelf
CPU	Central processing unit
CSNC	Canadian Nuclear Safety Commission
CSNI	Committee on the Safety of Nuclear Installations (OECD/NEA)
D/A	Digital/analog
DCS	Data communication software
DCU	Data Communication Unit (Data Communication Module)
DEMUX	Demultiplexer
DIGREL	Digital system reliability failure mode taxonomy
DIM	Digital input module
DLS	Data link configuration
DOM	Digital output module
EDF	Électricité de France
EF	Elementary function
EFW	Emergency feedwater system
ENEL	Ente Nazionale per l'Energia eLettrica, Italy
EPR	European Pressurized Water Reactor, product of AREVA
ESBWR	Economic Simplified Boiling Water Reactor, product of GE Hitachi Nuclear Energy
ESFAS	Engineered Safety Features Actuation System
FM	Failure mode
FMEA	Failure mode and effects analysis
FMECA	Failure mode, effects and criticality analysis
FMEDA	Failure mode, effects and diagnostics analysis
FPGA	Field Programmable Gate Array
FRS	Functional requirements specification
FTD	Fault-tolerant design
GRS	Gesellschaft für Anlagen- und Reaktorsicherheit, Germany
HVAC	Heating, ventilation, air conditioning
HW	Hardware
I&C	Instrumentation and control
I/O	Input/output
IAEA	International Atomic Energy Agency
IC	Integrated circuit
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 Organi ation
KAERI	Korea Atomic Energy Research Institute
MUX	Multiplexer
NEA	OECD Nuclear Energy Agency
NPIC-HMIT	Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies conference
NPP	Nuclear power plant
NRC	U.S. Nuclear Regulatory Commission
NRG	Nuclear Research and consultancy Group, the Netherlands
NRI	Nuclear Research Institute Rez plc
OECD	Organisation for Economic Co-operation and Development
OS	Operating system (software)
OSU	The Ohio State University
PLD	Programmable logic device
PRA	Probabilistic risk analysis
PSA	Probabilistic safety assessment
RAM	Random-Access-Memory
ROM	Read-Only-Memory
RPS	Reactor protection system
RPV	Reactor pressure vessel
SCM	Signal conditioning module
SSA, SSB	Subsystem A, Subsystem B of the example system
SSC	Systems, structures, components
SIL	Safety integrity level
SW	Software
TBL	(Data) Table
TXS	Teleperm XS, product of AREVA
V&V	Verification and validation
VEIKI	Institute for Electric Power Research, Hungary
VU	Voting unit
VTT	Technical Research Centre of Finland
WDT	Watch dog timer
WGRISK	OECD/NEA CSNI Working Group on Risk Assessment

EXECUTIVE SUMMARY

Digital protection and control systems appear as upgrades in older nuclear power plants (NPP), and are commonplace in new NPPs. To assess the risk of NPP 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 the many unique attributes of digital systems (e.g., functions are implemented by software, units of the system interact in a communication network, faults can be identified and handled online), a number of modelling and data collection challenges exist, and international consensus on the reliability modelling has not yet been reached.

The objective of the task group called DIGREL has been to develop a taxonomy of failure modes of digital components for the purposes of probabilistic risk analysis (PRA). An activity focused on the development of a common taxonomy of failure modes is seen as an important step towards standardised digital instrumentation and control (I&C) reliability assessment techniques for PRA. Needs from PRA has guided the work, meaning, e.g., that the I&C system and its failures are studied from the point of view of their functional significance point of view. The taxonomy will be the basis of future modelling and quantification efforts. It will also help to define a structure for data collection and to review PRA studies.

The proposed failure modes taxonomy has been 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 PRAs have been simple – typically a software CCF failing identical processing units.

The DIGREL task group has defined a new failure modes taxonomy based on a hierarchical definition of five levels of abstraction:

1. system level (complete reactor protection system),
2. division level,
3. I&C unit level,
4. I&C unit modules level
5. basic components level.

This structure corresponds to a typical reactor protection system architecture, which is the scope of the DIGREL work. The taxonomy that was developed provides a framework to classify digital system failure modes.

Failure propagation, which is essential for analysing failure modes and their effects, is described using a failure model. Four important elements of the failure model on which the taxonomy focuses stand out:

1. fault location,

2. failure mode,

3. uncovering situation,

4. failure effect and the end effect.

These concepts are applied in particular at the I&C unit and module levels of abstraction. I&C unit level failure modes are associated with the end effects at the I&C unit module level, depending on the fault location and uncovering situation.

The purpose of the taxonomy is to support PRAs and therefore focus is 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. At the level of system, division and I&C units, no significant distinction is made between hardware or software aspects. At the module and basic component levels, the taxonomy differentiates between hardware and software related failure modes.

This report can be seen as a step towards more harmonised approach to analyse and model digital I&C in PRA. There is a number of areas where further studies are needed, and many of the recommendations given in the previous digital I&C expert report NEA/CSNI/R(2009)18 are still valid. For instance, the following actions could be considered:

- Testing of the applicability of the taxonomy in modelling, including test with different I&C designs and modelling approaches.

- Testing of the applicability of the taxonomy in data collection. After the termination of the COMPSIS project, OECD/NEA International Common-cause Failure Data Exchange (ICDE) project has expressed a willingness to integrate computer failures as a new component type for data collection.

- Development of methods for software reliability quantification for nuclear PRAs. There are several past and ongoing R&D projects in this area. Benchmarking studies may be considered.

- Complementation of the failure modes taxonomy with issues that were left out of the scope, e.g., control systems, networks, programmable logic device (PLD) technology, field-programmable gate array (FPGA), application specific integrated circuit (ASIC)

- Development of methods to architecture level assessment, including defence-in-depth and diversity assessments. It is essential to account for the needs of both deterministic and probabilistic safety assessments.

- Development of methods for the evaluation of fault tolerance features in the hardware and software of the safety important I&C systems.

1. INTRODUCTION

With digital upgrades of safety-related protection systems at operating NPPs and use of digital electronic systems in nuclear safety applications at new reactors, it becomes very important to develop reliability methods for quantifying digital instrumentation and control systems.

In the last decades a variety of different safety-related digital I&C systems were developed and implemented in nuclear installations and facilities around the world. Digital I&C architectures are deployed in several reactors worldwide [NUREG/CR-6992], [IAEA-NP-T-3.12] such as the N4 reactors Chooz B and Civaux (France), Sizewell B (United Kingdom), Ringhals 1 and 2 units (Sweden), Temelin 1 and 2 units (Czech Republic), Tianwan (China), Kahiwazaki-Kariwa (Japan). Also new designs such as the EPR developed by AREVA, the APWR by Mitsubishi Heavy Industries, Ltd. (MHI) and the ESBWR by General Electric - Hitachi (GEH) demonstrate the recent state of digital I&C architectures in NPPs. In the United States, only few safety-related systems use digital control or actuation systems, for example, the core protection calculators and the diesel generator sequencers. Oconee and Wolf Creek are the two plants that have replaced their reactor protection system and main-steam isolation control system, respectively.

Modelling of digital systems in a PRA is an important challenge to the PRA community. Due to many unique attributes of digital systems, a number of modelling and data collection challenges exist and there is no full consensus on how the reliability models should be developed [DI&C-ISG-03, BNL-90571-2009-IR]. For example, whether or not a fault tree model adequately captures all dependencies and how software failures should be included in a reliability model remain to be investigated. Different methods for modelling digital systems have been proposed. In particular, fault tree method has been used. In one case study, it was recognized that different modelling methods generated very different results [RESS-1999-Rouvroye]. This is due to the different assumptions, levels of detail, and failure databases used (even if the same method is used).

The main reasons a consensus method has not been identified include (1) how to include and quantify software failures, (2) how to model the fault tolerance features and associated dependencies, (3) what is the right level of detail of modelling, and (4) availability of applicable failure data [NUREG/CR-6962, NPIC&HMIT-2004-Chu].

In 2007, the CSNI of OECD/NEA directed the Working Group on Risk Assessment (WGRisk) to set up a task group to coordinate activities related to digital instrumentation and control system risk [NEA/CSNI/R(2009)18]. A result of this work was recognition that several difficult technical challenges remained to be solved. One of the recommendations was to develop taxonomies of hardware and software failure modes of digital components for the purpose of PRA. This effort was seen as an important first step towards standardizing digital I&C reliability assessment techniques for PRA. This report provides a first effort at developing such taxonomy.

The objective of the digital system reliability failure mode taxonomy (abbreviated DIGREL) task group is to develop a failure mode taxonomy for the reliability assessment of digital instrumentation and control (I&C) systems for use in probabilistic safety assessment (PSA) or probabilistic risk assessment (PRA). In general, the purpose of a taxonomy is the organization of the set of concepts that underlies a particular discipline into elemental or functional units. The taxonomy further identifies the relationships between these units.

Taxonomies are used in many diverse scientific disciplines, from biology to library science. Principles for taxonomy construction have been defined. For example, the General Classification Theory [Marcella-1994] emphasises distinct and unambiguous descriptions of the elemental units; use of hierarchies; and progressions from the general to the particular. A summary discussion of these principles and an example of application in the field of PRA are discussed in [RiskAnalysis-2006-Li].

This report provides a foundation for developing commonly accepted reliability failure mode taxonomy (or taxonomies) for digital systems. It documents feasible failure mode taxonomy which can be used for PRA modelling, related data collection, and quantification of digital I&C system reliability. The failure mode taxonomy of this report intends to support the future development of a consensus method or methods by identifying and categorising the failure modes at different levels of abstraction.

The taxonomy has been developed by using experts knowledgeable in digital system hardware and software architecture and experts with PRA knowledge to systematically study digital systems failure modes from a functional significance point of view.

The document can be used as a reference material for various purposes but it is not a normative standard. Therefore the user of the report should pay attention to the following issues when referring to this document:

1. Purpose of using the taxonomy. The taxonomy has been defined for the purpose to support PRA for nuclear power plants. If the taxonomy is used for other purposes than PRA or other context than nuclear power plants, the user will most likely have to adapt or complement the taxonomy.

2. Analysis method. The taxonomy assumes a functional analysis approach integrating top-down and bottom-up views, but it does not define any specific analysis approach. Therefore, the user needs in any case to define the analysis method of his or her own.

3. Definitions. Definitions are always subject to discussion and can vary in different contexts. The meaning of the definitions and the structure of the taxonomy are important, but the user may redefine terms.

4. Technology assumptions. The taxonomy is based on an example system described in chapter 5. This may imply limitations in the taxonomy defined in chapter 6. The user shall verify the correspondence of his or her analysis target with the descriptions provided in this document. Especially, if the taxonomy is applied to control systems, there are a number of simplifications made in this report, which are presumably not valid for control systems. Nevertheless, the structure of the taxonomy should apply to any I&C system, but the user needs to take into account complementary issues.

5. Modelling method. The modelling method has great influence on the needs for failure modes. This taxonomy implicitly assumes fault tree modelling type of approach even though the intention is to leave the modelling method as an open choice for the user. As long as the modelling follows the Boolean logic or discrete state space modelling paradigm, there should not be obstacles in using this taxonomy. However, this general approach has been developed in the course of the DIGREL task and its applicability and usefulness need to be validated in further research efforts.

6. Data sources and quantification method. The taxonomy is based on a collection of experiences to perform quantitative reliability analysis. However, the proposed taxonomy is defined in a generic manner and does not refer to any specific data source or quantification method for, e.g., software reliability. The user may need to develop a link between the taxonomy used in the data source or quantification method and the taxonomy of this document.

Chapter 2 presents an overview of the role of taxonomy and failure modes in the context of PRA. Chapter 3 defines main terms used in the document. Chapter 4 describes the general approach and assumptions. Chapter 5 defines the example protection system. Chapter 6 is the core of the document, providing the failure modes taxonomy. Chapter 7 complements chapter 6 with examples. In chapter 8, the taxonomy of chapter 6 is evaluated against requirements defined in chapter 4. Data sources and collection related issues are discussed in chapter 9. Chapter 10 outlines future work, and finally chapter 11 concludes the report. Appendix A summarizes a survey of failure modes applied for digital I&C systems. Contributors to the report are listed in Appendix B.

2. USES OF THE FAILURE MODES TAXONOMY WITHIN PRA

A failure mode taxonomy can be seen as a framework of describing and classifying failure modes associated with a system. Failure modes are basic elements of PRA. They represent different ways in which structures, systems, components (SSCs), and humans can fail and their failure effects are accounted for in the PRA model. Important requirements of a failure mode and effects analysis in supporting PRA modelling include (1) completeness of failure modes, (2) failure effects are clearly defined and can be propagated, and (3) quantification of the associated failure rates and probabilities. Failure modes taxonomy can be used also for the modelling and evaluation of the CCFs of the components of the digital I&C systems.

Besides to support the system reliability analysis, failure modes taxonomy is needed in the collection and statistical analysis of operating experience (failure data) of technological systems. Data from operating experience of digital systems obtained and analysed to date has been found to be inadequate, lacking statistically significant quantity, detail, and pedigree, validity, or quality (e.g., lack of context conditions) to identify and analyse failure modes and causal factors adequately. This lack of adequate data is partly exacerbated by rapid technological changes. This fact does not, however, eliminate the fundamental need of having a taxonomy when analysing operating experience, and it is foreseen that in future PRA community will have even statistically significant quantity of data on digital I&C systems at NPPs.

In a PRA of an operating NPP, the failure modes are usually well established and databases have been developed for the failure modes of the SSCs; while human reliability analysis methods have been developed for quantifying human errors. The current PRA method for a level 1 PRA is considered well established except for a few areas that are still subjects of additional developments and research, for example, enhancement of human reliability analysis, modelling of passive systems, etc. As PRAs can be used in different applications or for specific PRA applications, certain level of detail and quality may be required [RG-1.200]. An example may be the modelling of analog protection systems that is often done in a simplified way. That is, in some cases a single basic event is used to model the single failures of a protection system or the single failures of the actuation signals of an ESFAS (Engineered Safety Features Actuation System) train. On the other hand, some PRAs do include detailed fault tree models of the I&C systems. The detailed model of analog protection systems can potentially be applied to addressing Technical Specifications related issues such as test frequency and allowed outage time of I&C safety-related systems [WCAP-10271, NEDC-30851p].

With a shift in I&C technology from analog systems to digital systems with their functional advantages, plants have begun such replacement, while new plant designs fully incorporate digital systems. Modelling of digital systems in a PRA is an important challenge to the PRA community. For new reactors, for design certification purpose, a PRA has to be prepared. As a part of the PRA, the reactor vendors have developed fault tree models of digital systems, for example, AP1000 has a circuit-board level fault tree model of its plant protection and monitoring system [NUREG/CR-6962, PSA2013 Westinghouse]. The U.S. NRC has sponsored a few studies in modelling of digital systems using dynamic methods (i.e., [NUREG/CR-6985]) and a simulation based method [NUREG/CR-6997]. Internationally, fault tree models have been developed for reactor protection systems of several nuclear power plants, for example, see e.g. [PSAM10-Authen] and [IJCAS-2006-Lee]. The number of examples is growing.

The level of abstraction of the failure mode taxonomy is driven by the potential use of the reliability analysis. I&C unit failure modes and I&C module failure modes have been applied in the context of fault tree analysis, while the basic component level failure modes have been applied in a Markov model based analysis and in a simulation type of modelling [NUREG/CR-6997]. The failure modes can be used in a failure modes and effects analysis (FMEA) of a digital system as a part of the efforts in developing a reliability model. They also might serve as the targets of failure data collection for the specific system or plant. Level of abstraction will be further discussed in the next chapters of the report.

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 PRA. For protection systems, as discussed in Section 6, the system level failure modes may include loss of the function or functions, and spurious actuations. 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). See references [MIL-STD-1629A, Rausand-Høyland] for general guidance on FMEA, and the standard [IEC-60812] for I&C FMEA.

3. DEFINITION OF TERMS

This chapter provides a summary of definitions used especially in section 5 where the features of digital I&C systems are discussed and section 6 where the failure modes taxonomy is developed.

The general approach taken is to use the standard [ISO/IEC/IEEE 24765] as the general source of the definitions of terms used in this document. In several cases more than one definition is given in that standard. Therefore, the standard that is the original source of that definition is also shown in parentheses.

However, several definitions needed are missing in this standard. Therefore, definitions from other standards are included where necessary. The respective standards are shown in parentheses in the following list. The remaining definitions have not been taken from other sources but are specific to this work. For some terms no formal definition is given but instead the relevant aspects are described.

3.1 I&C system description terms

Application Software: software or a program that is specific to the solution of an application problem [ISO/IEC 2382-1]

Architecture: the organisational structure of the I&C systems of the plant. Derived from [IEC 61513]

Channel: A channel is a pathway from sensors to generation of an actuation signal.

Diversity: realisation of the same function by different means [ISO/IEC/IEEE 24765]. Example: use of different hardware, network technologies and architectures, storage media, programming languages, algorithms, or development teams.

Division: part of a system that is separated both physically and electrically from other parts. Different divisions typically contain redundant trains. "Division" is usually applied at the plant level.

Functional diversity: application of the diversity at the functional level (for example, to have trip actuation on both pressure and temperature limit) [IEC 60880]

Functional unit: an entity of hardware or software, or both, capable of accomplishing a specified purpose [ISO/IEC 2382-1]. Functional unit is a universal term for all kinds of units from a larger system down to a small component. Functional unit should not be mixed with I&C unit (see below). I&C units are a subset of functional units.

Hardware: physical equipment used to process, store, or transmit computer programs or data [ISO/IEC/IEEE 24765]

Level of abstraction: High level of abstraction means few details. Low level of abstraction means the opposite. Levels of abstraction considered from high to low in this taxonomy report are:

- system level,
- division level,
- I&C unit level,
- module level and
- basic component level.

Different levels of abstraction are illustrated in Figure 1. A more detailed description of the level of abstraction contained in Section 0.

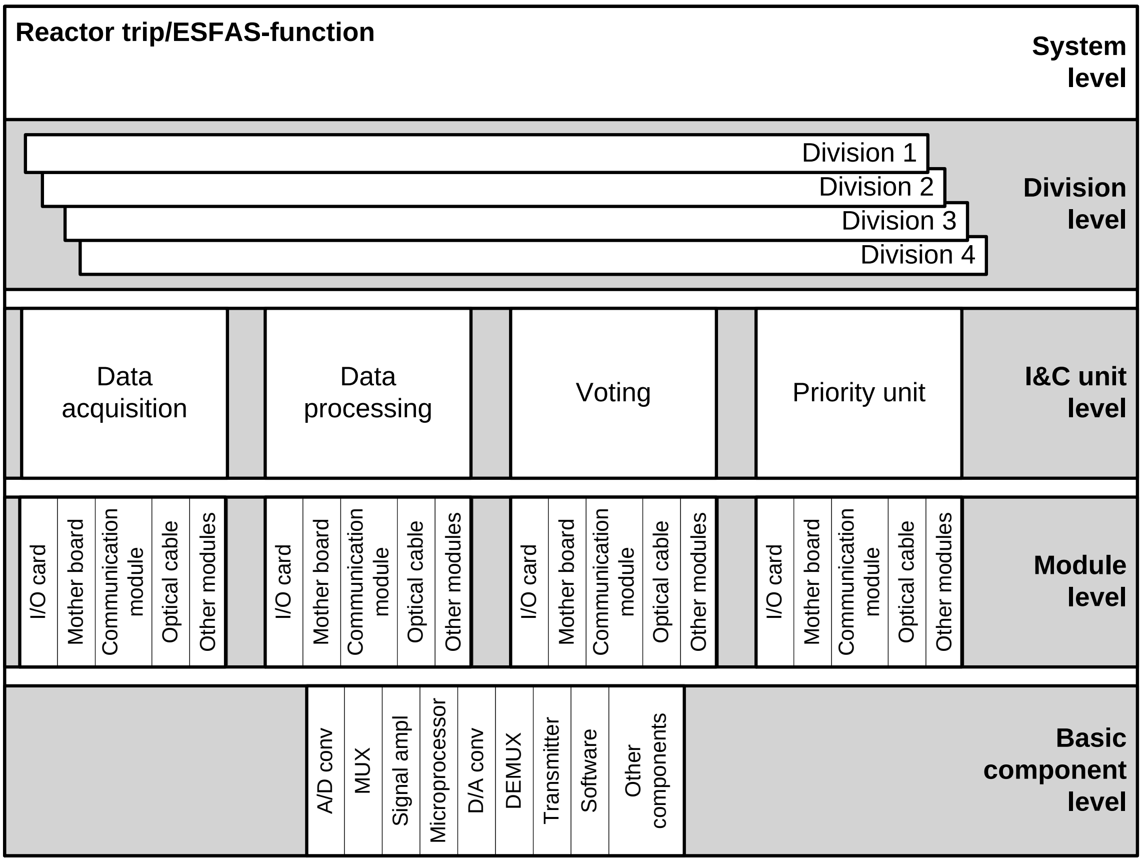


Figure 1. Principal structuring of a safety I&C system into different levels of abstraction.

Operating system: set of software that manages computer hardware resources and provides common services for computer programs.

Platform: set of hardware and software components that may work co-operatively in one or more defined architectures (configurations). The development of plant specific configurations and of the related application software may be supported by software tools. Platform usually provides a number of standard functionalities (application functions library) that may be combined to generate specific application software.

Reactor Protection System (RPS): those I&C systems which initiate safety actions to mitigate the consequences of design basis events. The protection systems include the reactor trip system (RTS) and the engineered safety features actuation system (ESFAS). [NUREG-0800]

Redundancy: the presence of auxiliary components in a system to perform the same or similar functions as other elements for the purpose of preventing or recovering from failures [ISO/IEC/IEEE 24765]

Software: all or part of the programs, procedures, rules, and associated documentation of an information processing system [ISO/IEC 2382-1]. Note: includes firmware, documentation, data, and execution control statements

System: a group of equipment that is configured as a system and operated to serve some specific set of I&C functions as defined in the I&C documentation of the plant.

Train: set of components providing totally or partially one or several functions of a system. A train is usually redundant to one or more similar trains, each with the same or similar capability to provide the specific function(s). “Train” is usually applied to mechanical systems.

3.2 Failure modes analysis terms

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.

Common cause failure (CCF): failure of two or more a structures, systems or components due to a single specific event or cause [IEC 62340].

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 (see Section 6).

Continuous detection: Detection by the monitoring feature, e.g., self-monitoring or alarms.

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.

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 [MIL-STD-1629A]. Note that this includes detection by the system (e.g. continuous detection, etc.).

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 24765], “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 25000]. Note: "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 60812] (“of the system” removed).

Failure mechanism: relation of a failure to its causes.

Failure mode: the physical or functional manifestation of a failure [ISO/IEC/IEEE 24765].

Fatal failure: I&C units or the hardware module stalls. It ceases functioning and does not provide any new computed signal. Fatal failures may be subdivided into:

> **Ordered fatal failure:** At time of failure, the outputs of the I&C unit or the hardware module are set to specified values. Equivalent to the definition “Halt/abnormal termination of function with clear message” [BNL NUREG 77124 2006 CP].
> Note: The means to force these values are usually exclusive hardware.
>
> **Haphazard fatal failure:** At time of failure, the outputs of the I&C unit or the hardware module have not been set to specified values. Equivalent to the definition “Halt/abnormal termination of function without clear message” [BNL NUREG 77124 2006 CP].

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 61508], “defect” added). The following definition is specific to software: An incorrect step, process, or data definition in a computer program (called also software development/implementation error) [ISO/IEC 25040]

Fault tolerance: The ability of a functional unit to behave despite the presence of hardware or software faults.
Note: 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-SSG3]. In this report, the standard PRA definition of an initiating event is followed.

Non-fatal failure: The I&C unit or the hardware module fails but continues to compute and communicate signals. Non-fatal failures may be subdivided into:

> **Failures with plausible behaviour:** I&C runs with wrong results that are not evident [BNL NUREG 77124 2006 CP]. An external observer (online or offline detection means) cannot determine whether the I&C unit or the hardware module has failed or not. The unit seems to be 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 [BNL NUREG 77124 2006 CP]. 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.

Periodic testing: periodic surveillance testing, e.g., defined in technical specifications. This is not same as the periodical (cyclic) self-testing of I&C systems, which corresponds to “continuous detection”.

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

Revealed by demand: It is an event in which the functional unit had failed, the failure had not been detected, and is revealed by demand.

Revealed by spurious actuation: It is an event in which the occurrence of the failure immediately triggers spurious actuation.

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 61508].

Spurious actuation: a failure where an actuation of an I&C function occurred without a demand.

Uncovering situation: The context where the failure becomes visible. The failure may become visible through dedicated “detection mechanisms” (see below), or failures may be discovered by a process event.

4. APPROACH

4.1 General approach

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

The activity has taken advantage from recent and on-going R&D activities carried out in the member countries in this field. This knowledge has been merged by inviting experts in the field to contribute to the activity. Example taxonomies have been collected from the member countries (for detailed information is referred to Appendix A), and analysed, and the conclusions from the taxonomy examples and workshop discussions have been taken into account when considering principles for the taxonomy. In chapter 5, a representative I&C system is outlined, and the fictive but realistic example is used as a reference to define and illustrate the failure modes.

In chapter 6, a failure model, the taxonomy itself and uncovering situations are introduced to define the failure propagation at different hierarchical levels of the I&C system. The proposed taxonomy is thus based on current practices to analyse and model I&C in PRA, features of typical digital reactor protection system and an analytical failure model.

The taxonomy has been developed jointly by PRA and I&C experts which have slightly different views and needs on defining the failure modes. The PRA experts' perspective follows the needs of PRA modelling in order to capture relevant dependencies and to find justifiable reliability parameters. I&C experts are focused on failure mechanisms and their recovery means. An important aspect in the development of the taxonomy has been for PRA and I&C experts to define the "meeting point" for the two perspectives. The "meeting point" means both agreeing on common terminology and defining the issues and levels of abstraction which the taxonomy shall address.

4.2 Scope

The taxonomy is deliberately focused on the reliability analysis of the reactor protection system, which reduces the scope of failure modes and failure effects considerably. This limitation can be justified by several arguments. Firstly, there is a general consensus that protection systems (reactor trip & ESFAS) shall be included in PRA, while control systems can be treated in a limited manner. Secondly, the system architecture and the mode of operation of protection systems versus control systems are different, which creates quite different basis for the reliability analysis and modelling. The I&C of the control systems is versatile having both on demand and continuous functions and they do not necessarily have a redundant structure. Even if the taxonomy is focused on the protection systems, it can be useful for control systems, too. The user of the taxonomy should be aware that some assumptions may not be valid for control systems.

One of the objectives for the taxonomy was to be as exhaustive as possible (see also ch. 4.4 for criteria). This is a target, which never can be completely fulfilled. The approach was to define failure modes at higher level of abstraction (system, division, I&C unit) in a generic manner. In this sense, the taxonomy should at least cover what is needed for PRA. At lower level of abstraction (module, basic component) a large number of example failure modes are given and they are associated with end effect at higher level. Example failure modes represent those typically applied in PRA, but exhaustiveness of failure mode taxonomy is not claimed.

4.3 Summary of collected taxonomies

One of the steps to develop a failure mode taxonomy was to collect current practices in the Nuclear Industry. A total of twelve organizations provided input:

- BNL (Brookhaven National Laboratory);
- CNSC (Canadian Nuclear Safety Commission);
- EDF (Electricity of France);
- IRSN (Institut de Radioprotection et de Surete Nucleaire);
- JNES (Japan Nuclear Energy Safety Organization);
- KAERI (Korean Atomic Energy Research Institute);
- NRG (Nuclear Research and Consultancy Group);
- NKS (Nordic Nuclear Energy Research); summarising input from three Nordic utilities
- OSU (Ohio State University);
- RELKO Ltd (Engineering and Consulting Services).

Each of the respondents provided a list of failure modes and a definition of the level of detail at which they performed the analysis. The responders were asked to include the software failure modes. Also a report from the Oak Ridge National Laboratories has been used as input [Korsah 2010].

In order to meaningful compare the failure modes, the taxonomies have been categorised in different levels of abstraction (see Figure 1). The distribution of the inputs over the different levels of abstraction is listed in Table 1.

Table 1. Distribution of the inputs over the different levels of abstraction

	System level/ Division level	**I&C unit level**	**Module level**	**Basic component level**	**Total**
Hardware	12	5	6	4	**27**
Software	4	-[1]	10[2]	4	**18**

Compared to the levels of abstraction defined for hardware, the I&C unit level is missing for software. This is not surprising, given the fact that also for the hardware levels of abstraction the distinction between I&C unit level and module level lead to discussion. However, the input of the responders on software did

[1] From the collected taxonomies there was no need to define an I&C unit level for the software failure modes.

[2] Four out of ten responders indicated software as a source of CCF.

not lead to a separate level of abstraction for an I&C unit level. This leads to the discussion if it is necessary to map software failures to hardware components and if so, how this is done correctly.

Most responders provided input for the module level, however the differences are small. Especially if it is taken into account that from the 10 inputs on software failure modes at module level, 4 responders only indicated software as a source of CCF. The collected input confirms that it is an open issue which level of abstraction is most appropriate for modelling digital I&C.

The most detailed taxonomies, both for hardware and software, were provided at the module level and basic component level. At these levels of abstraction it could be observed that responders did not have a homogeneous way of defining the failure modes. From the collected taxonomies two different types of failure modes have been identified:

- **Functional failure mode**: A failure mode regarding the effect on the function that is considered. For example, failure to actuate or spurious failure.
- **Structural failure mode**: A failure mode that includes the failure cause. For example, “frozen sensor” or “amplifier adjustment too low”.

The latter approach needs to be evaluated in order to determine their failure effects. The functional failure modes do give information about the effect, but not about the causes. In a reliability model they are likely to be combined, so that both the cause and the effect can be interpreted. A taxonomy needs to structure both approaches.

Additionally some responders defined detected and undetected failure modes. A taxonomy needs to include these attributes to reflect the practice of modelling in PRA.

In total 6 out of 18 software failure mode taxonomies indicated software to be a source of CCF. Which leaves 2 taxonomies at system level, 6 taxonomies at module level and 4 taxonomies at basic component level for further analysis. It is anticipated that the software failure modes need further development.

4.4 Requirements for the taxonomy

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 detail of hardware and 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 PRA: the wish to have a realistic and complete taxonomy (or PRA model) and on other hand to have a practical, usable and understandable taxonomy (or PRA model). There is a pressure both towards perfectionism and towards simplifications so that a balance must be decided between these targets.

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 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. In practical application, analysis is constrained to plausible failure modes, but from the taxonomy point of view, it may be difficult to define a general rule to judge which failure modes are relevant for certain components and which are not.

As a conclusion, the used approach to develop a taxonomy compromises between the simplicity and completeness targets. Plausibility arguments have also been used to discuss the relevance of some failure modes and failure effects.

Following the general principles of taxonomy construction and the particular requirements set by the domain of study, i.e. failure modes for digital I&C systems for application to PRA practice, the following set of criteria have been defined:

- **Criterion 1: Be defined unambiguously**
  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 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 not to 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 PRA 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. It should be noted that the meaning of the criterion is not to exclude failure modes for which there is no data available.

- **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/or have similar failure modes. For example, if the failure modes of a component at a given taxonomic level are “High” and “Low”, it would be preferable if the failure modes of another component at this same taxonomic level be also “High” and “Low”, if applicable, instead of “Very high”, “High”, “Low” and “Very Low” provided that important aspects of system operation for the PRA are still well represented. This concept is similar to the use of consistent failure modes for mechanical components in PRAs (e.g., use of “fail to open” and “fail to close” for motor operated valves). 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 PRA modelling**
  Dependencies can be divided into [IAEA-SSG3]: functional dependencies, physical or spatial dependencies, human interaction dependencies and component failure dependencies. 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 PRA practice, and fulfil PRA requirements/conditions**
  This criterion comprises of a wide range of aspects, which vary between PRA projects, e.g.,

  - Feasible for PRA experts to apply in the systems analysis

  - Possible to implement into existing tools

  - Possible to review by a PRA-expert

  - Allows living PRA, e.g. possible to maintain and update with reasonable resources

  - Available and maintainable failure data, i.e., allows collection and evaluation of operational events

  - Support PRA 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 demand 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, i.e., which uncovering situations are valid for the failure mode.

