Digital I&C systems in probabilistic safety assessment (PSA)

Ola Bäckström[1], Jan-Erik Holmberg[2], Stefan Authén[2], Markus Porthin[3], Tero Tyrväinen[3]

[1] Lloyd's Register Consulting
[2] Risk Pilot AB
[3] VTT (Technical Research Centre of Finland)

Lloyd's Register Consulting

Working together
for a safer world

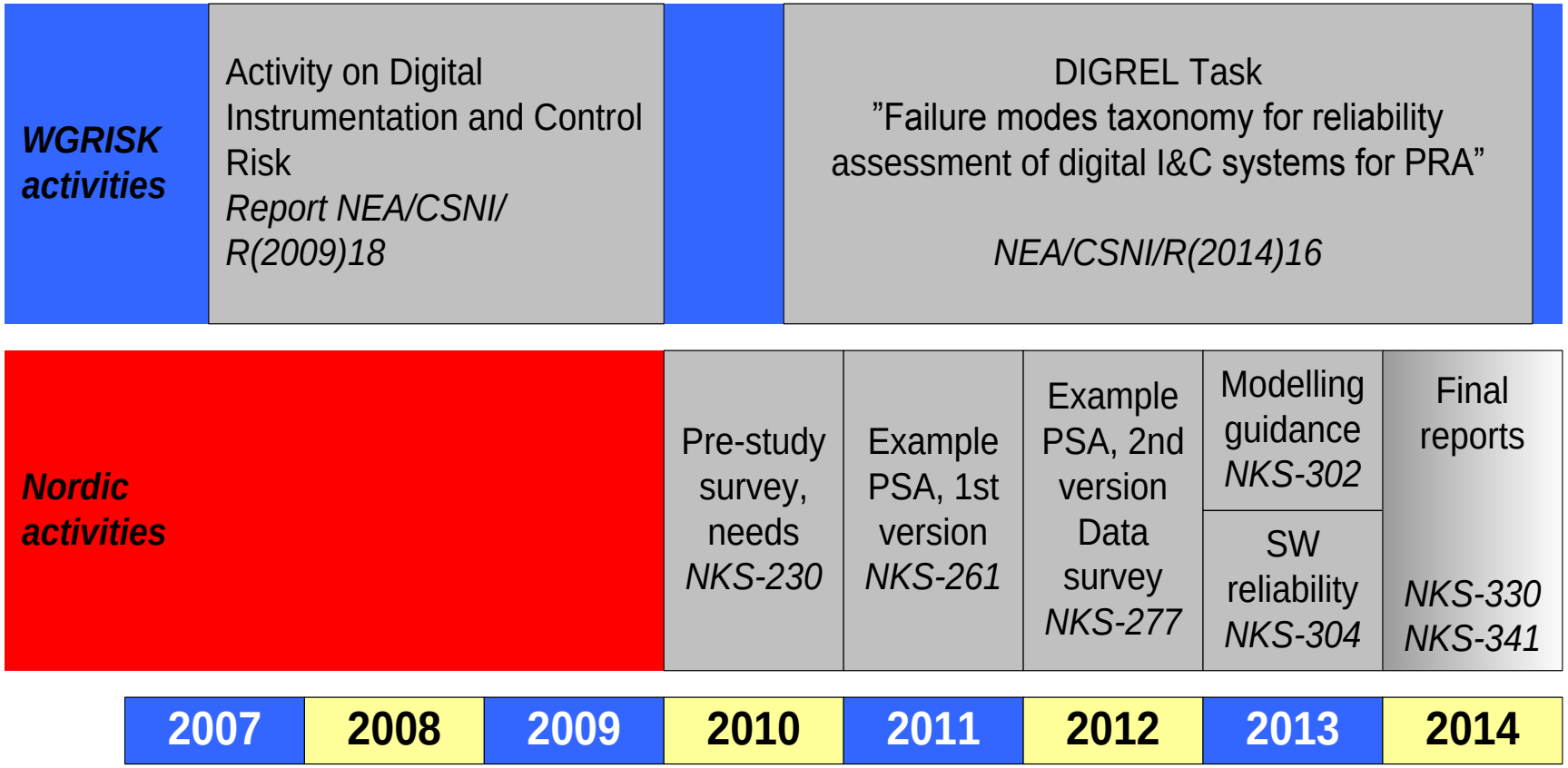
Contents

- Overview of DIGREL project
- Analysis of hardware
  - Hardware failure modes
- Analysis of software
  - Software failure modes
  - Suggested method for data analysis

