WGRISK Task DIGREL
"Development of best practice guidelines on failure modes taxonomy for reliability assessment of digital I&C systems for PSA"

DIGREL Seminar

Drawn up by: Jan-Erik Holmberg, VTT, November 10, 2011
Date: October 25, 2011
Place: VTT Digitalo, Espoo, Finland, meeting room AP107

Participants:

Name	Organisation	Country
Janne Valkonen	VTT	Finland
Ilkka Niemelä	STUK	Finland
Jan-Erik Holmberg	VTT	Finland
Stefan Authén	Risk Pilot	Sweden
Martti Välisuo	Fortum	Finland
Jiri Sedlak	NRI	Czech
Emil Ohlson	FKA	Sweden
Jan Stiller	GRS	Germany
Ewgenij Piljugin	GRS	Germany
Man Cheol Kim	KAERI	Korea
Nguyen Thuy	EDF	France
Johan Mangs	VTT	Finland
Atte Helminen	TVO	Finland
Kim Björkman	VTT	Finland
Keisuke Kondo	JNES	Japan
Wietske Postma	NRG	The Netherlands
Kalle Jänkälä	Fortum	Finland
Carol Smidts	Ohio State University	USA
Louis Chu	BNL	USA
Teemu Mätäsniemi	VTT	Finland
Tuomas Launiainen	Aalto university	Finland
Jan Tomas Bergström	Scandpower	Sweden

1 Opening of the seminar

Jan-Erik Holmberg, VTT, opened the seminar and welcomed the participants. Participants presented themselves.

The agenda of the seminar was accepted (Att. 1)

Jan-Erik Holmberg presented an overview of VTT and its nuclear safety related R&D, see Att. 2.

Kaisa Simola, VTT, gave an overview the Finnish nuclear safety research programme, SAFIR2014, see Att. 3.

2 STUK (Radiation and Nuclear Safety Authority in Finland) perspective

Ilkka Niemelä, STUK, presented the Finnish regulator's perspective on risk analysis and digital automation (Att. 4). Differences of demonstrative and descriptive analyses were discussed. In STUK's presentation, failure tolerance analysis is a demonstrative analysis, while FMEA and PRA are descriptive analyses. A criterion needs to be defined if the analysis is used for decision making. STUK's experience is that more emphasis should be put on the early phase of the system lifecycle when the requirements are defined.

3 Presentation of the WGRISK DIGREL task

Jan-Erik Holmberg presented the objectives and scope of the WGRISK DIGREL task (Att. 5).Objectives and scope of the WGRISK DIGREL task

4 Definitions

Man Cheol Kim, KAERI, gave an introduction to terms and definitions related to DIGREL task. He made a comparison of terms used in IEEE and IEC standards (Att. 6).

5 Survey of failure modes

Tsong-Lun Chu, BNL, presented a summary of taxonomy inputs from the Task group member organisations (Att. 7). Eleven organisations provided input. The results are presented in two tables: one for hardware failure and other one for software failure modes.

6 Hardware failure modes taxonomy

Stefan Authén, Risk Pilot, presented the status of WGRISK taxonomy development on hardware failure modes and a proposal for further work (Att. 8). List of criteria for taxonomy definition and choosing the level of details were presented. "Module level" was regarded as the most appropriate level.

7 Software failure modes taxonomy

N. Thuy, EDF, presented an approach to define and use software fault taxonomy in PSA (Att. 9). In top level, faults can classified into application and I&C platform related faults and on the other hand into specification, software or HW&SW interaction faults. With regard to category A I&C systems, some faults can be excluded.

8 Example DIC system

Ewgenij Piljugin, GRS, presented an example of a generic digital I&C system (Att. 10). The system will be decomposed into generic hardware structure, generic software structure and generic interfaces between them. Failure modes taxonomy will be tested with the example.

9 Contents of the guidelines

Carol Smidts, Ohio State University, presented the list of contents of the guidelines (Att. 11).

10 Discussion, conclusions of the seminar

A round table discussion was carried out. Jan-Erik Holmberg asked what the most critical (high priority) issues to be resolved by the DIGREL task are.

Difference in the viewpoints between PSA and I&C experts was recognised to be a possible obstacle. It is important to have common definitions.

A big question is where to get failure data. CCF should be addressed, too.

Attachments

1. Agenda
2. VTT, VTT Nuclear, VTT System research Overview
3. National Nuclear Power Plant Safety Research Programme 2011-2014, SAFIR2014 overview
4. NKS/DIGREL. STUK perspective. Digital Automation & Risk Analysis
5. WGRISK DIGREL task. Objectives and scope
6. Terms and Definitions for Reliability Assessment of Digital I&C Systems
7. Survey of Failure Modes
8. Hardware Failure Modes Taxonomy
9. Taxonomy for Software Faults
10. Example of a generic digital I&C System
11. Outline of the Guidelines

DISTRIBUTION Seminar participants, WGRISK DIGREL Task Group members, NKS/Karoliina Ekström

Attachment 1. Agenda

Tuesday October 25, 2011 **NKS/DIGREL seminar**		
08:30	Coffee, registration	
09:00	Opening of the seminar - Participants round table presentation - VTT short overview - SAFIR2014 Finnish nuclear safety research programme - Meeting logistics	Jan-Erik Holmberg, VTT Everybody Jan-Erik Holmberg, VTT Kaisa Simola, VTT Jan-Erik Holmberg, VTT
09:40	STUK (Radiation and Nuclear Safety Authority in Finland) perspective	Ilkka Niemelä, STUK
10:00	Presentation of the WGRISK DIGREL task - Objectives and scope - Definitions	 Jan-Erik Holmberg, VTT Man Cheol, KAERI
10:40	Break	
11:00	- Survey of failure modes - Hardware failure modes taxonomy	Louis Chu, BNL Stefan Authén, Risk Pilot
12:00	Lunch	
13:30	- Software failure modes taxonomy - Example DIC system - Contents of the guidelines	Nguyen Thuy, EDF Ewgenij Piljugin, GRS Carol Smidts, OSU
15:00	Break	
15:15	Discussion, conclusions of the seminar	
16:15	Seminar participants: Adjourn Workshop participants: Planning of the workshop	
17:00	Workshop participants: Adjourn	

Business from technology

VTT, VTT Nuclear, VTT System research Overview

NKS/DIGREL seminar, October 25, 2011

Espoo VTT

Jan-Erik Holmberg, VTT

VTT Technical Research Centre of Finland

VTT IS

- the biggest multitechnological applied research organisation in Northern Europe

VTT HAS

- polytechnic R&D covering different fields of technology from electronics to building technology
- clients and partners: industrial and business enterprises, organisations, universities and research institutes

VTT CREATES

- new technology and science-based innovations in co-operation with domestic and foreign partners

- Turnover 245 M€
- Personnel 2,700
- 77% with higher academic degree
- 6,200 customers
- Established 1942
- VTT has been granted ISO9001:2000 certificate.

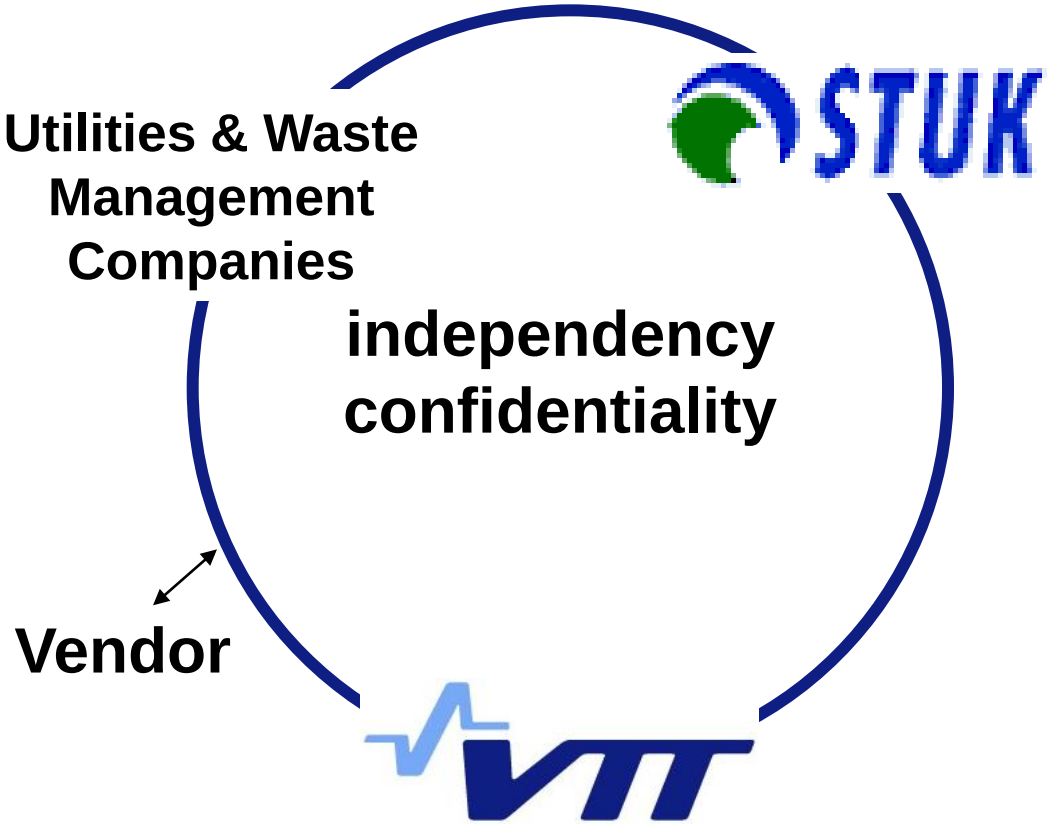
VTT nuclear energy R&D competencies

- VTT has 200 researchers in nuclear energy
- VTT research competencies cover
  - reactor safety
  - Gen-IV
  - waste management and
  - fusion
- VTT is the major technical support organisation for the authorities and the Finnish nuclear industry
- VTT performs contracted research on challenging topics related to nuclear safety, plant life management and nuclear waste management

Independent and confident VTT

☞ VTT serves both STUK and industry

- **Not the same analyses**
- **Not the same people**
- **Not the same equipment**
- **Not the same software**
- **In case of same application & same software:**
  Input data & assumptions are selected/given by STUK and sensitivity analyses are performed by VTT – preferably by a different person



Strategic R&D focus areas

1) Reactor safety - Existing plants and new builds

- Deterministic safety analyses
  - Fuel and reactor physics
  - Thermal hydraulics
  - Accident and transient analyses
  - Severe accident management
- Structural safety of reactor circuit and structures
- Risk-informed safety management
- Simulation products and services
- Automation and control room
- Radiological impacts and emergency preparedness
- Organisation and human factors

2) Waste management and geological disposal

- Performance analyses and experiments of technical and natural barriers of repositories
- Technology development of engineered safety barriers

3) New generation reactors

4) Nuclear fusion technology

VTT Systems Research

Models, analyses, simulation and software for better safety and productivity of nuclear power plants

Systems Analysis

- Probabilistic risk assessment (PRA)
- Assessment of safety critical automation (I&C)

Computer Simulation Models and Technology

- Plant-wide dynamic simulation models
- Simulation based training and testing of automation
- Semantic information models in industry: integration of simulation with design

Human Factors Engineering (HFE) and Systems Usability

- Human activity and Human-Technology Interaction (HTI) in control centres
- Development and evaluation of control room operations and technology
- Competence development and training

VTT creates business from technology

National Nuclear Power Plant Safety Research Programme 2011-2014

SAFIR2014 overview

Kaisa Simola
SAFIR2014 programme director

SAFIR2014 programme 2011-2014

SAFIR2014

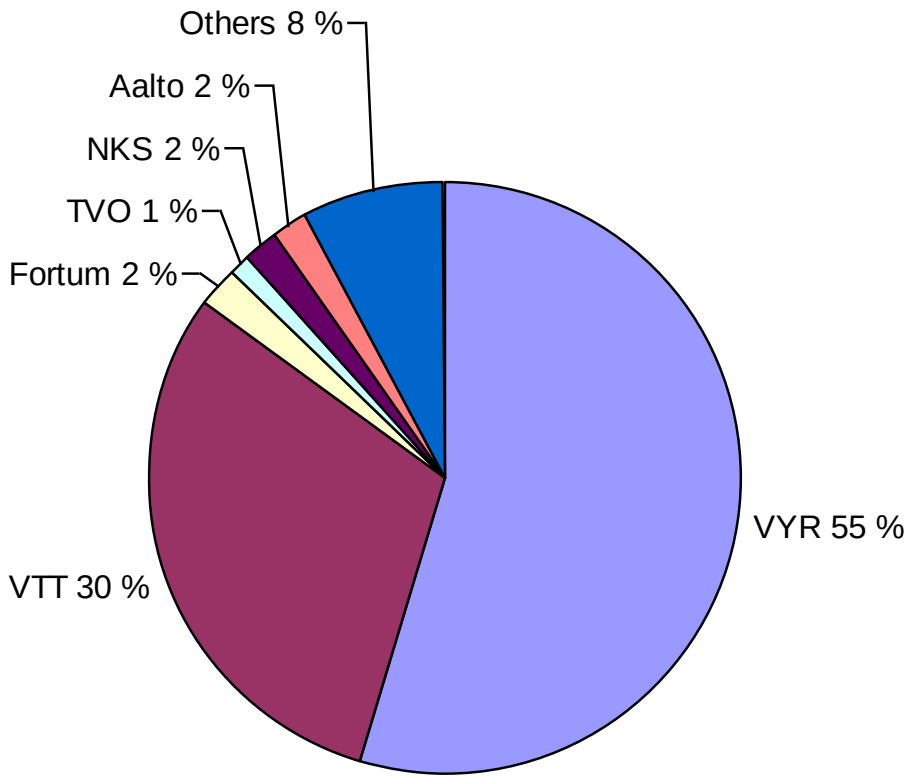
- Continuation to a series of national NPP safety research programmes (since 1990)

- Mission of the research programme is derived from the stipulations of the Finnish Nuclear Energy Act:

  The objective of the SAFIR2014 research programme is to develop and maintain experimental research capability, as well as the safety assessment methods and nuclear safety expertise of Finnish nuclear power plants, in order that, should new matters related to nuclear safety arise, their significance can be assessed without delay.