The failure modes taxonomy is evaluated against these criteria in chapter 8.

5. EXAMPLE SYSTEM

5.1 General assumptions

In general the design of the digital I&C of the safety important systems can be implemented on the basis of very different technologies (e.g. distributed micro-processor based automation system, supervisory control and data acquisition system, PLD (e.g. FPGA, ASIC) based instrumentation, logic and actuation equipment) and platforms (e.g. AREVA Teleperm XS, Rolls-Royce Spinline™, Invensys Triconex). The differences between different I&C platforms and softwares may be significant, not only by the physical design but also the functional design, e.g. using of different fault tolerant features, implementation of different architectures or topologies of the signal processing and of the voting logic.

A representative fictive digital protection system example has been developed to be used as a reference in the application and demonstration of the taxonomy. Even though there are technical differences between solutions provided by different vendors, many of the features of protection systems are similar for all vendors. Therefore the example is considered representative enough for the failure modes taxonomy purposes.

In putting together the hypothetical digital I&C system with safety important functions, several assumptions were made, which are discussed below.

The simplified model takes into account the following:

- Typical architecture of digital I&C systems performing safety functions of the Reactor trip system (RTS) and the Engineered Safety Features Actuation System (ESFAS), jointly called hereafter RPS functions
- Typical hardware components of the digital I&C platforms
- Typical operation modes of the RPS: ready to actuate a safety function on demand (maintenance, testing, etc. modes are not considered)
- Typical means and features for failure detection and recovery
- Typical majority voting for actuation of RPS functions.

5.2 Description of the example system

5.2.1 General remarks

The general principles for the evaluation of the failure modes taxonomy in Section 4 are very ambitious and require a clear picture of the functional relationships of a digital I&C system regarding pathway from a failure cause to the probable effects. It is not possible to propose a profound and meaningful failure modes taxonomy completely in the abstract and without any assumption regarding the I&C system architecture, hardware and software. Therefore, this section describes an example system on which the taxonomy proposed in Section 6 will be applied in order to derive a failure modes taxonomy suitable for PRA purposes.

This example system is not based on any specific real-life system. However, its features are representative of those of actual computer-based reactor protection system, though in a simplified manner to leave out unnecessary details that could obscure the proposed taxonomy construction approach.

The purpose of the example I&C system is the generation of the safety-important functions (e.g. start of the safety injection pumps, closing of the containment isolation valves) including actuation of the corresponding components (e.g. drives, motors, solenoid valves and switches) in a nuclear power plant based on signal processing of the inputs received from the processes situated at the field level (e.g. sensors, valve position indicators).

The safety standards define requirements regarding design for reliability of structures, systems and components. The highest quality of and best practices for hardware and software shall be used for digital I&C of safety systems, considering the following criteria [IAEA NP-T-3.12, IAEA NS-R-1]:

- Compliance with single failure criterion,
- Robustness concerning common cause failures,
- Principle of fail-safe design.

The architecture, the equipment (hardware) and software of the digital safety-important I&C (I&C platform) shall be designed, manufactured, qualified and used to meet all safety I&C requirements in nuclear power plants.

Consideration of equipment failure modes is given in the design of I&C systems (e.g. using of redundant signal processing, cyclic execution of the application software, implementation of diversity principles, automatic self-surveillance testing, implementation of the fail-safe principle, using of voting and priority logic) to make their functions more robust and tolerant of expected failures of systems or components. The design of systems and equipment should strive to ensure that the range of possible failure modes is predictable and that the most likely failures will always place the system in a safe state (see paragraphs 4.49-4.50 of [IAEA-NS-G-1.3]).

The dissimilarities between different I&C platforms may be significant. Not only the physical design but also the functional design, e.g. performance, fault tolerant features and voting logic, may differ. However 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.
In course of a system analysis (e.g. PRA) the most common way to subdivide the I&C by functional groups is the following:

- Sensors: to interface with the physical processes within a plant and to continuously take measurements of plant variables such as neutron flux, temperature, pressure, flow, etc.;
- Operational control, regulation and monitoring systems: to process measurement data, to manage plant operation, and to optimize plant performance. Surveillance and diagnostic systems that monitor sensor signals for abnormalities are important parts of operational monitoring systems;
- Automation of the safety systems: to keep the plant in a safe operating envelope in case of any postulated initiating event (design basis accident);

- Communication systems: to provide data and information transfer through wires, fibre optics, wireless networks or digital data protocols;

- Human-system interfaces (HSIs): to provide information to and interaction with plant operating personnel;

- Supply systems: to provide power supply of the I&C equipment, HVAC (Heating, Ventilation and Air Conditioning) of the I&C rooms and cabinets;

- Actuators (e.g., valves and motors): to adjust the plant's physical processes.

5.2.2 The example system architecture

The example system implements I&C safety functions (of Category A according to [IEC 61226], of Class 1E according to [IEEE-323-2003]. The overall system architecture describes its organisation in terms of divisions and I&C units.

The development of a generic digital I&C system was based on examples of the implementation of the following different platforms of digital I&C systems for safety functions:

- Teleperm XS (e.g. EPR reactor design, modernisation projects in several NPPs)

- Common-Q/Advant AC160 system (e.g. AP1000 reactor design)

- Tricon PLC system (e.g. ESBWR reactor design).

The architecture of a digital I&C system is established primarily by hardware (e.g. analog and digital circuit boards/modules, units, cabinets) and their communication paths (e.g. direct wired connections, network communications, signal distribution boards). The architecture determines essentially the propagation paths of the probable failures of the hardware and of the software.

The generic architecture of the example system presented here can consider different architectures of the digital safety-important I&C systems. The example system is organised into two separated subsystems SSA and SSB, which are based on the same I&C platform, but implement functions that are diverse. The two subsystems do not exchange information, and for shared actuators, their outputs are fed to simple hardwired logic to determine system-level outputs.

In one case the sub-systems A and B (SSA and SSB) can be interpreted as implementation of the functional diversity inside of a redundant architecture of a system based on a common platform (Note: this term suggests strong commonalities or similarities in the hardware and in the software). In the other case the sub-systems A and B can be interpreted also as redundant divisions of the diverse I&C systems based on different platforms (diverse hardware and software, see more in [NUREG/CR-7007]), such as the redundant divisions of a primary and of a secondary (diverse) protection system.

The overall example system architecture can thus be summarised as shown in Figure 2. Each subsystem is itself organised into four redundant divisions, each division of subsystem being composed of different types of I&C units, namely:

- Acquisition and processing units (or APUs): these units acquire process-related information from sensors, and perform calculations to determine the division outputs. Each subsystem 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.

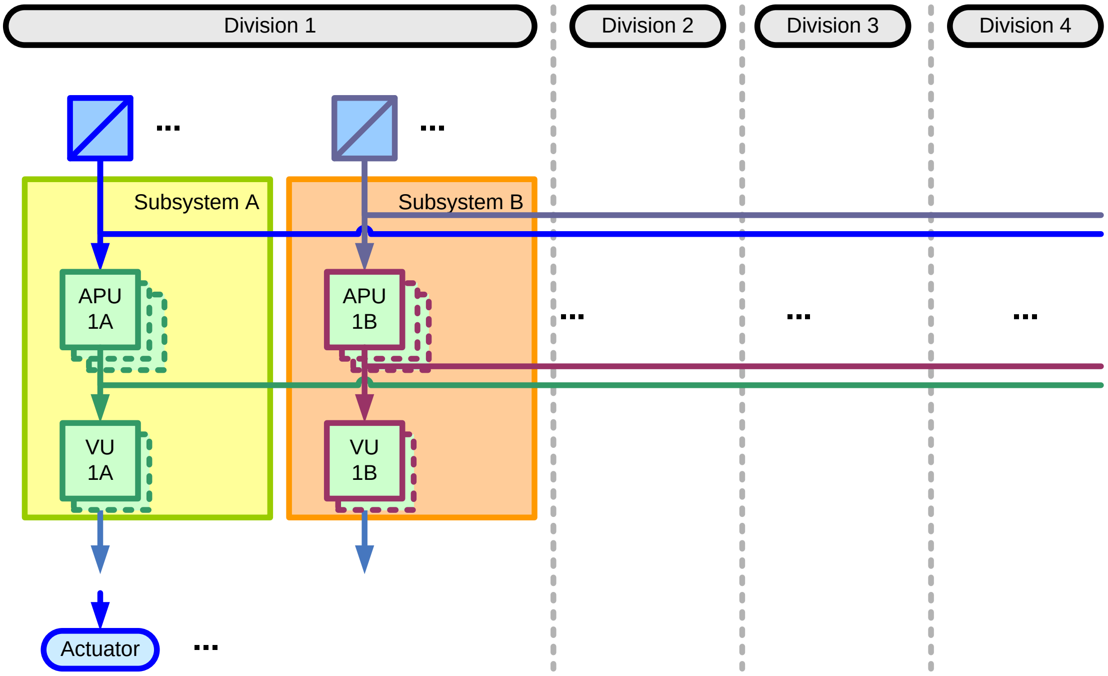


Figure 2. Overall architecture of the example system. APU = Acquisition and processing unit, VU = Voting unit. Data communication units are integral parts of APUs and VUs and not shown in the simplified figure. Details of the architecture are further discussed in next chapters.

5.2.3 Hardware architecture

The hardware architecture describes the hardware organisation of individual I&C units in terms of hardware modules and basic components. The generic structure of the hardware of the entire signal processing consists of the following kinds of hardware components [IAEA-NP-T-3.12]:

- Processor modules for signal processing,

- Communication modules,

- Input and output modules of digital or analog signals,

- Electrical items such as electrical connections, cables and power supply modules.

- Mechanical components such as subracks, fans and cabinets for housing and cooling the above modules.

Figure 3 presents a simplified hardware architecture of a generic example digital safety-related I&C system similar to Figure 2 (the difference is that in Figure 3 there is only one APU per each division and subsystem while Figure 2 assumes that the functions may be allocated between several APUs).

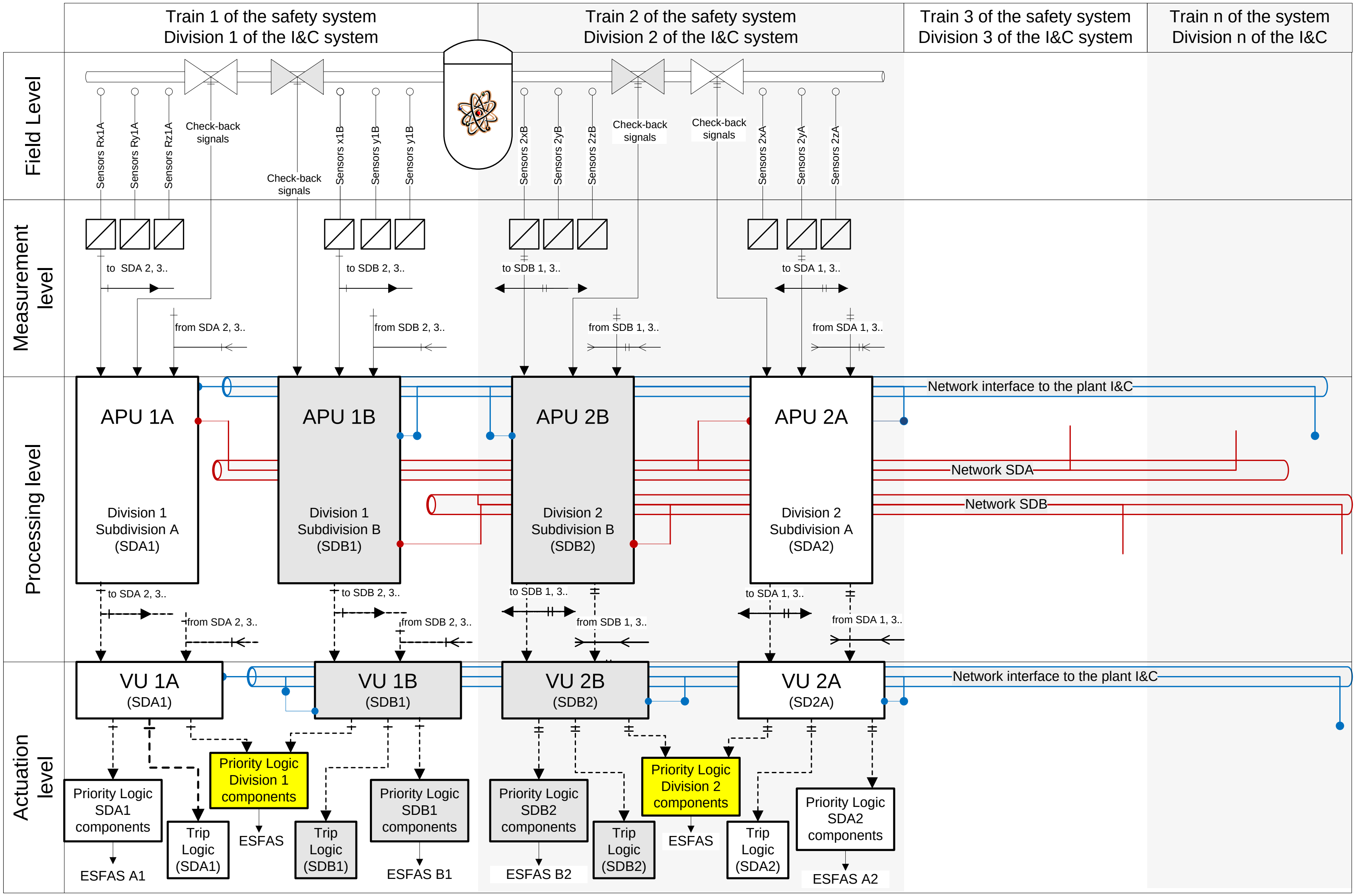


Figure 3. Hardware architecture of the example system (white blocks belong to subsystem A, grey boxes to subsystem B, yellow to both)

.

There are four levels of components described in this generic model:

- field level (e.g. sensors, actuators),
- measurement level (e.g. transmitters, signal distribution boards),
- signal processing level (e.g. I/O modules, processor modules, communication modules),
- actuation level (e.g. voting logic units, commands prioritization logic).

The field level includes instrumentation such as sensors that measure various characteristics of the processes in the nuclear power plant.

The measurement level includes devices such as analog to digital converters that convert the signals from the processes into digital signals to the processing units.

The processing level acquire the signals from various redundant paths and process the information into signals that are input to the voting logic of the actuation level.

The actuation logic includes the trip logic and the priority logic components for various redundant paths.

The signal processing from sensors to actuators in Figure 4 is simplified to allow the identification of the main features (items of hardware and their functions) and main signal pathways (e.g. networks) inside and outside of the divisions and subdivisions. For reasons of clarity the connections with the external systems (e.g. connection to the plant network, messaging and service interface) are not included.

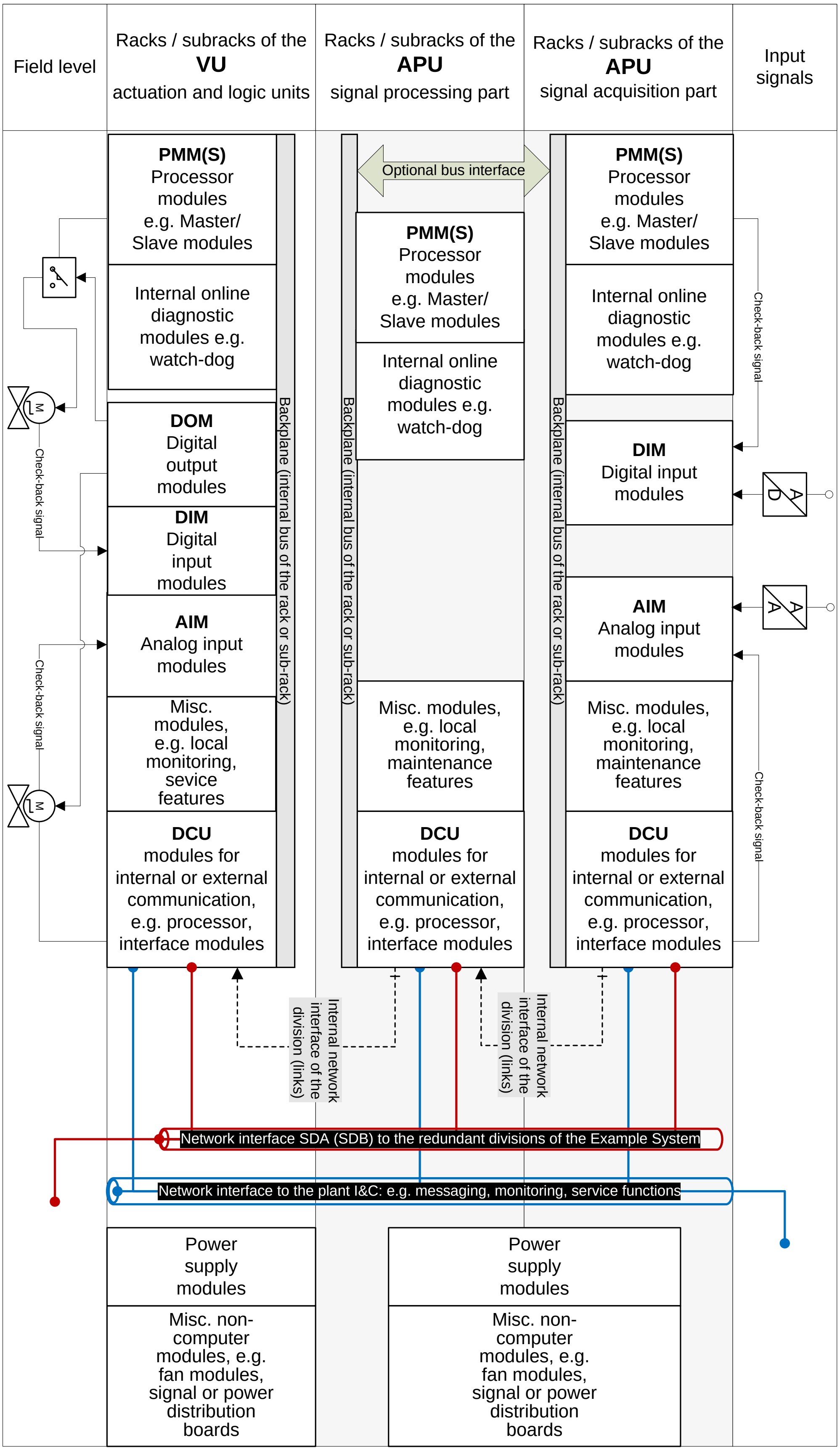

Field level
Racks / subracks of the VU actuation and logic units
Racks / subracks of the APU signal processing part
Racks / subracks of the APU signal acquisition part
Input signals
PMM(S) Processor modules e.g. Master/ Slave modules
Internal online diagnostic modules e.g. watch-dog
DOM Digital output modules
DIM Digital input modules
AIM Analog input modules
Misc. modules, e.g. local monitoring, sevice features
DCU modules for internal or external communication, e.g. processor, interface modules
Backplane (internal bus of the rack or sub-rack)
Optional bus interface
PMM(S) Processor modules e.g. Master/ Slave modules
Internal online diagnostic modules e.g. watch-dog
Misc. modules, e.g. local monitoring, maintenance features
DCU modules for internal or external communication, e.g. processor, interface modules
PMM(S) Processor modules e.g. Master/ Slave modules
Internal online diagnostic modules e.g. watch-dog
DIM Digital input modules
AIM Analog input modules
Misc. modules, e.g. local monitoring, maintenance features
DCU modules for internal or external communication, e.g. processor, interface modules
Check-back signal
M
A D
A A
Internal network interface of the division (links)
Network interface SDA (SDB) to the redundant divisions of the Example System
Network interface to the plant I&C: e.g. messaging, monitoring, service functions
Power supply modules
Misc. non-computer modules, e.g. fan modules, signal or power distribution boards

Figure 4. Example of hardware modules in APU and VU inside the divisions of the example system

.

The processor modules usually implement the following tasks:

- Execution of the operating system and application software (e.g. I&C functions),
- Data accesses to the input and output modules, to the interface modules,
- Monitoring features.

The components for communication provide signals, data and information transfer through wires or fibre optics by using networks and data protocols [IAEA-NP-T-3.12], [ORNL/NRC/LTR-07/05]. Typical components are:

- Communication processor modules,
- Communication modules (e.g. interface modules),
- Server and gateways.

The analog or digital input/output modules can provide following analog input/output capabilities:

- Single or multiple analog input/output channels
- Connection features to the internal bus,
- Integrating measuring principle,
- Power supply of the sensors,
- Isolation (decoupling) features.

The above modules and network components are typically organized in racks and cabinets. They are comprised of:

- Internal computer buses,
- Internal power supplies for the I&C modules from the external power supply,
- Packaging systems with the mechanical installation features for the I&C modules,
- Cooling systems (e.g. natural convection and/or forced ventilation by fans),
- Monitoring equipment (e.g. alarm, power supply, temperature).

Some ancillary units can be connected via their own data communication units to the example systems, e.g. the engineering computer or workstation. The engineering computer is used mainly to perform the maintenance work on the I&C units (e.g. update of the software or modification of configuration of the module, modification of setpoints or calibration parameters). The operator interface can also allow the modification of the specified changeable parameters such as setpoints. Both are usually physically disconnected during normal operation and are therefore not further taken into consideration.

Digital modules can be further decomposed into basic digital components, e.g., microprocessors and A/D and D/A converters, multiplexers (MUXs)/demultiplexers (DEMUXs). Regardless of vendors, the functions of individual basic components are clearly defined, e.g., A/D converter is always used to convert analog signals to digital ones.

The basic components from functional point of view are common to all digital systems and independent of specific system architecture, and higher level digital units, i.e., modules. All I&C units, and systems are composed of several electronic parts (e.g. diodes, amplifiers, memory chips, timers) and therefore their model principally can be constructed on basic components. For example, Figure 6 shows the "internal" essential components of an analog input module which converts analog inputs into digitalized signals (digitalized analog outputs of the figure). The analog backplane bus interfaces with all input and output signals of the module. An internal bus is used for the microprocessor of the module to interact with components connected to the backplane bus. A current-loop (CL) device produces a current output (usually 0–20 mA). Each analog input is assumed to use one current loop. The digitalized analog outputs will be delivered to other modules via the backplane bus. Other basic components in the figure, for example, basic input/output system (BIOS), are also standard in digital systems. The arrows represent signal flows between different components. Note that there are more active and passive components in a module, for example, resistors and capacitors that are not shown in the figure. These components are needed to support the essential components and are considered a part of the essential components that they support.

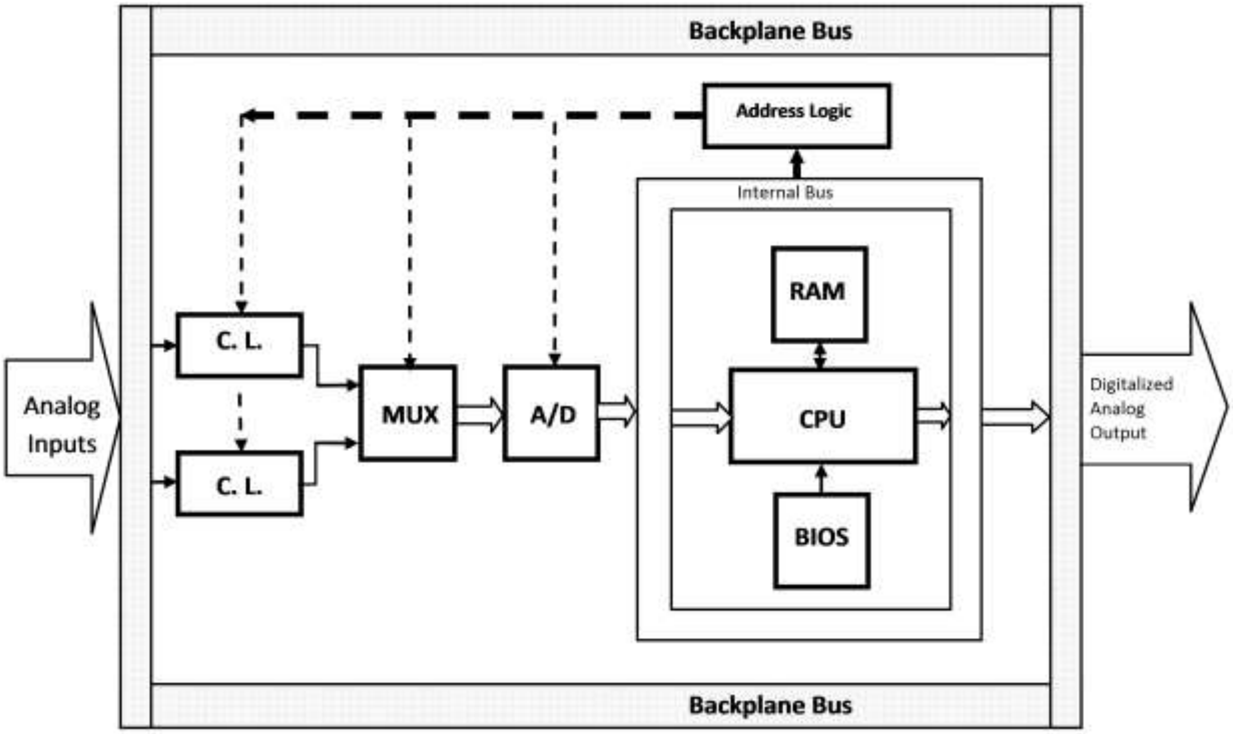


Figure 5. Major components of an analog input module (AIM).

Note, these are major components that are commonly used in generic digital systems. The reason of listing only major components in the diagram is the difficulty in claiming the completeness of basic components used in digital systems, e.g., a vendor may produce a new type of components of special functions for a particular digital system. (Table 11 provides a more complete list of generic components).

5.2.4 Software architecture

The software architecture describes the software organisation of individual I&C units in terms of software modules.

5.2.4.1 APUs

The software of an APU can be decomposed into the following software modules:

- Operating System (OS): this module controls the overall functioning of the APU. Its main functions are to initialise the unit, to manage input-outputs and data communication, to perform auto-tests, to process operator requests, and to activate the application-specific software (see Figure 6). It is part of the platform software: this means that the OS is the same for all the APUs of all divisions or of all sub-divisions (SSA or SSB) of the example system.

- Elementary Functions (EFs): these modules provide readily useable standard (library) functions such as Boolean logic, mathematical functions or delays. This means that they are the same for all the APUs of the example system. However, an important difference with respect to the OS is that a specific APU will use only a specific subset of all available EFs.

- Application-specific Software (AS): typically, an APU supports multiple application functions, and is an AS module. Homologous APUs in redundant divisions have the same sets of AS modules. There could be functional dependencies between application functions (functional requirements FRS-F, -G, -H, -J and application-specific functions AS-F, -G, -H, -J in Figure 6). In Figure 7, the function G is functionally dependent on function F, which is indicated by the downward arrows in the figure (information flow between modules). This relationship is not symmetric.

- Proprietary Elementary Functions (EF): Functions whom the implementation in software belong to a commercial company. It does not perform a function of its own. The source code is not freely nor publicly available. It is restricted from use, such as modification or V&V, for the end user. However he may use the elementary functions to design its own programs, e.g. application software. The Proprietary Elementary Functions are tightly linked with the technology of the platform and its vendor.

In addition to these "concrete" software modules, it is worthwhile to also include "virtual" software modules representing the Functional Requirements Specification (FRS) of each application function expected of an APU. This allows taking into consideration of errors in functional requirements specifications. Analyses of operating experience (cf. EPRI TR 1021077) show that such errors are not uncommon.

Further software parts which could be considered are the data table that specifies the hardware configuration of the APU, and also the data table that specifies the data communication allowed for that APU. These tables are used by the OS to "know" which types of I/O modules are available (and at which addresses), to which networks it is connected, and which data messages it can send or receive through each network. These tables are automatically generated by the I&C platform engineering tools, based on a specification of the overall system architecture. Although postulated errors in these tables might be detected during system tests, the data tables could be impacted by other system failure events such as a single event upset hardware failure. Due to the complexities of assessing these types of hardware/software interaction failures, this issue was considered to be outside the scope of this report. However, for PRA purposes, an assessment should be done to consider the importance of this type of APU failure to system safety function.

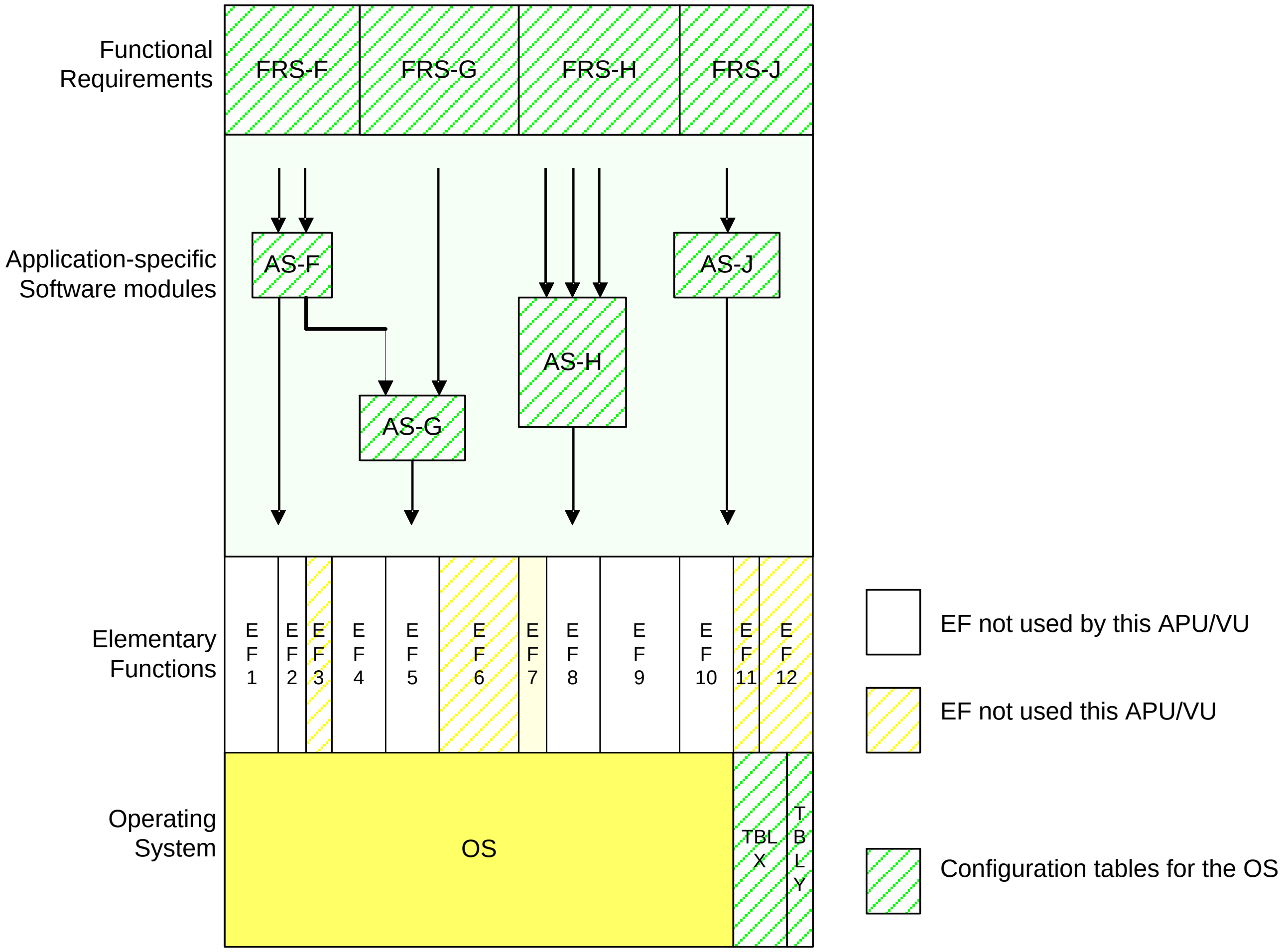


Figure 6. Software architecture of an APU or VU.