Acknowledgements

- The work has been financed by NKS (Nordic nuclear safety research), SAFIR2014 (The Finnish Research Program on Nuclear Power Plant Safety 2011–2014) and the members of the Nordic PSA Group: Forsmark, OskarshamnKraftgrupp, RinghalsAB and Swedish Radiation Safety Authority
- DIGREL project partners: Risk Pilot AB, Lloyd's Register Consulting -EnergyAB, VTT (Technical Research Centre of Finland)
- AREVA GmbH has contributed to the software reliability quantification task
- Siemens AG has participated in the discussion on software reliability quantification
- Failure modes taxonomy has been developed by a task group of OECD/NEA WGRISK
- NKS conveys its gratitude to all organizations and persons who by means of financial support or contributions in kind have made the work presented in this report possible

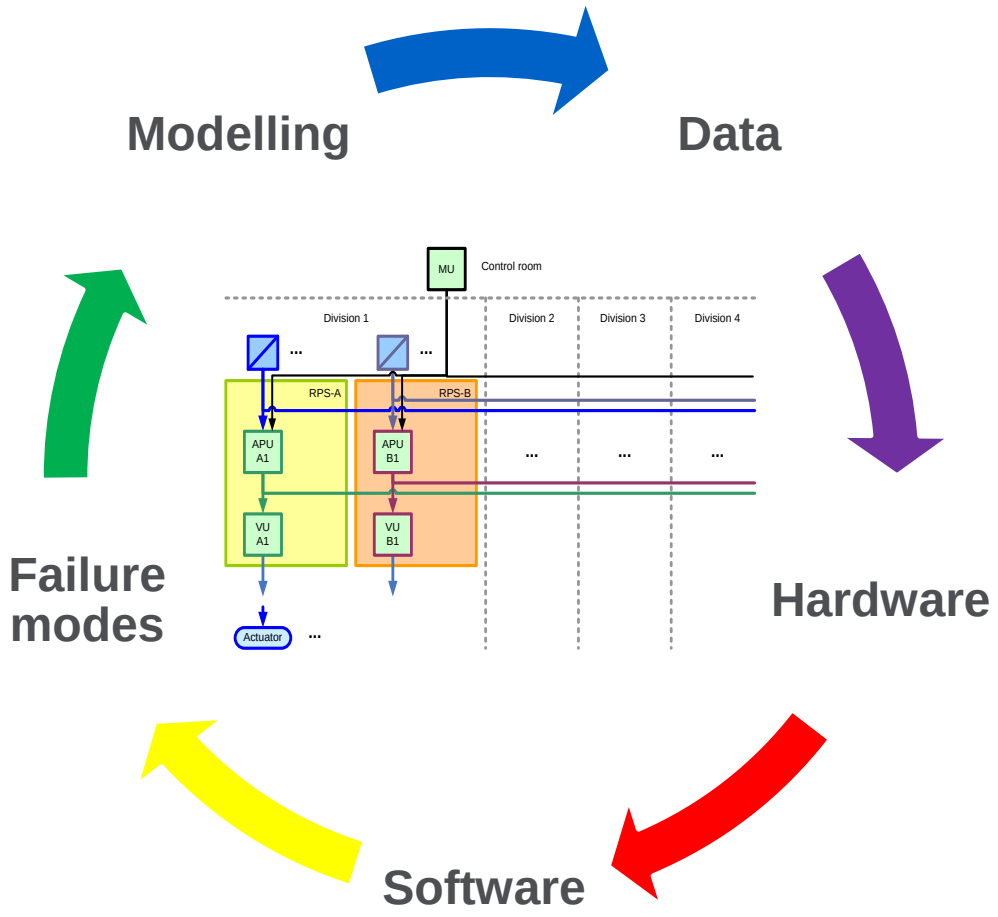
DIGREL activities



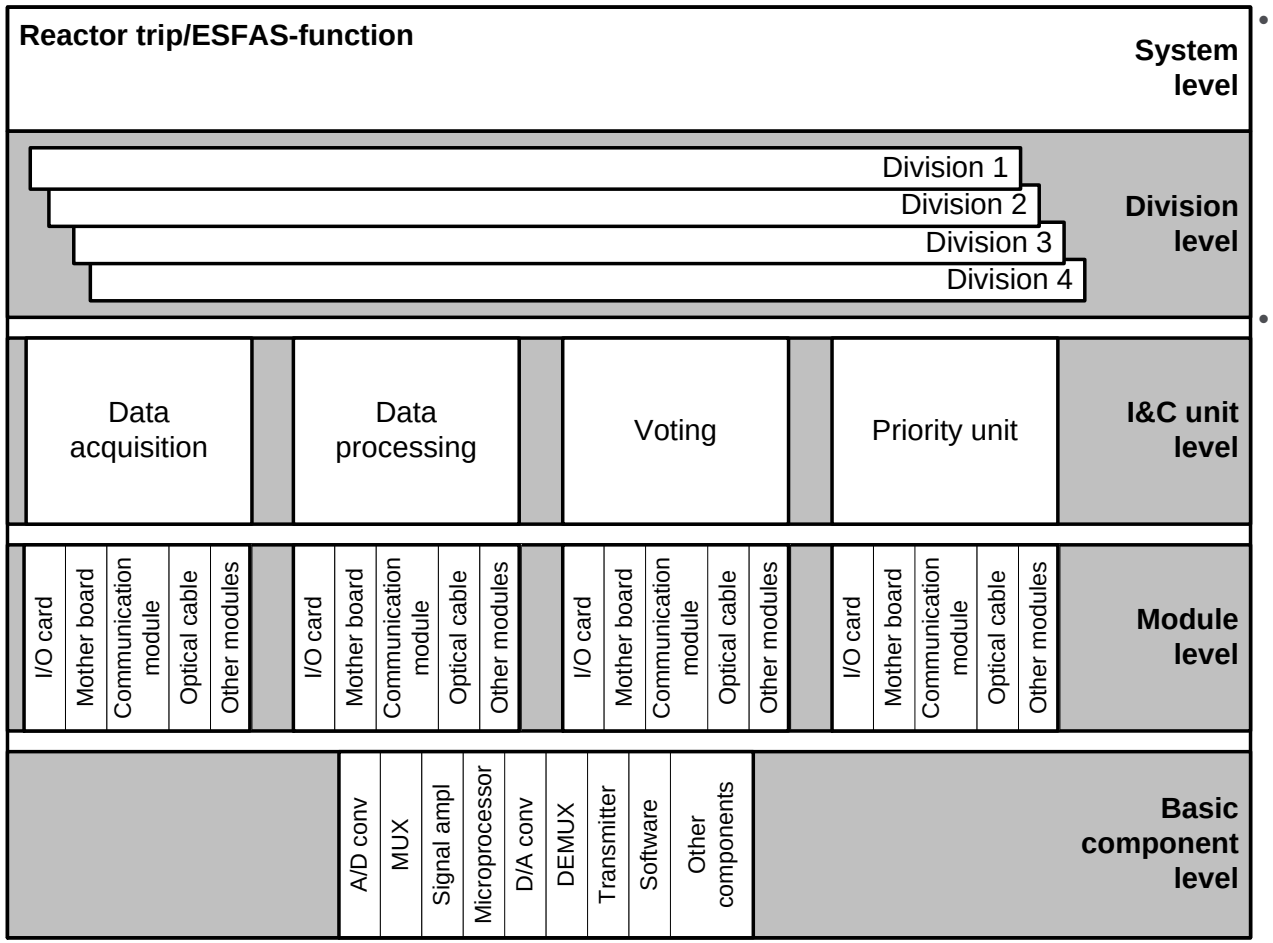
- Reports can be loaded from www.nks.org and http://www.oecd-nea.org/nsd/docs/indexcsni.html