SAFIR2014 programme in 2011
Planned volume 9.4 M€

SAFIR2014



Planned Funding shares in SAFIR2014 in 2011

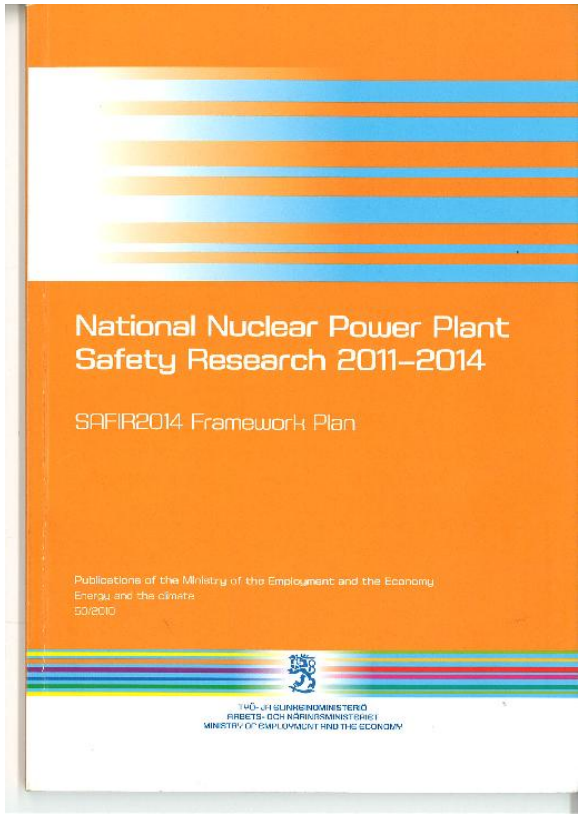
Main funding source: The Nuclear Waste Management Fund (VYR)

SAFIR2014

Framework for the programme

- SAFIR2014 **Framework Plan** published together with the call for proposals in Autumn 2010

- Supplements to the Framework Plan for the call for 2012
  - Fukushima-related issues
  - Topics for social research



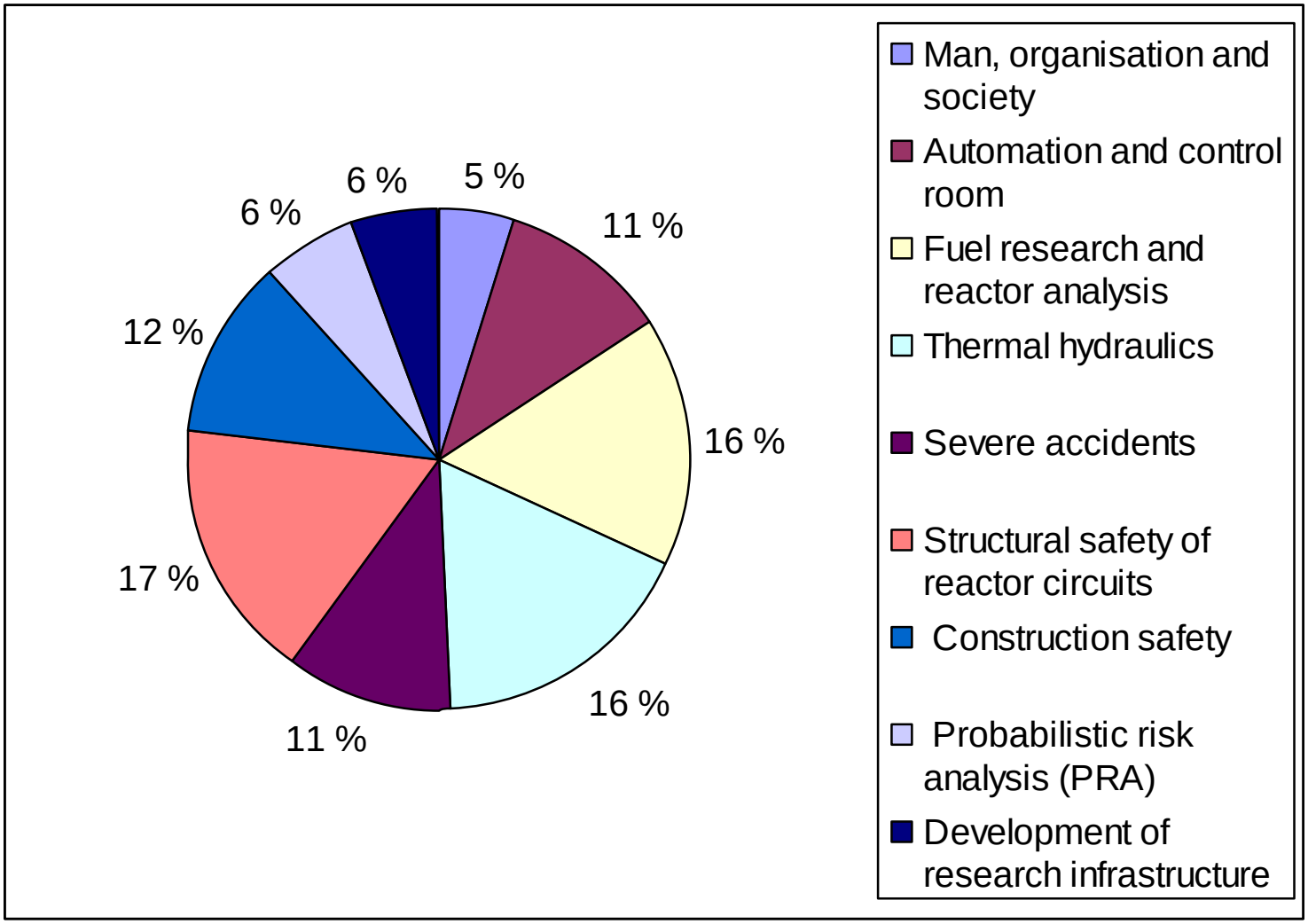
SAFIR2014 programme 2011-2014

SAFIR2014

Research areas	(# of on-going projects)
1. Man, organisation and society	(2)
2. Automation and control room	(4)
3. Fuel research and reactor analysis	(5)
4. Thermal hydraulics	(8)
5. Severe accidents	(4)
6. Structural safety of reactor circuits	(6)
7. Construction safety	(4)
8. Probabilistic risk analysis	(3)
9. Development of research infrastructure	(2)

SAFIR2014 programme in 2011

SAFIR2014



Planned total volume of research projects 9.4 M€

Research topics in 2011

SAFIR2014

2 – Automation and control room

- Coverage and rationality of the software I&C safety assurance (CORSICA)
- Human-automation collaboration in incident and accident situations (HACAS)
- Safety evaluation and reliability analysis of nuclear automation (SARANA)
- Safety requirements specification and management in nuclear power plants (SAREMAN)

International co-operation

SAFIR2014

Nearly all projects have international contacts:

- OECD/NEA experimental projects and database projects, NEA working groups
- EU networks and projects
- Nordic co-operation (NKS, NORTHNET, Halden)
- Co-operation with universities and research institutes
- Co-operation with nuclear industry and safety authorities abroad

SAFIR2014 programme 2011-2014

SAFIR2014

http://safir2014.vtt.fi

NKS/DIGREL
STUK perspective

Digital Automation & Risk Analysis

25.10.2011 / Ilkka Niemelä

with viewpoints of Mika Koskela

I&C Design and Analysis Problems

- I&C has been recognized as one of the major challenges on nuclear power generation field. Anyway, we feel that the discussion sometimes misses its mark.
- A kind of scapegoat has been software and software based technology. However, a major proportion of problems we have met is not technological but about lack of proper project control and use of ad-hoc human activity.
- Problems arise from inadequate design and analysis of technology, not from the technology itself.
- Our inspectors have seen (obvious!) design errors – independent of technology – which could have been sorted out before sending the documents to the authority.
- Should we focus on design and implementation processes or on the complexity and details of software based technology – or both? Are both equally possible?

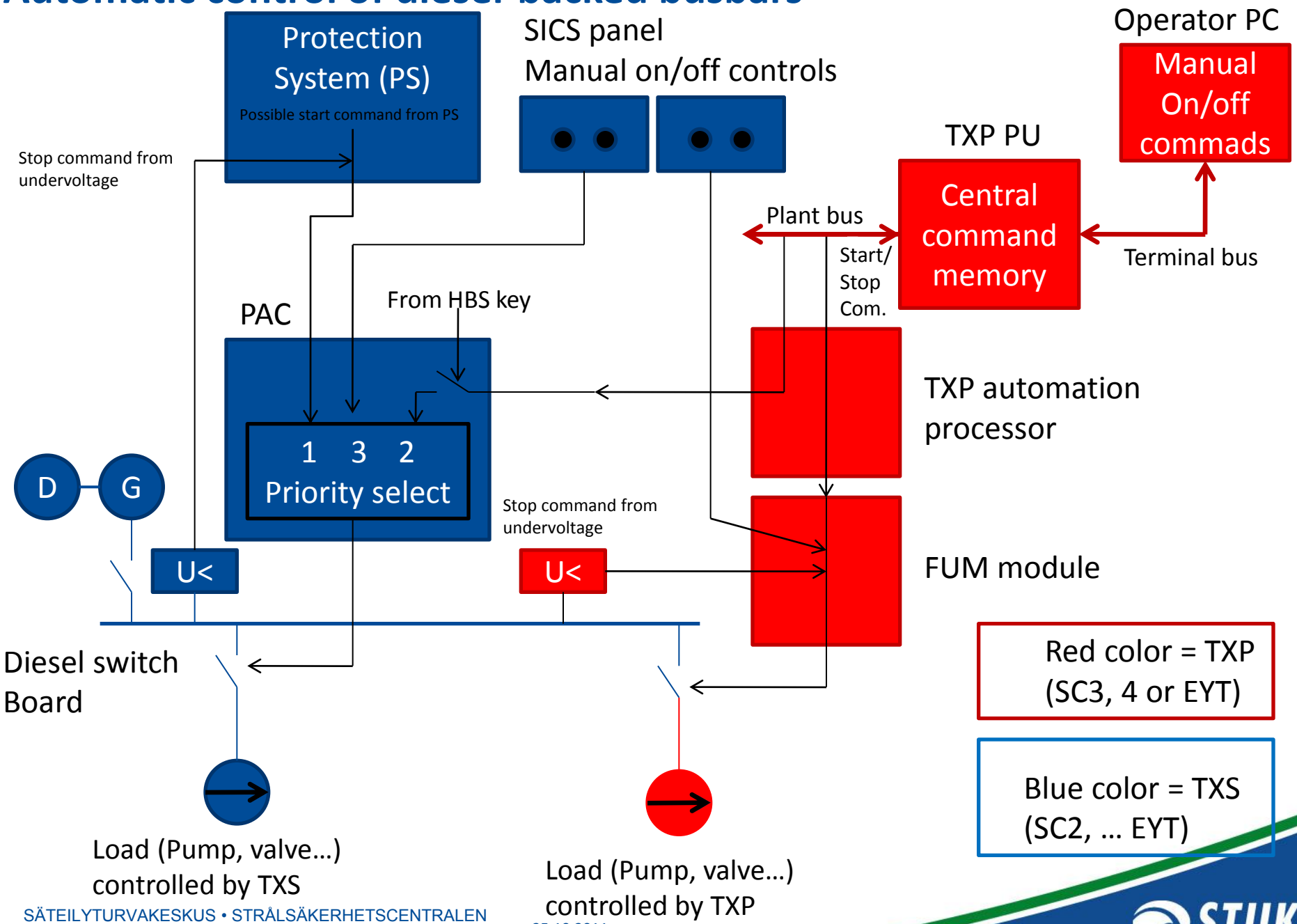
Which kind of analyses reveal problems in the following architecture?

More specifically:

In which requirement specification is the correct behavior of the process-automation-protection entity defined?

Can we find the errors by testing each individual I&C system against its own requirements specification only?

Automatic control of diesel-backed busbars



Requirement oriented design

- Is it possible to find the above design flaw without requirements?
  - Discovery outside requirement specification by NPP professional is more or less random and depends on the insights of individuals.