5.2.4.2 VUs

The organisation of the software of a VU is to a large extent very similar to that of an APU:

- Operating system (OS): the OS of a VU is the same as the OS of the APU.

- Elementary functions (EFs): the set of EFs at the disposal of a VU is the same as for the APU. However, as voting logic is mostly Boolean, VUs are less likely than APUs to use EFs which use integer or floating-point arithmetic operations.

- Application-specific software (AS): the various AS modules of a VU implement the specific voting logics required for the I&C functions implemented by the subsystem to which the VU belongs. The two subsystems use different voting logics, and therefore the AS modules of their VUs tend to be different.

- Functional requirements specification (FRS): as for APU, these virtual modules allow the representation of errors in the specification of voting logic.

5.2.4.3 DCUs

The organisation of the software of a DCU can be decomposed into the following software modules (Figure 7):

- Operating system (OS): the OS of a DCU is the same as the OS of the APUs and VUs.
- Data communication software (DCS): this module implements the data communication protocol. It is part of the platform software, and all DCUs of the example system have the same DCS.
- Data link configuration (DLC): this module is provided in the form of a data table. It specifies the nodes that can be part of a given network, and the data messages that can be exchanged between the nodes of the network. The two subsystems use different networks, and therefore the DLCs of their DCUs are different.

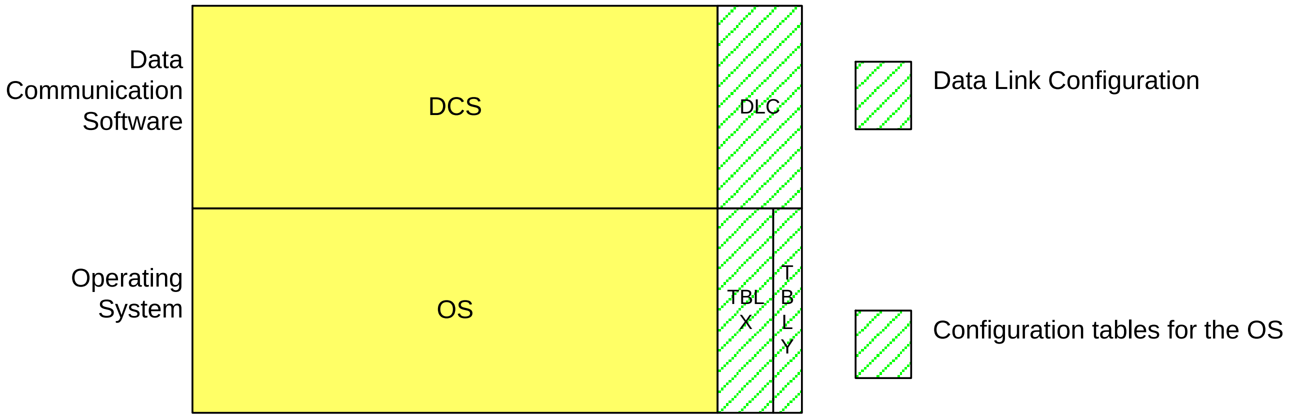


Figure 7. Software architecture of a DCU.

5.2.4.4 Software in other hardware modules

Some hardware modules such as input or output boards may be microprocessor or microcontroller-based, and may contain their own software. Such software is part of the platform software and cannot be altered by the I&C system designers.

Such software is designated here as “software in COTS”. It is defined as a code that is embedded in a specific hardware module, different from microprocessor module of APU, VU and DCU, and that performs a function of its own. For example, functions such as power supply control, signal processing in a smart sensor are performed by the software in a hardware module. This software is proprietary and is generally not available for the end user. Except for some parameters that may be changed, it is not reusable or reprogrammable by the plant operator, and often even not by the I&C vendor. Such hardware modules are of the COTS category (Commercial Off the Shelf).

The level of complexity of such software is usually significantly reduced compared to the application software of the APU, VU and DCU. The development of the embedded software is fundamentally different from that of non-embedded software. Technologies for the development of embedded software should address specific constraints such as hard timing constraints, limited memory and power use, predefined hardware platform technology. Therefore this type of software is not comparable with the EF regarding handling of the faults, pattern, errors during the development and V&V process. As such, development of a taxonomy covering embedded software is considered out of scope of this report. However, the PRA analyst should consider the risk importance of COTS components in a holistic manner that treats the hardware and software as a single functional unit.

6. TAXONOMY

6.1 Introduction

This chapter presents a taxonomy that supports the representation of digital I&C protection systems in PRA. One of its purposes is to help the PRA analysts work closely with I&C experts.

The review of existing taxonomies (ch. 4 and appendix A) demonstrates that the failure modes and mechanisms they consider differ in level of detail, and represent large sets, especially for software related failure mechanisms. Thus, to be usable while still being comprehensive, a taxonomy requires assumptions regarding its scope and the I&C system, its architecture, including hardware and software aspects. Also, as the purpose of the taxonomy is to support PRA, it therefore focuses on functional aspects with little details rather than structural aspects with many details. These assumptions are summarized in a digital I&C architecture model and a failure model. Thus, in some way, the "Failure Mode Taxonomy" is practically a series of classifications supporting a model of I&C failures.

The taxonomy is based on a failure propagation model and the hierarchical definition of five levels of abstraction. Four important elements of the failure model on which the taxonomy focuses stand out:

1. fault location,

failure mode,

uncovering situation,

failure effect and the end effect.

These concepts are applied in particular at the I&C unit and module levels. Module level failure modes are associated with the end effects at the I&C unit level, depending on the fault location and uncovering situation. The assessment of the End Effect of the propagation requires in addition the notions of Failure Origin, Maximum Potential Effect, and Most likely Potential Effect. All these terms are defined in this chapter.

Assumption about the fault location in I&C units and modules is necessary to assess the end effects. Only four categories of failure local effects are used at the different levels of abstraction, such that they can be used to develop reliability models of digital systems without too much additional work on identifying them.

These assumptions permit to handle variability of failure modes and mechanisms of I&C components. They reduce the difficulties associated with the complex structural aspects of software in redundant distributed systems.

The taxonomy is based on the I&C example presented in Chapter 5, typical of existing digital I&C systems platforms which have been used to implement safety functions, such as Teleperm XS, Common-Q/Advant AC160 system, Tricon PLC system, etc. I&C protection systems may differ from the example used here and adaptations of the approach have to be done. It may also be used for other categories of I&C systems (control systems, systems with sequential outputs), with some adaptations.

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

6.2 I&C architecture and taxonomy

The architecture determines the locations of software elements and the propagation for failures of the hardware and software. As defined in chapter 3, different levels of abstraction can be considered (see also Figure 1):

- The *System Level* is the abstraction level corresponding to the complete I&C system, e.g., the reactor protection system, the diverse reactor protection system or an entity defined as a system that provides a set of protection functions. It is divided in redundant and possibly diverse divisions. Subsystems are considered to be at this level for the purpose of this taxonomy. In the example (see ch. 5), it is assumed that the system (and both subsystems) has four more or less identical *divisions*.

- The *Division Level* is the abstraction level corresponding to physical separations of the I&C system. A division consists of *I&C units*.

- The *I&C Unit Level* is the abstraction level corresponding to elements implementing specific functions that are essential for an I&C system in rendering its intended services. I&C unit categories are defined by their generic functions in a protection system, e.g., data acquisition, data processing, voting and data communication. An I&C Unit consists of *modules*.

- The *Module Level* is the abstraction level corresponding to sets of hardware elements supporting specific tasks like input/output-cards, CPU boards, backplanes, communication cards, etc., and software elements performing well-defined specific functions like operating system, application specific software, power supply management, etc. They are implemented on hardware modules, and may be distributed. Note that this definition of software module is specific to this taxonomy for PRA and does not correspond to the definition used in computer science.

- The *Basic Component level* is the abstraction level corresponding to hardware components like resistors, CPUs, RAM chips or ASICS mounted on a circuit board, and software components implemented on them. At this level of abstraction, software is assumed to be of proprietary nature and a black box for further analysis (see discussion on “SW in COTS” later in this chapter).

Failure effects at one level of abstraction become failure modes at the next (higher) level of abstraction as illustrated in Figure 8.

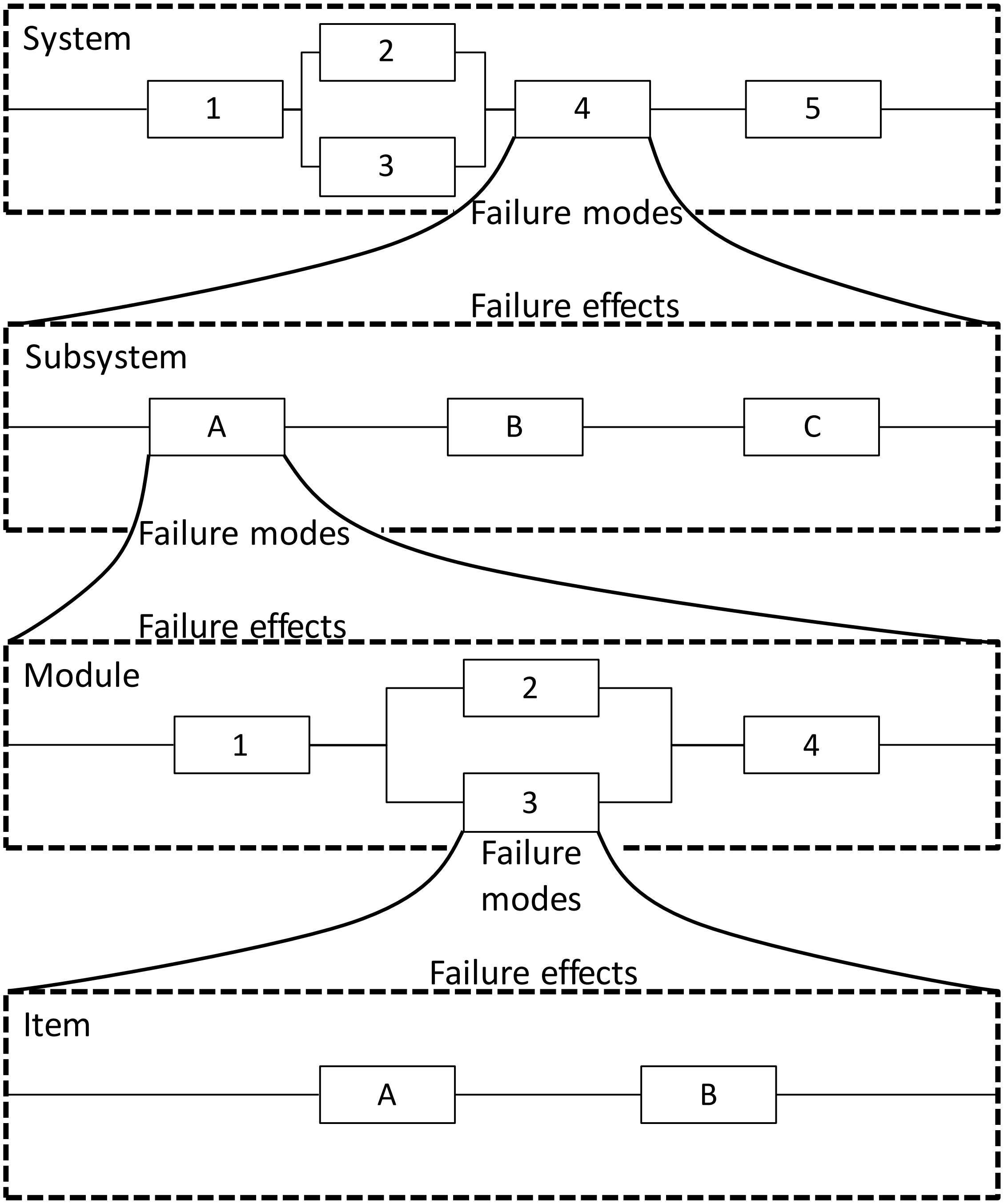
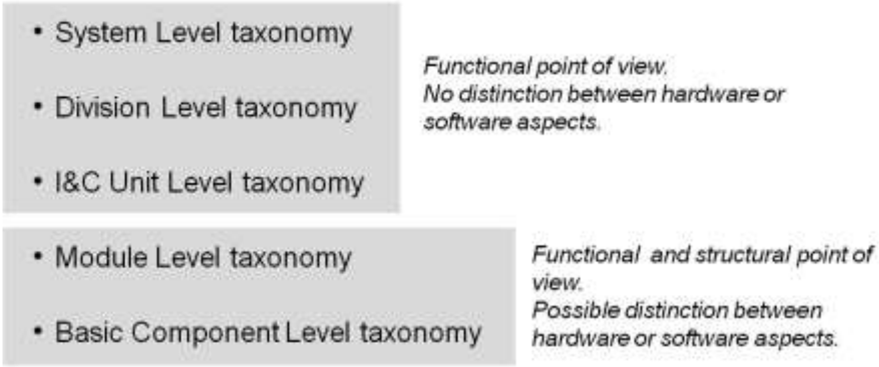


Figure 8. Relationship between failure modes and failure effects (based on [IEC 60812]). Note that "item" is equivalent to "basic component" and "subsystem" is equivalent to "division/unit" levels defined in this report.

To handle complexity, at the level of system, division and I&C units, failure modes are considered only from the functional point of view. No significant distinction is made between hardware or software aspects at these levels. At module and basic component levels, the taxonomy may differentiate between hardware and software related failure modes (see Figure 9).



- System Level taxonomy
- Division Level taxonomy
- I&C Unit Level taxonomy

Functional point of view.
No distinction between hardware or software aspects.

- Module Level taxonomy
- Basic Component Level taxonomy

Functional and structural point of view.
Possible distinction between hardware or software aspects.

Figure 9. Taxonomy levels and relevant point of views.

The taxonomy has been developed to be used in a top-down approach or a bottom-up approach. The top-down approach begins with the system functions, identifies failure effects on the functions and continues down to units, modules and even to basic components, if needed. The bottom-up approach starts from the failure modes and effects of the modules or basic components, and can be formalized by a FMEA. The PRA practitioner has to choose a suitable level of detail for each individual PRA and PRA application.

6.3 The failure model

6.3.1 Representation of an I&C function

The failure model is based on the representation of the actuation of a Line of Defence (see Figure 10). The actuation of the physical line of defence, that prevents an initiating event becoming a hazardous event, is the purpose of the protective functions of the I&C system. Such functions are called in this taxonomy “I&C Safety functions”. Only these functions are usually of practical interest for PRA.

The actuation of a line of defence is due to an initiating event which occurs within a given state of the plant, designated hereafter by the term “plant condition”. The output of the I&C system is determined by the plant condition. Also, some plant conditions require the set up of operational bypasses. There is a possible dependency between the initiating event and the plant condition: for example some events may occur only under some plant conditions, e.g. during low power and shutdown.

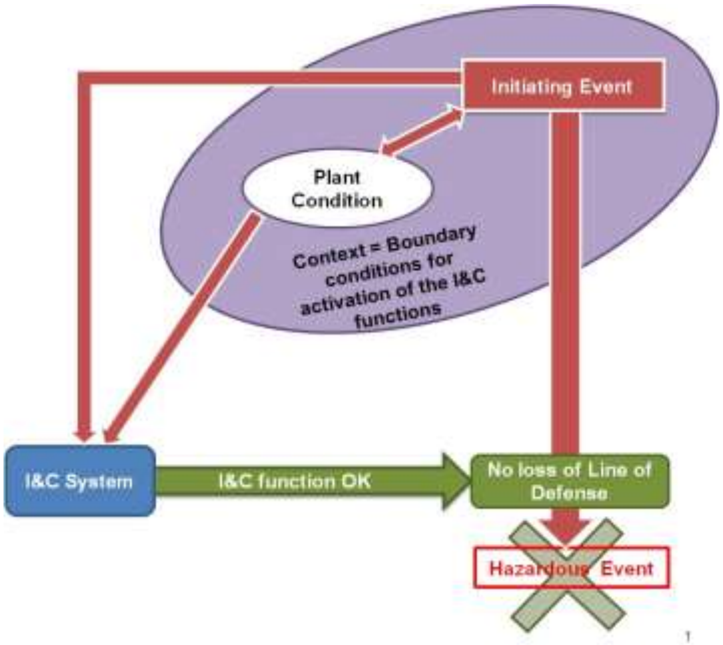

Initiating Event
Plant Condition
Context = Boundary conditions for activation of the I&C functions
I&C System
I&C function OK
No loss of Line of Defense
Hazardous Event

Figure 10. Representation of the actuation of a line of defense.

In this representation, the initiating event and plant condition constitute the "context" of the I&C system. It is expected from I&C safety functions that they function in accordance with the contexts they are specified to handle with.

6.3.2 Effect of context on I&C system

On the basis of this representation and I&C architecture, a "failure propagation" model is developed, with the basic assumption that failures are activated by an interaction between the context and the I&C system.

According to the definitions (see chapter 3) a fault is a state that exists in a place of the hardware or the software, the fault location. A fault becomes a failure under the occurrence of external events, the so called "activation conditions". The effects of the fault activation can be significant or not, depending on the fault location.

The fault location is the place where the failure is triggered. It may be a gate in an IC, an open circuit on an electronic board, a fault in the software, and so on. It may include multiple physical locations, especially in a situation of fault in a software that is implemented in redundant boards. In the model, the fault location is the division, unit, module, component in which a fault (as defined in 3.2.) is postulated.

Various events may correspond to activation conditions: high loads, high data burst, buffer overflow, unusual/unexpected data or signal constellations, time dependency, a failure in a support system due to an initiating event, a signal due to a plant state change, etc. In this situation, the initiating event, plant condition and activation conditions constitute a new, sometime unexpected context of the I&C system, different from normal conditions. Moreover, these three elements may be correlated. For example, the frequency of an initiating event may be influenced by activation conditions. The output of the I&C system is thus determined by the context as a whole, not only by the initiating event. The output may be a failure to actuate the line of defence, a spurious actuation of the line of defence, etc. (Figure 11).

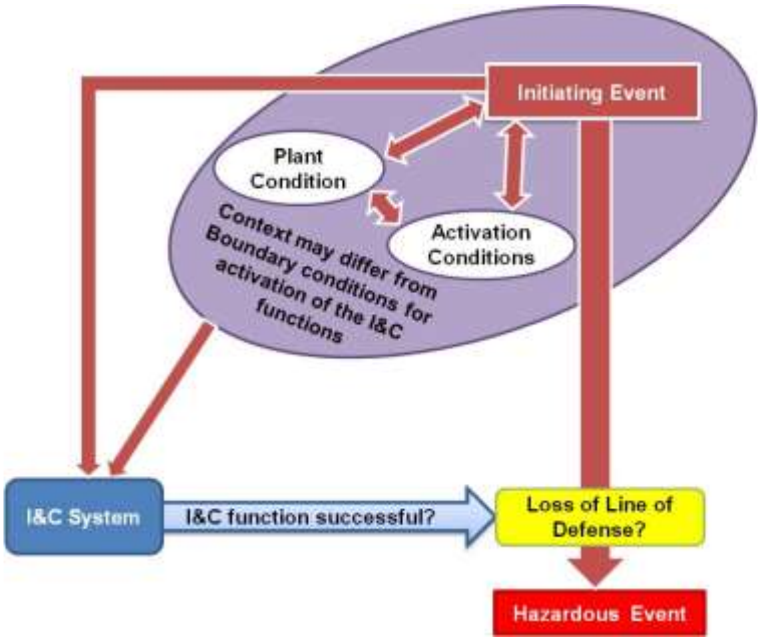


Figure 11. Representation of the effect of context on I&C system.

6.3.3 Representation of a Failure Propagation

Once activated, the Failure may propagate. Two basic modes of failure propagation are defined for this model and the end effect assessment:

- Propagation through common cause failure (CCF). In this model, CCFs are considered as a particular kind of failure propagation. Two or more failures occurs simultaneously in the I&C system because of the presence of a given context and a same fault in different physical locations. For example, physical stresses affecting different components of the system hardware at the same time, systematic software failures, hardware failures due to design and manufacturing may be the CCF.

- Propagation through cascade failures. A failure causes a wrong output in a part of a system that becomes a wrong input in another part of the system, and so on. Random hardware failures, systematic software failures, hardware failures due to design and manufacturing may be the cause of such propagations.

These basic modes may combine. Also, the propagation model does not exclude single failures, as they may affect a critical single point or combine with partial loss of redundancies due to another failure propagation occurring independently in a same time period. This variety of cases is a particularity of an I&C system combining software and hardware in a distributed, redundant architecture, and has to be analysed as part of the assessment of failure propagation.

The I&C architecture determines the failure propagation of the hardware and software. The context, in particular the activations conditions may affect multiple levels of the system and aggravate the propagation of the failure (Figure 12).

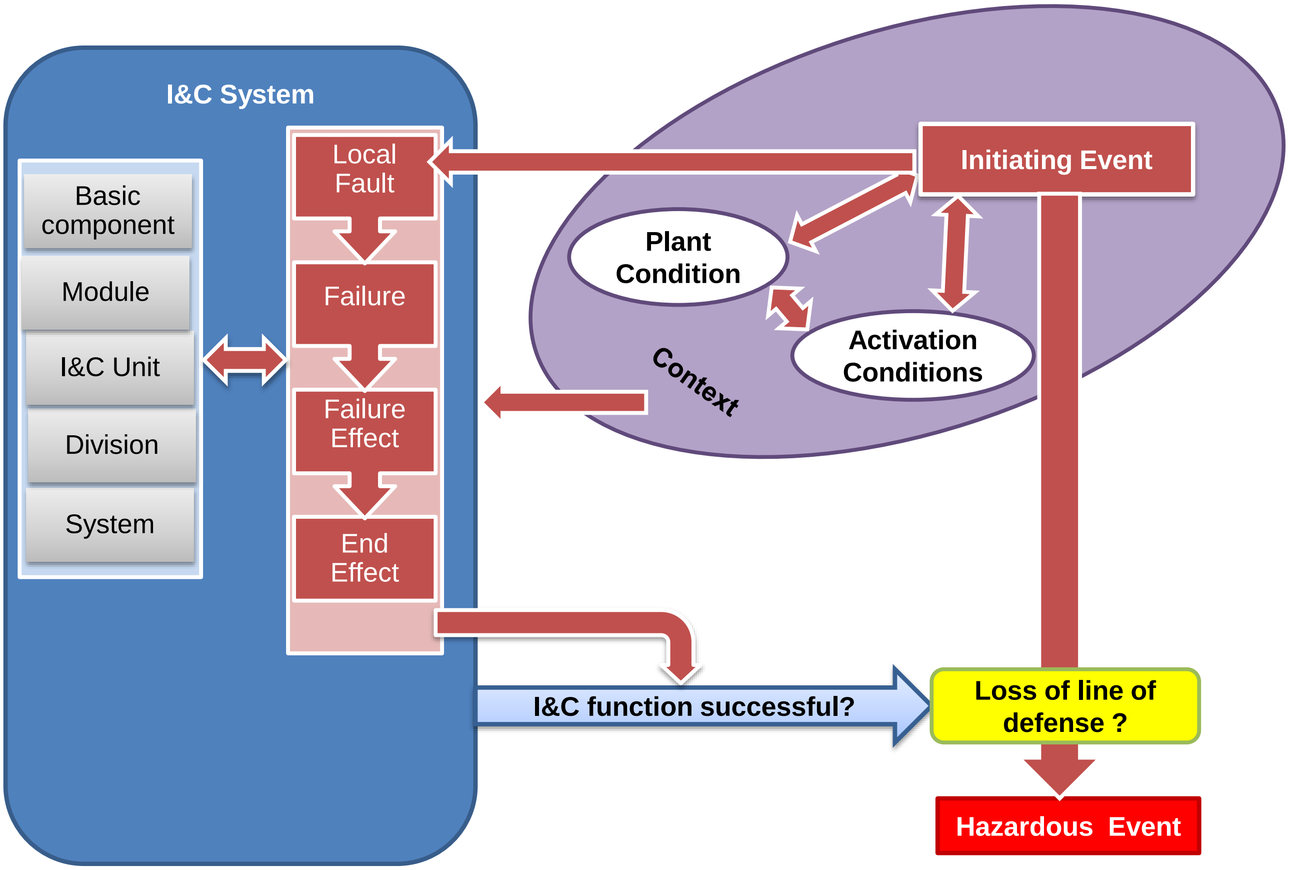


Figure 12. Representation of the Failure Propagation in an I&C system.

6.3.4 Complete representation of the failure model

There are many possibilities of failures, the so-called failure modes. They depend on the fault location, failure activation condition, failure origin, etc. Also, there are many possible combinations. For example, a given activation condition may eventually induce a “fail-low” failure mode of a module, while inducing a “fail-high” failure mode of another module unit. There are too many possible situations to build an exhaustive description of failures for all levels of abstraction.

Therefore, the model focuses on a limited but exhaustive set of failure effects. It is sufficient to describe the effect of wrong output in a location of I&C on another location. The taxonomy uses the same set of generic failure effects for different I&C levels (units, modules, basic components), so-called failure effects. The classification described in this chapter shows how the set of failure effects substitute to an exhaustive representation of failure modes. At a given level, a failure effect is local: for example, a stall at a level corresponds to a wrong output at another level.

To compensate or prevent failure propagation, fault tolerance design (FTD) features (diversity, monitoring, etc.) are organized at multiple levels of the system according to a defence in depth strategy and may stop the failure propagation. The context, in particular, the activations conditions, influence the ability of the system to handle the failure properly.

The end effect describes the final propagation of the failure, taking into consideration all these elements.

- If FTD features are present and effective, the failure will be detected and handled. The end effect of the failure propagation is predetermined.
- If FTD features are not effective or do not exist, the failure propagates up to having a more or less hazardous, more or less unknown in advance, end effect at various possible I&C architecture levels.

There are too many possible FTD features and potential failure propagation to build an exhaustive taxonomy from them, in the context of this study.

Thus, the following four elements form the basis of the taxonomy:

- fault location,
- failure effect,
- uncovering situation,
- end effect (maximum possible and most likely).

The end effect is identified after a specific analysis, the so-called propagation assessment. The taxonomy offers the guidelines to achieve this assessment. Three additional concepts are used for this analysis:

- Failure origin
- The maximum possible end effect assumes that FTD features are not effective or do not exist
- The most likely end effect assumes that FTD features are present and efficient.

They are described and defined in 6.11.3.With these elements and assumptions, the model handles the various cases of propagation presented in 6.3.3.

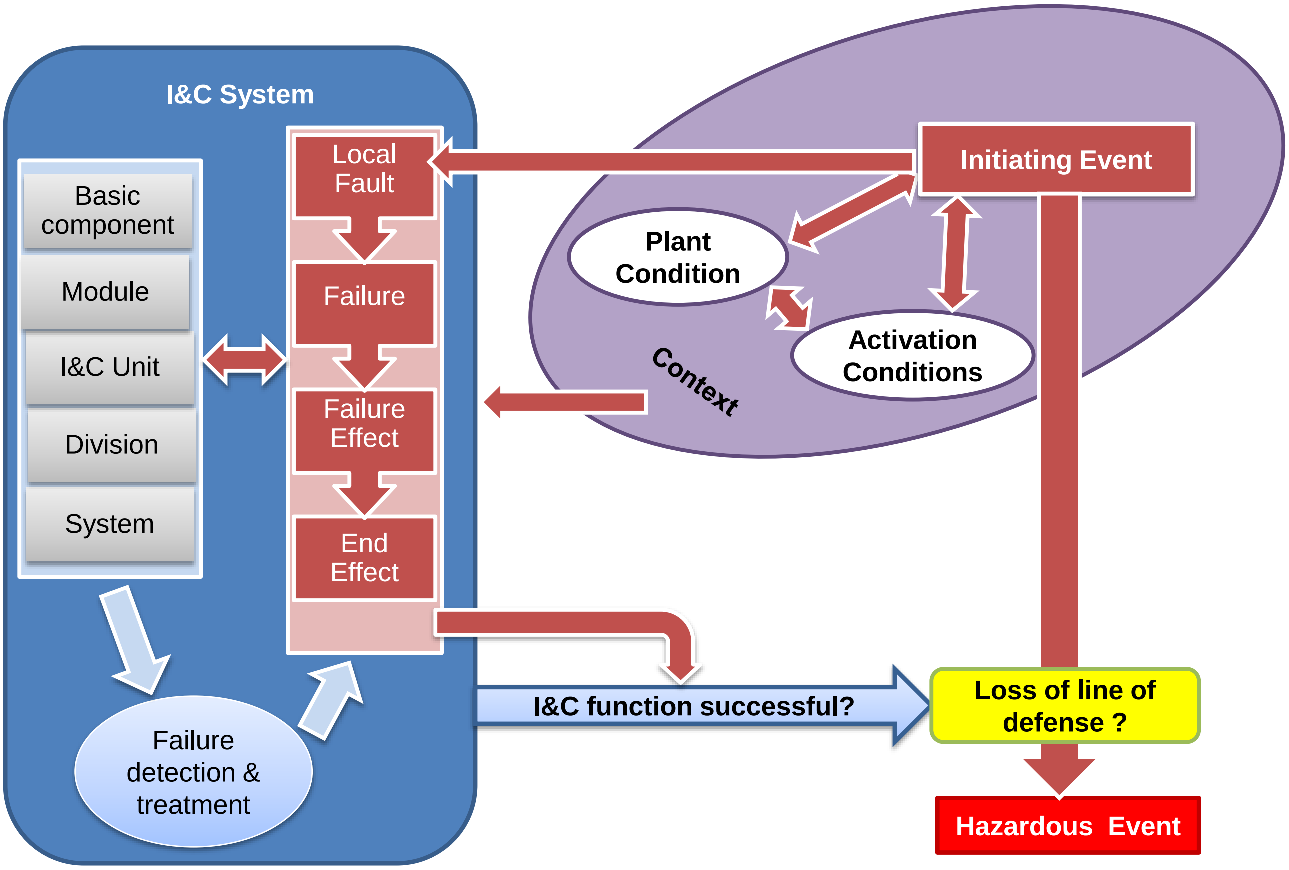


Figure 13. Complete representation of the failure model.

6.3.5 Example of a random hardware failure

Let us assume that a hardware fault is present in one single location of the protection system, due to a random cause. The activation condition may be a temporary external stress (like an electrical overstress), or a lasting environmental condition (moisture, heat...). By definition, this failure occurs, in term of time and location, independently from other failures.

Once detected, fault tolerance features may compensate or prevent the propagation of the failure. Fault may also remain latent and be uncovered by demand. As the failure is unique, there is no effect at system level, the line of defence is activated and the initiating event has no consequence. In PRA model, single failures are taken into account by corresponding basic events and the effect of combinations of single failures and CCFs is the natural part of the fault analysis.

There is the possibility of multiple failures may be activated independently in a short time interval in many locations, so that the redundancy strategy of the system may not manage the failures. Also, non-detected random failures may accumulate up to cause a loss of redundancy if the detection capabilities are not adequate.