DIGREL project objectives

- The objective with the project is to provide guidelines to analyse and model digital systems in PSA context, including
  - a taxonomy of hardware and software failure modes of digital components for common use (part of the international OECD/NEA Working Group RISK task)
  - guidelines for failure modes analysis and fault tree modelling of digital I&C
  - an approach for modelling and quantification of software



Principal structuring of safety I&C into different levels of details Hardware failure mode taxonomy



- System / Division level
  - Failure to actuate
  - Spurious actuation
- I&C Unit / Module level
  - Fault location
  - Failure effect
    - Fatal (ordered, haphazard)
    - Non-fatal (plausible, non-plausible)
  - Uncovery situation
    - Online, offline, demand, spurious

Hardware failures, example for module level

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

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

Example of application of taxonomy

Hardware module	Failure mode examples	Failure effect	Uncovering situation	Functional impact on I&C units
Processor module	Hang	Fatal, ordered	Online Detection	Loss of APU or VU functions (all)
	Communication dropout	Non-fatal, implausible	Online Detection	Loss of APU or VU functions (all)
	Delayed signal	Non-fatal, plausible	Latent revealed by demand	Loss of APU or VU functions (all)
	Random behaviour	Non-fatal, plausible	Latent revealed by demand	Loss of APU or VU functions (all)
		Non-fatal, implausible	Online Detection	Loss of APU or VU functions (all)
			Spurious effect	Spurious APU/VU function(s)

Hardware reliability data

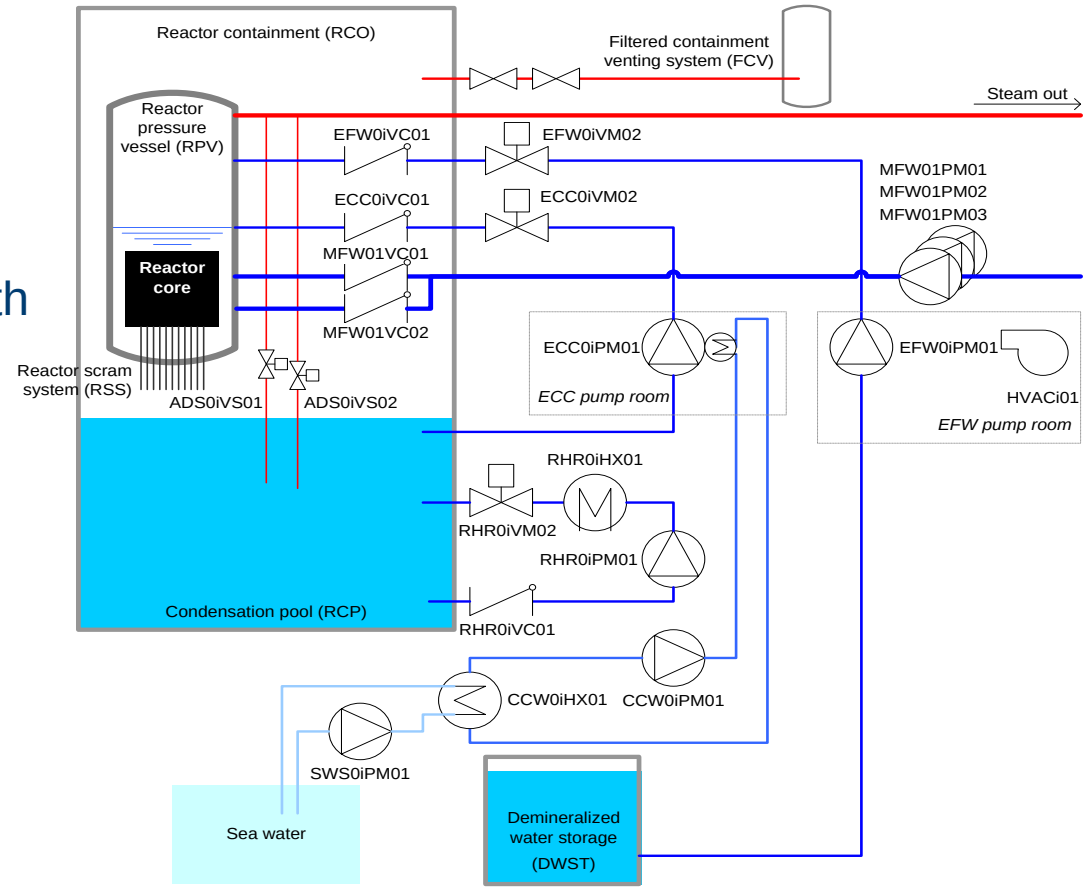
- Vendors usually provide
  - Operating experience (preferable)
  - Part counting method
    - Siemens SN 29500
  - Generic data bases
    - Military handbook MIL-HDBK-217 (outdated)