- Requirements are <u>the</u> starting point
  - design basis; reference models; failure behavior; functionality...
  - for configuration mgmt: the situation picture present; “What we have?” “What has to be done?”
  - for testing: how we expect system to behave?
  - for failure analysis: what is a failure; what is a success

- Requirement oriented approach is the only way to survive with complexity
  - large scale applications
  - software intensive systems

Risk-related Failure Analyses in New YVL Guides

- Failure tolerance analysis
  - Shows that the system fulfills its function in presence of failures according to success criteria (N+1, N+2, D+1)
  - Demonstrative

- FMEA
  - Descriptive analysis of system under failures

- PRA
  - Descriptive numerical risk analysis, based on FMEA

I&C Failure Tolerance Analyses Levels

1. Control diagram / protection diagram level
   - Analysis of logic independent of implementation
   - HAZOP on signals/parameters: too little, too much, no change, too early, too late, contradictory
   - E.g. CCF analysis for sensors of OL3 protection system showed that simultaneous erroneous signals from all sensors performing the same task does not prevent from reaching safe state
   - Identifies signals/parametersw where incorrect values are critical or non-critical
2. Signal / System level
   - Identifies reasons for incorrect signals/parameters (HW/SW)
3. System level
   - Identifies dependencies within the system and dependencies on support systems

Versus Analyses on One Level

One-level I&C failure analyses typically

- Assume a failure and then describe its effect to a complex system without presenting the mechanism of propagation
- Thus, are not traceable
- Can not be reviewed, can only be believed
- Are of little use
- Are too many!

Requirement Oriented Design

- Because of the difficulty of reaching adequate confidence on software intensive systems only by testing, the method for licensing software intensive systems is in general two-step approach:
  - Assurance that the requirements are correct (starting point and target of the activities)
  - Assurance that the actor has proper capability to reach the target (processes, organizational issues)
  - Additional elements are
    - independence (included in safety systems development) (licensee/supplier scope)
    - checking verification efficiency by targeted inspections/checks using diverse verification methods etc. (authority scope)

- Emphasized in future STUK YVL design guides

Requirement Oriented Design & Analysis

- Focus must be set on early phases of lifecycle/project
  - the most important decisions are concerning requirements, design bases, high level design etc.
  - designer / analyst may not assume/invent, but refer to requirements specification
- Strict holdpoints
  - clear motivation for licensee, suppliers and authorities
- Traceable analyses based on requirements
  - Credibility in design AND analyses

Summary

Can you define acceptance criteria before analysis?

- If not, there is no need to perform analysis
  - This could be a sign of unclear/ambiguous situation
- If yes, make a traceable analysis
  - This is a sign of mastered/well defined situation

Business from technology

WGRISK DIGREL task
Objectives and scope

NKS/DIGREL seminar, October 25, 2011

Espoo VTT

Jan-Erik Holmberg, VTT

DIGREL Project aim

- The objective with the project is to provide guidelines to analyse and model digital systems in PSA context
  - International OECD/NEA WGRISK task
    - focused on failure modes taxonomy
  - Nordic “NKS/NPSAG” effort
    - aims to cover wider scope of issues

WGRISK task background

- In 2007, the OECD/NEA CSNI directed the Working Group on Risk Assessment (WGRisk) to set up a task group to coordinate an activity in this field
- One of the recommendations was to develop a taxonomy of failure modes of digital components for the purposes of probabilistic safety assessment (PSA)
- Task report NEA/CSNI/R(2009)18, OECD/NEA/CSNI, Paris, 2009 http://www.oecd-nea.org/nsd/docs/2009/csni-r2009-18.pdf

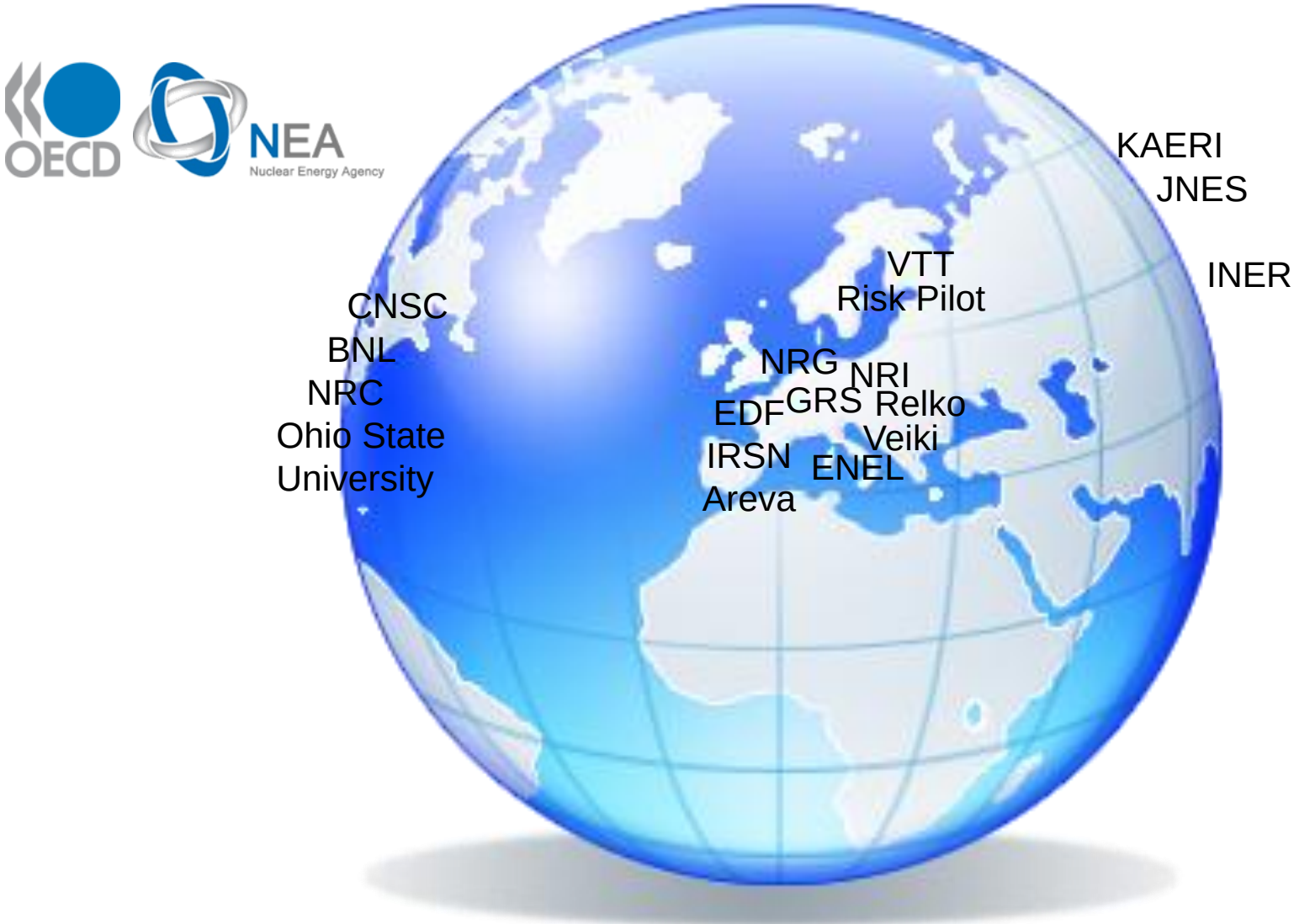
Nordic project background

- Variety of experience of analysing digital I&C in PSA context
  - Most plants do not yet have digital RPS, but will have in future
  - Turbine plant I&C and diverse other safety-related systems are already digital, but have minor role in PSA context
  - New-builts (in Finland) will have complete digital I&C
- No common approach (yet)
  - However, there is a tradition to try find harmonised approaches for PSA and its applications
- Generally strong interest to find solutions and guidelines how assess safety and reliability of digital I&C and how to meet regulatory requirements

WGRISK Task objectives

- To develop technically sound and feasible failure modes taxonomy for reliability assessment of digital I&C systems for PSA
- To provide guidelines on the use of taxonomy in modelling, data collection and quantification of digital I&C reliability

WGRISK/DIGREL Task Group



Scope

- The activity focuses on failure modes taxonomy and its application to modelling, data collection and impacts on quantification

- The following items will be considered (but not limited to):
  - Protection systems and control systems
  - Hardware and software
  - Development, operation and maintenance
  - Failure detection and recovery means

- Needs of PSA are addressed

Perspective of the task

- I&C of nuclear power plants
- PSA/PRA
- Reliability analysis
- Taxonomy
- Failure modes, FMEA
- Guidelines

Overall activities 2011-13

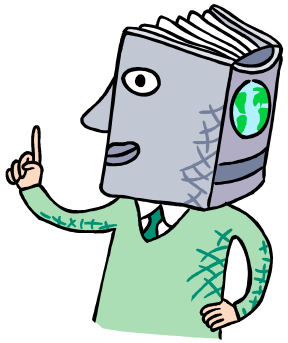
- Collection, comparison and analyses of failure modes taxonomies for hardware and software of digital components
- Development of generic I&C system example for demonstration and benchmarking purposes
- Guidelines regarding level of detail in system analysis and screening of components, failure modes and dependencies
- Approach for modelling of CCF between components (including software)

WGRISK DIGREL actions so far

- Task group established
- Several phone meetings, a few physical meetings, one workshop
- DIGREL working file repository created http://proxnet.vtt.fi/digrel/ for TG members only
- Input to failure mode taxonomies collected from TG members and compared
- Draft fictive I&C system examples
- Preliminary proposals for taxonomies
- Outline of the guidelines
- Conferences abstracts

Dissemination

- WGRISK guidelines: draft 2012, final 2013 (WGRISK approval March 2013, CSNI meeting June 2013)



- NKS/NPSAG guidelines
  - pre-study 2010 [NKS-230]
  - interim reports 2011, 2012
  - final 2013

- Conference papers
  - PSAM11 (June 2012) – four abstracts submitted
  - NPIC-HMIT (July 2012) – several abstracts considered

DIGREL Guidelines

- Document for PSA practitioners and reviewers
- Should be useful for I&C experts, too

- We should aim to have a consensus
- Guidelines will be a proposal of an expert group
- The conclusions and viewpoints presented in the guidelines are those of the authors and do not necessarily coincide with those of OECD/NEA or those of the organisations experts come from

Seminars/workshops:

Past

- WGRISK workshop May 16-19, 2011, hosted by U.S.NRC

Present

- NKS/NPSAG seminar October 25, 2011 in Espoo + task group workshop

Future

- WGRISK workshop May 2012 in Munich hosted by GRS
- NKS/NPSAG workshop 2012 in Espoo or Stockholm

국가 미래 에너지를 책임지는 연구원

Terms and Definitions for Reliability Assessment of Digital I&C Systems

Man Cheol KIM Ph.D.
Integrated Safety Assessment Division
Korea Atomic Energy Research Institute

KAERI
Korea Atomic Energy Research Institute

Outline

1. Introduction
2. IEC and IEEE Standards
3. Example I&C System
4. Conclusions

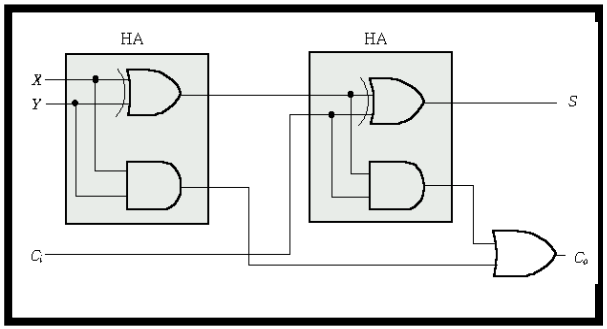
1. Introduction

- Challenges with developing consensus definitions (Dr. T. L. Chu, BNL)
  - "The WGRisk DIGREL appears to be trying to re-define some of the terms that are considered more suitable for PRA uses."
  - "Otherwise, we need to investigate the existing definitions that may differ significantly."
  - "Trying to reach a consensus on the definition of the terms may be as difficult as developing a guidance or standard."

1. Introduction

- Project/Activity Title
  - Development of best practice guidelines on failure modes taxonomy for **reliability assessment** of **digital I&C systems** for PSA

- Question
  - Are we covering all digital I&C systems?
    - Safety-critical digital I&C system
    - (Digital) computer-based safety system
    - Digital computer in safety systems
    - Software-based safety system
    - Programmable system
    - Electric equipment important to safety
    - Class 1E digital computer system



1. Introduction

- Safety classification (1/2)

Term	Standard	Definition
safety system	IEEE Std 603-1998 IEEE Std 7-4.3.2-2003	A system that is relied upon to remain functional during and following design basis events
safety-related system	10 CFR 50.2	
	IEC 61508	designated system that both • implements the required safety functions necessary to achieve or maintain a safe state for the EUC; and • is intended to achieve … the necessary safety integrity for the required safety functions

Notes from NUREG/CR-6101

"The term *safety critical* refers to a system whose failure could cause an accident."

1. Introduction

- Safety classification (2/2)

National or International Standard	Classification to the Importance to Safety			
IAEA	Systems Important to Safety			Systems Not Important to Safety
	Safety System	Safety Related System		
IEC 61226	Category A	Category B	Category C	Unclassified
France N4	1E	2E	IFC/NC	
European Utility Requirements	F1A (Automatic)	F1B (Automatic and Manual)	F2	Unclassified
Russia	Class 1 (beyond DBA*)	Class 2 (Safety System, DBA)	Class 3	Class 4
UK	Category 1	Category 2	Unclassified	
USA (IEEE)	1E	Non-nuclear Safety		

2. IEC and IEEE Standards

- IEC Standards
  - IEC 61508
    - Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems
- IEEE Standards
  - IEEE Std 603-1998
    - IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations
  - IEEE Std 7-4.3.2-2003
    - IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations

2. IEC and IEEE Standards

- Same definitions in IEEE standards

Term	Standard	Definition
safety system	IEEE Std 603-1998	A system that is relied upon to remain functional during and following design basis events
	IEEE Std 7-4.3.2-2003	

Notes in IEEE Std 603-1998

1. The electrical portion of the safety systems, that perform safety functions, is classified as Class 1E.
2. This definition of "safety system" agrees with the definition of "safety-related systems" used by the American Nuclear Society (ANS) and IEC 60231A (1969-01).
3. Users of this standard are advised that "Class 1E" is a functional term. Equipment and systems are to be classified Class 1E only if they fulfill the functions listed in the definition.