6.3.6 Example of a spurious actuation

The model handles also the cases of spurious actuation. In such case, there is no initiating event requiring the I&C function. But some activation conditions are sufficient for the initiation of a latent fault and propagation of the failure. For example, a software fault addressing the management of priority orders in some plant conditions; a calendar condition, the mishandling of a periodic test or maintenance operation, an extreme data burst generated by unusual operational mode or transient. In addition, the detection means may not be capable of identifying the spurious actuation as a failure, as it corresponds to a safety function.

Thus the failure may propagate up to the system level with a spurious actuation of a line of defence as an end effect.

6.4 Classification of failure effects

Given the large variety of possible failure modes at I&C units, modules and basic component levels, the taxonomy does not pretend to provide any classification of failure modes at every level of the architecture. They have to be identified on a case per case basis. A classification of failure effects is used instead. In particular, at I&C unit, module and basic component levels, the failure effect classification is based on the effect of the output failures on other units, modules and basic components.

Four categories of failure effects are defined. Each of these failure effects are considered to have a significant effect on safety functions by either affecting the actuation of a line of defence, having a cascading effect, or otherwise affecting a defined safety function. . These four categories are not only sufficient but also exhaustive way to represent the functional effects and the apparent behaviour of a failed I&C unit, module or basic component. This classification permits to make the relation between PRA at plant level and the failure analysis done specifically for I&C. The classification is independent of the architecture, defence in depth and diversity strategies of the protection system.

- **Fatal:** The functional unit ceases functioning and does not provide computed output anymore. In other words, the I&C “stalls” [BNL-NUREG-77124-2006-CP]. Although the term “fatal” is associated to a complete and definitive failure, it may be in fact, in the case of a system using defensive safety measures, a relatively benign failure, as it is easy to detect online by external watchdog, for example. In some cases, the absence of a signal could be the right response of the element to specified input conditions.

  – **Ordered fatal failure:** At time of failure, the outputs of the functional unit are set to pre-determined values. The means to force these values are usually based on hardware. Functional requirements specify how to set the values of the outputs. An example is the failure effect “Halt/abnormal termination of function with clear message” [BNL-NUREG-77124-2006-CP].

  – **Haphazard fatal failure:** At time of failure, the outputs of the functional unit have not been set to predetermined values, the element is in an unpredictable states. In other words, there is no supervision of the failure. An example is the failure effect “Halt/abnormal termination of function without clear message” [BNL-NUREG-77124-2006-CP]

- **Non-fatal:** The functional unit fails but continues to compute and communicate outputs. In other words, the *I&C functions with wrong outputs*. Non-fatal failures may have worse consequences than Fatal Failures, as they are more difficult to detect.

  – **Failures with plausible behaviour:** the I&C runs with wrong results that are not evident [BNL-NUREG-77124-2006-CP]**.** An external observer (with online or offline detection means) cannot decide whether the functional unit has failed or not, with a given plant condition. The element seems to be still in a state that is compliant to its specifications, or compliant to the context perceived by the observer.

  – **Failures with implausible behaviour:** the I&C runs with evidently wrong results [BNL-NUREG-77124-2006-CP]. An external observer can decide that the functional unit has failed. It is clearly in a state that is not compliant to its specifications, or not compliant to the context perceived by the observer.

6.5 Classification of uncovering situations

Uncovering situations describe how the failure can be perceived by an observer external to the I&C system. The failure perception may not necessarily be at the place and the moment of the failure activation.

Uncovering situations can be divided into two cases:

1. identified without an actual demand

   – the failure is detected by a detection mechanism

   – spurious actuation, i.e., the activation of the fault triggers spurious actuation. It should be noted that spurious actuation may be caused by a detection mechanism. Spurious actuation may also cause an actual demand for other I&C functions (e.g. a plant trip).

2. identified during an actual demand, i.e., the failure occurred when the system was called upon to perform its intended function.

Depending on the failure mode and underlying fault, one or more uncovering situations may be relevant. Some failure modes are detected immediately by a detection mechanism or spurious actuation. For others uncovering may happen during an actual demand.

In this report, the cases are further divided into four possible uncovering situations, as described in Figure 14 and following chapters.

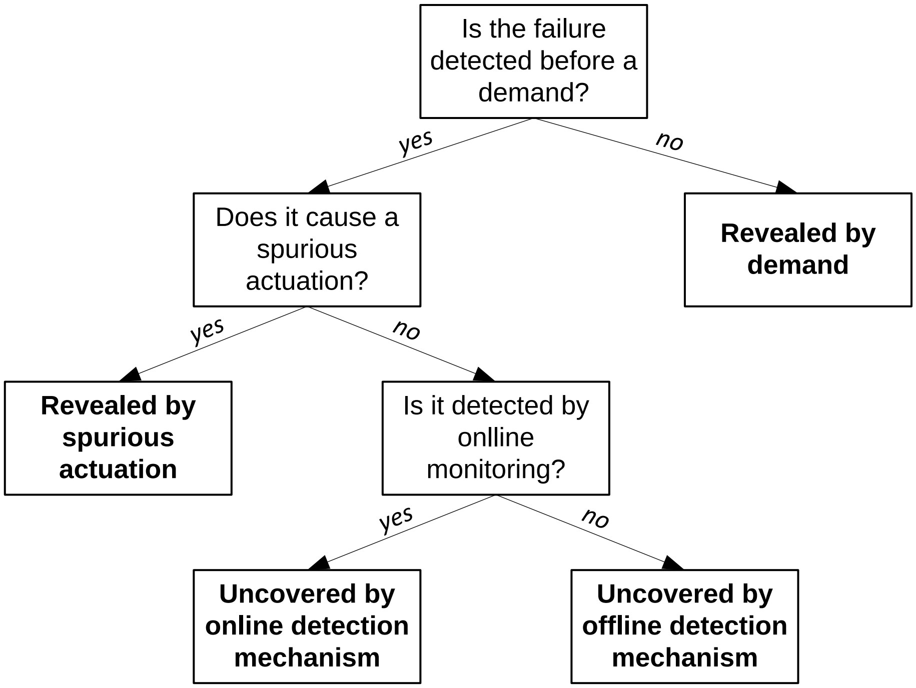

Is the failure detected before a demand?
yes
no
Does it cause a spurious actuation?
Revealed by demand
yes
no
Revealed by spurious actuation
Is it detected by onlline monitoring?
yes
no
Uncovered by online detection mechanism
Uncovered by offline detection mechanism

Figure 14. Classification tree for uncovering situations.

6.5.1 Uncovering by detection mechanisms

Detection mechanisms are classified according to their capacity to detect a latent failure online (e.g. build-in automatic test features) or offline (e.g. periodic test, maintenance). These two attributes influence the test coverage (that is more exhaustive for offline detection) and the duration between two tests (that is longer for offline detection). They are important parameters for the calculation of the probability of failure on demand. The detection mechanisms lead to predetermined states:

- *Online detection mechanisms*. They permit a fast failure treatment by the system itself. Transient failures are covered by these mechanisms.
  - Self-monitoring: the detection mechanism is in the same location as the fault.
  - External monitoring: the detection mechanism is in another location than the fault.
- *Offline detection mechanisms*, i.e. periodic testing and other controls (maintenance). They complete on line detection but with a relatively low frequency.

In both situations, there is a risk of failure on demand, if the demand occurs between two periodic tests or before repair. Automatic actuation by FTD features may eliminate this risk for online detection.

6.5.2 Revealed by spurious actuation

Revealed by spurious actuation is an event in which the occurrence of the failure triggers spurious actuation. Spurious actuation may happen before a demand or it may cause a demand for other I&C functions (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 a detection mechanism has time to take place. This situation covers two variants:

- Spurious actuation due to functional failure, including voting logic.
- Spurious actuation due to failure of FTD features.

The failure effect is different from the case with online detection mechanism since FTD has not taken place.

Other root causes that may lead to a spurious actuation are not directly related to digital I&C and are not considered in the taxonomy:

- Failure of some part of hardware inside an actuator leading to a spurious actuation
- Failure of sensors.

6.5.3 Revealed by demand

Revealed by demand is an event in which the functional unit had failed, the failure had not been detected, and is revealed by demand. In some cases, failures could have been detected by offline detection mechanism but the demand happens before the periodic test. Some failures cannot be detected by offline detection mechanism, e.g., specification errors.

6.6 Taxonomy at the system level

At this level, the fault location is the system as a whole. To identify the failure effects at the system level, the safety function assigned to the system should be clearly identified first. Practically, a safety function of the system is defined as the generation of one or several safety actuation signals in a predefined time interval only when required. The four possible combinations shown in Table 2 can be found by considering the required system output, the actual system output and their possible states, i.e. actuation, no actuation, late or partial actuation.

The example considers safety functions with a “Boolean output”. It should be noted that a safety-related actuation signal generated late is considered equivalent to no actuation because the above definition of the safety-related function includes time constraints. In this case, failures occur when the safety-related actuation signal is not generated when required (failure-on-demand) or when the safety-related actuation signal is generated when not required (spurious actuation). Therefore, the failure modes at the system level are combinations of required output and system output at system level and are described from a functional point of view (see Table 2).

This table addresses reactor shutdown function and actuation of safety features of a RPS. The taxonomy is identical for subsystems. Note that for some categories of functions, that are not Boolean, outputs may have to be redefined.

6.7 Taxonomy at the Division level

At this level, the fault location is the division as a whole. Practically, the safety-related function of a division is also defined as the generation of the safety-related actuation signal within a predefined time interval only when demanded. Thus, the failure modes in the division level are similar with those of the system level (See Table 2).

Table 2. Failure Modes at system and division level.

Required output Actual output	Actuation	No actuation
Actuation	Success	Failure (Spurious actuation)
Late actuation, Partial actuation	Failure (Failure-on-demand)	Failure (Spurious actuation)
No actuation	Failure (Failure-on-demand)	Success

6.8 Taxonomy at the I&C Unit level

6.8.1 Rationale of the taxonomy at I&C Unit level

It is based on the failure model, and use the following set of attributes:

- I&C unit category
- Failure effects at I&C unit level
- I&C unit failures uncovering situation.

The taxonomy at I&C units level does not differentiate explicitly between hardware and software.

Reminder: The PRA of a plant represent failures of I&C outputs (signals, commands, orders) produced by channels (see definition in chapter 3). As the outputs of channels are built at the I&C unit level, a great attention has to be put on the interaction of the I&C unit and the module levels. The connection of the unit and module levels corresponds to different kind of modelling and is a key issue.

6.8.2 I&C unit category

This classification defines the possible fault locations at I&C units level and contribute to assess failure propagation. It is based on the example system. At this level, the fault location is an I&C Unit as a whole. A failure may affect different I&C units if it is propagated through CCF or cascade failure (see 6.3.3.). An I&C unit may belong to one of the three categories hereafter:

- An a*cquisition & processing unit (APU)* acquires process-related information from sensors, and performs calculations to determine the division outputs. It may also process operator requests related to the functions they implement (such as the modification of a setting point).
- A *voting unit (VU)* receives results determined by the APUs of their division and subsystem and for which voting is required. It also receives decisions made by the APUs regarding operational bypasses. It exchanges information with other VUs across division boundaries, but not subsystem boundaries).
- A d*ata communication unit (DCU)* allows 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.

There is a limited variety of I&C units in such a protection system, this classification is practically independent of the architecture, defence-in-depth and diversity strategies [see ISO/IEC/IEEE 24765] of the protection system.

In the example system, all VUs and DCUs in a same subsystem have identical hardware and software. There are five groups of APUs (APU1–APU5) in sub systems A and B (See Figure 15) that have identical hardware and software.

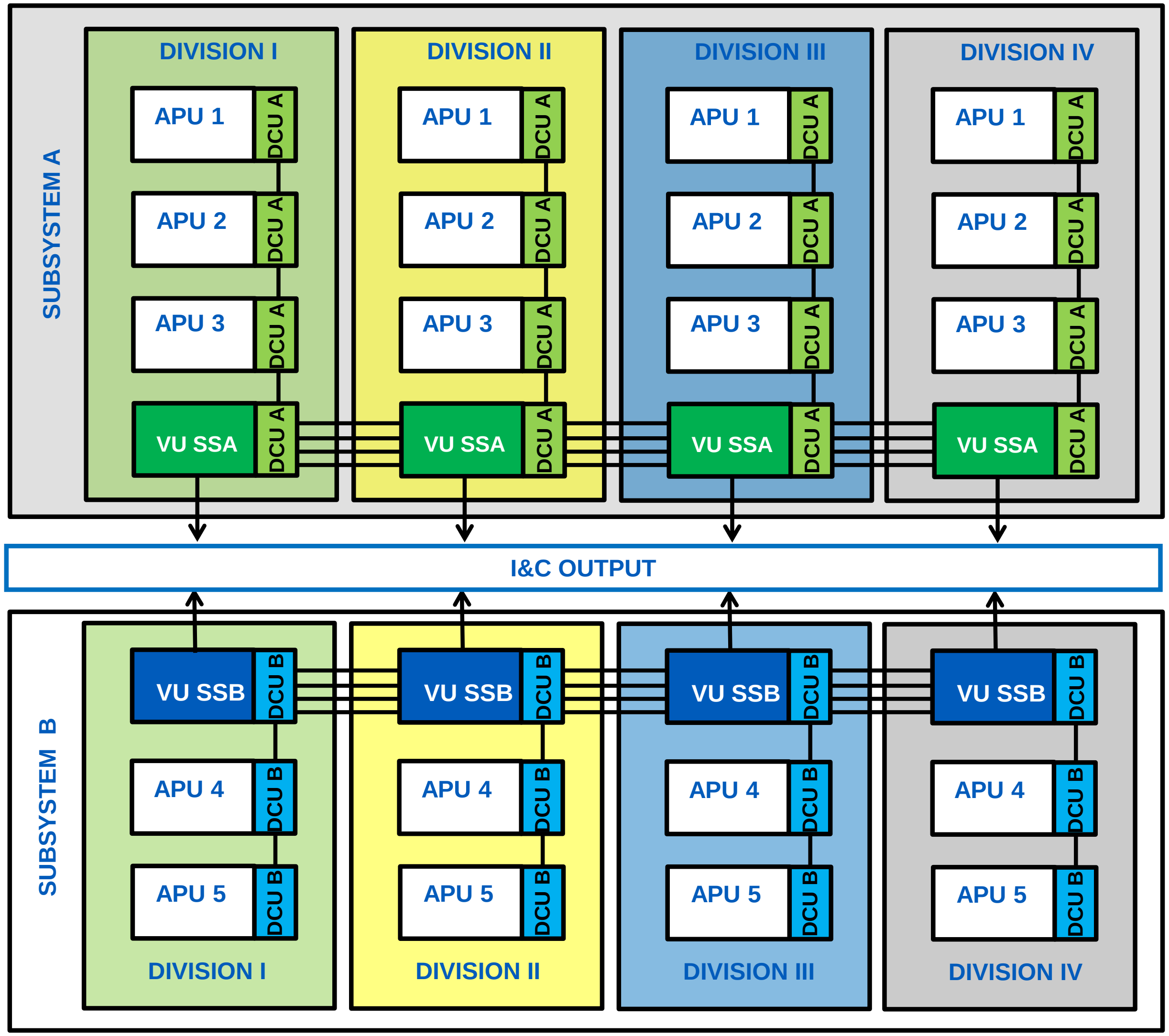


Figure 15. Location of I&C units within the architecture of the example I&C system. The representation of communication paths is simplified. The I&C inputs are not represented. APU = acquisition and processing unit, VU =voting unit, DCU = data communication unit. Colouring of the boxes indicate similarities between the I&C units. Both subsystems relies on same platform, but uses different application functions.

6.8.3 Failure effects at I&C unit level

As the fault location is the I&C unit as a whole, all functions of a given I&C unit can be considered to be affected with a same effect. This simplified assumption is often conservative. If the different functions implemented on an I&C unit are not affected the same way, it is recommended to go to module or basic component levels.

The failure effects defined at I&C units level are the same than those defined at I&C modules and basic components level. A fatal failure of an I&C unit by definition affects all functions of the unit, and the end effect depends on the fault tolerant design. Non-fatal failure affects specific I&C function(s) while the others remain unaffected. In both cases, the analysis must be done specifically for each I&C function relevant to PRA and to judge, e.g., whether the effect is failure to actuate or spurious actuation for the given I&C function.

6.8.4 Uncovering situations of I&C units failures

Table 3 provides examples of uncovering situations at I&C unit level. Note that the effect of an I&C unit failure at division or system levels depends on the fault tolerance features of the division or system.

Table 3. Examples of uncovering situations.

Uncovering situation	**Examples of detection or manifestation of the failure modes of the uncovering situation**
Online detection mechanisms	Crash of the CPU board of an APU detected by the watchdog and immediately taken into account by the voting logic
Offline detection mechanisms	The APU in a frozen state detected by a periodic test, and immediately set under repair.
Latent revealed by demand	Progressive saturation of a memory block. Measure in a frozen sate of a Unit used only during some incidents. The failure is detected only during the incident requiring to use it.
Triggered by demand	The APU triggers action not appropriate given the plant condition, or fail to actuate in a situation not considered by the specifications,
Spurious	The APU triggers the opening of a valve that is not required in a normal plant condition.

6.9 Taxonomy at the I&C module level

6.9.1 Rationale of the taxonomy at I&C module level

The taxonomy is based on the failure model, and use the following set of attributes:

- I&C module category
- Failure effects at I&C module level
- I&C module failures uncovering situation.

I&C units consist of multiple modules, which may be hardware, software or combination of hardware and software. The taxonomy at I&C module level does differentiate between hardware and software faults or failure modes.

6.9.2 I&C modules category

This classification defines the possible fault locations at I&C module level and permit to assess failure propagation. It is based on the example system. At this level, the fault location is a module as a whole. A failure may affect different I&C modules if it is propagated through CCF or cascade failure (see 6.3.3.). Fault locations have to consider two points of view, or “aspects” of the system at module level:

- Hardware aspect
- Software aspect.

6.9.2.1 Hardware aspect

In practice, there is a limited variety of hardware modules in a protection system, that correspond to typical functionalities of a data processing and are associated with I&C Units (see Table 4). This classification is independent of the I&C protection system considered, its architecture, and its fault tolerance features.

Table 4. Classification of I&C modules - hardware aspect.

I&C modules (fault locations) - hardware aspect	Relevant I&C unit category
APU input module hardware (digital or analog) APU processing module hardware APU subrack hardware APU output modules hardware (digital or analog)	Acquisition & processing unit (APU)
VU output module hardware VU processing module hardware VU subrack hardware	Voting unit (VU)
Link module hardware DCU processing module hardware DCU subrack hardware Network wires Network optical cables APU/DCU interface	Data communication unit (DCU)

6.9.2.2 Software aspect

The classification of software modules is potentially more complex than the classification of hardware modules, given the variety of functions that software can perform. Also, the delimitation of a software module may be difficult in a complex distributed architecture. To handle this issue, the classification is done in a pragmatic scope. As its purpose is to help a PRA analyst to represent I&C, the risk of failure propagation and the effects of defensive measures, the classification identifies as modules software elements that a PRA analyst can identify and assess in term of benefits of defensive measures.

The level of detail that is accessible to the PRA analyst is dependent of the management and maturity of a project, and the regulatory framework. In the example, the PRA analyst can identify the details and get access to the specification of application-specific software modules. However, there has no access to the details of a program running on a particular microcontroller in a communication module or a subrack power supply module, for example. Thus, the application software is detailed in various modules, but, data communication software, or software implemented in microcontrollers can only be treated as black boxes.

Under these considerations, the example system features the following typical list of software modules, associated with I&C units (see Table 5):

- Operating system (OS). Its main functions are to initialise the units, to manage input-outputs and data communication, to perform auto-tests, to process operator requests, and to activate the application-specific software. It is part of the platform software: this means that the OS is the same for all the APUs, VUs and DCUs of all divisions or of all subsystems of the example.

- Elementary functions (EFs). They provide readily useable standard (library) functions such as Boolean logic, mathematical functions or delays. A unique set of EF is at the disposal of the application-specific software, as part of the platform software. However, a specific APU will use only a specific subset of all available EFs.

- Application-specific software modules in APUs (APU-AS). An APU supports multiple application functions. The software of an application function is an AS module. Similar APUs in redundant divisions contain the same sets of AS modules. Also, there could be functional and asymmetric dependencies between application functions. There is one module of this category per application function implemented in an APU. As an option for the PRA practitioner, an APU-AS module may be created to represent possible faults in the data tables specifying the hardware configuration and the data communication of the APU. APU-AS allows also representation of the risk of faults in the implementation of application-specific acquisition and processing software.

- Application-specific software modules in VUs (VU-AS). There is one module of this category per voting function implemented in a VU. VU-AS allows representation of the risk of faults in the implementation of application-specific voting software. If needed by the PRA practitioner, a module may also be created to represent possible faults in the data tables specifying the hardware configuration and the data communication of the VU.

- Data communication software (DCS). There is one module of this category per network in the system. In the example, each subsystem has its own network.

  - Data link configuration (DLC). There is one module of this category per network in the system.

In addition to these software modules, it is worthwhile to take into consideration faults in functional requirements specifications. Operating experience shows that such faults are not very unlikely and not specific to programmed digital systems [EPRI TR 1021077]. This may be done using particular modules that are neither hardware nor software, representing the functional requirements specification (FRS) of each application function expected of an APU or au VU.

- APUs functional requirements specification modules (APU-FRS). There is one module of this category per application function implemented in the APU. APU-FRS allows representation of the risk of faults in functional requirements specifications of the acquisition and processing functions.

- VUs functional requirements specification modules (VU-FRS). There is one module of this category per voting function implemented on a VU. VU-FRS allows representation of the risk of faults in functional requirements specifications of the voting functions, especially operational bypasses and modification of the voting logic.

At last, the category SW in COTS modules (COTS-SW) represent specific pieces of software present in hardware modules, independently of the applicative functions. Since these modules are often designed by the component or hardware supplier and not the platform vendor, the PRA analyst has to assess their behaviour and the effect of defensive measures in another way than for other software modules. SW failure modes can be associated with the HW module failure modes.

Table 5. Classification of I&C modules - software aspect.

I&C modules (fault locations) - software aspect	Relevant I&C unit category
Operating system (OS) Elementary functions (EF) Application specific software (APU-AS) Functional requirements specification (APU-FRS)	Acquisition & processing unit (APU)
Operating system (OS) Elementary functions (EF) Application specific software (VU-AS) Functional requirements specification (VU-FRS)	Voting unit (VU)
Operating system (OS) Data communication software (DCS) Application specific software (Data Link Configuration) (DLC) Functional requirements specification (DCU-FRS)	Data communication unit (DCU)
SW in COTS modules (COTS-SW)	Any kind of I&C unit

6.9.2.3 Mapping of software and hardware modules

The locations of software modules in the hardware modules (the mapping) is dependent of the design of the system and the I&C units. In addition, there may be no one-to-one mapping between hardware and software modules.

6.9.3 Failure effects at I&C module level

The failure effects classification defined at modules level is the same at I&C unit level (See Table 6 for hardware modules and Table 7 modules).

Table 6. Failure modes and failure effects of hardware modules.

I&C module output	Module types	Failure modes	Failure effect
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 7. Failure modes and failure effects of software modules.

Module types	Failure modes	Failure effect
Operating system	Hang, crash (no output). – 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 – These failures are trapped by the microprocessor exception features	Fatal failure
Elementary functions, application specific software, functional requirements specification	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	Failure modes are protocol dependent	Protocol dependent
Proprietary modules	Failure modes are function dependent	Function dependent

To link taxonomy and PRA, 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 FMEAs and reliability studies which are normally carried out on the component and module level by suppliers constitute very important input data. They typically provide failure descriptions that have to be put in equivalency with the failure effects used in this taxonomy. To do that, it is sufficient to focus on categories of outputs, that are relevant to assess the functionality of I&C modules. There are in the example only two categories: modules with digital outputs (hardware and software) and with analog outputs (hardware).

6.9.4 Uncovering situations at I&C modules level

The uncovering situations defined at modules level are defined in the failure model. Table 8 provides examples of uncovering situations at I&C modules level. Note that the uncovering situation of a I&C module depends on the fault tolerance features of the module, I&C unit, division or system. For example; in a case of fatal failure, the failure may be detected by a watchdog and treated at module level. In case of a plausible non-fatal failure, even with no detection by the watchdog, the redundancy strategy of the system would have prevented or compensated the failure.

Table 8. Examples of uncovering situations at I&C module level.

Uncovering situation	Example of fault tolerance feature
Online detection mechanisms	Crash of the microprocessor of a processing module detected by the watchdog and immediately taken into account by the voting logic Fatal failures trapped by the microprocessor exception features
Offline detection mechanisms	The microprocessor of a processing module in a frozen state detected by a periodic test, and immediately set under repair.
Latent revealed by demand	Progressive saturation of a memory block. Failure in a frozen sate of an input module used only during some incidents. The failure is detected only during the incident requiring to use it.
Triggered by demand	A VU module triggers an action not appropriate given the plant condition , in a situation not considered by the specifications An electronic board fails due to a thermal shock or/and electrical overstress out of specification when a motor is actuated. There is no detection, before triggering.
Spurious actuation	An output module triggers the opening of a valve that is not required in a normal plant condition. There is detection, before actuation.

6.10 Taxonomy at the basic component level

6.10.1 Rationale of the taxonomy at basic component level

The taxonomy is based on the failure model, and use the following set of attributes:

- basic component category
- failure effects at basic component level
- basic component failures uncovering situation

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. In this chapter only hardware elements are discussed. Software elements is not currently a notion that has a sufficient consensus to make a classification. Thus, for software, the module level is the most detailed level considered.

6.10.2 Basic component categories

This classification defines the possible fault locations at hardware basic component level and can contribute to assess failure propagation.

It is difficult to provide an exhaustive list of hardware basic components categories; and practically infeasible for software basic components. Thus only a classification of basic hardware components is defined (Table 9). As there is a one-to-one correspondence of hardware and software basic components, that is not the case at module level, hardware categories may also be used for fault locations of basic software components, e.g., the proprietary software modules.

This classification is independent of the I&C protection system considered, its architecture, and its fault tolerance features.

Table 9. Classification of basic hardware components.

Basic component	Comment
Microprocessor	
Associated components of a microprocessor	Associated components of a microprocessor, such as the internal bus, RAM, ROM, BIOS, flash disk, buffer, and serial port
RAM	
Current Loop	The major components of the current I/O devices are current loops that essentially are linear transmitters/receivers.
Voltage Regulator	The voltage regulators are assumed to be the major component of the voltage input module.
Multiplexer	Multiplexer (MUX) and demultiplexer (DEMUX)
Demultiplexer	
A/D converter	
D/A converter	
Address logic	Address logic: This is a basic digital component, also called a decoder. A microprocessor uses the address logic to access the information transmitted on the backplanes and other components.
Solid-state switch	

6.10.3 Failure effects at basic component level

The failure effects classification defined at basic component level is the same than those defined at I&C units and module level by the failure model. A failure may affect different basic components if it is propagated through CCF or cascade failure (see 6.3.3.).

The effect of a basic component failure at I&C module level is dependent of the function of the component. For example, a crash in a CPU usually leads to a crash of a module; a signal stuck to current value in an output component leads to a failure with plausible behaviour of an APU output module, etc.

The failure modes and effects for individual components of a digital module are summarized below, and the sources of the failure modes are cited in Table 10. Failure modes of components discussed here may not perfectly match all component failure modes of digital systems; nevertheless, they were the best approximation found at present. For example, “no output” and “short-circuit” failures modes of a linear IC, such as A/D and D/A converters, are interpreted as “fails low.” Also, only one failure mode is postulated for some components, such as the internal bus, RAM, address logic etc.

The failure modes are represented in relation with the function, e.g. the signals processed by the components. Regardless of vendors, the functions of basic components of digital systems, are well-defined, e.g., A/D converter is always used to convert analog signals to digital signals. This facilitates the definition of failure effects. Failure effects defined at the basic component level are assumed to be independent of design or vendor of I&C systems. Thus, a same set of failure effects can be applied to components of the same category, even if they are of different models.

The FMEAs and reliability studies normally carried out at the component and module level by suppliers constitute important input data for the failure analysis at this level. Usually, reliability data for the basic components exist and are available, at least among vendors who manufacture these components.

6.10.4 Uncovering situations at basic component level

The uncovering situations defined at component level are the same than those defined at I&C modules and units level (see Table 11).

Table 10. Examples of failure modes of basic hardware components.

Component	Failure mode	Failure effect	Comments
Microprocessor	The microprocessor seems to be running normally but sends erroneous output	Failure with plausible or implausible behaviour.	This is considered an undetectable failure of the module, i.e., the module will send wrong output signals.
	The microprocessor stops updating output	Ordered or haphazard fatal failure.	
Associated components of a microprocessor	See comment	Ordered fatal failure	It is conservatively assumed that each component has only one failure mode, i.e., a failure of the component, which entails the loss of the functions performed by the component.
RAM	Loss of RAM	Ordered fatal failure	
	False memory content	Non-fatal failure	
Current loop	Fail (drift) high or fail (drift) low of current loop device	Non-fatal failure	1. The current loop may fail high or low, resulting in the associated I/O signal failing high or low. Fail low includes failures of fail to zero. 2. Further analyses of individual input/output signals based on the design information of the module are needed to determine their impacts on the module(s). 2. If the current loop is for an output, then the analog output of the module will fail (drift) high or fail (drift) low.
Voltage regulator	Fail (drift) high or fail (drift) low of voltage signal	Non-fatal failure	The failure modes are fail-high and fail-low of the associated voltage input signal [RAC-1997]. 1. The voltage regulator is a major component for the voltage signal I/O. It may fail high or low, and effectively, causes the voltage signals to fail high or low. 2. If the regulator is for an output signal, then the analog output of the module will fail (drift) high or fail (drift) low.
Multiplexer	Loss of all signals from MUX	Non-fatal failure	Loss of a signal means that the signal fails low. All analog inputs share the MUX. This failure mode indicates that all analog input signals related to this MUX fail low.
	Loss of one signal from MUX	Non-fatal failure	The failure mode indicates a loss of a specific analog signal.
Demultiplexer	All analog output signals (related to this DEMUX) of the module fail low.	Non-fatal failure	The DEMUX is shared by all analog outputs. Loss of an output signal means that the signal fails low. Responses of the receiving module(s) to this failure depend on signals they received and their FTDs.
	An analog signal (related to this DEMUX) of the module fails low.	Non-fatal failure	The failure mode indicates a loss of a specific analog output signal. Responses of the receiving module(s) to this failure depend on the individual signal(s) and their FTDs.