- Detected vs. undetected failures: fractions depend on failure detection features
  - Vendors provide usually

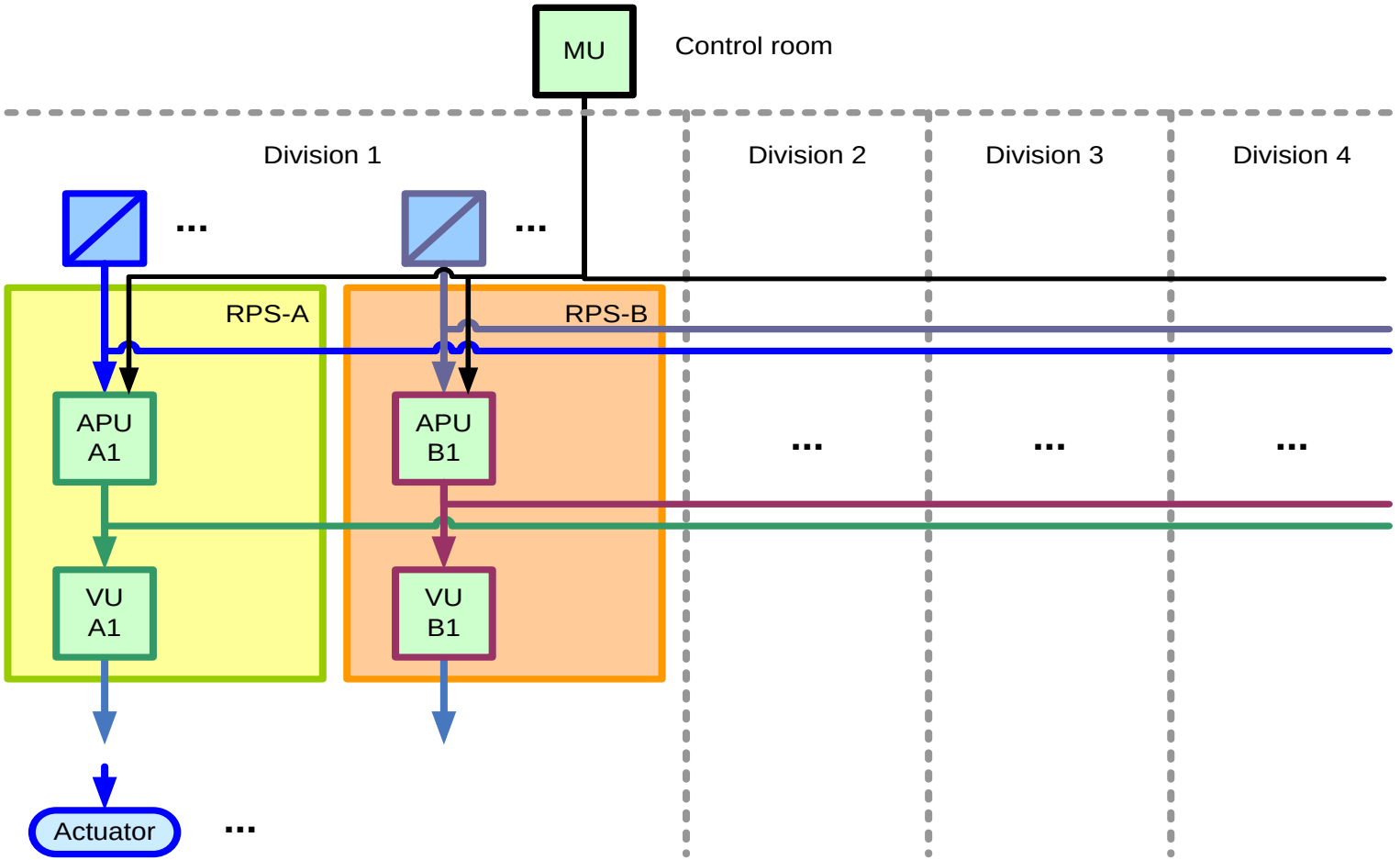
- CCF parameters
  - Generic values
  - IEC 61508-6