2. IEC and IEEE Standards

- Different definitions in IEEE standards

Term	Standard	Definition
component	IEEE Std 603-1998	discrete items from which a system is assembled
	IEEE Std 7-4.3.2-2003	one of the parts that make up a system
module	IEEE Std 603-1998	any assembly of interconnected components that constitutes an identifiable device, instrument, or piece of equipment
	IEEE Std 7-4.3.2-2003	a program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading

2. IEC and IEEE Standards

- Same definitions in IEC and IEEE standards

Term	Standard	Definition
functional unit	IEC 61508	entity of hardware or software, or both, capable of accomplishing a specified purpose
	IEEE Std 603-1998	an entity of hardware, software, or both capable of accomplishing a specified purpose

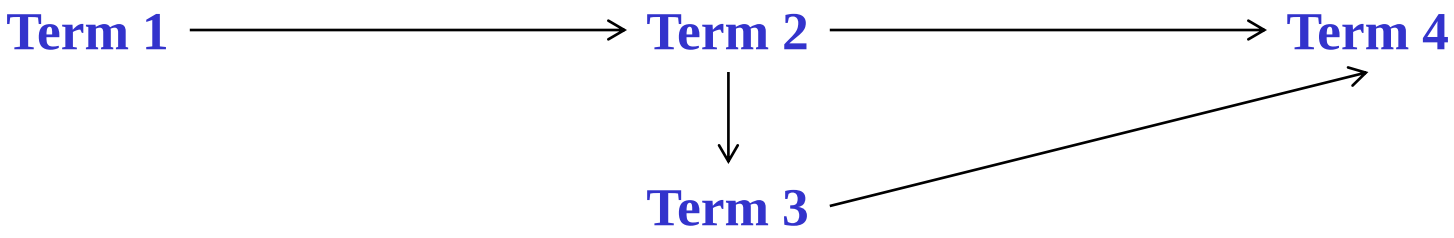
2. IEC and IEEE Standards

- Different definitions in IEC and IEEE standards

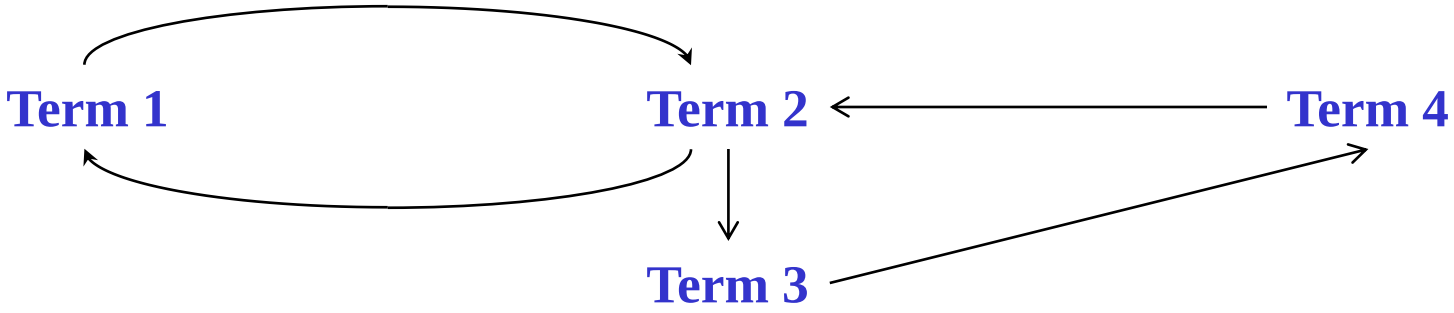
Term	Standard	Definition
architecture	IEC 61508	specific configuration of hardware and software elements in a system
	IEEE Std 7-4.3.2-2003	the organizational structure of a system or component
channel	IEC 61508	element or group of elements that independently perform(s) a function
	IEEE Std 603-1998	an arrangement of components and modules as required to generate a single protective action signal when required by a generating station

2. IEC and IEEE Standards

- Dependencies between terms
  - IEC Standards

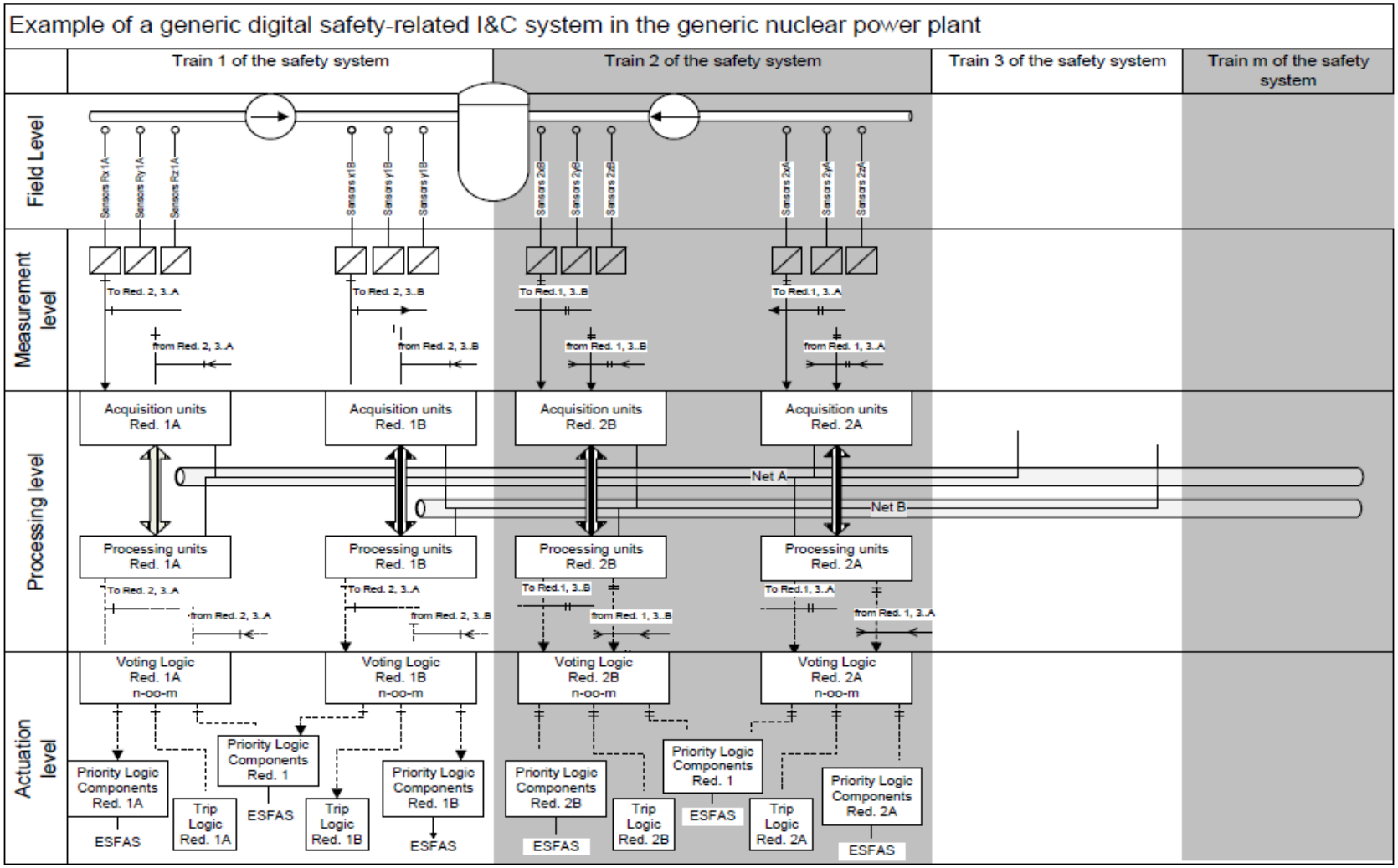


  - IEEE Std 603-1998 and 7-4.3.2-2003



3. Example I&C System

- Example I&C system (Ewgenij Piljugin, GRS)



4. Conclusions

- IEC and IEEE standards
  - IEC 61508
    - Acyclic
    - Hierarchical
  - IEEE Std 603-1998 and 7-4.3.2-2003
    - Different definitions in different IEEE standards
    - Cyclic
    - Descriptive
- Proposed integration
  - Based on IEC standards
  - Add descriptive explanations in IEEE standards

WGRISK DIGREL Task Group Workshop
Helsinki, Oct. 25 – 28, 2011

Survey of Failure Modes

Tsong-Lun Chu
Energy Sciences and Technology Department
(631-344-2389, Chu@BNL.GOV)

BROOKHAVEN NATIONAL LABORATORY
a passion for discovery

Office of Science
U.S. DEPARTMENT OF ENERGY

Outline of Presentation

- Objectives
- Summary of failure mode taxonomies provided by participants
- Definition of levels of detail for hardware and software
- Consensus failure modes for software from DC worksh op
- Persisting scope and technical Issues
- Path forward

Objectives

- Summarize the failure mode taxonomies provided by participants.
- Clarify definitions of different levels of detail for hardware and software.
- Discuss scope and technical issues.

Eleven Organizations Provided Inputs

- NRG (Nuclear Research and Consultancy Group)
- EDF (Electricity of France)
- JNES (Japan Nuclear Energy Safety Organization)
- BNL (Brookhaven National Laboratory)
- Nordic survey (PSAs for Ringhals, Olkiluoto and Loviisa NPPs)
- OSU (Ohio State University)
- ORNL (Oak Ridge National Laboratory)
- KAERI (Korean Atomic Energy Research Institute)
- CNSC (Canadian Nuclear Safety Commission)
- IRSN (Institut de Radioprotection et de Surete Nucleaire)

Summary of Taxonomy Inputs from Participants

- Two tables summarizing hardware and software failure modes including the failure modes identified by the software group at DC workshop were prepared.
- Each table further classifies the failure modes in terms of their levels of detail.
  - For hardware: system, channel, module, circuit board, and generic component levels
  - For software: system, channel, microprocessor, and sub-level
- Two additional columns provide information on modeling methods and data sources.

Levels of Detail for Failure Mode Definition: Hardware

- For a digital protection system, the levels of detail at which failure modes were defined and used in the failure mode survey may include
  - the entire system, e.g., an RPS.
  - a channel or a division: an RPS may consist of multiple (redundant or diverse) channels or divisions;
  - a module, e.g., input, output, and processing modules;
  - a circuit board: a module may be implemented using circuit board(s); and
  - the generic components, e.g., A/D and D/A et al.
- In principle, failure modes of a control system can be defined at the same levels of detail
  - Often control systems do not have redundant channels.

Levels of Detail for Failure Mode Definition: Software

- System Level:
  - For a digital protection system, at the system level, the software consists of the collection of software running on various microprocessors of the system and failure modes can be defined at this highest level.
- Channel Level:
  - For the redundant or diverse channels of an RPS, the collection of software running on the microprocessors of a single channel may also fail and cause the failure of that channel. Failure modes of all software belonging to a single channel can also be defined at this level as channel level failure modes.
- Microprocessor Level:
  - For the software program running on a particular microprocessor, the software is treated as an individual component like the microprocessor of a module.
- Sub-level:
  - The software that runs on a microprocessor may be complicated enough such that it can be further decomposed, to a so-called sub-level.

Observations on Participant Inputs

- Some participants included information on failure effects (as a part of an FMEA); others did not.
- In general, most of the failure modes are developed specifically for protection systems only.
- There is more information on hardware failures than on software failures but a set of consensus software failure modes were developed at DC Workshop.
- Levels of detail at which failure modes were defined are very different, and at the same level of detail failure modes from different participants may still be different.
- Although there is no consensus modeling methods, event tree/fault tree approach appears to be the most popular one.
- CCF did not receive too much attention.

Levels of Detail for Software Failure Mode Definition from DC Workshop, May 2011

- Failure modes can be defined at different levels of detail, i.e., RPS function level, trip signals level (high reactor pressure level), individual signal level (starting individual pump, not applicable to RPS but may be necessary for ESFAS).
- RPS software consists of data acquisition (e.g., communications, filtering, data validation, scaling), logic processing (e.g., standard elementary functions such as comparison or Boolean manipulation, application specific logic), voting (e.g., voting algorithm and communication), priority actuation logic (many be implemented only in hardware).

Example Software Failure Modes Identified at DC Workshop, May 2011 (1)

- System Level Failure Modes *
  - For an RPS: failure to actuate (including failure to hold); spurious failure; possible others dependent upon additional functions judged to be safety related.
  - For load sequencing: failure to activate in time.
  - For an ESFAS: failure of trip signals such as a high reactor pressure level.

*: failure modes at this level of detail may also be categorized as channel level failure modes.

Example Software Failure Modes Identified at DC Workshop, May 2011 (2)

- Microprocessor Level
  - For Data Acquisition*: 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.
  - 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).

*It may be worthy considering failure modes for communication logic (which seems to be considered a part of data acquisition here) separately, considering its importance.

Example Software Failure Modes Identified at DC Workshop, May 2011 (3)

- Sub-level failure modes are defined for software functional modules related to individual signals to hardware components such as pumps and valves.
- Example sub-level failure modes include failure of individual signals from an ESFAS to actuate pumps and valves.