Component	Failure mode	Failure effect	Comments
A/D converter	All 16 bits of A/D converter stuck at zeros or ones	Non-fatal failure	1. The A/D converter is shared by all analog inputs, and its failure will entail the failure of all analog inputs.
	Random bit failure of A/D converter	Non-fatal failure	
D/A converter	All analog signals (related to this D/A) of the module fail (drift) high.	Non-fatal failure	The digital/analog converter is shared by all outputs of the module, and its failure will result in a failure of all outputs.
	All analog signals (related to this D/A) of the module fail (drift) low.	Non-fatal failure	
Address logic	Loss of address logic	Failure with plausible behaviour	The failure mode is assumed as a loss of the address logic, so that the microprocessor cannot access the intended information upon loss of the address logic. This is considered an undetectable failure of the module, i.e., the module will send wrong output signals.
Solid-state switch	Failure to operate or false operation of solid-state switch	Non-fatal failure	A solid-sate switch carries a digital I/O signal. Its failure to operate indicates that the digital signal fails as is. False operation indicates that the digital signal fails to the opposite state.
Other	Require a case par case failure mechanisms assessment and failure modes analysis		

Notes:

- Failure modes of MUXs and DEMUXs are defined in [AN8500-1] in terms of the analog signals they transmit, which include a loss of one or all signals. No other failure modes of MUXs or DEMUXs were mentioned in [AN8500-1], and, therefore, a loss of signal is modelled as signal fails low.
- Digital/Analog (A/D) and Digital/Analog (D/A) converters: Both A/D and D/A converters are linear integrated circuits (ICs), i.e., the I/Os are proportional to each other; all analog I/Os of the same module share them. The failure modes of an A/D converter include all bits of the A/D stuck at zeros, all bits stuck at ones, and a random bit-failure of the A/D converter [Meeldijk 1996]. The failure modes of a D/A converter include output fails (drifts) high or low [Meeldijk 1996]. It is assumed that if the D/A converter output starts drifting, it will eventually reach the high or low detection threshold.
- Current I/O devices: They also are linear ICs and their failure modes are current signal fails (drifts) high or low [Meeldijk 1996]. It is assumed that if the current starts drifting, it will eventually reach the high or low detection threshold.
- Digital I/O devices: Digital I/O is implemented via a solid-state switch [Eurotherm-2000]. The status of a digital signal is controlled by opening or closing the switch. The solid-state switch may fail to operate (fail as is) and spuriously operate (fails to the opposite state), as stated in [RAC-1997].

Table 11. Examples of uncovering situations of basic hardware components.

Component	Failure mode	Uncovering situation	Comments
Microprocessor	The microprocessor seems to be running normally but sends erroneous output	Revealed by demand or spurious	This is considered an undetectable failure of the module, i.e., the module will send wrong output signals.
	The microprocessor stops updating output	Any (FTD dependent)	Depending on FTDs of the module such as a WDT. It is assumed that this failure may be detected by the WDT if its status is normal[3].
Associated components of a microprocessor	Loss of component	Any (FTD dependent)	Depending on FTDs of the module such as a WDT.The input and output of the module rely on the internal bus. It is assumed that this failure may be detected by the WDT if its status is normal.
RAM	Loss of RAM	Any (FTD dependent)	Depending on FTDs of the module such as a WDT. It is assumed that this may be detected by the WDT if its status is normal.
Current loop	Fail (drift) high or fail (drift) low of current loop device	Any (FTD dependent)	Signal and FTD dependent if the current loop is for an input signal. Further analyses of individual input/output signals based on the design information of the module are needed to determine their impacts on the module(s).
Voltage regulator	Fail (drift) high or fail (drift) low of voltage signal	Any (FTD dependent)	Signal and FTD dependent if the regulator is for an input signal. Further analyses of individual input/output signals are needed to determine their impacts on the module(s).
Multiplexer	Loss of all signals from MUX	Any (FTD dependent)	Depending on FTDs, in particular the application software logic, of the module.
	Loss of one signal from MUX	Any (Signal and FTD dependent)	The response of the module to this failure is signal specific.
Demultiplexer	Loss of all signals from DEMUX	Any (FTD dependent)	The DEMUX is shared by all analog outputs. Loss of an output signal means that the signal fails low. Responses of the receiving module(s) to this failure depend on signals they received and their FTDs.
	Loss of one signal from DEMUX	Any (FTD dependent)	The failure mode indicates a loss of a specific analog output signal. Responses of the receiving module(s) to this failure depend on the individual signal(s) and their FTDs.
A/D converter	All 16 bits of A/D converter stuck at zeros or ones	Any (FTD dependent)	Depending on FTDs of the module, in particular the application software logic.Stuck at zeros or ones indicates that all analog signals fail low or high. The module may detect failures of some input signals and handle the failures according to its FTDs.
	Random bit failure of A/D converter	Revealed by demand or spurious	This is considered an undetectable failure of the module, i.e., the module will send wrong output signals.
D/A converter	Output of digital/analog converter fails (drifts) high	Any (FTD dependent)	Responses of the receiving module(s) to this failure depend on signals they received and their FTDs.
	Output of digital/analog converter fails (drifts) low	Any (FTD dependent)	Responses of the receiving module(s) to this failure depend on signals they received and their FTDs.

[3] WDT detectable means the update of the toggling signal to WDT is stopped

Component	**Failure mode**	**Uncovering situation**	**Comments**
Address logic	Loss of address logic	Revealed by demand or spurious	The failure mode is assumed as a loss of the address logic, so that the microprocessor cannot access the intended information upon loss of the address logic. This is considered an undetectable failure of the module, i.e., the module will send wrong output signals.
Solid-state switch	Failure to operate or false operation of solid-state switch	Any (Signal and FTD dependent if the signal is an input signal)	If the signal is an output, the digital output of the module will fail as is or fail to the opposite state. Further analyses on individual input/output signals are needed to determine the impacts.

FTD: Fault tolerance design; WDT: Watch dog timer

6.11 Relation between taxonomy and PRA

Two activities require to use the taxonomy and related classifications in relation to PRA. In this context:

- Failure propagation assessment and CCF effects
- Representation of failures as basic events in the PRA

Only the first activity is presented in this chapter. The second one is detailed in chapter 7.

6.11.1 Failure propagation assessment and CCF effects

Failure propagation assessment is presented here as a guideline, to be adapted on each case with classifications that support it. The approach has three main steps:

- Successively postulate a single fault (hardware or software related) in fault locations. Fault location may be assumed at least at I&C Unit level, and often at a higher level of abstraction. At this step of the approach, there is no prior assumption on the likelihood of presence of a fault. Fault may be first postulated in the various modules or some basic components present in the unit to make the analysis as much exhaustive as possible. Single failures have to be considered as they affect non redundant parts of the system. Also, propagation assessment requires assumptions about the failure origin. Three origins have to be considered: random hardware failures, software failures due to activation of faults, hardware failure due to design and manufacturing. Two cases of system/division level end effects have to be considered for each origin: failure of a required safety actuation (failure on demand), or spurious actuation.
- Given assumption about the fault location within the unit, determine first its maximum possible end effect, regardless of the measures taken by design or operation to limit the propagation or vulnerability to CCF
- Determine then its most likely end effect, taking into account the V&V, defence in depth and diversity features actually implemented.

The assessment requires information to understand the hardware and software design to propagate a component failure that is usually not available for a PRA analyst. This activity usually associates the I&C and the PRA practitioners. It is out of scope of this document to provide a full taxonomy of all possible and activation conditions and defensive measures. The analysts have to refer to existing literature for this purpose ([EPRI TR 1007997], [IEC 62340], etc.).

According to the model presented at the beginning of this chapter, a failure propagates from one or many failures locations to other levels of the I&C architecture, under influence of variable combinations of failure mode and effect, failure origin, plant condition, initiating event, and activation conditions. The assessment of the end effect of the propagation requires the combined knowledge of following attributes defined by the failure model:

- Fault location
- Failures mode

- Uncovering situation
- Failure Origin
- Maximum potential end effect
- Most likely potential end effect.

Also, the assessment has to consider the digital I&C architecture and its fault tolerance design. The classifications of failures modes, fault location, and uncovering situations have been presented previously at the different I&C levels.

6.11.2 Combinations of failures modes and uncovering situations

Uncovering situations have to be combined with failure effects to analyse the propagation of a local failure to another location or level. Table 12 presents 20 theoretical combinations (5 uncovering situations x 4 failure modes), from which 6 are not relevant, due to logical considerations. For example, a plausible behaviour is not detected online, a failure detected by online detection does not need to be considered in combination with Periodic Testing. These combinations are independent of the architecture, defence in depth and diversity strategies of the protection system.

Table 12. Relevance of failure effect and uncovering situations.

	Uncovering Situation				
Failure effect	Online detection	Offline detection	Revealed by spurious action	Latent, revealed by demand	Triggered by demand
Fatal, ordered	**R**	**NR**	**R**	**R**	**R**
Fatal, haphazard	**NR**	**R**	**R**	**R**	**R**
Non-fatal, plausible behaviour	**NR**	**R**	**R**	**R**	**R**
Non-fatal, implausible behaviour	**R**	**NR**	**R**	**NR**	**R**

R: Relevant. Combination to be considered in analysis of the effects.
NR: Not Relevant. Combination that has does not need to be considered for the analysis of the effects. Non relevance is due to logical considerations.

6.11.2.1 Online detection

This detection mechanism is especially relevant for hardware failures. It is also the detection mechanism relevant for self-healing or transient failures. It is especially efficient against “fatal, ordered” and “non-fatal, implausible” failure modes. Thus, two combinations are relevant for the analysis of online uncovering situations:

- Online detection and fatal, ordered
- Online detection and non-fatal, implausible failure.

These situations lead to pre-determined states. The effect of these failures is thus limited, as they are detected before a failure on demand, spurious, or trigger. Note that the failure of the online detection mechanism itself may be not safe.

6.11.2.2 Offline detection

It is especially relevant for HW faults or failures. It can only detect permanent failed state. It complements online detection mechanisms, as it is efficient for some non-fatal failures with plausible behaviours. Self-healing or transient failures cannot be detected by offline detection mechanisms.

Other uncovering situations are covered by online detection that occurs before offline detection, or may not be covered by offline detection (example: specification errors may be not detected by periodic testing since tests are based on the same specifications). Thus only two combinations are relevant:

- Offline detection and non-fatal failures with plausible behaviours
- Fatal, haphazard.

These uncovering situations lead to safe failures. The effect is usually limited, as the failure is detected before failure on demand, spurious actuation. The case of a similar failure that would not be detected before failure on demand or spurious actuation is addressed hereafter.

6.11.2.3 Spurious actuation

The effect is a spurious actuation of some protective actions, consistent or not from the system point of view. All combinations are relevant. In particular:

- Fatal, ordered and spurious. Example: a watchdog (online detection) detects a fatal ordered failure and launch a protective, spurious action, although the physical process is not in a state that justifies it. The fatal ordered failure may be due to a failure of the watchdog or a disturbance of power supply, loss of communication network, etc.
- Non-fatal, plausible and spurious actuation. These combinations are possible and may lead to the propagation of a failure. Example: the diesel sequencer launches a normal series of orders, in absence of demand.
- Non-fatal, implausible and spurious actuation. These combinations are possible and may lead to the propagation of a failure. Examples: the diesel sequencer launches a random series of orders in absence of demand; failures leading to non-consistent false alarms; etc.

6.11.2.4 Latent, revealed by demand.

The I&C unit does not fulfil the demand; it was not correct before the demand, but the failure was not detected. Depending on the failure location within the system, it may lead to a Failure on Demand, a delay or a fail-safe state. As fatal, ordered and non-fatal, implausible failures are detected online and offline, only two combinations are relevant:

1. Latent, revealed by demand and non-fatal, plausible haphazard failures. The plausible failure are often non detectable online.
2. Fatal haphazard.

Latent, revealed by demand uncovering situations may have various end effects. They may eventually lead to a failure on demand, a delay or a fail-safe state.

6.11.2.5 Triggered by demand.

The demand itself causes the failure of the I&C unit while no failures were experienced before. Typical causes for software are demand out of the range of a specification or untested, error in a procedure for the setting of parameters, specification error. These specification or procedural errors are usually not detectable by periodic testing or self-detection (oracle issue).

Typical causes for hardware are failures by an abnormal electrical overstress caused by the demand signal. They are neither detectable by periodic testing or self-detection. Triggered by demand failures are represented in PRA like Failures on demand.

6.11.3 Assessment of end effects

6.11.3.1 Classification of end effects.

The I&C units and modules categories are necessary for the assessment. Also, the assessment of the propagation of a failure requires the knowledge of the I&C architecture and FTD. We provide here definitions that are relevant for the architecture presented in chapter 3. Table 13 presents the classification of end effects.

Table 13. Classification of end effects.

End Effect	Definition	Relevant for hardware	Relevant for software
Single failure	Failure of a single I&C module or basic component	Yes	Yes
FF-1SS	Failure of one application function including elementary function[4] (or more) in one subsystem	No	Yes
FF-1D-1SS	Failure of one function (or more) in only one division in one subsystem	No	Yes
FF-AllSS	Failure of one application function including elementary function (or more) in all subsystems	No	Yes
1APU	Failure of 1 group of redundant similar APUs in all divisions	Yes	Yes
1VU	Failure of 1 group of redundant similar VUs in all divisions	Yes	Yes
MAPU-1SS	Failure of multiple groups of redundant similar APUs in only one subsystem	Yes	Yes
MVU-1SS	Failure of multiple VUs in only one subsystem	Yes	Yes
MDCU-1SS	Failure of multiple DCUs in only one subsystem	Yes	Yes
1SS	Failure of only one subsystem	Yes	Yes
MAPU-AllSS	Failure of multiple groups of redundant similar APUs in both subsystems. Possible system failure, depending on APU-AS allocation	Yes	Yes
1SS-APU	Failure of one subsystem and of group(s) or redundant similar APUs in all divisions in the other subsystem. Likely system failure.	Yes	Yes
SYSTEM	Failure of both subsystems.	Yes	Yes

The classification covers the case of function without vote. Function failure in one division only (FF-1D-1SS) and Function failure in multiple or all divisions have the same effect as FF-1SS.

[4] An application function is a combination of elementary functions

In addition to the end effects classification, for failure propagation assessment, it is necessary to make assumptions about the failure origin. Three origins have to be considered:

- Random hardware failures
- Hardware failure due to design, manufacturing, maintenance errors, harsh operation condition (stress)
- Software failures due to activation of faults, maintenance failures, operator failures

The two last origins cover human failures during operations, maintenance and building, set up and test.

Figure 16 Propagation model in the case of failure on demand and Figure 17 represent the model of the propagation assessment, in the cases of failure on demand and spurious actuation.

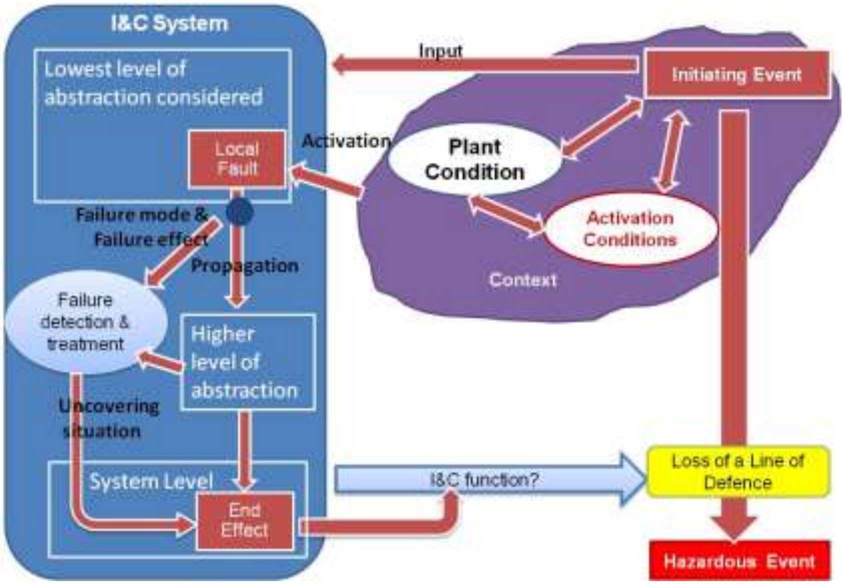


Figure 16. Propagation model in the case of failure on demand.

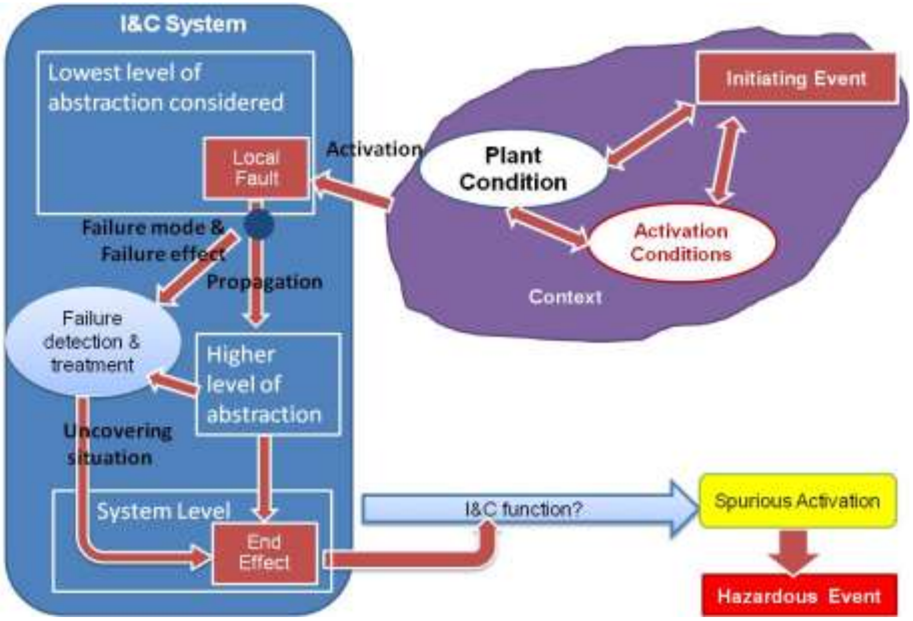


Figure 17. Propagation model in the case of spurious actuation.

6.11.3.2 Propagation of random hardware failures.

FTD handles most random hardware failures. The exceptions are random failures that affect a critical single point or combine with another failures occurring independently in a same time period. The failures modes at module or at basic component level are sufficient to analyse the end effect.

Table 14 and Table 15 summarise the relevant end effects, in absence of other coincidental random failures or failure propagations, depending on the fault location in the hardware modules present in the example system.

The cells in grey correspond to end effects affecting the system level. In the example system, failures of the output signals of the Voting Units are the only potential critical single points. Their failures may lead to spurious actuations.

For most applicative I&C functions implemented by the APUs, the VUs will perform a vote to reduce the potential for spurious actuation and provide protection against random failures. For such functions, only failure propagation involving multiple divisions will have system/subsystem consequences and need to be considered in the assessment. However, voting is feasible mainly for functions the output of which is a single, latched Boolean signal. Also, functions with sequential complex outputs, like for example diesel load sequencers, require a specific assessment.

This assessment may lead to different conclusions given another I&C architecture.

Table 14. End effect of a random hardware failure (case of actuation as required output).

End effect	Random failure location in hardware modules	Note
	All fault locations	
Single failure	R	Local effect (level of one APU, DCU or VU)
Other end effects	NR	The system is designed to avoid a propagation of a random failure affecting availability of the protective functions

Table 15. End effect of a random hardware failure (case of spurious actuation).

End effect	Random failure location in hardware modules		Note
	Fault located in a VU module	Other fault locations	
Single failure	R	R	Local effect (level of one APU, DCU or VU)
1SS	R	NR	For some failures affecting VUs, a random failure may cause a spurious actuation
SYSTEM	R	NR	
Other end effects	NR	NR	

R: Relevant. Combination to be considered in analysis of the effects of a random hardware failure.
NR: Not Relevant. Combination that has not to be considered for propagation assessment. Non relevance is due to logical considerations.
1APU: Failure of 1 set of redundant similar APUs in all divisions
MAPU-1SS: Failure of multiple sets of redundant similar APUs in only one subsystem
1SS : Failure of only one subsystem
MAPU-AllSS: Failure of multiple sets of redundant similar APUs in both subsystems.
1SS-APU : Failure of one subsystem and of set(s) or redundant similar APUs in all divisions in the other subsystem.
SYSTEM: Failure of both subsystems.

6.11.3.3 Propagation of hardware failures due to design, manufacturing.

The failures originating from hardware design or manufacturing issues have potentially important effects. As they have systematic root causes, they have a potential for a large propagation that may extend up to the system level. For example, a design flaw in a memory used in boards mounted in a series of APU, a mounting error of a protection diode used in a subrack of all APUs, etc.

Table 16 and Table 17 summarise the potential end effects, depending on the fault location in the hardware modules, in absence of cumulative coincidental random hardware failures. These end effects may have to be adapted for other architectures. The lines in gray correspond to end effects affecting the system level, and differ in the failure on demand or spurious cases.

Some design consideration lead to NR propagations. For example, the Module in DCU is Non Relevant for an effect at MDCU-AllSS because, in the example of architecture, there are two different DCU per division (one per sub-system).

Systematic hardware failure in APU affects at least one set of redundant similar APUs in all divisions. Wider propagation is possible, however it is very unlikely it may affect all APU in both subsystem with no prior online or offline detection. Failure in Voting Units may have an effect at subsystem level. As APU in a same subsystem provide data to the VU through DCU, a DCU failure may have an effect on multiple APUs or all of them in one subsystem or both subsystems, if the hardware is the same in both subsystems.

An assessment has to be done for every I&C architecture and may lead to different conclusions. To assess the most likely end effect, note that the likelihood of large propagation of such failures is smaller than in the case of software failures, as the detection means are more efficient in addressing hardware failures than software failures.

Table 16. Example of end effects of a systematic hardware failure (case of actuation as required output).

End effect	Design and manufacturing faults location in hardware modules			Notes
	Fault located in APU module	Fault located in VU module	Fault located in DCU module	
1APU	R	NR	NR	This category is only relevant for APUs
1VU	NR	R	NR	This category is only relevant for APUs
MxU-1SS	R	NR	R	
1SS	R	R	R	
MAPU-AllSS	R	NR	NR	This category is only relevant for APUs
1SS-APU	R	NR	NR	This category is only relevant for APUs
SYSTEM	R	R	R	

Table 17. Example of end effects of a systematic hardware failure (case of spurious actuation).

End effect	Design and manufacturing faults location in hardware modules			Notes
	Fault located in APU module	Fault located in VU module	Fault located in DCU module	
1APU	R	NR	NR	This category is only relevant for APUs
1VU	NR	R	NR	This category is only relevant for APUs
MxU-1SS	R	NR	R	
1SS	R	R	R	
MAPU-allSS	R	NR	NR	This category is only relevant for APUs
1SS-APU	R	NR	NR	This category is only relevant for APUs
SYSTEM	R	R	R	

R: Relevant. Combination to be considered in analysis of the effects of a random hardware failure.
NR: Not Relevant. Combination that has not to be considered in analysis of the effects. Non relevance is due to logical considerations.

1APU: Failure of one set of redundant APUs
MxU-1SS: Failure of one group of DCUs, VUs or redundant APUs in only one subsystem
MAPU-AllSS: Failure of multiple sets of redundant APUs in both subsystems
1SS : Failure of only one subsystem
1SS-APU : Failure of one subsystem and of set(s) or redundant APUs in the other subsystem
SYSTEM: Failure of both subsystems.

6.11.3.4 Software failures due to activation of faults

Software failures occur when software faults are activated under particular conditions. These failures have potentially important effects. As they have systematic root causes, they have a potential for a large propagation that may extend up to the system level.

Under the exact same conditions (same inputs), the failures will always appear and in this manner software failures can be defined as being systematic. The end effects of a fault activation cannot be isolated from the origin of the fault (specification or implementation), the input conditions, the reuse of particular software components throughout the architecture (same OS in the different components, same elementary functions used throughout). A same fault may have been reproduced in various portions of the architecture and its activation in one location may coincide with its activation in other locations.

The assessment has to consider the software modules and some basic components, if they contain specific complex pieces of software.

Given the system example, software failure propagation from an elementary function in an APU affects at least one set of redundant similar APUs in all divisions. Wider propagation is possible, however it is very unlikely it may affect all APU or all VU in both subsystem with no prior online or offline detection. Failure propagation from an elementary function in voting units has an effect at subsystem level. As APU in a same subsystem provide data to the VU, through similar DCUs, a systematic failure affecting a DCU has an effect on multiple APUs or all of them in one subsystem or both subsystems.

Table 18 and Table 19 summarise the potential end effects, depending on the fault location in the software or hardware modules of the example system, in absence of other cumulative coincidental random failures or failure propagation. These end effects may have to be adapted for other architectures. The lines in grey correspond to end effects affecting the system level, and differ in the failure on demand or spurious cases.

An assessment has to be done for every I&C architecture and may lead to different conclusions. To assess the most likely end effect, note that the likelihood of end effect of such failures is higher than in the case of hardware failures, as the detection means are more efficient in addressing hardware failures than software failures.

Table 18. Example of end effects of a software module failure (case of actuation as required output).

End effect	Fault location in software modules									
	OS	**EF (in APU)**	**APU-FRS**	**APU-AS**	**COTS-SW**	**VU-FRS**	**VU-AS**	**EF (in VU)**	**DCS**	**DLC**
FF-1SS	R	R	R	R	NR	R	R	R	NR	NR
FF-1D-1SS	R	R	R	R	NR	NR	NR	NR	NR	NR
FF-allSS	R	R	NR	NR	NR	NR	NR	NR	NR	NR
1APU	R	R	R	R	R	NR	NR	NR	NR	NR
1VU	R	NR	NR	NR	NR	R	R	R	NR	NR
MxU-1SS	R	R	NR	NR	R	NR	NR	NR	NR	NR
1SS	R	R	R	NR	R	R	R	R	R	R
MAPU-AllSS	R	R	NR	NR	R	NR	NR	NR	NR	NR
1SS-APU	R	R	NR	NR	R	NR	NR	NR	NR	NR
SYSTEM	R	R	NR	NR	R	R	R	R	R	NR

Table 19. Example of end effects of a software module failure (case of spurious actuation).

Category of End Effect	Fault locations in software modules									
	OS	EF (in APU)	APU-FRS	APU-AS	COTS-SW	VU-FRS	VU-AS	EF (in VU)	DCS	DLC
FF-1SS	R	R	R	R	NR	R	R	R	NR	NR
FF-1D-1SS	R	R	R	R	NR	NR	NR	NR	NR	NR
FF-allSS	R	R	NR	NR	NR	NR	NR	NR	NR	NR
1APU	R	R	R	R	R	NR	NR	NR	NR	NR
1VU	R	NR	NR	NR	NR	R	R	R	NR	NR
MxU-1SS	R	R	NR	NR	R	NR	NR	NR	NR	NR
1SS	R	R	R	NR	R	R	R	R	R	R
MAPU-AllSS	R	R	NR	NR	R	NR	NR	NR	NR	NR
1SS-APU	R	R	NR	NR	R	NR	NR	NR	NR	NR
SYSTEM	R	R	NR	NR	R	R	R	R	R	NR

R: Relevant. Combination to be considered in analysis of the effects of a random hardware failure.
NR: Not Relevant. Combination that has not to be considered in analysis of the effects. Non relevance is due to logical considerations.

FF-1SS: Failure of one function (or more) in one subsystem
FF-1D-1SS: Failure of one function (or more) in only one division in one subsystem
FF-AllSS: Failure of one function (or more) in all subsystems
1APU: Failure of one set of redundant APUs.
MxU-1SS: Failure of one group of DCUs, VUs or redundant APUs in only one subsystem
MAPU-AllSS: Failure of multiple sets of redundant APUs in both subsystems
1SS: Failure of only one subsystem
1SS-APU: Failure of one subsystem and of set(s) or redundant APUs in the other subsystem
SYSTEM: Failure of both subsystems

7. DEMONSTRATION OF THE TAXONOMY

7.1 PRA modelling example

7.1.1 Introduction

The main purpose of the developed failure mode taxonomy is to serve as basis for the modelling of digital I&C reliability in PRA:s. The intent of this chapter is to demonstrate the usage of the developed taxonomy for PRA modelling. The example is taken from the reference [NKS-277]. In chapter 0, failure modes of a typical I&C module will be defined at the basic component level.

The task of incorporating a reliability model of a digital I&C based RPS into a traditional PRA 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 PRA model structure. Another challenge due to the complexity and number of components within a digital I&C RPS is to keep the PRA model comprehensive at a reasonable size, e.g., the number of FT:s and basic events, and to meet requirements regarding realism, quality assurance, maintainability, etc.

Since the dominating tool for performing state-of-the-art PRA is fault tree-event tree analysis, it will be the focus of this example. It is however recognised that other, more advanced, tools can be considered and that these in certain situations may be better suited for reliability analysis of digital I&C than traditional fault tree-event tree analysis. It should be noted that the developed taxonomy of chapter 6 does not exclude the use of other tools than fault tree-event tree analysis.

7.1.2 Failure modes taxonomy for modelling

Chapter 6 presents generic failure mode taxonomy at different levels of abstraction. The required level of detail to apply in the PRA depends on several factors such as complexity of the digital I&C design and the RPS architecture, purpose of the PRA, diversity of the reactor protection system and safety systems in general. The failure mode taxonomy for the *module level* will be applied in this example. Reference [NKS-277] shows example results on the comparison between module level and I&C unit level of modelling.

At the module level, a distinction is made between hardware and software related failure modes. There are a number of reasons to do it in this manner. Firstly, failure modes are explicitly associated with specific hardware and software modules (“fault location” as discussed in chapter 0). Secondly, there is a rather good consensus on how to analyse hardware module failure modes, and this chapter demonstrates that the taxonomy is compatible with that practice. Regarding software failure modes, the state-of-the-practice has been rather simple. In this context, a step forward is made based on the failure propagation assessment outlined in chapter 6.11.

7.1.2.1 Hardware failure modes

Figure 18 shows the assumed hardware modules of a typical module structure of acquisition, signal processing and voting units appearing in a digital RPS. In addition, measurement sensors are included in the example. Software modules are assumed in similar manner as in Figure 18.

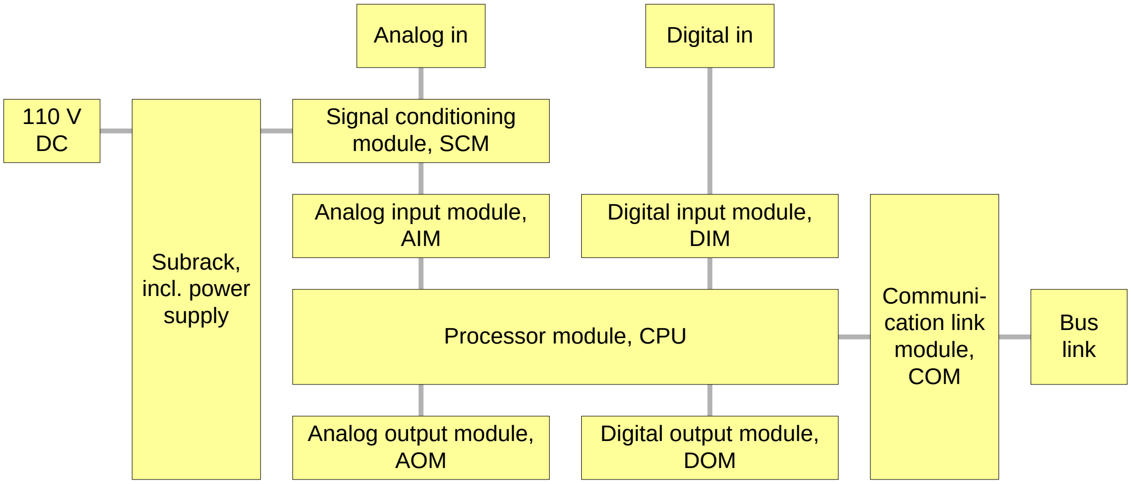

Analog in
Digital in
110 V DC
Subrack, incl. power supply
Signal conditioning module, SCM
Analog input module, AIM
Digital input module, DIM
Processor module, CPU
Communi-cation link module, COM
Bus link
Analog output module, AOM
Digital output module, DOM

Figure 18. Modules included in the example I&C unit.

The taxonomy presented in chapter 7 is for the module level will in most cases be of unnecessary high level of detail to apply in a PRA model. The high level of detail is necessary initially to classify the failure modes of each digital I&C module into one of the defined generic failure modes (called failure mode and local effect in ch. 0), in order to decide the effect of the failure on a functional level.