Example nuclear power plant

- Fictive boiling water reactor
- The protection system has two subsystems
- Diversification of safety functions the whole path from sensors to actuators
  - E.g. the emergency core cooling system (ECC) is controlled by RPS A and the emergency feedwater system (EFW) is controlled by RPS B
- The protection system is designed with fault tolerant features

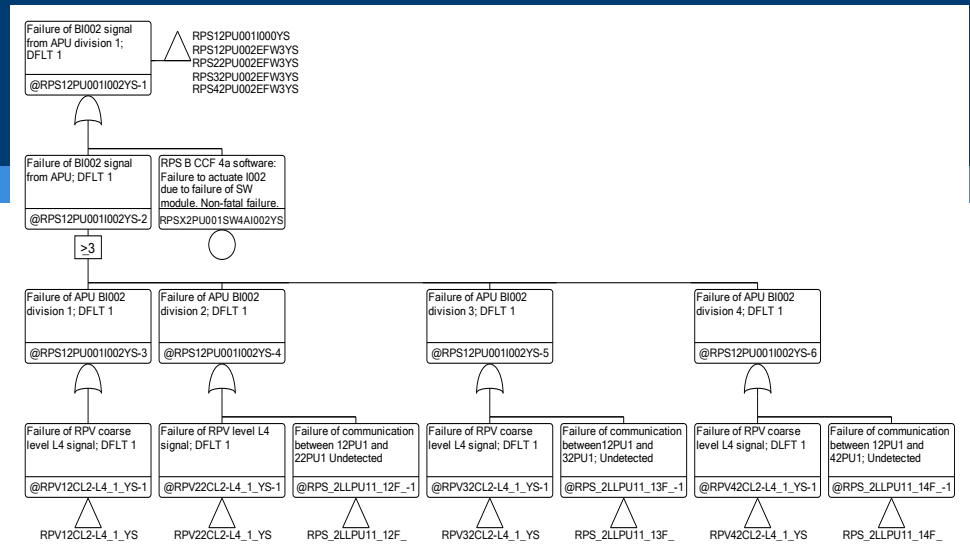


Example I&C architecture



Lessons learnt – modelling

- Example PSA model was used and needed for experiments



- Choice of level of detail for the modelling was of high importance for the result
  - Module level is recommended
- Both undetected (latent) and detected failures contributed significantly for all cases (default values and intelligent voting logic)
- The risk contribution from Digital I&C was mainly due to CCF events
- In order to develop a realistic fault tree model for a digital I&C protection system it is vital to correctly represent the chosen fault tolerant design
- SW faults have a non-negligible effect on the results due to their functional impact on all divisions — one or more safety functions can be lost

Software - definitions

- System software (SyS)
  - operating system and runtime environment (interaction between application and operating system). System software is plant independent.
- Elementary functions (EFs).
  - Modules provide readily useable standard (library) functions.
  - Elementary functions can be also called Library Functions or Function(al) Blocks. Elementary functions are plant independent.
  - An important difference with respect to the SyS is that a specific I&C unit will use only a specific subset of all available EFs
- Application software (AS)
  - Software modules which are executed by the operating system during an operating cycle of the processing module. These modules implement specific I&C functions in I&C units. AS modules are plant-specific and are constructed using elementary function modules.