Persisting Issues: Modeling Methods and Data Sources

- "*The taxonomy will be the basis of future modeling and quantification efforts. It will also help define a structure for data collection*." ~ The CAPS
- Most participants appear to have the fault tree modeling 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)?
- Different modeling methods may require different levels of detail. Before a standardized method for modeling is agreed upon, can we determine the preferred level of detail of the failure modes or should we leave the level of detail as an open item?
- Software failures are conditional, i.e., on the environment it is being executed. How can this be captured by the existing modeling methods?

Persisting Issues: How to Determine the Appropriate Level of Detail for Failure Mode Definition

- Modeling methods often define the levels of detail.
- Data availability is a realistic constraint on the level of detail.
- At the selected level of detail, the defined failure modes should be physically meaningful and the failure effects should be propagatable.
- Capability of capturing fault-tolerance features is desirable.
- Should CCF be defined at the same level as individual failures?

Persisting Issues: Should Failure Effects Be Included in the Taxonomy?

- Failure effects are needed in order to include failure modes in a reliability model (so that the failure effects can possibly be propagated).
- Can generic failure effects be identified and described or should they only be determined based on a specific application when developing a reliability model?

Persisting Issues: Definition of Failure Modes, Effects, Causes, and Mechanisms

- In the literature, these terms are often used inconsistently.
- Is there a standard that defines them? Should we endorse such a standard (or standards)?
- Should failure modes be defined in terms of functionality? In terms of functionality, is there any difference between software and hardware failure modes.
- Should failure modes be generic (i.e., apply to any digital protection or control system), be specific to a class of digital systems (e.g., reactor protection systems), or be specific only to a particular system?

Persisting Issues: Completeness and Genericness of Failure Modes

- Do we want to discuss the issue using the inputs from the participants at each level of detail to ensure some kind of completeness and genericness?
- Common cause failure modes of hardware and software should be further discussed. What should be the level of detail?
- Can communication and synchronization, which are sometimes only auxiliary functions of a system, introduce CCF or dependent failures?

What Additional Research is Needed on Failure Mode Taxonomy?

- Capturing interactions between hardware and software
- Do we need to treat application, platform, and operating system software separately?
- Assessment of “coverage”, for example, the ability/probability of a watchdog timer to detect failures
- Software CCF between diverse digital protection systems
- Use of reliability physics modeling to arrive at data

Path Forward

- Reach consensus on hardware failure modes and levels of detail.
- Should software failure modes of communication logic also be reviewed?
- For guideline development, should we define failure modes at a specific level of detail or at all levels of detail?
- How to assess the completeness and genericness of failure modes of hardware and software at each level of detail?
- Application of the failure mode taxonomy to case study.

Risk Pilot
Your Risk Navigator

NKS/DIGREL Seminar

Hardware Failure Modes Taxonomy

Helsinki October 25, 2011

Stefan Authén, CEO, Risk Pilot AB

Status of WGRISK taxonomy development

- Prerequisites for the HW taxonomy agreed
  - Criteria for definition
  - Scope
  - Conditions
- Existing HW taxonomies collected
  - Input for level of detail and failure modes
- Example system developed
  - Basis for taxonomy and test case

Criteria for taxonomy definition

- Required:
  - Defined clearly
  - Be organized hierarchically
  - Analogy between failure modes of different components
  - Supporting data should be potentially available
- Desired:
  - Defined unambiguously
  - Form a complete/exhaustive set
  - Be orthogonal/exclusive

Scope of taxonomy

- The taxonomy shall support PRA practice, i.e. appropriate level for PRA, fulfill PRA requirements/conditions
- The taxonomy shall primarily consider safety related systems, i.e. RPS and not control systems
- The taxonomy shall be on a level of detail such that all critical dependencies and design features are captured.
  - Requirement depends on PRA application
  - Taxonomy for more than one level of detail may be needed

Scope of taxonomy, cont´d

- The taxonomy shall cover both undetected and detected failures
- The taxonomy shall define failure modes
  - Not failure causes or mechanisms
  - Failure modes shall be RPS specific, not generic
- CCF needs to be addressed

Misc. conditions for the taxonomy

- Will be developed based on a function view
  - Component view not needed for taxonomy
- Different failure mode categories for HW, SW and SW/HW interaction
- Issues of definition of fault, trigger, failure are relevant but not considered at this point

Levels of detail WGRISK taxonomy

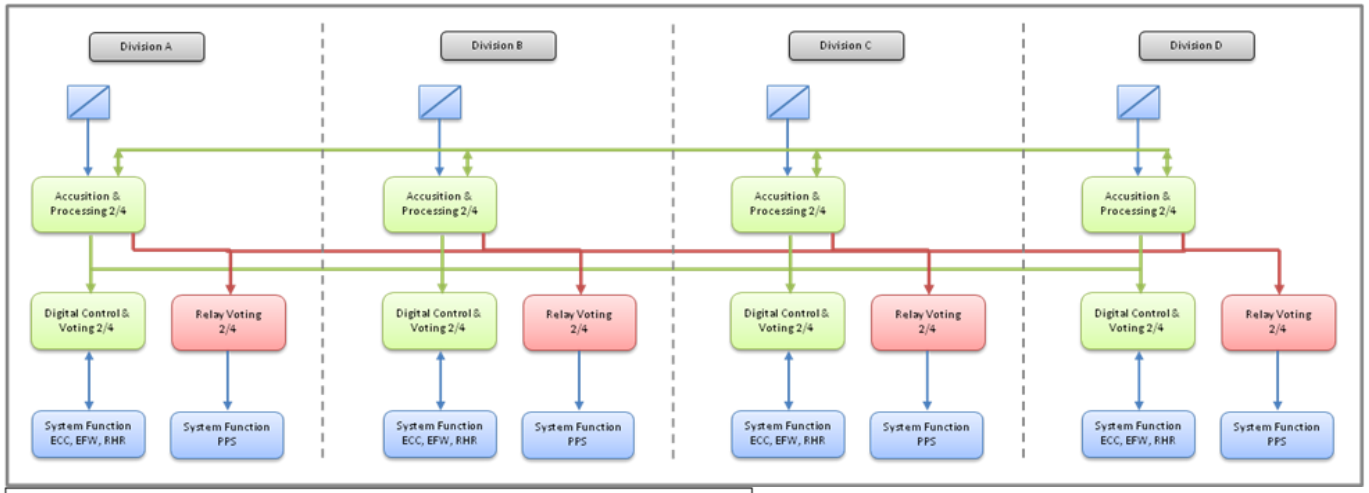
- The following possible levels of detail has been developed:
  - System Level
  - Channel/division
  - I&C module
  - Circuit board
  - Generic component

Level of detail NKS/DIGREL

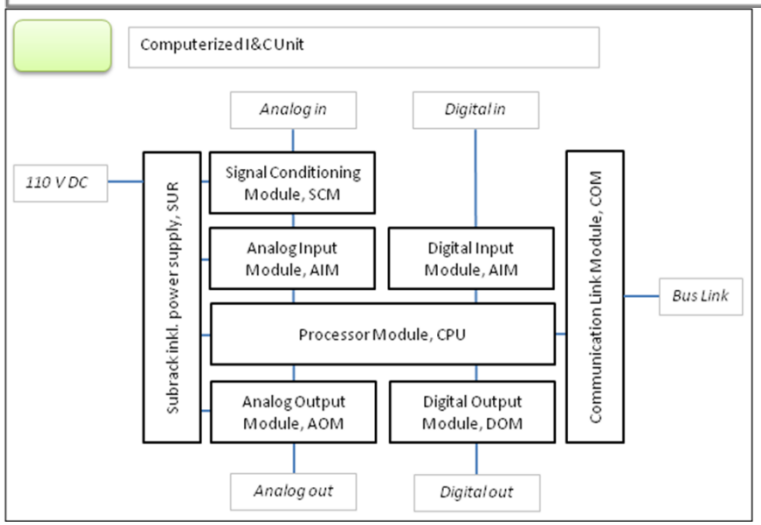
- The following possible levels of detail has been developed:
  - System Level
  - Division Level
  - Processing Level
  - Sub-component/Module Level
  - Generic component Level (I&C expert level)

Level of detail NKS/DIGREL, cont´d

- System
- Division
- Processing



- Module
- Gen. Comp.



PRA and I&C expert perspectives

PRA expert

- Follows the needs of PRA modelling
  - to capture relevant dependencies
  - to find justifiable reliability parameters
- Estimation of the system reliability P
- Level of details can be kept in rather high level

I&C expert

- Focused on failure mechanisms and their recovery means
  - to demonstrate that the system fulfil the safety and reliability requirements, including P < P* (residual risk is acceptably low)
- Needs to analyse failure initiation possibilities and failure propagation methods in a far more comprehensive manner

Requirements on level of detail

- Shall support PRA practice, i.e. appropriate level for PRA, fulfill PRA requirements/conditions
- Shall cover undetected and detected failures
- Shall capture all critical dependencies and design features
- Shall be appropriate for safety related systems
- Shall support definition of failure modes, not mechanisms
- Shall be based on function view, not component
- Shall support modeling of CCF:s at necessary level

Support PRA practice

- This means (among others):
  - Feasible analysis for PRA experts
  - Possible to implement into existing tools (fault tree / event tree)
  - Possible to review by PRA-expert
  - Possible to Quality Assure with reasonable effort
  - Allows living PSA, e.g. possible to maintain and update with reasonable resources
  - Available and maintainable failure data, i.e. allows collection and evaluation of operational events
  - Supports PRA applications
  - Other?

- Appropriate level: Module level(?)
  - May not fulfill requirement regarding applications
  - Failure data available at a level of higher detail (gen. comp., vendor)

Cover undetected and detected failures

- This means:
  - Fault detection, treatment of faulty signals and complex voting logic needs to be considered in the model
  - Spurious activation of safety functions due to detected failure of components needs to be covered

- Appropriate level: Module level

Capture critical dependencies/features

- This means:
  - Functional dependencies
    - incl. voltage and signaling dependencies
  - Area dependencies
  - Treatment of faulty equipment/signals
    - Fault detection, default values, output control
  - Voting logic
    - Ignore, trip, no trip, etc.
    - (2:nd max, 2:nd min)
  - Coverage of failure detection
  - CCF:s
  - SW failure modes
  - HW/SW interaction?
  - Tests, maintenance, AOT
  - Other?

- Appropriate level: Module level?
  - Do we capture all dependencies?
  - Coverage calculated at a level of higher detail (gen. comp., vendor)

Appropriate for safety related systems

- This means:
  - Relatively high level of detail in line with state of the art PRA
    - Level of detail should correspond with the general level of a detailed PRA
    - simplifications as for e.g. operational systems not acceptable
    - generic component level is too detailed
  - Issues regarding digital I&C control functions are secondary
  - Other?

- Appropriate level: Module level

Support definition of failure modes

- This means:
  - Failure mechanisms will not be addressed
  - A lower level of detail than gen. comp. level can be chosen
  - Chosen level must cover all critical failure modes, or facilitate aggregation from levels of higher detail
  - Other?

- Appropriate level: Module level
  - Module level should capture all relevant failure modes and will fit with SW failure modes

Based on function view

- This means:
  - Failure modes can be grouped/modularized to a higher extent
  - Lower level of detail than gen. comp. can be chosen
  - Other?

- Appropriate level: Module level
  - Processing level possible but excluded by other requirements

Support modelling of CCF

- This means:
  - Critical dependencies shall be covered
    - Grouping decided by the safety functions and/or controlled equipment
      - Different allocation of modules for different safety functions
      - Different modules for control of different components within a safety system
    - Should cover intra-system CCF
  - CCF parameters must be available or possible to estimate

- Appropriate level: Module level
  - Channels within a module may be important to cover depending on failure data and testing procedures

HW failure modes level summary

- System level => diversity
- Division level => physical separation (compartments), power supply
- Processing level => network topology, power supply, physical separation (room, cabinet), preliminary design info, system-specific I&C functions
- Module level => fail-safe-principle, software allocation, communication, spurious functions, CCF, tests, maintenance
- Generic component => failure mechanisms, failure data, failure detection, “vendor level”

Summary on requirements for level of detail

- Module level seems to fulfill the main part of the requirements
  - Level of CCF important: will dominate HW risk contribution
- Drawbacks:
  - All PRA applications may not be feasible at full extent, e.g. test interval optimization
  - Failure data and coverage needs (presently) to be aggregated to module level and supplied by vendor
  - Do we need a taxonomy for more than one level of detail?
- Some I&C included in “special solutions” may fall outside the taxonomy: inevitable

Failure modes at Module level

- The PRA expert wants to know:
  - The consequence of the failure on the safety function
    - Loss of function or spurious function
  - Will the failure be detected or pass undetected?

- In the PRA failure modes should be aggregated to as high level and as large groups as possible
  - Without loosing critical dependencies or consequences

Failure modes at Module level, cont´d

- From PRA practitioners point of view:
  - Detected functional failure
  - Undetected functional failure
  - Spurious failure

- Measurements special case:
  - Fails high, Fails low, Drift, Freeze

- Is this possible or do we miss dependencies and/or consequences?
  - If so, where do we need to go in further detail?