From the PRA modelling perspective, it is more beneficial to compress the failure modes and to define them by functional effect rather than local effect, since this not only will keep down the number of events and the model size, but also will simplify the modelling efforts and make the fault tree structure and the dependencies more comprehensible to the PRA user. Functional effect of main interest is how APU respectively VU behave from the protection signals actuation point of view in the presence of the fault. These functional effects are defined for different failure modes at the module level.

Based on the above reasons 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 PRA to further categorize and group failures of the different digital I&C modules according to the developed taxonomy to achieve a more compressed set of failure modes adapted for use in a PRA.

The main steps in developing the PRA adapted failure modes are:

1. Failure effects according to the failure modes taxonomy at the module level (chapter 0) are assigned to the different failure modes of the digital RPS example system hardware modules. Then the uncovering situation and local functional impact can be defined for the system.

2. Failure mode types describing the functional impact on I&C unit level are defined based on the failure effect and uncovering situation for the different failure modes. The failure mode types distinguish between failures detected by the fault tolerant design (detected failures) and failures that are not detected (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 PRA terminology, by defining generic failure detection means. See Table 20.

3. Based on the knowledge of functional impact on I&C unit level, whether a failure will be covered by the fault tolerant design or not and the location of the detection, makes it possible to define the failure end effect, i.e. the impact on RT/ESFAS actuation signals for a given module failure, see Table 21.

4. The last step in defining the failure modes for the digital RPS modules of the example PRA is to group all basic failure modes of a I&C module that have the same attributes for generic failure mode, generic failure detection and failure end effect. The PRA adapted failure modes is presented in Table 22.

Table 20. Demonstration of the taxonomy for PRA modelling of hardware modules, step 1.

Hardware modules	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/freeze	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, single channel	Signal stuck to current value	Non-fatal, plausible	Latent revealed by demand	Loss of specific module application function
		Non-fatal, implausible	Online detection	Loss of specific module application function
	Signal fails to opposite state	Non-fatal, implausible	Spurious effect	Spurious module application function
Digital output module, single channel	Signal stuck to current value	Non-fatal, implausible	Online detection	Loss of specific module application function
		Non-fatal, plausible	Latent revealed by demand	Loss of specific module application function
	Signal fails to opposite state	Non-fatal, implausible	Spurious effect	Spurious module application function

Table 21. Demonstration of the taxonomy for the PRA modelling of hardware modules, steps 2 and 3.

Hardware modules	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[1]	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[2]	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[3]	1oo4 conditions of specific[4] 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, single channel	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	Spurious module application function	Spurious function	Self-revealing	Spurious 1oo4 conditions of specific APU/VU outputs
Digital output module, single channel	Online detection	Loss of all module application functions	Loss of function	Self-monitoring	Specific APU/VU output acc. to FTD
	Latent revealed by demand	Loss of all module application functions	Latent loss of function	Periodic test	Loss of specific APU/VU output
	Spurious effect	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

[1] Detected by monitoring functions in the next level of I&C-units, i.e. units communicating with the faulty unit.

[2] Periodic tests according to technical specifications

[3] Detected by the self-monitoring functions implemented in the module, or by monitoring mechanisms, provided by controlling modules

[4] 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 22. Demonstration of the PRA adapted failure modes of hardware modules, step 4.

Hardware modules	Compressed failure modes	Failure detection	Failure end effect (RT or ESFAS)
Processor module	Loss of function	Monitoring[1]	All outputs of APU or VU acc. to FTD
	Latent loss of function	Periodic test[2]	Loss of all APU/VU outputs
	Spurious function	Self-revealing	Spurious APU/VU output(s)
Analog input module	Loss of function	Self-monitoring[3]	1oo4 conditions of specific[4] 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
	Latent 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 output acc. to FTD
	Latent loss of function	Periodic test	Loss of specific APU/VU output
Communication module	Loss of function	Monitoring[1]	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[1]	All outputs of APU or VU acc. to FTD
Measurement	Loss of function	Monitoring[3]	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

[1] Detected by monitoring functions in the next level of I&C-units, i.e. units communicating with the faulty unit.

[2] Periodic tests according to technical specifications

[3] Detected by the self-monitoring functions implemented in the module, or by monitoring mechanisms, provided by controlling modules

[4] 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

7.1.2.2 Software failure modes

Table 23 summarises software faults which could be considered in PRA. As discussed in chapter 0, faults can be postulated in different software modules and different end effects may be postulated. In PRA it is reasonable to consider a limited number of end effects. The selection should be large enough to cover all relevant cases (i.e. end effects)

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 effetcs since they are covered by the larger case. Large end effects include complete CCF of the system (SYSTEM), and CCF of one subsystem (1SS).

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 as long as it is impossible to refine the probability judgement, it is meaningless to refine the set of modelled software fault.

Table 23. Example screening of SW faults cases for PRA modelling. Cases 1–4 are explained below.

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

The proposed screening of software faults includes the following four cases from the end effect point of view:

1. Software fault causing loss of both subsystems (SYSTEM). This is a complete CCF covering all platforms that have the same OS. 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 leading to a crash of the prosessing units, i.e., no output signals coming from the processors. This maximal end effect covers all the other principally possible end effects. Software fault can be located in OS, EFs, COTS-SW modules, VU-AS or DCS, but it can be represented in a model by a single basic event.

2. Software fault causing loss of one subsystem (1SS). This is a complete CCF causing a fatal failure which crash the prosessing units in one subsystem, i.e., no output signals coming from the processors. There are subcases:

   [2a] Software fault is located in OS, COTS-SW modules in APUs/VUs or VU-AS. The consequence is that there is no output signals coming from the VUs.

   [2b] Software fault is located in DCS or DLC. The consequence is loss of communication between APUs and VUs. The outputs are set to specified default values

   Both subcases can be represented by a single basic event per subcase and per subsystem.

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

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

   This case can be modelled by a single basic event per set of redundant APUs or VUs.

4. Software fault causing a failure of one ore more application function. This a non-fatal failure and can lead to a failure to actuate or spurious actuation. Fault is associated with application functions in APUs (4a) and VUs (4b). In special cases, it may be worth considering application function fault only in one division (4c). For instance, there can be safety functions which only exist in one division or which have success criteria N-o-o-N in some context. Cases 4a–4c are modelled by application function and failure mode specific basic events.

In this example, the following assumptions have been made to justify the selection of software faults:

- Fault in OS can in principle cause any type of end effect. Fatal failure of identical redundant units may be relevant to take into account, if we cannot justify that this is extremely improbable. Non-fatal failure is covered by corresponding AS-fault. Non-fatal failure is assumed to affect only one I&C function (and other I&C functions functionally dependent on it).

- Fault in EF can in principle cause any end effect. The case ”fatal failures affecting redundant units” is covered by the OS fault. Non-fatal failures are covered by corresponding AS-fault. It may be of interest to study whether some extra complex EF is used in several AS, which causes a dependency between AS-modules. The most likely fault is not EF fault itself but that the EF is used in a wrong way in the AS – use of EFs is thus part of analysis P(AS-fault). Therefore there is no need to explicitly model EF faults.

- Faults in COTS-SW modules are covered by HW faults from the end effects point of view. Therefore there is no need to explicitly model these COTS SW module faults.

- Faults in DCS and DLC may require some special treatment, due to possibly unique end effect, not necessarily covered by cases 1 and 2. However, the case ”fatal failures affecting redundant units” is covered by OS fault.

- Faults in AS and FRS are tied to together, since the probability of an AS fault can be modelled conditionally

  P(AS fault) = P(AS fault | FRS fault)P(FRS fault) + P(AS fault | no FRS fault)P(no FRS fault)

FRS may be common to several AS leading to a dependency. “AS faults” are the basic events that are modelled, but “FRS faults” are taken into account in the quantification and modelling of CCF between AS-modules. The end effect of an AS fault can be fatal or non-fatal failure, which can be taken into account by a factor (similar to fault detection coverage factor). Therefore AS faults must be split into several cases depending on the end effect.

7.1.3 Procedure to develop fault trees

Based on the failure modes defined in section 0 and the safety I&C protection functions and fault tolerant design, the fault tree model can be developed in two ways. The main tasks of the procedure in a *bottom-up* perspective are:

- Grouping of failures of each module 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 function 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 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 *top-down* procedure starts from actuator functions. As an example, the emergency feedwater system (EFW) pump and the safety function core cooling is considered. In this example, the safety function, division and system levels are skipped, the procedure starts from the component. The failure modes of the pump, which is a stand-by component, are

- failure to start
- spurious stop.

The above failures may be caused by several reasons, among others failures of the safety I&C, making the link to the fault tree models of RPS. We denote the start signal of EFW-pump by EFW-ON and the stop signal by EFW-OFF.

Assuming similar RPS architecture as introduced in chapter 0, the signal path from the measurements to the pump goes via APUs and a division specific VU. The design principle of RPS is that given the critical input signals from the measurements, 2-o-o-4 is enough to create the actuation signals in APUs (EFW-ON). APUs send the signal to all VUs, which vote again by 2-o-o-4 principle, causing the start signal EFW-ON.

EFW-pumps may also be supervised by a pump leakage protection function. If a leakage is detected in the pump room, the protection system shall stop the pump (only the specific pump). The signal path is the same as for EFW-ON signal, but the measurements are different and the output signal is designated as EFW-OFF. The difference is also that EFW-OFF is division specific safety function (only the leaking train is stopped).

The failure modes of APUs and VUs are analysed in Table 24. The following fail-safe principles have been assumed:

- Voting units are assumed to fail to provide EFW-ON and EFW-OFF signals if power supply fails or if there is an internal voting unit failure (i.e. the default value is 0).
- At loss of communication between VU and APU due to a detected failure in the APU, EFW-ON will be failed to activate in a 3/4 condition and EFW-OFF will be activated spuriously in a 2/4 condition.

- In case of APU safety functions, detected failures of input signals from measurements or from other APU:s cause an actuation (i.e. the default value is 1) in an 2/4 condition.
  - EFW-ON is actuated by 2-o-o-4 low water level condition in the reactor pressure vessel, denoted by the acronym RPV-LL. There are four measurement sensors, one in all four divisions, which information is shared by all divisions.
  - EFW-OFF is actuated by 2-o-o-2 leakage protection signal in each EFW train, denoted by EFW-LEAK-x, x = 1, 2, 3, 4. There are two measurement sensors per division, and this information is *not* shared between divisions. EFW-OFF stops the affected EFW train in case of pipe break in an EFW train.

Table 24. Failure modes and causes at the I&C unit level with respect to the EFW function.

Unit	EFW function failure mode	Failure causes
VU	Failure to actuate EFW-ON	VU internal failure • undetected failure • detected failure Power supply failure No EFW-ON from APU:s (3-o-o-4)
	Spurious EFW-OFF	VU failure causing spurious signal • detected failure VU-APU communication link failure • detected failure Spurious EFW-OFF from APU
APU	No EFW-ON from APU	APU internal failure • undetected failure Failure of EFW-ON actuation from APU:s (3-o-o-4) • undetected failure Failure of measurements for EFW-ON (3-o-o-4) • undetected failure
	Spurious EFW-OFF	APU internal failure • detected failure Spurious EFW-OFF actuation from APU • detected failure Spurious EFW-OFF measurements (1-o-o-2) • detected failure

In the top-down approach, the next step is to analyse the I&C unit failures at the module level. Each item listed in the column “Failure causes” is further associated with the module level failure modes. The hardware modules are the same as listed in Table 22. The following application software modules are considered in this example:

Table 25. Example application software (AS) modules in VUs and APUs.

Unit	AS module	Condition
VU	EFW-ON	2-o-o-4 EFW-ON from APUs 1–4
	EFW-OFF	EFW-OFF from the same division's APU
APU	EFW-ON	2-o-o-4 RPV-LL from APUs 1–4
	EFW-OFF	EFW-LEAK from the same division's APU
	RPV-LL	2-o-o-4 RPV water level below "very low level" measurement from division 1–4 RPV level measurement sensors
	EFW-LEAK	1-o-o-2 water level in the EFW pump room over the leakage criterion from the same division's leakage detection sensors

A schematic fault tree for the failure to actuate EFW-ON in one division is shown in Figure 19. Fault tree is developed down to boxes of hardware and software modules failure modes, which are listed in Table 26. Only one redundancy (division 4) is developed at the APU level. The other divisions are identical.

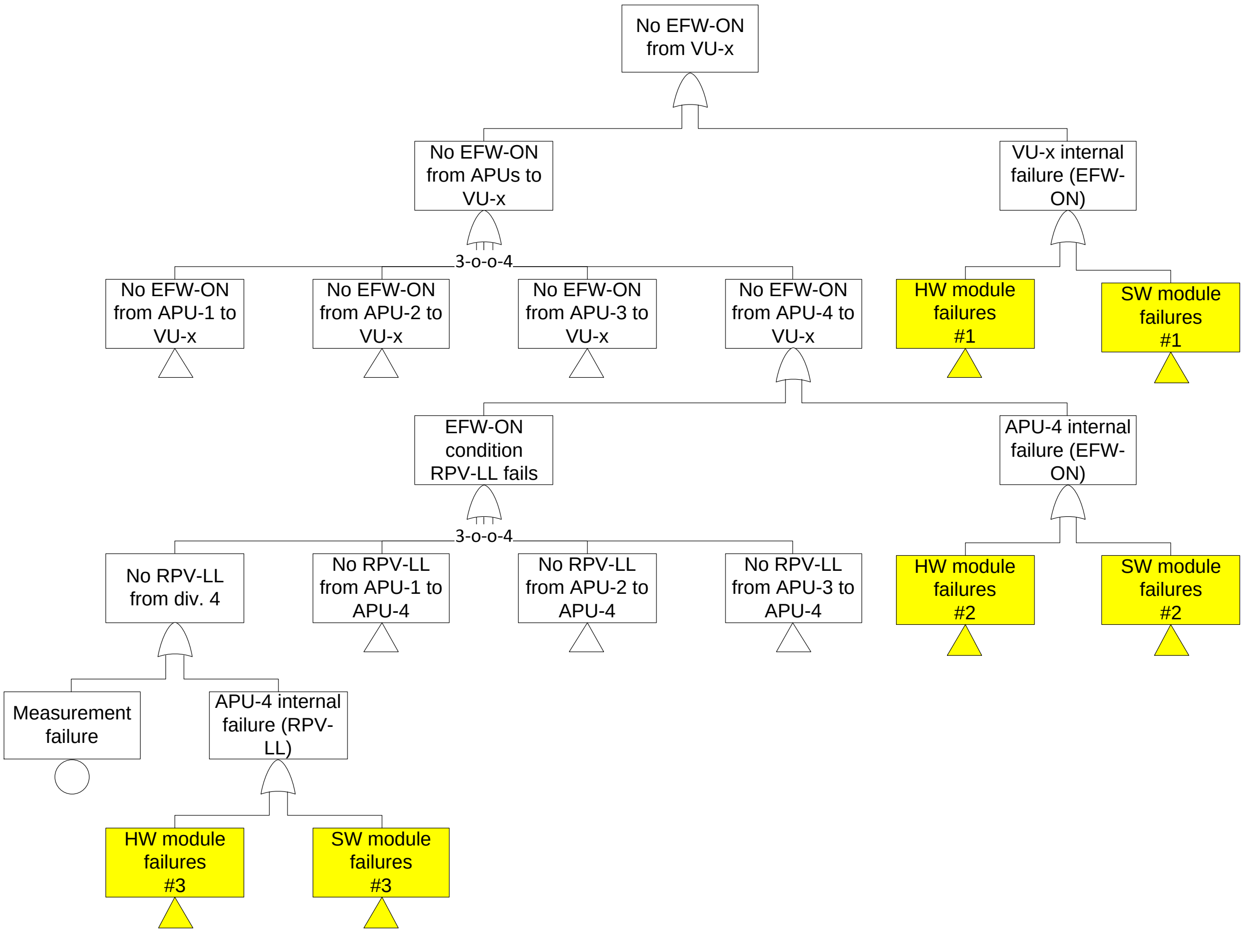

No EFW-ON from VU-x
No EFW-ON from APUs to VU-x
VU-x internal failure (EFW-ON)
3-o-o-4
No EFW-ON from APU-1 to VU-x
No EFW-ON from APU-2 to VU-x
No EFW-ON from APU-3 to VU-x
No EFW-ON from APU-4 to VU-x
HW module failures #1
SW module failures #1
EFW-ON condition RPV-LL fails
APU-4 internal failure (EFW-ON)
3-o-o-4
No RPV-LL from div. 4
No RPV-LL from APU-1 to APU-4
No RPV-LL from APU-2 to APU-4
No RPV-LL from APU-3 to APU-4
HW module failures #2
SW module failures #2
Measurement failure
APU-4 internal failure (RPV-LL)
HW module failures #3
SW module failures #3

Figure 19. Schematic fault tree for failure to actuate EFW-ON in division x. Yellow boxes correspond to hardware and software module level failure modes listed in Table 26.

Table 26. Failure modes at the module level with respect to the end effect "failure to actuate EFW-ON". Indexes #1–#3 refer to the transfer gates of the fault trees above.

Unit	EFW function failure mode	Module level failure modes
VU	#1 No EFW-ON from VU	Hardware modules: • digital output module, loss of function • digital output module, latent loss of function • processor module, loss of function • processor module, latent loss of function • backplane, loss of function • power supply, loss of function • digital input module, loss of function • digital input module, latent loss of function Software modules: • SYSTEM level CCF in SW modules (case 1) • 1SS level CCF in SW modules (case 2a) • 1VU level CCF in SW modules (case 3b) • EFW-ON application SW fault in VU (case 4b)
APU	#2 No EFW-ON from APU to VU	Hardware modules: • processor module, latent loss of function Software modules: • SYSTEM level CCF in SW modules (case 1) • 1SS level CCF in SW modules of DCU (case 2b) • 1APU level CCF in SW modules (case 3a) • EFW-ON application SW fault in APU (case 4a)
	#3 EFW-ON condition RPV-LL fails in APU (3-o-o-4)	Hardware modules: • analog input module (providing RPV-LL), latent loss of function (own division) • measurement signal (RPV-LL), latent loss of function (own division) Software modules: • RPV-LL application SW fault in APU (case 4a)

A schematic fault tree for the spurious EFW-OFF in one division is shown in Figure 20. Fault tree is developed down to boxes of hardware and software modules failure modes, which are listed in Table 27. The other divisions are identical.

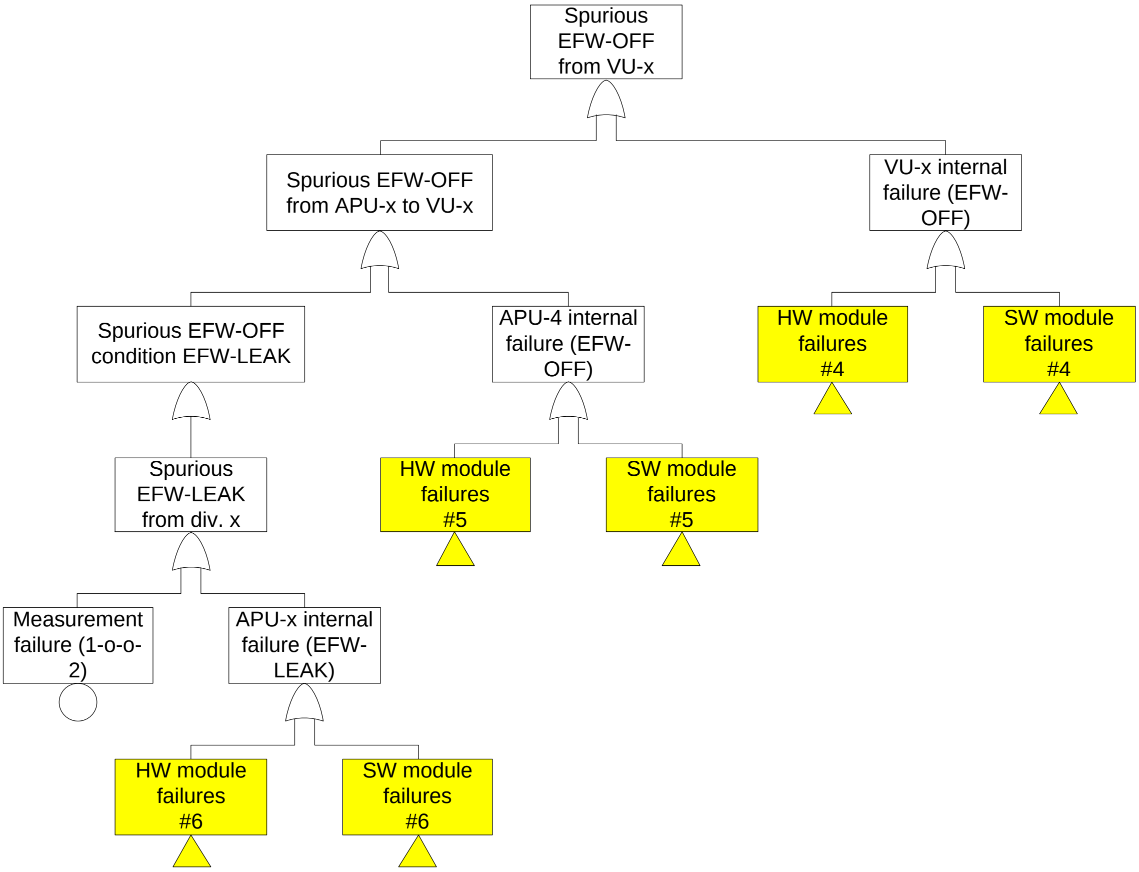

Spurious EFW-OFF from VU-x
Spurious EFW-OFF from APU-x to VU-x
VU-x internal failure (EFW-OFF)
Spurious EFW-OFF condition EFW-LEAK
APU-4 internal failure (EFW-OFF)
HW module failures #4
SW module failures #4
Spurious EFW-LEAK from div. x
HW module failures #5
SW module failures #5
Measurement failure (1-o-o-2)
APU-x internal failure (EFW-LEAK)
HW module failures #6
SW module failures #6

Figure 20. Schematic fault tree for spurious EFW-OFF in division x. Yellow boxes correspond to hardware and software module level failure modes listed in Table 27.

Table 27. Failure modes at the module level with respect to the end effect "spurious EFW-OFF". Indexes #1–#3 refer to the transfer gates of the fault trees above.

VU	#4 Spurious EFW-OFF	Hardware modules: • VU-APU communication link failure, detected failure Software modules: • EFW-OFF application SW fault in VU (case 4b)
APU	#5 Spurious EFW-OFF from APU to VU	Hardware modules: • processor module, loss of function • backplane, loss of function • power supply, loss of function Software modules: • EFW-OFF application SW fault in APU (case 4a)
	#6 Spurious EFW-OFF condition EFW-LEAK in APU	Hardware modules: • analog input module (providing EFW-LEAK), loss of function • measurement signal (EFW-LEAK), loss of function (1-o-o-2) Software modules: • EFW-LEAK application SW fault in APU (case 4a)

7.2 Basic component level example

7.2.1 Example system diagram

The example system in Section 0 will be used to demonstrate the application of failure mode taxonomy at a level of basic components to a digital module. The structure of a generic digital safety-related I&C system is shown in Figure 5 of Section 0.

7.2.2 The selected example module for demonstration

For demonstration purpose, one of the modules of the example system will be selected. Note, in Figure 4 of Section 0, racks/subracks of processing units or acquisition units may contain multiple modules as defined in the summary table of inputs from different organizations in Appendix A, and the modules in each column of that figure are not necessarily all microprocessor-based. For example, a fan module is merely a mechanical device with electrical/electronic circuits, and therefore, can be considered a basic component.

For the example system, analog and digital input modules, processing modules, and communication modules are all based on microprocessors. In this demonstration, the analog input module (AIM) of the acquisition unit is selected. In order to come up with failure modes of the selected module, the failure modes of components of the analog input module will be identified first. A set of generic failure modes of those components has been defined in Section 0. The failure modes of the components are represented in terms of the associated input signals that are processed through the module, and the corresponding failure effects will become the failure modes of the module based on status of module, which may be determined by assessing the output signals. Apparently, the application software of the selected module will play a major role in determining the status of the module.

For the selected example system, analog input modules send digitalized input data to the processor module via the backplane bus and thus, output devices such as current loop and demultiplexer. The communication is not involved in the AIM based on our understanding of the design demonstration. Note, communication related components and the associated failure modes were not discussed in NUREG/CR-6997. The components related to communication need to be identified and their failure modes have to be developed if a communication module is selected.

The major components of the analog input module were shown in Figure 5 and the associated failure modes were identified in Table 10 summarized below:

1. Current I/O loops: Their failure modes are current signal fails (drifts) high or low. It is assumed that if the current starts drifting, it will eventually reach the high or low detection threshold.

2. MUX: Failure modes of MUXs and DEMUXs are defined in terms of the analog signals they process, which include a loss of one or all signals (same as signal fails low).

3. A/D converters: All analog inputs of the same module share them. The failure modes of an A/D converter include all bits of the A/D stuck at zeros, all bits stuck at ones, and a random bit-failure of the A/D converter. The selected AIM module does not have a D/A converter.

4. Microprocessor of the AIM: Failure modes considered are (1) the microprocessor seems to be running normally but sends erroneous output and (2) the microprocessor stops updating output.

5. Associated components of a microprocessor, such as the internal bus, RAM, ROM, BIOS, and buffer: It is conservatively assumed that each component has only one failure mode, i.e., a loss of the component, which entails the loss of the functions performed by the component.

6. Address logic: The failure mode is assumed as a loss of the address logic, so that the microprocessor cannot access the intended information upon loss of the address logic.

7. Hardware common-cause failure (CCF): The hardware of the same type of modules of different diversion is identical. The occurrence of a hardware CCF may fail the entire system.

8. Power supply: The failure mode is a loss of power supply. CCF of the power supplies for different channels may also occur. Here it is assumed that a single power supply is provided for the AIM. A loss of the power supply will cause the AIM to send no output signals.

In this demonstration, the applicable failure modes are further evaluated by making assumptions about design details of the AIM. Due to resource constraints, we can only evaluate impacts of single failures of individual components on the module where the application software is running normally. By doing so, the failure rate of the module can be approximated by summing failure rates of all the component failure modes that will fail the module. Combinations of individual component failure modes, i.e., high order failure sequences, are ignored assuming that they are rare events. By considering combinations of component failure modes and propagating them, many different module level failure mode can be defined in terms of the output signals of the module that may not be captured by the module level failure modes of this report that are based on only one output signal.

7.2.3 Major assumptions

The following information is needed for the demonstration: (1) individual input signals (both analog and digital) of process parameters, e.g., temperature or pressure signals, and individual output signals, e.g., a trip signal; (2) a high level description of the overall architecture of the module hardware design including interface with other modules; (3) how individual signals are processed by the software or firmware on the module, e.g., filtering, before output signals are generated; and (4) a description of fault-tolerance features provided by both software/firmware and hardware, e.g., sanity check of signals via deviation and/or out-of-range comparison.

Some assumptions had to be made for a demonstration purpose in addition to some generic assumptions made in Section 3.3 of [NUREG/CR-6997], e.g., the system is running all the time, a component can only fail once, and failure of a component will not cause failures of other components that are physically connected to that component et al. The detailed discussions on these generic assumptions were provided in [NUREG/CR-6997]. In this case study, some additional assumptions about the hardware architecture are listed below:

(1) Since there are multiple input signals to the analog input module, it is assumed that all the input signals will be treated or processed in a similar manner by the AIM. The process parameters (e.g., pressure or temperature signals) that are used to determine the generation of a trip signal are connected to the AIM.

(2) It is further assumed that redundant signals are provided to the AIM for each parameter, e.g., two pressure signals that provided by redundant pressure sensors are fed to the AIM. This is a common practice in digital system design.

(3) The input signals are current signals and carried, e.g., by 4–20 mA current loops in this study. A current loop is the most popular device carrying an input signal.

(4) The input signals will be processed and digitalized. The AIM does not have output devices (current or voltage). Instead, the digitalized input signals will be sent by the microprocessor of the AIM to the processor module via backplane bus. All input/output and processor modules will be exchanging data via the backplane bus in the same cabinet.

The following assumptions are made for the application software running on the AIM. These assumptions are applicable to different types of input signals, e.g., pressure, temperature etc.

(5) If a signal is out-of-range, the signal is considered a faulted signal by the application software of the AIM microprocessor and will be marked as a faulted signal. If the other (redundant) signal is not faulted, it will be used and processed. However, an alarm will be sent to the operator and/or a local alarm device.

(6) If the other signal is also faulted, the AIM is assumed to enter a pre-defined state, which is very design specific. However, it is reasonable to believe this failure will not cause any significant impact on the entire system.

(7) If both redundant signals are not faulted, they will be compared to determining the deviation between them. If the deviation is within an acceptable range, they will be averaged and processed by the AIM microprocessor. If the deviation is large enough, the AIM is assumed to enter a pre-defined state. This is based on the reasoning that both signals may be faulted or invalid and the AIM microprocessor should not trust either of the signals.

In addition, the watchdog timer of the AIM microprocessor may also be able to detect some failures:

(8) If a CPU halt, either due to a hardware or software failure, is detected by the watchdog timer of the microprocessor, it is assumed that the AIM enters a pre-defined state.

(9) It is assumed that a single power supply is provided for the AIM. A loss of the power supply will cause the AIM to send no output signals.

Based on the above assumptions about the hardware and software of the AIM, Table 8 presents the component failures of the AIM and the failure effects of these failure modes on the AIM.

Table 28. Failure modes and effects analysis for the example AIM.

Failure mode	**Failure mode detected by**		**Failure effects on the AIM**	**Comments**
	Application software	**WDT**		
Software CCF	-	-	Incorrect output signals from the AIM[5].	It is assumed that the CCFs of software or hardware will fail all of the analog input modules of the RPS channels. The output of the AIMs ca nnot be decided deterministically and therefore, are assumed to be incorrect. This is considered an undetectable failure of the AIM.
Hardware CCF	-	-		
The software on the main CPU seems to be running normally but sends erroneous output	No	No	Incorrect output signals from the AIM.	This is considered (conservatively) an undetectable failure of the AIM.

[5] Note that "incorrect signals" may also be "valid signals".

Failure mode	**Failure mode detected by**		**Failure effects on the AIM**	**Comments**
	Application software	**WDT**		
Software halt (CPU stops updating output)	No	Yes	The AIM enters a pre-defined state.	When the WDT no longer receives a toggling signal, it should act in a pre-defined manner, e.g., reset the microprocessor, according to Assumption (8). This is a watchdog detectable failure. Note that the WDT has to function normally.
The microprocessor seems to be running normally but sends erroneous output	No	No	Incorrect output signals from the AIM.	This is considered (conservatively) an undetectable failure of the AIM.
The microprocessor stops updating output	No	Yes	The AIM enters a pre-defined state.	When the WDT no longer receives a toggling signal, it should act as pre-defined, e.g., reset the microprocessor. The AIM should be in a pre-defined state, according to Assumption (8). Note that the WDT has to function normally.
Loss of internal bus of the AIM	No	Yes	The AIM enters a pre-defined state.	The input and output of the CPU rely on the internal bus. It is assumed that this failure is detected by the WDT, if the WDT functions normally, because the microprocessor of the AIM may not be able to update the output including the toggling signals to the WDT. The AIM should enter a pre-defined state, according to Assumption (8).
Loss of RAM	No	Yes	The AIM enters a pre-defined state.	Application software has to be loaded into RAM to run it. Thus, the application software cannot run upon a loss of RAM. It is assumed that the WDT can detect the loss of RAM because the application software of the AIM will no longer run and send out toggling signals to the watchdog timer. The AIM should enter a pre-defined state, according to Assumption (8).
Loss of BIOS	No	No	Incorrect output signals from the AIM.	The input and output operations of the CPU rely on BIOS routines. However, it is unknown whether a loss of BIOS will cause a complete loss (or a partial loss) of inputs to and outputs from the application software; hence, it is assumed (conservatively) that the output signals become incorrect when this failure occurs.
Fail (drift) high or fail (drift) low of current loop device	Yes	No	AIM does not fail but with degraded redundancy. This may also be considered a pre-defined state of the system.	The current loop is a linear device that may fail high or low, resulting in the associated I/O signal failing high or low. Fail low includes failures of fail to zero. As discussed in Assumption (5), the other signal will be used and the AIM functions as designed. An alarm signal is assumed to be generated.