Next step

- Decide on level(s) of detail to be covered
  - Agreement between PRA experts and I&C experts
  - Apply to decided criteria and conditions
- Define failure modes for decided level(s)
  - A strong basis is given by the collected existing taxonomies
- Address CCF:s
- Define/agree on terminology

WGRisk DigRel - Taxonomy for Software Faults

N. Thuy
EDF R&D

October 25th, 2011

CHANGER L'ÉNERGIE ENSEMBLE

Use of SW Faults Taxonomy for PSA

- Sensitivity analysis to identify 'critical' values
  - PFDs, beta-factors, frequencies of initiating events triggered by I&C failures
- Dividing possible software faults / failure mechanisms into types
  - Types may overlap but should cover all reasonably possible faults / mechanisms
- Identification of measures taken to protect against each type
  - Fault avoidance, detection & removal, Avoidance of fault activating conditions, Avoidance of concurrent fault activating conditions (CCF)
- Assessment of their effectiveness
  - Particular modes could be removed from consideration
- Evaluation of contributions from the residual modes
  - Applying the most appropriate methods

Definitions 1/2

- **Fault**: abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function, or that may cause a functional unit to perate in any way other than required
  - Based on ISO/IEC 2382-14
  - **Software fault**: fault that can be squarely attributed to software
- **Failure**: termination of the ability of a functional unit to perform a required function, or operation of a functional unit in any way other than required
  - Based on IEV 191-04-01
- **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
  - Based on IEV 191-04-19
  - **Software failure**: failure caused by the activation of a software fault. Software failures are usually systematic

Definitions 2/2

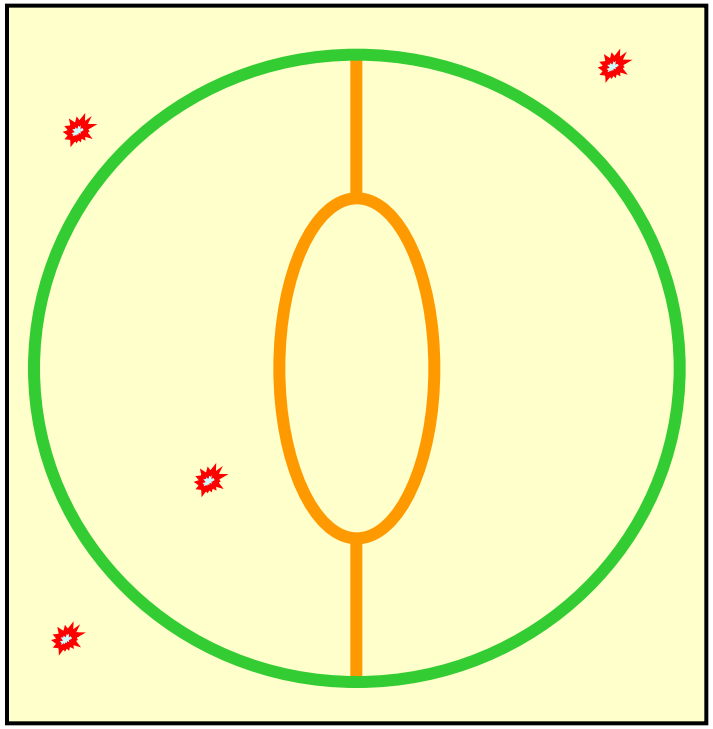
- **Failure mechanism**: event or chain of events occurring during operation and / or maintenance that can lead to the failure of a component, system or function
  - Failure mechanisms generally manifest themselves internal to the component or system
  - Possible failure mechanisms are dependent on the design decisions made and on the implementation technology used
- **Failure mode**: incorrect **external** behavior of a component, system or function
  - The component, system or function is viewed as a black-box
  - A priori independent from design and implementation technology

Scope

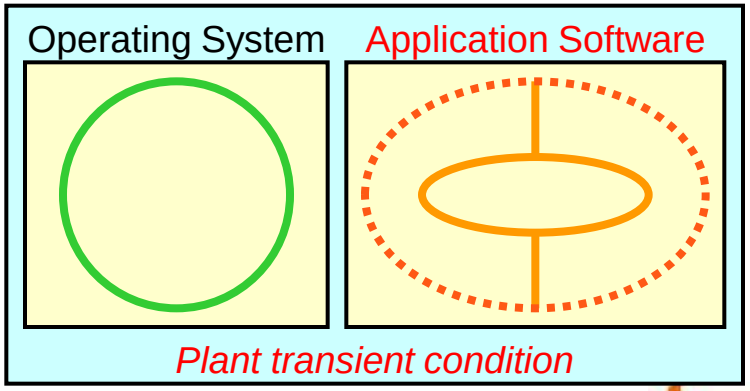
- *Failure mode* is a system-level notion that cannot be addressed at software level
- *Software faults* and *software failure mechanisms* are more meaningful notions
  - But there are cases where a fault or a failure mechanism can be squarely attributed neither to software nor to hardware, but to their interaction
  - Requirements specification faults need also to be considered
- The proposed faults taxonomy is mostly applicable to digital systems implementing Category A / 1E functions
  - Similar design principles
- It covers only faults made during I&C system development
  - Mistakes made during manufacturing & installation (e.g., incorrect default parameters), or operation & maintenance (e.g., incorrect setpoints, cyber attacks) are not covered
- Objective: help I&C systems developers and assessors to minimise the number of residual software faults
  - May also help PSA analysts identify the dominant software failure mechanisms

Assumptions 1/2

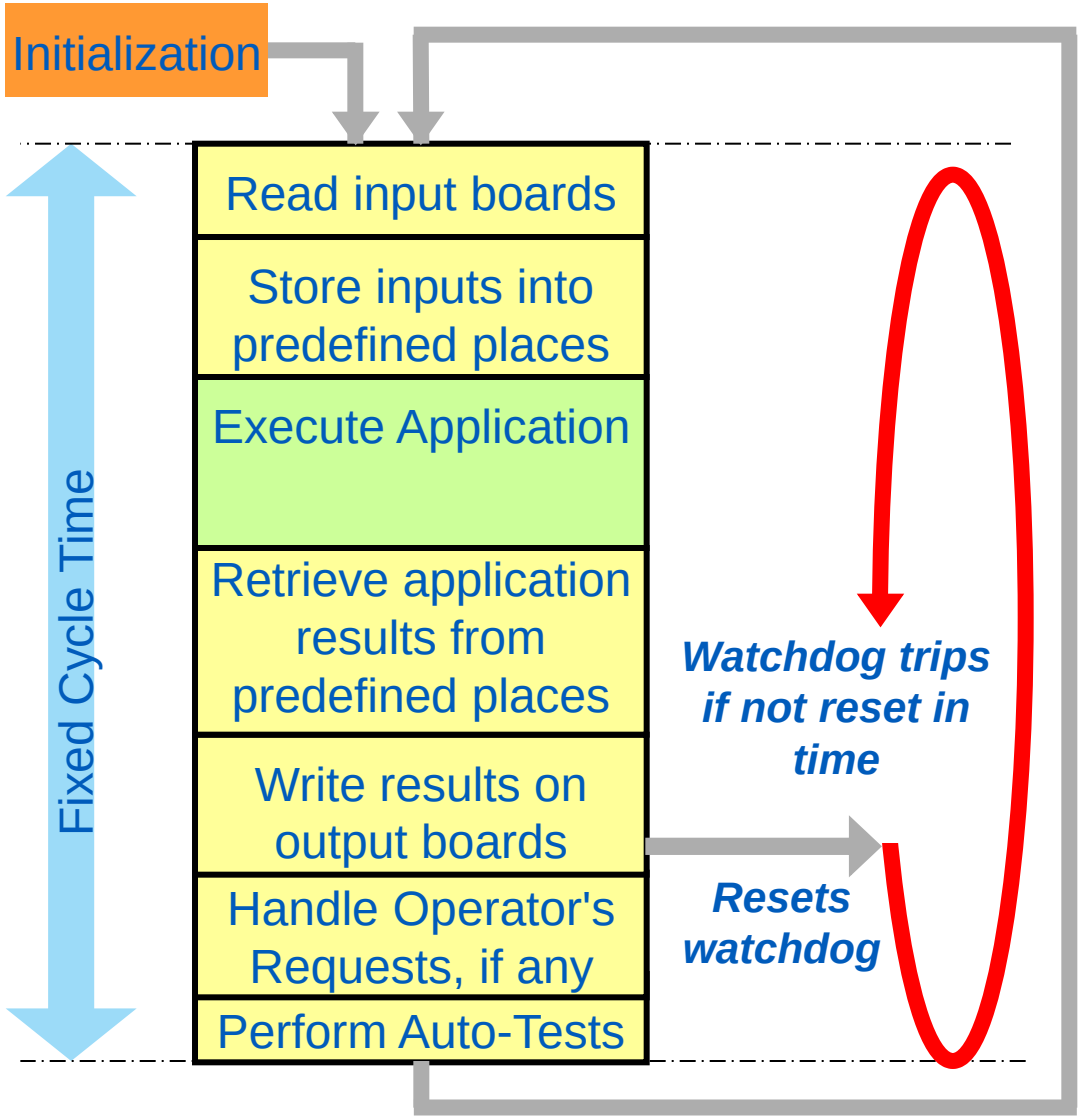
- Distributed I&C system architecture
  - Redundant, independent divisions
  - Diverse, segregated functions within each division
- Properties important to defence against Common Cause Failure (CCF)
  - Of identical redundant divisions
  - Or that could be caused by data communication
- Software designed to minimise influencing factors



- Main software components in individual computing units
  - Operating System (OS)
  - Application software
- Software designed to behave cyclically
  - Time-based design, not event-based
  - Also true of data communications
- OS designed not to be affected by plant conditions



Assumptions 2/2



Basis for the Proposed SW Faults Taxonomy

- **When** the fault at the origin of the failure was made
  - Development
  - Manufacturing, Installation, Operation, Maintenance
- **Location** of the fault
  - Application-specific functional requirements specification
  - Application-specific code or logic
  - Pre-developed I&C platform used to implement the system
- **Type** of the fault
- **Activating or revealing condition** of the fault
  - Particular dates and times (e.g., January 1st, 2000)
  - Data communication avalanches

Top Level View of the Proposed Taxonomy

- Application and I&C platform are subject to the same types of faults, but possibly with different interpretations

	Specification Faults	*Software Faults*	*Faults in SW / HW Interactions*
Application	•Incorrectness •Incompleteness •Out-of-date •Ambiguity	•Forgotten requirements •Inadequate algorithms or design •Programming faults •Incorrect SW version & configuration management •Translation tools faults •Non-compliance to claimed structural properties	•Insufficient system HW resources •Inconsistent system HW configuration
I&C Platform	•Noise •Over-specification •Excessive ambition		•Inadequate handling of HW failures •Insufficient detection of HW faults

Specification Faults in Applications 1/2

- Outright **incorrectness**: what is specified is not what the system really needs to do
  - Expression faults: what is specified is not exactly what the specifier had in mind
  - Understanding errors: the specifier did not have a fully correct understanding of what the system needs to do
- **Incompleteness**: necessary requirements or important characteristics are left unspecified
  - Intrinsic incompleteness: missing requirements can be identified without knowledge of the purposes of the system
    - Examples: incomplete look-up table, incomplete consideration of inputs combinations, unspecified timing accuracy
  - Incompleteness with respect to a given background
  - Others

Specification Faults in Applications 2/2

- **Out-of-date** requirements: what is specified might have been correct and complete at some time, but no longer reflects the current needs
  - E.g., due to changes in the system environment, regulation or plant operation
- **Ambiguity**: some requirements specification can be understood differently by different stakeholders / participants
  - Use of ambiguous terms, use of imprecise terms

Other Weaknesses in Application Specification

- **Noise**: genuine requirements are lost in a sea of (often well-meaning) comments
  - Some requirements could be ignored
- **Over-specification**
  - Specification of means without specifying the true goals of the system (its 'raison d'être')
  - Specification of requirements that are not really necessary to achieve the true goals of the system
- **Excessive ambition**
  - Could lead to unwarranted complexity and difficulty in achieving and justifying the required reliability level

Faults & Weaknesses in I&C Platform Documentation

- Outright **incorrectness**: what is documented is not what the platform really does
- **Incompleteness**: important characteristics or features of the platform are left undocumented
- **Out-of-date** documentation: what is documented might have been correct and complete at some time, but no longer reflects the current version of the platform
- **Ambiguity**: some statements in the documentation can be understood differently by different stakeholders / participants
- **Noise**: genuine platform characteristics or features are lost in a sea of comments
- **Excessive ambition**: the platform far exceeds what is really needed

Types of Software Faults 1/2

- **Lack of implementation** of some required functions or features
- **Use of inappropriate algorithms**
  - Leading to incorrect outputs and / or timings, not necessarily affecting all output signals
- **Inadequate software design**
  - Incorrect allocation of system requirements to individual parts of a distributed system
    - E.g., of timing requirements
  - Incorrect interaction between software components
    - Within a computing unit. Examples: incorrect assumptions regarding the use of a given software component
    - Between computing units
  - Incorrect allocation of priorities
- **Programming errors**
  - Some appear as **intrinsic faults**, i.e., faults that can be recognised without any knowledge of the requirements
    - Examples: Division by zero, out-of-bound index, numerical overflow / underflow, use of non-initialised variable
    - Examples for more complex systems (not likely for Cat A/1E): deadlocks, memory leaks
    - Could affect and cause the failure of a complete computing unit

Types of Software Faults 2/2

- **Incorrect software version & configuration management**
  - Within a single computing or communication unit, or among the multiple computing and communication units that constitute the system
- **Translation tools faults**
  - Example: faults introduced by inadequate compilers
- **Non-compliance to claimed structural property**
  - Could jeopardize the argument for defence against particular CCF mechanisms
  - Example: OS not transparent to plant conditions, …