Failure mode	**Failure mode detected by**		**Failure effects on the AIM**	**Comments**
	Application software	**WDT**		
Loss of all signals from MUX	Yes	No	The AIM enters a pre-defined state.	Loss of a signal means that the signal fails low. All analog inputs share the MUX. This failure mode indicates that all analog signals related to this MUX fail low. The AIM should enter a pre-defined state, according to Assumption (6).
Loss of one signal from MUX	Signal Dependent	No	AIM does not fail but with degraded redundancy. This may also be considered a pre-defined state of the system.	The failure mode indicates a loss of a specific analog signal. The other signal, if valid, will be used and AIM continues to function, according to Assumption (5).
All 16 bits of A/D converter stuck at zeroes or ones	Yes	No	The AIM enters a pre-defined state.	1. Both A/D and digital/analog converters are linear ICs. The A/D converter is shared by all analog inputs, and its loss will entail the loss of all analog inputs. 2. Stuck at zeros or ones indicates that all analog signals fail low or high. The AIM application software can detect failures and then enters a pre-defined state, according to Assumption (7).
Random bit failure of A/D converter	No	No	Incorrect output signals from the AIM.	Although the AIM application software may detect some random failures, they are conservatively assumed to be undetectable and will send out incorrect output signals.
Loss of address logic	No	No	Incorrect output signals from the AIM.	Although some failures of address logic might be detected by the application software, it is conservatively assumed that a loss of the address logic will result in an undetectable failure of the AIM and generate incorrect output signals.
Loss of power supply	No	No	No output signals from the AIM.	It is assumed here that this failure causes an absence of output signals from AIM. Note, this assumption is conservative because an absence of output from the AIM might be detectable by other modules, e.g., the Processor Module.

It should be pointed out that only single failures are considered in most of the cases discussed in the above table. The importance of the considering the combinations of individual failures should be highlighted since they may have significant contribution to the overall system reliability, as demonstrated in [NUREG/CR-6997]. A detectable failure mode in the table is detectable only when the fault tolerant mechanisms (software or hardware based) are normally operating. A fault tolerant device may also fail and needs to be considered in the reliability modelling.

The propagation of the component failure modes stops in 0 at the module level, i.e., the failure effects of the component become the module level failure modes. In fact, the component failure modes can be further propagated through all of the intermediate levels of abstraction, i.e., I&C units and division level, until the entire RPS system level, provided that the required design information is available for performing the analyses. This will be illustrated below by using an example failure mode of an A/D converter based on some assumptions on the system.

Taking failure mode "random bit failures" of the A/D converter" as an example, the effects will be the incorrect output signals from the AIM. Note, the signals are incorrect but still valid, which therefore cannot be detected by the acquisition I&C unit since there is no comparison to the redundant signals. The acquisition I&C unit will send out the incorrect but valid signals to the processing I&C unit. If each acquisition I&C unit sends the outputs to all of the processing I&C units in the four divisions, and the processing units performs a 3 out of 4 logic vote to determine the correct signal values, then this A/D converter failure mode will be contained in the division containing that A/D converter and the outputs of the four processing units in the four divisions will all be correct. Therefore, the processing units, the voting units, the four divisions, and the entire system are all working correctly. All of the individual failures and the combined failure sequences can be analysed similarly, which is, of course, very difficult to do manually.

8. EVALUATION OF THE FULFILMENT OF THE REQUIREMENTS

Section 6 proposed a taxonomy that classifies and organizes digital I&C failure modes for the purposes of NPP PRAs or PSAs. Section 4.4 identified 9 desirable criteria that the proposed taxonomy should meet to support application to PRA or PSA. This section summarizes evaluation results of this taxonomy against these 9 criteria.

This taxonomy classifies and organizes digital I&C failure modes using a three elements tuple (<location, effect, uncovering condition>) scheme. The location is determined by the levels of detail and the example system. Failure effects are classified into "fatal/non-fatal". "Fatal" is further grouped into "ordered/haphazard", and "non-fatal" into "plausible/implausible behavior". Uncovering situations are either "revealed by spurious actuation", "revealed by demand" or "uncovered by online/offline detection mechanisms".

Each of the nine criteria was evaluated as either "Met", "Open" or "Not Met". "Met" means the taxonomy fully satisfies the criterion under study. "Not Met" indicates the taxonomy, as currently defined, cannot fulfill requirements specified by the criterion. "Open" reflects situations where further review is needed to determine if the taxonomy fulfills a criterion. "Open" is used for cases where nothing was identified that would unequivocally demonstrate that the criterion is "Not Met"; however, more effort is required to demonstrate that the criterion is "Met".

Table 29. Evaluation of the fulfilment of the requirements.

Criterion (Section 4.4)	Description	Evaluation	Comments
Criterion 1	Be defined unambiguously	Met	The location, effect and uncovering situation are all defined unambiguously
Criterion 2	Form a complete / exhaustive set	Met	The taxonomy, as currently defined, provides a complete/exhaustive set in that any postulated failure can be categorized within the taxonomy.
Criterion 3	Be organized hierarchically	Met	The "location" element is defined hierarchically by 5 levels, e.g. system, division, unit, module and basic component; The "effect" element is organized as "fatal/non-fatal" and "fatal", "non-fatal" are further hierarchically defined. Similarly the "uncovering situation" element is classified into "detected/non-detected", "detected" and "non-detected" are further organized hierarchically. All three elements are organized hierarchically. This criterion is met.
Criterion 4	Be mutually exclusive	Met	The three elements characterize failure modes from three different aspects and they are mutually exclusive. The "location" element is defined by the 5 levels and they are mutually exclusive. Uncovering situations are classified by whether failures detected before demand. If "no", then into "revealed by demand"; if "yes", then into either "revealed by spurious actuation", or "uncovered by online/offline detection mechanisms". This classification scheme is mutually exclusive.
Criterion 5	Data to support the taxonomy should be available now or in	Open	Although data there is only a limited amount of data currently available that aligns with the taxonomy, future data collection efforts could be designed to collect data in

Criterion (Section 4.4)	Description	Evaluation	Comments
	the future		a manner consistent with this taxonomy. However, because additional data collection development effort is required in the future, this criterion is “Open”.
Criterion 6	There should be analogy between failure modes of different components	Met	Because any postulated failure mode is categorized by <location, effect (fatal/non-fatal), uncover situation>, similar components will have analogous failure modes under the taxonomy.
Criterion 7	At the very least, the lowest level of the taxonomy should be sufficient to pinpoint existing dependencies of importance to PRA modelling	Open	Although the taxonomy covers an appropriate level of detail to capture postulated dependencies (e.g., down to the “basic component” level), modeling of dependencies was beyond the scope of this effort. This criterion is considered to be “Open” pending additional effort to develop dependency models for the postulated I&C system.
Criterion 8	Should support PRA practice, and fulfil PRA requirements/conditions	Open	Chapter 7 of the report demonstrates that the taxonomy can be successfully applied to a pilot study. However, there may be practical limitations for the taxonomy in supporting a full range of potential PRA applications. For example, the taxonomy may not adequately address the following cases: • The taxonomy appears to be insensitive to cases where the timing of a failure may be important to the PRA evaluation. For example, a function performed too early or too late may lead to different PRA accident sequences. However, the taxonomy, as defined, may not be able to make a meaningful distinction between these cases. • Some postulated failures may result in different functional behaviour, even though they may be classified the same using the current taxonomy. For example, a stuck bit error (i.e., bit stuck at “1” or “0”) could result in different functional behaviour(s). For practical PRA applications, it may be desirable to make a distinction between these cases. Therefore, pending additional effort to resolve the above issues, the criterion is considered to be “Open”.
Criterion 9	Should capture defensive measures against fault propagation (detection, isolation and correction) and other essential design features of digital I&C	Not Met	The “uncovering condition” element characterizes whether a failure is detected, and how is detected (i.e., the detection portion of fault propagation). However, the “isolation” and “correction” features of fault propagation are not covered in this taxonomy. In addition, “other essential design features” are not identified in this study. Therefore the evaluation against this criterion is “Not Met”. Although this criterion is not fully met, it is expected that this issue will not significantly limit the PRA application of the failure taxonomy. The needs for further research in this area are discussed in Section 10.5.

9. POSSIBLE DATA SOURCES AND DATA COLLECTION NEEDS

9.1 Data source overview

The basic goal of the taxonomy developed is to prepare the basis for PRA modelling. Since the PRA is a qualitative and quantitative method, it needs to be supported by relevant credible set of input qualitative and quantitative data. In another words, failure effects as defined by the taxonomy could serve as a basic events in a Fault Tree part of the PRA model. They need to be well-defined and quantified by probabilistic parameters to allow calculation.

In the opposite of the mechanical components, the progress in the electronic components development is very rapid and the market lifetime of electronic products is very low. This makes the electronic components data gathering very uneasy. Additionally, the real failure causes is very often complicated to allocate (cascade, avalanche effects). To make it worse, digital components are usually subjects of different levels of diagnostic tests (either online or offline) and according to the test issues different probabilistic parameters are used. Particular tests success rate estimation is very tricky but it has severe impact on the calculation results. All in all, the uncertainty of the probabilistic calculation of digitally based systems is very high.

In general, sources of probabilistic data are:

- Experiments
- Operational experience.

The practically used data sources are:

- Predictive systems/database
- Information from producer/vendor
- Information from operators/users
- Engineering judgement.

9.2 Predictive systems/database

The level of detail typically differs in particular sources. The predictive database, e.g. the most famous “Military Handbook for Reliability Prediction of Electronic Equipment" [MIL-HDBK-217] is mostly oriented on the basic electronic components like diodes, condensers, etc. Even if some databases, like [RDF 83] and [UTE C 80 810] provide statistics about failure modes (like short circuit, open circuit, leak diode etc.), most do not distinguish between them.

The common technique that establish probabilistic reliability characteristic of complex electronic components is the parts count reliability prediction method. This method is often incorporated into prediction software codes along with databases containing generic data. In the simplest case probabilistic parameters (e.g. failure rates) of all basic components are summed regardless of their failure mode and impact on the function. This approach is rather conservative and can be refined with the use of FMECA (respectively FMEDA) at basic component or module levels. The probabilistic values obtained have a high level of uncertainty.

Furthermore, the level of detail in PRA model is lower, typically at least electronic board (the so called module level).

9.3 Information from producer/vendor

Taking information from vendor seems to be much more credible way how to get data for electronic components and electronic boards. Unfortunately, both producers and vendors very rarely perform any tests to simulate performance in operational conditions. Very often they try to utilize results from lifetime tests, which correlates with operational conditions faintly if at all. They are also sometimes rather reluctant to share operational experience. As a consequence, the data obtained from vendors are frequently based on prediction methods and surprisingly rather conservative (i.e. worse than reality), because vendors are satisfied when they demonstrate the reliability required by the equipment quality category (e.g. given level of Safety Integrity Level – SIL). Recent reliability databases propose guidelines to correlate reliability data from vendors with operational conditions [UTE C 80 811, IEC 62380].

Some vendors (like AREVA NP GmbH for TELEPERM XS platform) provide a quantitative assessment of the failure modes and effects of their modules. These FMEA reports typically give in a first step an overview of the operating principles of the modules and their self-test and self-monitoring features. They present, in a second step, the failure modes of the modules, their causes and their effects on a functional basis. They indicate the internal mechanisms that are typically used to detect malfunctions.

In a third step, quantitative assessment is made. In this calculation, the module may be subdivided in several sub modules or function blocks, with one individual failure rate per block. If the result shows different failure modes within the same function block, additional reasoning is provided in the text explaining the distribution of the failure rate to the various failure modes and, if applicable, to the different failure effects.

For each failure mode, percentages of coverage by online mechanisms are given based on the analysis. Percentages of coverage by some engineered monitoring features implementable in the design of the I&C system are also sometimes given. The overall failure rates can be calculated based on some standards like Siemens Standard SN 29500 [SN 29500] with the “part count” or “part stress” method, similar to [IEC 61709].

9.4 Information from operators/users

Definitely best results can be obtained by gathering information from operational experience from utilities. This approach however has a lot of constraints. As stated before, the production life of electronic systems is fairly low and it is problem to get information from the implementation of the same system of the same generation. However, it is acceptable to use data from the previous generation in the case it is based on the same or at least similar technology. Moreover it is also acceptable to use data from other similar types of application, e.g. fossil plants. The big advantage of operational experience data gathering is that it usually concerns rather complex modules (not basic components), which could be similar to those used as basis for PRA modelling. On the other hand, the disadvantage is often missing realistic root cause analysis and even the failure mode could be uncertain. Similarly to all other electric components, the failure could reveal at the different place to the real cause. Without forensic analysis it is sometimes impossible to disclose the real causes of cascade or avalanche failures.

9.5 Engineering judgement

Even if does not look so, engineering judgement could give a very realistic estimate of probability of failure behaviour of electronic (but not limited to) components. However it requires systematic approach and unbiased experts. The advantage of it is that the analyst can define boundary and failure mode of investigated module according to the future use in the PRA model.

9.6 Data sources gathering and incoherency treatment

The obvious possibility is to try to gather probabilistic parameters from different sources and test them how they mutually correlate. There are some possibilities what to do if there is a significant gap between numbers obtained using different sources. The more sophisticated is to use appropriate mathematical apparatus like Bayesian method. However, the more usual in PRA is a conservative approach – to use the worse number and check whether it has a significant impact on the overall results (e.g. CDF). The conservative approach is acceptable for e.g. licensing acceptance but it is improper for risk informed decision making based on the instantaneous risk calculations.

9.7 Time related data

The failure rate or probability of failure on demand are however not the only parameters which determine the event probability. The probabilistic model of basic events corresponding to the failure effects and parameters like mean time between periodical tests and test coverage could have much bigger influence than the failure rate itself.

For the standard running mechanical technological equipment we treat intervals between periodically tested equipment as firmly done as well as considering of their reliability status “as good as new” after the test is considering being an acceptable approximation. This is not true in the case of digital components. The wide range of tests with different “success detection rate” (also called “Failure detection coverage factor”) and different period exits starting from online diagnostic to the tests performed once per campaign or even several campaigns. Moreover, there is no reason to consider the digital components after the test just like error free. The very important data regarding electronic devices is the effectiveness of particular levels of diagnostic (test detection success rate). To establish the ratio between detected and latent unrevealed failures is a tricky issue and usually the expert judgement of the designer is the only source of information and uncertainty of such estimate is pretty high. The influence on the overall failure probability (unavailability of the component) could be in the order of magnitude and even higher.

Another time dependency which could have a serious impact on the reliability of I&C is time needed for maintenance. It is generally undervalued according to operating manuals prescribed by vendors. In the operational practice, the plant maintenance producers are much more restrictive in order to protect systems against human fault (e.g. it is better to switch off the whole cabinet for the exchange of a single signal diode then to risk partial trip made by slipped out screwdriver). This is however mostly issue of a preoperational PRA of either new-build plants or modernized I&C. Otherwise this type of data is relatively easy to collect from the operational history.

9.8 CCF modelling influence

Finally, CCF parameters are by far the most influential type of data. It is sometimes questionable to which extent are the CCF probabilities correlated with basic event probabilities. Moreover current PRA software is not able to handle big groups of CCFs as well “multidimensional common dependencies” (e.g. electronic boards in the same cabinet versus electronic boards in different rooms). The SINTEF reliability method and data collection provides estimates for CCF parameters of I&C safety systems [SINTEF-2010].

9.9 Conclusion

There is no general clue how to procure data for basic events related to electronic components or modules. Obviously the operational experience data gathering is the best option if applicable. The taxonomy of failure modes proposed in this document represents a sound basis to establish effective and reasonable data gathering.

Fortunately, the safety I&C functions do not usually dominate the list of minimal cut sets in the most of PRAs. It allows to be rather conservative in the I&C components reliability data estimation and check the quality of the I&C design. On the other hand, one could have been more accurate in the cases, when the PRA is used as a basis for risk informed decision making applications like risk monitoring or accident sequence precursors analysis.

Furthermore, the importance of failure modes *failure rates/probabilities of failures on demand* accuracy should not be overestimated since there is a big influence of other parameters of basic events reliability models as well as CCF parameters on the overall results.

10. FUTURE WORK

10.1 Task objectives and scope

The objective of the task was to contribute to the definition of a structured and community shared digital I&C failure modes taxonomy. It has to be highlighted that, today, there is no international consensus in this matter and the existing PRA uses a variety of models (more or less complex, functional or component related) to take into account the failures of digital I&C hardware and software in the PRA models. The existing PRA models techniques may need to be upgraded in order to make possible the modelling of digital I&C failures in a sound and transparent manner.

In this context, and taking into account the dimension of the problem, the task objective was principally to start the international cooperation on this field, based on the participants experience and expertise. In the future, more and more PRA for plants having digital I&C systems are expected to be developed, especially for new reactors and for renovated exiting reactors. The task may be later completed to include wider experience. Possibly the analysis of the experience related to the implementation of the failure modes taxonomy suggested by this document, may be also performed.

The variety of the digital I&C systems can be very large (architecture, platforms, applications). However, as shown by the safety analysis (deterministic and probabilistic), the safety importance of different systems is not same. Taking into account the limited resources allocated for the task, it was decided to treat in priority the most sensitive systems, i.e. the reactor trip system and the ESFAS systems. The regulation systems, the communication devices (plant networks), the human interfaces, etc. are not included in this task and may be addressed in the future. Nevertheless, the taxonomy is not meant to be just for the protection systems, but the user of the taxonomy should be aware that some assumptions made may not be valid for control systems and for other systems not included in the protection systems.

The task focused at system level and below, the higher plant I&C architecture level related failure modes may need to be addressed in the following actions.

10.2 Modelling methods

Though modelling was not within the scope of this project. Most participants appear to have the fault tree modelling method in mind, though it is not clear if this method can capture all dependencies (e.g., communications between channels), fault tolerant features (e.g. self-diagnosis), and software-hardware interactions (e.g., changes to the software logic upon detection of a hardware failure).

Capturing all dependencies may make a model much more complicated than a model for an analogy system. Therefore, if some dependencies are not explicitly modelled in a model, they have to be treated conservatively or shown to be negligible. However, to show that dependencies are negligible they need to be explicitly modelled. Some dependencies might be difficult to model in a traditional fault tree model. Whether or not a fault tree model adequately captures all dependencies and how software failures should be included in a reliability model remain to be investigated. In this context, several possibilities to assess dependencies outside of particular PRA project, e.g. simulation of the failure propagation inside of network topologies, root cause analysis of the maintenance induced failures of the software and hardware, were mentioned by the participants during the discussions. The subject of improving the methods for comprehensive identifying and modelling of dependencies can be a subject for future works.

The discussion during the task development leaded to the conclusion that the modelling of digital I&C in the PRA will require some simplifications and grouping. This is acceptable if the model is traceable and realistic. The proposed taxonomy then implicitly includes several simplifications, which need to be taken into account and explained while using it for practical modelling. A desirable method might treat the following challenges: (1) how to include and quantify software failures, (2) how to model the fault tolerance features and associated dependencies, (3) what is the right level of detail of modelling, and (4) availability of applicable failure data

The task results presented in this report represents the basis for the modelling of the digital I&C in the PRA. Next step might be represented by the development of guidelines to effectively integrate the digital I&C in PRA models (even tree level and fault tree level), including, beside protection systems, the regulation systems, the communication devices (plant networks), the human interfaces (actuations, alarms, indications) as well as the support systems (electrical power supply, HVAC, etc.). The guidelines may address the important points highlighted in this report, like balance between simplifications and complexity, appropriate level of details, identification and modelling of dependencies, identification and modelling of CCF, modelling of software failures. The development of this kind of guidelines will allow to further proceed, as continuation and as complement of the work already done (DIGREL as well as previous WGRISK related tasks), to the better understanding and modelling of digital I&C in the PRA.

10.3 Modelling of software

How to model and quantify software failure probability is an area of on-going research. Software failures are conditional, i.e., on the environment it is being executed. How can this be captured by the existing modelling methods? It is still the question whether it is possible to model software failures by existing methods. Software failures consist only from systematic failures, although the environment in which this failure will be activated will occur randomly. It is commonly recognized that it is not possible to demonstrate that a software program is bug free. So methods to quantify software failure probability are very important.

Even the intrinsic probability of failure of application software may be very small, the causes of the failure of application software are often misleading of the specification or specification errors as for example ARIANE accident (ARIANE), failures of the safety I&C of the refuelling machine, etc. However in case of a specification error the test coverage is more as questionable.

Additionally, some hardware modules such as input or output boards may be microprocessor or microcontroller-based, and may contain their own software. Such software is part of the platform software and cannot be altered by the I&C system designers, and that the level of complexity of such software is significantly reduced compared to the software of the various I&C units. The future work may also consider the programmable hardware, such as, gate arrays, complex programmable logic devices, or application-specific integrated circuits.

The interaction between software and hardware may need also to be explored further (for example, the software failures induced by hardware failures and vice versa are not considered, the failures related to data losing or unavailability are included in the software failures).

10.4 Modelling of common cause failures

Should CCF be defined at the same level as individual failures? Preferably the CCF is defined at the same level of detail as the individual failures, because this improves the consistency of the model and prevents that CCFs are accounted for more than once. However, this may unnecessarily increase the number of events significantly that have the same effects in the model. Although software CCF might form an exception because it gives a ground for common cause failure of components which are otherwise may be considered as not been dependent based only on the hardware functional analysis. For example a platform software fault can affect the whole system and therefore will be defined at system level and not for example at basic component level. The subject of software induced common cause failures can be a subject for future works.

Also, it is possible that communication and synchronization can cause a CCF. Further research might also be necessary.

10.5 Failure detection coverage factor

Assessment of “coverage”, for example, the ability/probability of a watchdog timer to detect failures is needed. The ability to detect some failures and provide possibilities for coping with the failures is an important feature of digital systems. In general, each failure mode of a digital component/system has to be evaluated so that reliability models can be developed. That is, not every failure can be detected, and the detectability of a detectable failure mode may not be 100%. Suppliers may claim that a diagnostic coverage of 99,9% is easily achievable and is even conservative. This type of claims needs to be verified for specific failure modes that are considered detectable.

The assessment of the coverage factor is not fully covered by the developed taxonomy. This aspect may be taken into account while applying it to the real models.

10.6 Data collection

Failure modes taxonomy is also needed in the collection and statistical analysis of operating experience. As the collected data may be used to derive reliability data for the modelling and quantification of the digital I&C failures, the coherence between the modelling taxonomy and data collection taxonomy may be one of the subjects for future work.

11. CONCLUSION AND RECOMMENDATIONS

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 PRA 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 PRA, while control systems can be treated in a limited manner.

The objective of OECD/NEA DIGREL task was to develop a failure mode taxonomy for reliability assessment of digital I&C systems for use in PRA. The failure modes taxonomy has been developed to support modelling and quantification efforts. It will also help define a structure for data collection and to review PRA studies.

The proposed failure modes taxonomy has been 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 software part of I&C, failure modes defined in NPP PRAs have been simple – typically a software CCF failing identical processing units.

The DIGREL task group has defined a new failure modes taxonomy based on a failure propagation model and the hierarchical definition of five levels of abstraction:

1. system level (complete reactor protection system),
2. division level,
3. I&C unit level,
4. I&C unit module level
5. basic component level.

This structure corresponds to a typical reactor protection system architecture, which is the scope of the taxonomy. 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. Four important elements of the failure model stand out, on which the taxonomy focuses: fault location, failure mode, uncovering situation, failure effect and the 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 PRA, 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 PRA. There are a number of areas where further studies are needed, as discussed in Section 0, and many recommendations given in the previous expert report [NEA/CSNI/R(2009)18] are still valid. For instance, the following actions could be considered:

- Testing of the applicability of the taxonomy in modelling, including test with different I&C designs and modelling approaches.

- Testing of the applicability of the taxonomy in data collection. After the termination of the COMPSIS project, OECD/NEA International Common-cause Failure Data Exchange (ICDE) project has expressed a willingness to integrate computer failures as a new component type for data collection.

- Development of methods for software reliability quantification for nuclear PRAs. There are several past and ongoing R&D projects in this area. Benchmarking studies may be considered.

- Complementation of the failure modes taxonomy with issues that were left out of the scope, e.g., control systems, networks, PLD technology (FPGA/ASIC)

- Development of methods to architecture level assessment, including defence-in-depth and diversity assessments. It is essential to account for the needs of both deterministic and probabilistic safety assessments.

- Development of methods for the evaluation of fault tolerance features in the hardware and software of the safety important I&C systems

12. REFERENCES

AN8500-1	Reliability Failure Mode Effects and Predicted Failure Rate Analysis for the ACT8500 64-Channel Multiplexer Module, Application Note AN8500-1, Aeroflex, September 15, 2005.
ARIANE	ARIANE 5. Flight 501 Failure. Report by the Inquiry Board, Paris, 1996. http://esamultimedia.esa.int/docs/esa-x-1819eng.pdf
BNL-90571-2009-IR	Chu, T.L., Martinez-Guridi, G., Yue, M., Samanta, P., Vinod, G., and Lehner, J., Workshop on Philosophical Basis for Incorporating Software Failures into a Probabilistic Risk Assessment," Brookhaven National Laboratory, Technical Report, BNL-90571-2009-IR, November 2009.
BNL-NUREG-77124-2006-CP	Chu, T.L., Martinez-Guridi, G., Yue, M., Lehner, J. A review of software induced failure experience., 5th NPIC HMIT meeting, November 2006, BNL-NUREG-77124-2006-CP
DI&C-ISG-03	Review of New Reactor Digital Instrumentation and Control Probabilistic Risk Assessment, Interim Staff Guidance, DI&C-ISG-03, USNRC, August 11, 2008.
EPRI TR 1007997	Torok, R. Guideline for performing defence in depth and diversity assessments for digital I&C upgrades, EPRI TR 1007997, December 2003
EPRI TR 1021077	Estimating Failure Rates in Highly Reliable Digital Systems. EPRI TR 1021077, EPRI 2010. Limited distribution.
Eurotherm-2000	Eurotherm Ltd., Using 2604/2704 Fixed Digital I/O, Technical Information, No. TIN 137, pg. B 113, 2000.
IAEA- NP-T-3.12	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, Austria, 2011.
IAEA-NS-R-1	Safety of nuclear power plants: design, safety requirements, IAEA Safety Standards Series No. NS-R-1, International Atomic Energy Agency. Vienna, 2000.
IAEA-NS-G-1.3	Instrumentation and Control Systems Important to Safety in Nuclear Power Plants, IAEA Safety Standards Series No. NS-G-1.3, International Atomic Energy Agency. Vienna, 2002.
IAEA-SSG3	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, 2010.
ISO/IEC 25040	Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process, ISO/IEC 25040:2011.
IEC 60880	Software aspects for computer-base systems performing category A functions, IEC 60880, ed. 2.0. International Electrotechnical Commission, Geneva, 2006.
IEC 61226	Nuclear power plants — Instrumentation and control important to safety — Classification of instrumentation and control functions, IEC 61226, ed. 3.0 International Electrotechnical Commission, Geneva, 2009.
IEC 61508	Functional safety of electrical/electronic/programmable electronic safety-related systems. Part 6: Guidelines on the application of IEC 61508:2 and IEC 61508:3. International Electrotechnical Commission, Geneva, 2010.
IEC 61513	Nuclear power plants — Instrumentation and control important to safety — General requirements for systems, IEC 61513. International Electrotechnical

	Commission, Geneva, 2009.
IEC 62340	Nuclear power plants – Instrumentation and control systems important to safety – Requirements for coping with common cause failure (CCF), IEC 62340, ed. 1.0. International Electrotechnical Commission, Geneva, 2007.
IEC 60812	Analysis techniques for system reliability, Procedure for failure mode and effects analysis (FMEA), International standard IEC 60812:2006(E), Second edition, International Electrotechnical Commission (IEC), Geneva, 2006.
IEC 61709	Electric components - Reliability - Reference conditions for failure rates and stress models for conversion, IEC 61709, ed. 2.0. International Electrotechnical Commission, Geneva, 2011.
IEC 62380	Reliability Data Handbook — Universal model for reliability prediction of electronics components, PCBs and equipment, IEC/TR 62380, 1st edition, 2004.
IEEE-323-2003	IEEE Standard for Qualifying Class 1E Equipment for Nuclear Power Generating Stations," IEEE Std. 323-2003, Institute of Electrical and Electronics Engineers, 2003.
IJCAS-2006-Lee	Lee, D. Y., et al., A Safety Assessment Methodology for a Digital Reactor Protection System, International Journal of Control, Automation, and Systems, Vol, 4, no. 1, pp. 105-112, February 2006.
ISO/IEC 2382-1	Information technology--Vocabulary--Part 1: Fundamental terms, ISO/IEC 2382-1:1993, 01.01.08
ISO/IEC 25000	Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 4.2, ISO/IEC 25000:2005
ISO/IEC/IEEE 24765	Systems and software engineering – Vocabulary, ISO/IEC/IEEE 24765:2010
Korsah 2010	Korsah, K., Cetinerm, S. M., Muhlheim, M. D., and Poore III, W. P., "An Investigation of Digital Instrumentation and Control System Failure Modes," ORNL/TM-2010/32, March 2010.
Marcella-1994	Marcella, R., & Newton, R. A New Manual of Classification. Ashgate Publishing Company, 1994.
Meeldijk 1996	Meeldijk, V., Electronic Components Selection and Application Guidelines, John Wiley & Sons, 1996.
MIL-STD-1629A	Procedures for performing a failure mode, effects and criticality analysis, Military standard, MIL-STD-1692A, US Department of Defense, Washington D.C., 1980.
MIL-HDBK-217	Reliability Prediction of Electronic Equipment, Notice 2, MIL-HDBK-217F(2), US Department of Defense, Washington D.C., 1995.
NEA/CSNI/R(2009)18	Recommendations on assessing digital system reliability in probabilistic risk assessments of nuclear power plants, NEA/CSNI/R(2009)18, OECD/NEA/CSNI, Paris, 2009.
NEDC-30851p	Sullivan, W.P., et. al., "BWR Owners-Group Technical Specification Improvement Analysis for BWR Reactor Protection System," General Electric, NEDC-30851p, May 1985.
NKS-230	Authén, S, Björkman, K., Holmberg, J.-E., Larsson, J. Guidelines for reliability analysis of digital systems in PSA context — Phase 1 Status Report, NKS-230 Nordic nuclear safety research (NKS), Roskilde, 2010.
NKS-277	Authén, S., Holmberg, J.-E., Guidelines for reliability analysis of digital systems in PSA context, Phase 3 Status Report, NKS-277, NKS, Roskilde, 2013. .
NPIC&HMIT-2004-Chu	Chu, T.L., Martinez-Guridi, G., Lehner, J., and Overland, D., Issues Associated with Probabilistic Failure Modeling of Digital Systems, NPIC&HMIT 2004, Columbus, Ohio, September 2004.
NUREG/CR-6962	Traditional Probabilistic Risk Assessment Methods for Digital Systems. U.S. Nuclear Regulatory Commission, Washington, DC 20555, July 2008.
NUREG/CR-6985	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, Q., Dion, J., A Benchmark Implementation of Two Dynamic Methodologies for the Reliability Modeling of Digital Instrumentation and Control Systems, NUREG/CR-6985, February 2009.
NUREG/CR-6992	Instrumentation and Controls in Nuclear Power Plants: An Emerging Technologies Update. NUREG/CR-6992, United States Nuclear Regulatory Commission, 2009.
NUREG/CR-6997	Chu, T.L., Yue, M., Martinez-Guridi, G., Mernick, K., Lehner, J., Kuritzky, A., Modeling a Digital Feedwater Control System Using Traditional Probabilistic Risk Assessment Methods, NUREG/CR-6997, September 2009.
NUREG/CR-7007	Wood, R.T., Belles, R., Cetiner, M.S., Holcomb, D.E., Korsah, K., Loebl, A.S., Mays, G.T., Muhlheim, M.D., Mullens, J.A., Poore III, , W.P., Quails, A.L., Wilson, Jr., T.L., Waterman, M.E., Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems, NUREG/CR-7007, February 2010.
NUREG/IA-0254	International Agreement Report Suitability of Fault Modes and Effects Analysis for Regulatory Assurance of Complex Logic in Digital Instrumentation and Control Systems, NUREG/IA-0254
NUREG-0800	Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition, U.S. Nuclear Regulatory Commission.
ORNL/NRC/LTR-07/05	Kisner, R., Mullens, J., Wilson, T., Wood, R., Korsah, K., Qualls, A., Muhlheim, M., Holcomb, D., Loebl, A.. 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, Prepared for the U.S. Nuclear Regulatory Commission, August 2007.
ORNL/TM-2010/32	Korsah, K., Cetiner, S. M., Muhlheim, M. D., Poore III, W. P., An Investigation of Digital Instrumentation and Control System Failure Modes, ORNL/TM-2010/32, 2010
PSAM10-Authen	Authen, S., Wallgren, E., Eriksson, S., Development of the Ringhals 1 PSA with regard to the Implementation of Digital Reactor Protection System, PSAM 10, Seattle, Washington, June 6-11, 2010.
PSA2013 Westinghouse	Davis, S.A., Detar, H.L., Masset, Y., Lessons learned from the digital I&C system modeling of the AP1000 plant PRA. ANS PSA 2013 International Topical Meeting on Probabilistic Safety Assessment and Analysis, Columbia, SC, September 22-26, 2013, on CD-ROM, American Nuclear Society, LaGrange Park, IL, 2013, paper 61.
RAC-1997	Reliability Analysis Center, Failure Mode/Mechanism Distributions, A Department of Defense Information Analysis Center, FMD-97, December 1997.
Rausand-Høyland	Rausand, M., Høyland, A., System Reliability Theorie, Models, Statistical Methods, and Applications, Wiley, 2004.
RDF 83	Recueil de Données de Fiabilité du CNET. Composants Electroniques. Edition 1983.
RESS-1999-Rouvroye	Rouvroye, J.L., and Brombacher, A.C., New Quantitative Safety Standards: Different Techniques, Different Results, Reliability Engineering and System Safety, 66 (1999) 121-125.
RG-1.200	An Approach For Determining The Technical Adequacy Of Probabilistic Risk Assessment Results For Risk-Informed Activities, Regulatory Guide 1.200, USNRC, March 2009.
RiskAnalysis-2006-Li	Li B., Li M., Chen K., Smidts C., Integrating Software into PRA: A Software-Related Failure Mode Taxonomy, Risk Analysis, Vol. 26, No. 4, 2006

SINTEF-2010	Reliability Prediction Method for Safety Instrumented Systems – PDS Example Collection, 2010 Edition. SINTEF, 2010
SN 29500	Siemens Standard SN 29500-1: Ausfallraten Bauelemente; Teil 1 Allgemeines, Erwartungswerte - Failure rates of electronic components, part 1: General, expected values.
UTE C 80 810	Reliability data handbook – Universal model for reliability prediction of electronics components, PCBs and equipment. August 2005.
UTE C 80 811	Reliability Methodology for Electronic Systems, FIDES Guide 2004 issue A
WCAP-10271	Andre, G.R., et. Al., “Evaluation of Surveillance Frequencies and Out of Service Times for the Engineered Safety Feature Actuation System,” Westinghouse Nuclear Energy Systems, WCAP-10271, February 1986.