Faults in Application SW Interactions with HW

Insufficient system HW resources

- The application software consumes more resources than the hardware configuration of the system provides
  - CPU, memory, data communication bandwidth

Inconsistent system HW configuration

- The HW configuration given to the application software does not correspond to the real system HW configuration
  - I/O, data communication, …

Faults in OS Interactions with HW

- Inadequate handling of HW failures
- Insufficient detection of HW faults

Conclusion

- For each item in the fault taxonomy, what are the measures taken to
  - Avoid faults (prevention)
  - Detect & remove faults
  - Avoid occurrence of their activating conditions, or avoid **concurrent** occurrences (defence against CCF)
  - Mitigate the effects at I&C system level (tolerance)
- Failure modes need to be analysed and 'taxonomied' from a functional standpoint
  - Could be caused either by software or hardware, or by both

Reminder: Activating or Revealing Conditions

- Specific plant / process conditions
  - Unlikely to affect the OS
- Interactions with other systems
  - Normally, such interactions are strictly limited
  - Interactions with systems less important to safety should not be able to adversely affect safety function
- Human-System interactions
  - Normally, one division at a time
- Elapsed time
  - Since power-up / initialisation of the computing unit
  - Usually, counter overflows
- Random hardware events

Reminder: Digital CCF Taxonomy

- Design faults in single-point vulnerabilities
- Identical or similar design faults in multiple systems or subsystems
- Failure propagation through data communication
- Shared susceptibility to global events
  - E.g., particular dates and times, malicious attacks

WGRISK Task DIGREL
Development of best practice guidelines on failure modes taxonomy for reliability assessment of digital I&C systems for PSA

Example of a generic digital I&C System (GenDIC)

NKS/DIGREL seminar
October 25, 2011, Espoo, Finland

Ewgenij Piljugin, GRS

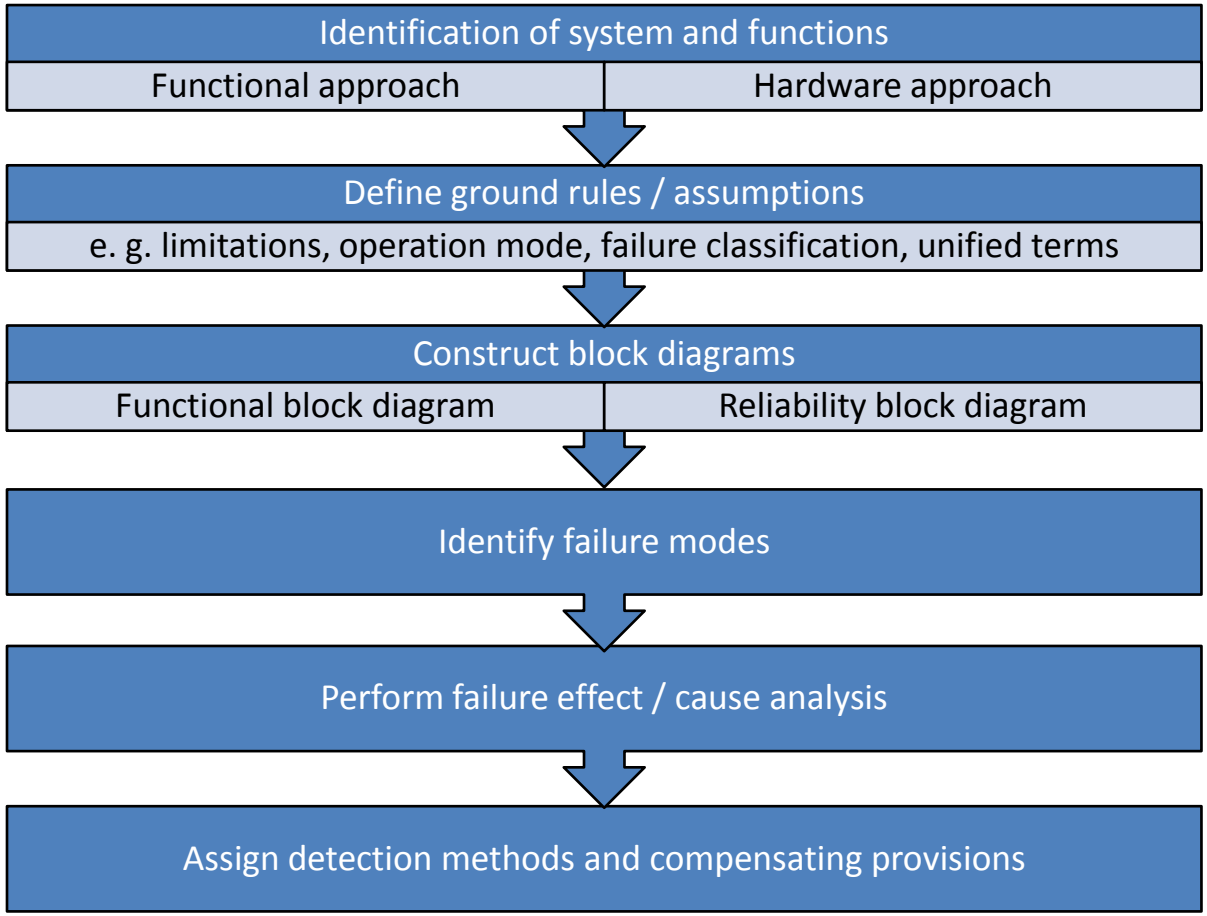
Content

- Objective and approach of the FMEA
- Proposal for the evaluation of the taxonomy approach
- Example of the structure a generic digital safety-related I&C system in the nuclear power plant
- Proposal for identification of generic feature of safety relevant communication of the DIC
- Concept for identification of generic failure modes of hardware
- Proposal for identification of generic issues regarding taxonomy of failures of software
- Examples of the FMEA of the digital I&C system
- Outlook

Objective and approach of the FMEA (1)

- The Failure Mode and Effects Analysis (FMEA) is a reliability evaluation/design technique which examines the potential failure modes within a system and its equipment, in order to determine the effects on equipment and system performance.
- The FMEA:
  - Determines the effects of each failure mode on system performance
  - Provides data for developing fault tree analysis and/or reliability block diagram models
  - Provides a basis for identifying root failure causes and developing corrective actions
  - Provides a foundation for qualitative reliability, maintainability, safety analyses
- The taxonomy approach should support the FMEA of a particular digital I&C system on the basis of the generic FMEA issues of the different types of the digital I&C systems.

Objective and approach of the FMEA (2)



Main steps of the evaluation of the FMEA

Proposal for the evaluation of the taxonomy approach (1)

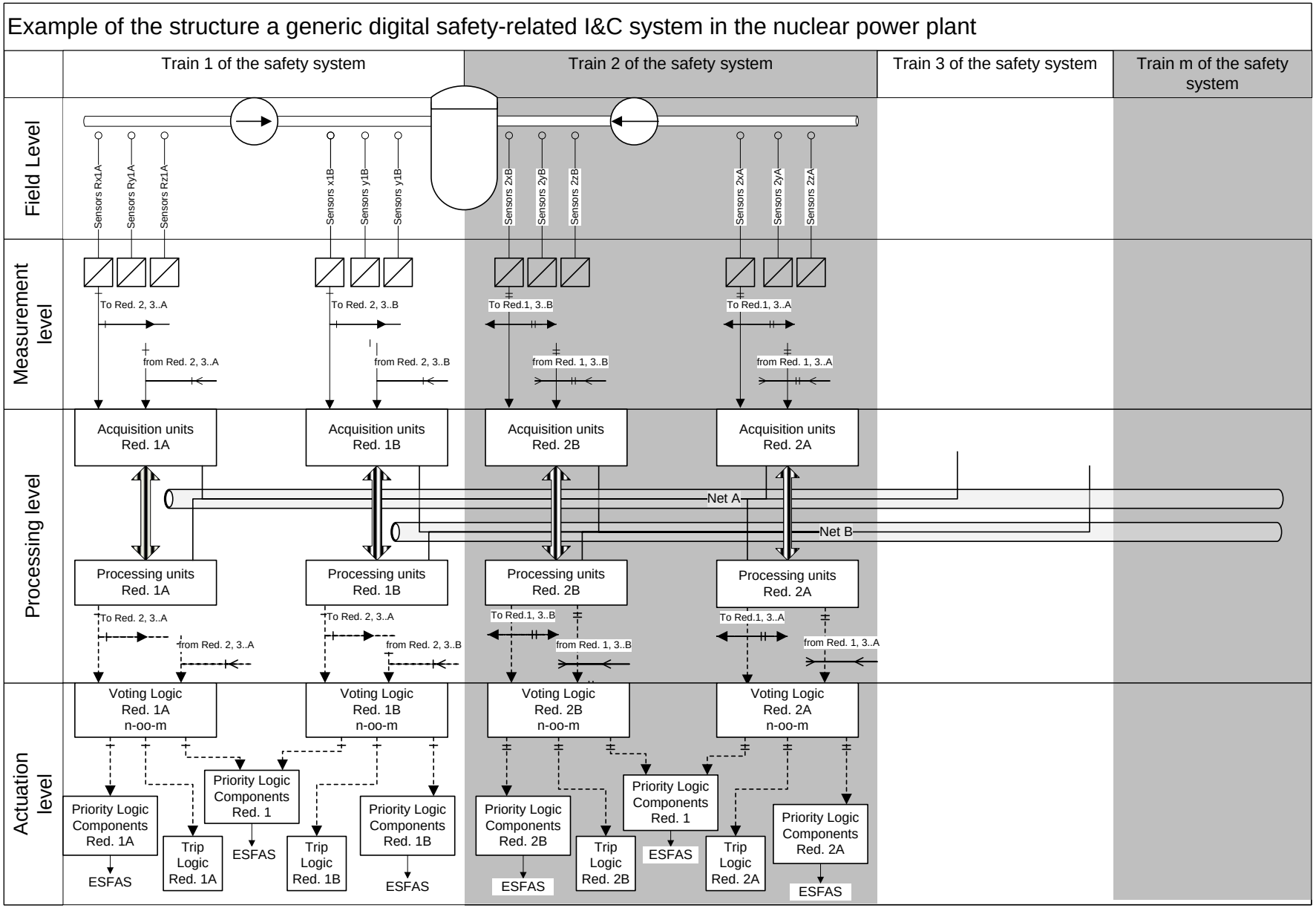
- Phase 1 – Definition of a generic DIC model:
  - draw up a flexible generic architecture of a simplified model of a typical digital (safety or safety relevant?) I&C system
  - break down a simplified model of the generic DIC system into generic parts (components) of the hardware and of the software under consideration of
    - needs of PSA : e.g. modeling of the functional interactions between technological process and essential functions of the DIC, definition of basic events, modeling of internal and external dependencies of the DIC
    - Consensus of the contributors to the taxonomy of the DIC regarding level of detail of the generic DIC and content of the FM/FE inputs

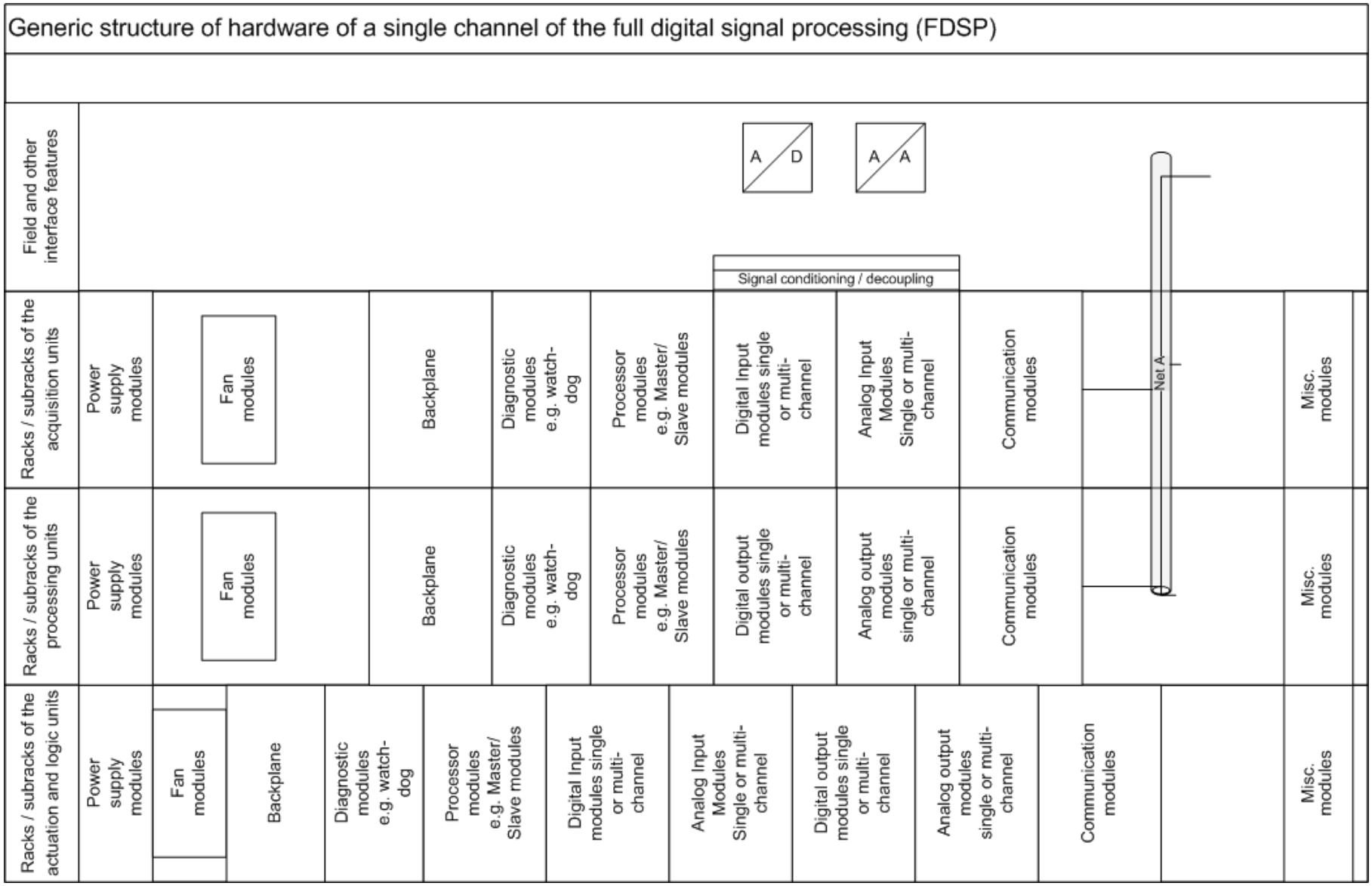
Proposal for the evaluation of the taxonomy approach (2)

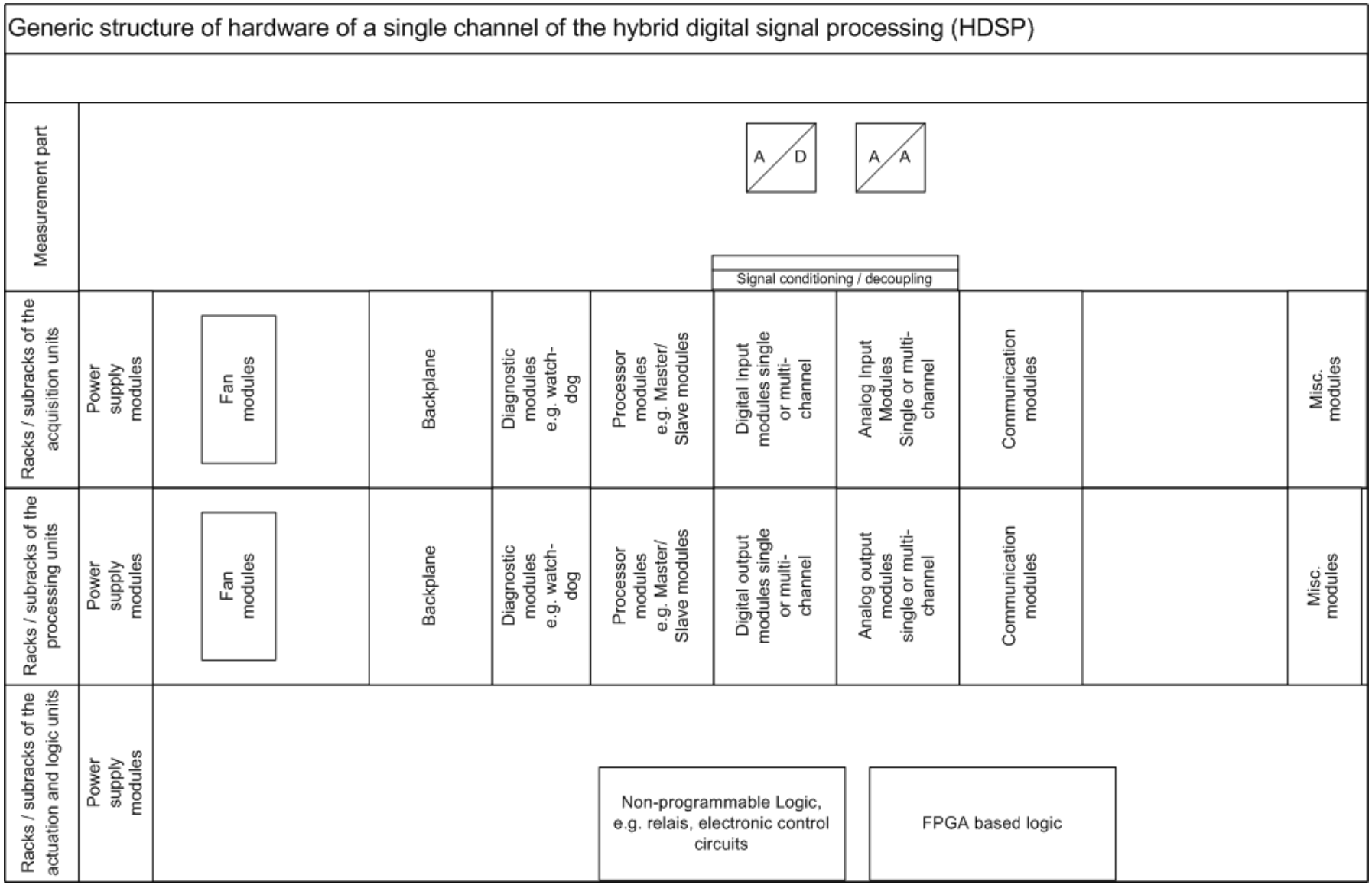
- Phase 2 – Evaluation of the FMEA issues and contribution to the taxonomy of a generic DIC model:
  - Decomposition of an individual I&C system according to
    - generic hardware structure,
    - generic software structure and
    - generic interfaces between them
  - Identification of generic issues on the basis of
    - results FMEA of the components (functions) of particular I&C systems,
    - operating experience,
    - PSA modeling,
    - manufacture data,
    - qualification,
    - tests e.c.t.

Definitions of the type of a generic digital I&C

Type of the process control system	Description
Distributed control system (DCS)	A DCS refers to a control system, in which the controller elements are not central in location but are distributed throughout the system with each component sub-system controlled by one or more controllers. The entire system of controllers is connected by networks for communication and monitoring.
Programmable logic controller (PLCs)	PLCs have a modular design so that they can be expanded to cover more aspects of process operation. PLCs can carry out a sequence of actions and incorporate single-loop controllers along with more advanced types of controller.
Single-loop controllers (SLCs)	simple on-off controllers are used to sequence, for example, valve movements and carry out other mechanical operations involved in process start up and shut down.
Supervisory control and data-acquisition systems (SCADA)	The term SCADA usually refers to centralized systems which monitor and control entire sites, or complexes of systems spread out over large areas


Example of the structure a generic digital safety-related I&C system in the nuclear power plant
Train 1 of the safety system
Train 2 of the safety system
Train 3 of the safety system
Train m of the safety system
Field Level
Measurement level
Processing level
Actuation level
Sensors Rx1A
Sensors Ry1A
Sensors Rz1A
Sensors x1B
Sensors y1B
Sensors y1B
Sensors 2xB
Sensors 2yB
Sensors 2zB
Sensors 2xA
Sensors 2yA
Sensors 2zA
To Red. 2, 3..A
from Red. 2, 3..A
To Red. 2, 3..B
from Red. 2, 3..B
To Red.1, 3..B
from Red. 1, 3..B
To Red.1, 3..A
from Red. 1, 3..A
Acquisition units Red. 1A
Acquisition units Red. 1B
Acquisition units Red. 2B
Acquisition units Red. 2A
Net A
Net B
Processing units Red. 1A
Processing units Red. 1B
Processing units Red. 2B
Processing units Red. 2A
Voting Logic Red. 1A n-oo-m
Voting Logic Red. 1B n-oo-m
Voting Logic Red. 2B n-oo-m
Voting Logic Red. 2A n-oo-m
Priority Logic Components Red. 1A
Priority Logic Components Red. 1
Priority Logic Components Red. 1B
Priority Logic Components Red. 2B
Priority Logic Components Red. 1
Priority Logic Components Red. 2A
Trip Logic Red. 1A
Trip Logic Red. 1B
Trip Logic Red. 2B
Trip Logic Red. 2A
ESFAS


Generic structure of hardware of a single channel of the full digital signal processing (FDSP)
Field and other interface features
Racks / subracks of the acquisition units
Racks / subracks of the processing units
Racks / subracks of the actuation and logic units
Power supply modules
Fan modules
Backplane
Diagnostic modules e.g. watch-dog
Processor modules e.g. Master/ Slave modules
Digital Input modules single or multi-channel
Analog Input Modules Single or multi-channel
Digital output modules single or multi-channel
Analog output modules single or multi-channel
Communication modules
Misc. modules
Signal conditioning / decoupling
A
D
A
A
Net A


Generic structure of hardware of a single channel of the hybrid digital signal processing (HDSP)
Measurement part
A D
A A
Signal conditioning / decoupling
Racks / subracks of the acquisition units
Power supply modules
Fan modules
Backplane
Diagnostic modules e.g. watch-dog
Processor modules e.g. Master/ Slave modules
Digital Input modules single or multi-channel
Analog Input Modules Single or multi-channel
Communication modules
Misc. modules
Racks / subracks of the processing units
Power supply modules
Fan modules
Backplane
Diagnostic modules e.g. watch-dog
Processor modules e.g. Master/ Slave modules
Digital output modules single or multi-channel
Analog output modules single or multi-channel
Communication modules
Misc. modules
Racks / subracks of the actuation and logic units
Power supply modules
Non-programmable Logic, e.g. relais, electronic control circuits
FPGA based logic

Proposal for identification of generic features of safety relevant communication (network) of the DIC (1)

- FMEA requires usually identification the all relevant pathways for the propagation of the initiating failures, therefore it is necessary to analyze the structure and the essential features of the internal and external communication of a digital I&C system.
- Topology and functionality of the network can affect fault propagation, failure detection and fault handling properties, therefore it should be consider by evaluation of generic issues of the FMEA.
- The network of the specific DIC can be configured as any one of several topologies the result being successful transmission of data from source to one or more receiver. A network of the generic DIC should consider at least three types of topology:
  - physical topology - the physical connections among the nodes,
  - signal topology - paths taken by the physical network signals among the nodes, and
  - logical topology - the flow of information between the nodes

Proposal for identification of generic feature of safety relevant communication of the DIC (2)

Level of signal processing	Internal/ inside of a channel Examples:	External 1 between redundant channel Examples:	External 2 To the other systems Examples:
Measurement	Hard-wired HART-link to data acquisition unit	Hard-wired No connections	Hard-wired No connections
Signal Acquisition	Hard-wired input Internal bus (backplane) Point-to-point	Point-to-point Ethernet bus No connections	Ethernet bus No connections
Signal Processing	Internal bus (backplane) Hard-wired output Point-to-point PROFIBUS link to the priority logic	No connections	No connections Monitoring & service interface of the system Ethernet-Gateway to plant bus
Actuation logic	Hard-wired Point-to-point PROFIBUS link from processing unit	No connections	No connections Hard-wired Ethernet-Gateway to plant bus

Concept for identification of generic FMs of hardware
Proposal of a FMEA Worksheet of the signal processing units

Hardware Components of the signal processing units (racks or subracks)	Failures of hardware modules FM	Failure effect of the function of the unit	Detection of the failure: e.g. on-line monitoring, test	Relevance regarding execution of the SW
Processor Modules				
Digital Input Modules				
Analog Input Modules				
Digital Output Modules				
Analog Output Modules				
Communication Modules				
Misc. Modules				
Rack (subrack) with built-in modules, e.g. backplane, self-diagnostic modules, watchdog power supply, fans				

Proposal for evaluation of impacts of the potential errors of software of the function of the affected hardware

Level of signal processing	Type of software	Allocation of the software	Potential errrors of software (FM)	Failure effect of the function of the HW	Detection of the failure: e.g. on-line monitoring, test
Measurement	none or firmware	none or internal memory	firmware integrity error	?	?
Signal Acquisition	executive & application software	memory of the main processor module, communication processor module	CRC checksum error	stop of the signal processing	e.g. on-line monitoring,
Signal Processing	executive & application software	memory of the main processor module, communication processor module	SW integrity error	stop of the signal processing	e.g. automatic integrity test, exception handler initiate a reset
Actuation logic	none or firmware or executive & application software	none or memory of the main processor module, communication processor module or embedded system (firmware)	firmware integrity error	?	?

Examples of the FMEA of the digital I&C system

- FMEA results derived from PSA Ringhals 1, RiskPilot (Sweden)

- FMEA results derived from PSA Study BWR 69, GRS (Germany)

Outlook

- Continuation of the development of the FMEA taxonomy:
  - Cross-check and correlation of the FMEA results of similar DIC systems from different sources regarding their application for the proposed generic DIC
  - Further completion of the FMEA worksheets with new inputs

Never give up!

Outline of the Guidelines

Presented By: Carol Smidts

WgRisk DIGREL

Helsinki, Finland

October 2011

DOE Academic Center of Excellence (ACE)

Instrumentation, Control and Safety

Proposed Outline

- Table of Contents
  - Executive Summary
  - Objective and Scope
  - Motivation
  - Uses of the Taxonomy within PRA
  - Definition of Terms
  - Approach and Assumptions
  - Taxonomies
  - Example System
    - Hardware Architecture
    - Software Architecture
  - Demonstration (?) of the Taxonomies Using the Example System
  - Possible Data Sources and Data Collection Needs
  - Open Issues- Limitations
  - User Guidelines
  - Conclusion and Recommendations
  - References
  - Appendix-Detailed Taxonomies