APPENDIX A. DETAILED TAXONOMIES

A.1 Introduction into the collected taxonomies

The basis for developing the failure mode taxonomy of digital systems as presented in section 7 is the collection of taxonomies as provided by the respondents. Twelve organizations provided input:

- BNL (Brookhaven National Laboratory);
- CNSC (Canadian Nuclear Safety Commission);
- EDF (Electricity of France);
- IRSN (Institut de Radioprotection et de Surete Nucleaire);
- JNES (Japan Nuclear Energy Safety Organization);
- KAERI (Korean Atomic Energy Research Institute);
- NRG (Nuclear Research and Consultancy Group);
- NKS (Nordic nuclear safety research) summarising input from three Nordic utilities;
- OSU (Ohio State University);
- RELKO Ltd (Engineering and Consulting Services).

Each of the respondents provided a list of failure modes and a definition of the level of abstraction at which they performed the analysis. The respondents were asked to include the software failure modes. Also a report from the Oak Ridge National Laboratories has been used as input (Korsah 2010). Together the respondents provided 27 failure mode taxonomies on hardware failure modes and 18 failure mode taxonomies on software failure modes.

In this appendix the taxonomies provided by the respondents will be summarized. The detailed taxonomies can be found in Appendix A.4. Hardware and software failure modes will be discussed separately. Not all respondents provided a separate failure mode taxonomy for software. Evidently, information provided is heterogeneous and as such are examples of different approaches and applications. At the same time it provides insights in current practice and the need for development.

It is noted that the contributions of the respondents were used by the task group as input for the discussions on definitions, failure modes and underlying concepts, the terms used in this section may therefore deviate from the definitions in section 3. It is stressed that the appendix reflects the definitions and failure modes as provided by the respondents.

The FMEAs of the respondents considered different kinds of automation functions:

1. Protection functions;

2. Functions with complex output;

3. Functions with sequential output.

The type of functions that is analysed, has inevitable consequences on the failure modes that need to be considered and therefore the failure modes that are defined. For example a protection function with binary output can fail either high or low, in contrast to a sequential output which can also fail by providing a faulty sequence.

Looking at the input of the respondents, they had also different ways of defining the failure modes, which are in this section grouped into two categories:

4. **Functional failure mode**: A failure mode regarding the effect on the function that is considered. For example, failure to actuate or spurious failure.

5. **Structural failure mode**: A failure mode that includes the failure cause. For example, “frozen sensor” or “amplifier adjustment too low”. These failure modes do not describe the effect on the functional level, as this might be dependent on the architecture that is considered.

The latter approach needs to be evaluated in order to determine their failure effects. The functional failure modes do give information about the effect, but not about the causes. To obtain a clear picture of all inputs of the respondents, the summary of the summary of the failure mode taxonomies will include both the functional and the structural failure modes.

A.2 Collected hardware failure modes

In order to meaningful compare the failure mode taxonomies the inputs have been classified according to their level of abstraction. Based on the definitions of the respondents (see figure section 3):

In Table A-1, the contributions of the different respondents are summarized. For every level of abstraction, the components that are considered are listed (in Column 2) as well as the failure modes at this level of abstraction (in Column 3).

Table A-1. Summary of the modules/components considered at every level of abstraction and the corresponding failure modes.

Level	Modules/components	Failure modes / effect / causes
System level	Entire system (RPS)	• Failure to actuate • Failure to actuate in time • Spurious actuation Structural failure modes: • Failure of support system, e.g. loss of power supply; • Failure of acquisition • Failure of treatment and communication
Division level	Single channel of RPS	• Undetected failures o Loss of function o Spurious function • Detected failures o Loss of function • CCF[6] • Unavailable due to corrective maintenance[7]

[6] Although not defined by the participant, CCF can be both detected and undetected.

Level	Modules/components	Failure modes / effect / causes
I&C unit level	• Acquisition and processing unit (APU) • Logic processing module • Signal conditioning module • Actuation logic unit (ALU) • Voting processing module • Hardwired output logic for actuation • Digital trip module • Trip logic unit • Safety logic unit	• Undetected failures ○ Loss of function ○ Spurious function • Detected failures ○ Loss of function • CCF[6] • Unavailable due to corrective maintenance[7]
Module level	• Remote multiplexing unit • Input/output devices ○ Digital I/O module ○ Digital I/O channel ○ Analog I/O module ○ Analog I/O channel • Load driver • Optical cable • PLC module • Communication card • Termination module • DC power supply • Power battery • AC power supply • Subrack	• Undetected failures ○ Loss of function ○ Spurious function ○ Malfunction[8] • Detected failures ○ Loss of function ○ Malfunction • CCF • Unavailable due to corrective maintenance Structural failure modes: • Loss of one sensor input • Intermittent sensor signal failure • Loss of an output • Loss of internal power supply • Internal power overshoot • Round-off/truncation/sampling rate errors • Unable to meet response requirements • Skipping software functions due to hardware/software faults or too fast triggered WDT • WDT fails to activate • WDT activates when computer has not failed • Arbitrary value output • Setpoint corrupted • Malfunction alarm of the PLC module of station blackout diesel generator system • Termination module D/I fails to close/open when energized/de-energized. • Card failure detected/undetected by software. • Card failure detected on panel check
Basic component level	• Current loop • A/D converter and D/A converter • Multiplexer • Demultiplexer • Software • Sensors • Signal amplifier • Transmitter	Failure modes defined for individual component according to their output. Software running on a microprocessor is modelled as a basic component. Structural failure modes: • Network interface Card fails to establish communication • Transmitter fails/drifts high/low • Amplifier output fails low • Amplifier output fails low due to CCF • Amplifier adjustment too low

[7] Unavailability due to corrective maintenance is a term that is more likely to be used in a PRA unavailability model than as a failure mode in a FMEA.

[8] Malfunction might be captured by loss of function or spurious function, depending on the situation.

Level	Modules/components	Failure modes / effect / causes
		• Power supply output fails low • Sensor signal fails low • Transducer spuriously fails high • Termination module A/I fails/drifts high/low

The defined functional failure modes are almost identical at every level of abstraction except the basic component level. Most respondents defining functional failure modes, categorized the failure modes based on detectability and effect, so that nine functional failure modes can be distinguished:

1. Undetected failures
   a. Loss of function
   b. Malfunction
   c. Spurious function
2. Detected failures
   a. Loss of function
   b. Malfunction
   c. Spurious function
3. Failure to function in time
4. CCF
5. Unavailable due to corrective maintenance.

Compared to the detection mechanisms defined in chapter 3, in the collected taxonomies the terms "detected" and "undetected" failures have been used, having the following interpretations:

6. **Detected failure**: a failure detected by (quasi-) continuous means or by plant behaviour, e.g. self-monitoring, or by plant behaviour through indications or alarms in the control room.

- **Undetected failure**: A failure that can only be detected by periodic surveillance testing or by demand. Also called latent failure or hidden failure.

Although considered one of the most important contributors of digital systems, CCFs are not mentioned separately by every respondents. This is due to the fact that CCF is generally not considered as a failure mode, but as a combination of failure modes due to a common cause. In this way CCF does not get too much attention in the taxonomies, but is an important factor in modelling.

Both loss of function and spurious function can be an effect of several failure modes of the components (e.g. modules, basic components). The signals of a component can fail high, fail low, give erroneous outputs or get stuck (detailed failure mode/module or basic component level). Depending on the design and the plant condition, these failure modes will result in either a spurious function or a loss of function (functional failure mode / system level). The functional impact of the structural failure modes is dependent on the design of the system. Although in a FMEA taxonomy the different type of failure modes can be used, in the PRA model the cause, the failure mode and the failure effect should be interpreted adequately. Therefore it is important to interpret a structural failure mode from the taxonomy to model the failure effects of the particular equipment or functions, i.e., to determine the behaviours of the system or the functional unit.

A.3 Collected software failure modes

The levels of abstraction for software as defined by the respondents deviate from the levels of abstraction as defined for hardware. Compared to the hardware levels of abstraction the I&C unit level is missing for the software failure modes. Which is not surprising, given the fact that also for the hardware levels of abstraction the distinction between I&C unit level and module level lead to discussion. However, the input of the responders on software did not lead to a separate level of abstraction for an I&C unit level. Which leads to the discussion if it is necessary to map software failures to hardware components and if so, how this is done correctly.

In Table A-2, the software failure modes at different levels of abstraction are listed. The detailed taxonomies can be found in Appendix A.4. Some respondents indicated different levels of CCF due to software failure:

- Loss of the complete system;
- Loss of (multiple) division(s);
- Loss of one or more specific modules.

In Table 3 the summary of the failure mode taxonomies on failure modes as provided by the respondents is shown.

Table A-2: Summary of the software failure modes defined by the participants

Level	**Failure modes**
System level[9] / Division level[10]	For reactor trip • Failure to actuate (including failure to hold) • Spurious failure • Adverse effects on other functions (systems, operators) • (and others, dependent upon additional functions judged to be safety related) For diesel generator load sequencing: • Failure to actuate in time For ESFAS: • Failure of trip signals such as a high reactor pressure level; The above failure modes can also be subdivided the location where they occur: Function failures, attribute failures, function set failures, input/output failure modes, multiple interaction failure modes, support failure modes. For systems with multiple functions there are also a few additional failure modes: single function failure, correlated multiple function failures, complex multiple function failures.
Microprocessor	Erroneous operation for data acquisition:

[9] Failure to actuate, spurious failure and failure to actuate in time might be more failure mode effect. At microprocessor level, the failure modes are more detailed, but the effect of the failure mode is context dependent. So also the failure modes defined at microprocessor level or at sub-level, might lead to a failure to actuate or a spurious failure.

[10] Generally failure modes defined for the system level can also be defined for the channel level

Level	**Failure modes**
level / module level	• Incorrect value • Incorrect validity • Incorrect value and incorrect validity • No value • No validity • Above failure modes may be subdivided, e.g. incorrect high or low Erroneous operation for logic processing: • Failure to actuate (including failure to hold) • Spurious failure Erroneous operation for voting logic: • Incorrect voting • No vote • Above failure modes can lead to a failure to actuate (including a failure to hold) or to a spurious failure Erroneous operation for priority actuation logic: • Incorrect priority • No priority • Above failure modes can lead to a failure to actuate (including a failure to hold) or to a spurious failure Other: • Software runs with misleading commands to the user, incomplete or incorrect display of information; • Software stalls; • Missing operation; • Extra operation; • Software aborts;
Sub-level	Failure modes are defined for software functional modules related to individual signals to hardware components such as pumps and valves. • Timing/order failure • Interrupt induced failure • Omission of a function or an attribute • Unintended function or attribute • Incorrect implementation of a function or an attribute • Data error

One respondent also included the sources from which software common cause failure could result:

- Faulty specifications for the application software;

- Faulty code generation for the application software;

- Faulty identical software components in the firmware (operating system, driver or compiler, etc.).

A.4 Detailed collection of taxonomies

Organization	**System/Division Level**	**I&C unit level**	**Module level**	**Basic component level**
	Definition	**Definition**	**Definition**	**Definition**
	Failure modes	**Failure modes**	**Failure modes**	**Failure modes**
NRG	The entire RPS system	I&C units include: 1) data acquisition from sensors; 2) network sensor; 3) Acquisition processing unit (APU); 4) Actuation logic unit (ALU); 5) network APU-ALU 6) hard-wired output logic for actuation; 7) actuator; 8) DC-power supply; 9) power battery; 10) AC-power.	Each APU has a main computer and a backup computer; each with a external watchdog timer. Main computer is used as an example for failure mode definition.	
	1. Output of 1 instead of 0; 2. Output of 0 instead of 1.		1) Loss of one sensor input; 2) intermittent sensor signal failure; 3) loss of an output; 4) loss of internal power supply; 5) internal power overshoot; 6) round-off/ truncation/ sampling rate errors; 7) unable to meet response requirements; 8) skipping software functions due to hw/sw fault or too fast triggered WDT; 9) WDT fails to activate; 10) WDT activates when computer has not failed; 11) arbitrary value output; 12) setpoint corrupted.	
EDF R&D	The entire system			

Organization	**System/Division Level**	**I&C unit level**	**Module level**	**Basic component level**
	Definition	**Definition**	**Definition**	**Definition**
	Failure modes	**Failure modes**	**Failure modes**	**Failure modes**
	For RPS: 1) Output of 1 instead of 0; 2) Output of 0 instead of 1; 3) No output in the required time frame. For system with redundant and/or diverse: Full failures and partial failures For system with multi-functions: 1. Single function failures; 2. correlated multiple function failures; 3. complex multiple function failures, or Preferred failure modes/dreaded failure modes/other failure modes, or Innocuous/mildly dangerous/dangerous failure modes.			
JNES		digital trip module (DTM), trip logic unit (TLU), safety logic unit (SLU), remote multiplexing unit panel (RMU), output logic unit (OLU), input/output devices (PI/O), load driver (LD), and optical cable (OC).		
		1. Loss of function; 2. malfunction; 3. No actuation.		

Organization	System/Division Level	I&C unit level	Module level	Basic component level
	Definition	**Definition**	**Definition**	**Definition**
	Failure modes	**Failure modes**	**Failure modes**	**Failure modes**
BNL	Entire digital feedwater control system	CPU modules and controller modules		Current loop, a/d and d/a converters, multiplexer/demultiplexer, software etc
	Loss of automatic control	Intermediate level of failure modes are not of interest although example failure modes can be undetected failures or detectable failures.		Failure modes defined for individual components according to their output. Software running on a microprocessor is modeled as a basic component.
Ringhals 1 PSA	Digital Reactor Protection System		Sub-components of computer units: Processor, Communication Module, Digital I/O Module, Digital I/O Channel, Analog I/O Module, Analog I/O Channel, Signal Conditioning Module, Subrack	
	Loss of RPS/ESFAS actuation and Spurious RPS/ESFAS actuation depending on: - Detected, undetected or spurious sub-component failure - Appliance of default values at detected failures - Effects of detected failures due to type of voting logic - Fail-safe actions applied to output channels at detected		Failure modes for each sub-component: 1. Loss of function due to detected failure, 2. Loss of function due to undetected failure, 3. Spurious function, 4. CCF 5. Corrective maintenance R1 PSA describes in detail the characteristics of the fail-safe design, e.g.: - Fail-safe design only covers detected failures - Undetected failures will	

Organization	System/Division Level	I&C unit level	Module level	Basic component level
	Definition	**Definition**	**Definition**	**Definition**
	Failure modes	**Failure modes**	**Failure modes**	**Failure modes**
	failures		challenge the RPS sequences - Detected failures might cause “spurious” actuations	
Ringhals 2 PSA	Entire RPS/ESFAS.	Mainly super component approach : Processor, communication module and misc. modules. Processor super component contains input, output, processing, subrack etc. In a few cases I/O modules and watchdog is modeled.		
	Failure modes are: - “no activation signal”	Failure modes: 1) Loss of function due to undetected failure, 2) CCF, 3) Corrective maintenance		
Olkiluoto 1/2 PRA	There are a few safety-related automation systems based on digital technology, which are accounted in the PSA: - the turbine automation, - the main circulation pump control system. In addition, there are programmable logic components in some systems included in PSA, e.g. the neutron flux monitoring system.	Mainly supercomponent approach.[11] Processor super component contains input, output, processing, subrack etc. Subcomponents (modules) considered in FMEA		
	Failure modes: - “failed safety			

[11] A safety-related system may have multiple channels.

Organization	**System/Division Level**	**I&C unit level**	**Module level**	**Basic component level**
	Definition	**Definition**	**Definition**	**Definition**
	Failure modes	**Failure modes**	**Failure modes**	**Failure modes**
	function”			
Loviisa 1/2 design phase PRA for the automation renewal project	Short-term accident management, which is considered in the PSA, consists of five different task categories: - Normal process control NPC (SPPA-T2000, OM690, FUM) - Preventive protection PREV (TXS, QDS, AV42) - Reactor protection RPS (TXS, QDS, AV42) - Manual Backup of Reactor protection RPSMBU (non-programmable TXS, hard-wired) - Automatic Backup of Reactor protection ABU (SPPA-T2000, OM690, AV42)	Mainly supercomponent approach. Subrack and priority units modeled, too.		
	Both “loss of system functions” and “spurious system actuation” are considered in PSA.			

Organization	System/Division Level	I&C unit level	Module level	Basic component level
	Definition	Definition	Definition	Definition
	Failure modes	Failure modes	Failure modes	Failure modes
Oak Ridge report ORNL/TM-2010/32 [12]	A collection of equipment that is configured and operated to serve some specific plant function, as defined by terminology of each utility (e.g., auxiliary feedwater system, containment spray system).	A subsystem, which is a collection of multiple components, that performs specific tasks or functions that are essential for a system in rendering its intended services.		A system is composed of a set of components bound together in order to interact, where each component is another system. This recursion stops when a component is considered atomic, i.e., any further internal structure cannot be discerned, or is not of interest and can be ignored. Consequently, the total state of a system is the set of the (external) states of its atomic components.
	System failure modes were not explicitly defined in Appendix A of ORNL/TM-2010/32. However, failure effects at system level were provided, e.g., No. 27, blackout diesel system unavailable.	Module level failure modes were generally not explicitly defined in Appendix A of ORNL/TM-2010/32. However, failure effects at module level were provided, e.g., No. 27, Malfunction alarm of the PLC module of blackout diesel system.	Failure mode of No. 27, blackout diesel system. PLC module, communication card is communication dropout.	Failure mode of No. 67, Fuel and Reloads System: Load Weighting System, PLC, Network Interface Card is failure to establish communication.
Canadian Nuclear Safety Commission (CSNC)	The entire SDS1 system Each channel includes data acquisition, termination modules, and trip computers		Termination module	Sensors or signal amplifier and transmitters.
	Failure to shutdown reactor via SDS1 after react. increase. Failure to open trip digital output on trip parameter.		1. Termination module D/I fails to close/open when energized/de-energized. 2. Card failure detected/undetected by software check/test.	1. Transmitter fails/drifts high/low. 2. Amplifier output fails low. 3. Amplifier output fails low due to CCF. 4. Amplifier adjustment too low.

[12] The table in Appendix A of this report contains detailed information about level-of-detail associated failure modes. Table 6 of the report intends to provide common failure modes, however, it completely removes the level of detail information, and is therefore not adopted in this summary. Instead, example failure modes at different levels of detail are extracted from the table in Appendix A of report ORNL/TM-2010/32 and shown here.

Organization	**System/Division Level**	**I&C unit level**	**Module level**	**Basic component level**
	Definition	**Definition**	**Definition**	**Definition**
	Failure modes	**Failure modes**	**Failure modes**	**Failure modes**
			3. Card failure detected on panel check.	5. Power supply output fails low. 6. Sensor signal fails low. 7. Transducer spuriously fails high. 8. Termination module A/I fails/drifts high/low.
KAERI	Sensors + Entire RPS system + trip circuit breakers + Manual actuation by human operators A single channel of RPS		Analog input module, digital input module, logic processing module, voting processing module, digital output module, etc.	
	Failure to actuate on demand		Failure to actuate on demand	
IRSN	The entire system (Each main specific automate is modelled, including the networks and the communication) Each division and sub-division are modelled separately but with potential CCF			Only the acquisition (sensors) is modelled at the level of basic components, including CCF between the identical sensors
	- Global failure of the automate -Failure of support system (ventilation, electrical support) - Failure of acquisition - Failure of treatment - Failure of communication The software related failures are considered as			-No output -Fail to detect change

Organization	**System/Division Level**	**I&C unit level**	**Module level**	**Basic component level**
	Definition	**Definition**	**Definition**	**Definition**
	Failure modes	**Failure modes**	**Failure modes**	**Failure modes**
	CCF between different I&C systems or subsystems			
RELKO (presented in the DIGREL workshop May 2011)	Failure of reactor trip function, Failure of safeguard actuation functions. Spurious actuation of RPS or ESFAS can have influence on frequency of initiation events		Analog input module, digital input module: - detected failure of all channels - undetected failure of one channel - detected failure of one channel - blocking of the back panel bus - monitoring module (no failure effects considered) Processing module: - detected function failure - undetected function failure - blocking of the back panel bus - monitoring module (no failure effects considered) Signal acquisition module: - detected failure (1 of 2 channels) - loss of overvoltage protection Relay module: - detected failure - undetected failure Converter 220/24 V: - detected failure Level transmitter: - detected failure - undetected failure (signal frozen)	

Table II: Software failure modes

Organization	System/Division level[13]	Microprocessor level[14]	Sub-level
OSU	1. Software-related failures[15]		Sub-levels means whether the failure occurs within the boundary of the software (internal failures, i.e., Level 2) or at the interface between the software and the outside world (interaction failures, i.e., Level 3). In a summary, all levels of detail other than system level, channel level, and microprocessor level are called sub-level in this study.
			Under Level 2[16]: 1. Function failures; 2. Attribute failures; 3. Function set failures. Under Level 3: 1. Input/output failure modes; 2. Multiple interaction failure modes; 3. Support failure modes.
EDF R&D	For RPS: 1. Output of 1 instead of 0; 2. Output of 0 instead of 1; 3. No output in the required time frame. For system with redundant and/or diverse: Full failures and partial failures For system with multi-functions: Single function failures; correlated multiple function failures; complex multiple function failures. Or 1. preferred failure modes/dreaded failure modes/other failure modes Or innocuous/mildly dangerous/dangerous failure modes.		
BNL		1. Software stalls; 2. Software runs as usually but with wrong	Sub-level means the level of software elements, which include input, output,

[13] System level failure modes indicate the failure modes of the collection of all software of a digital system.

[14] Microprocessor level failure modes indicate the failure modes of a software running on a microprocessor of a digital system.

[15] This could also be microprocessor level failure mode and needs to be clarified.

[16] These definitions give the place where the failure modes occur, but do not describe the failure modes itself?

Organization	System/Division level[13]	Microprocessor level[14]	Sub-level
		outputs; 3. Software runs with misleading commands to the user, incomplete or incorrect display of information.	communication, resource allocation, and processing elements.
			1. Timing/order failure; 2. Interrupt induced failure; 3. Omission of a function or an attribute; 4. Unintended function or attribute; 5. Incorrect implementation of a function or an attribute; 6. Data error.
Software Reliability Workshop at BNL	1. Failure to generate signal in time (omission failure); 2. Spurious signal; and 3. Adverse effects on other functions (systems, operators).	1. Hang; 2. Abort; 3. Missing operation; 4. Extra operation; and 5. Erroneous operation.	
Ringhals 1 PSA [NKS-230]		Presently software failures are described only by CCF events considering failures within redundant TXS units.	
Ringhals 2 PSA [NKS-230]		The impact of SW CCF is judged to be small and only included for use in sensitivity analyses	
Olkiluoto 1/2 PSA [NKS-230]		Software CCF (application software)	
Loviisa 1/2 PRA [NKS-230]		Software CCF (application software)	
Oak Ridge report ORNL/TM-2010/32 [17]		See BNL study above.	See BNL study above.
KAERI	N/A	Logic processing module and voting processing module.	N/A
		Failure to actuate on demand of logic processing module and voting processing module.	N/A

[17] In addition to generic software failure modes adopted from the BNL software FMEA study, some example failure modes were available in the table of Appendix A of ORNL/TM-2010/32. These example failure modes are generally at the microprocessor level. See Appendix A of ORNL/TM-2010/32 for details.

Organization	System/Division level[13]	Microprocessor level[14]	Sub-level
IRSN	The failure of the software is considered to be a potential source of CCF of the I&C systems and subsystems. The considered failure modes are the same as the independent failure modes(spurious actuation, failure of treatment or acquisition, ...)		
DC Workshop Software Group[18]	For RPS: Failure to actuate (including failure to hold); Spurious failure; Possible others dependent upon additional functions judged to be safety related.[19] For load sequencing: Failure to activate in time. For an ESFAS, failure of trip signals	For Data Acquisition[20]: Incorrect value, incorrect validity, both, no value, no validity (may be subdivided, e.g., incorrect low or high). For Logic Processing: Failure to actuate (including failure to hold), spurious failure.	Software functional modules related to individual signals to hardware components such as pumps and valves.

[18] At the workshop, the levels for defining software failure modes include (1) RPS function level, (2) trip signal level (e.g., high reactor pressure level), (3) individual signal level (e.g., individual ESFAS signals to start a pump or open a valve), and (4) data acquisition, logic processing, voting, and priority actuation logic. In this table, the failure modes at trip signal levels are considered system level failure modes of an ESFAS. The signals to individual pumps and valves are assigned sub-level assuming that the associated failure modes are those of individual software modules that generate the signals..

[19] The failure modes at this level of detail may also be categorized as channel level failure modes.

[20] It may be worth considering failure modes for communication logic (which is considered a part of data acquisition here) separately, considering its importance.

Organization	System/Division level[13]	Microprocessor level[14]	Sub-level
	such as a high reactor pressure level.	For Voting Logic: Incorrect voting, no vote (will lead to failure to actuate [including failure to hold] and spurious failure). For Priority Actuation Logic: Incorrect priority, no priority (will lead to failure to actuate [including failure to hold] and spurious failure). Other Systemic Failures, such as failures of watchdog timers.	Failure of individual signals from an ESFAS to actuate pumps and valves.
RELKO (presented in the DIGREL workshop May 2011)		The independent software failure probability is applied for the reliability model "undetected failure of software installed in a single computer" Common cause failures in the software can result from: - Faulty specifications for the application software, - Faulty code generation for the application software or - Faulty identical software components in the firmware (operating system, driver or compiler, etc.) The applied CCFs lead to failure of all corresponding computers in the same redundancy or diversity.	

APPENDIX B. CONTRIBUTORS TO DRAFTING AND REVIEWING

Amri, A.	OECD/NEA Nuclear Energy Agency
Authén, S	Risk Pilot AB, Sweden
Betancourt, L.	United States Nuclear Regulatory Commission, USA
Björkman, K.	VTT, Finland
Blundell, N.	OECD/NEA Nuclear Energy Agency
Brinkman, H.	Nuclear Research and consultancy Group, the Netherlands
Bruneliere, H.	AREVA, France
Chirila, M.	Canadian Nuclear Safety Commission, Canada
Chu, L.	Brookhaven National Laboratory, USA
Coyne, K.	United States Nuclear Regulatory Commission, USA
Delache, J.	Institut de Radioprotection et de Sûreté Nucléaire, IRSN, France
Deleuze, G.	EDF, France
Georgescu, G	Institut de Radioprotection et de Sûreté Nucléaire, France
Gheorge, R.	Canadian Nuclear Safety Commission, Canada
Halverson, D.	United States Nuclear Regulatory Commission, USA
Holmberg, J.-E.	Risk Pilot AB, Finland
Kim M.C.	Chung-Ang University, Korea
Kondo, K.	Nuclear Regulation Authority, Japan
Kuritzky, A.	United States Nuclear Regulatory Commission, USA
Li, M.	United States Nuclear Regulatory Commission, USA
Mancini, F.	ENEL Ingegneria e Innovazione S.p.A. , Italy
Piljugin, E.	Gesellschaft für Anlagen- und Reaktorsicherheit, Germany
Postma, W.	Nuclear Research and consultancy Group, the Netherlands
Quatrain, R.	EDF, France
Sedlak, J.	ÚJV Řež, Czech Republic
Smidts, C.	Ohio State University, USA
Sopira, V.	RELKO, Slovakia
Stiller, J.	Gesellschaft für Anlagen- und Reaktorsicherheit, Germany
Taylor, G.	United States Nuclear Regulatory Commission, USA
Thuy, N.	EDF, France
Yue, M.	Brookhaven National Laboratory, USA