Definitions, cont´d

- Data communication units have the following software modules:
  - System software (SyS), which is the same as the SyS of the APUs and VUs.
  - Data communication software (DCS) which implements the data communication protocol. It is part of the platform software, and is plant independent.
  - Data link configuration (DLC) which specifies the nodes that can be part of a given network (subsystem), and the data messages that can be exchanged between the nodes of the network. DLC is plant-specific.
- In addition, there are specific pieces of software present in other hardware modules than processor modules. These are called here as Proprietary SW.

Software

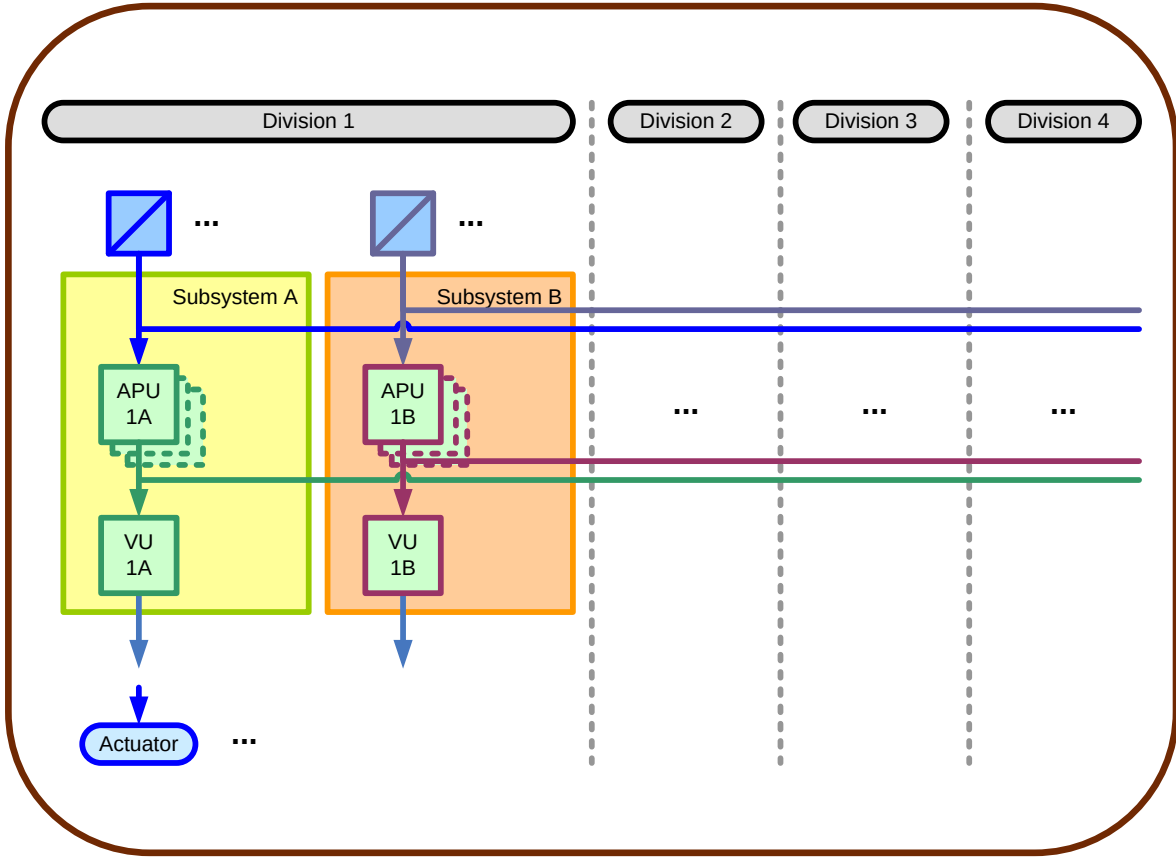
- A method for quantification of (mainly reactor protection system (RPS)) software failures in nuclear PSA context
- The purpose is to define a simple yet sufficient model
  - To describe the software failure impacts (a main objective)
  - To provide a quantification model

Effects	Definition of effects	Software fault location					
		SyS	APU-FRS	APU-AS	VU-FRS	VU-AS[7]	DCS
SYSTEM	Loss of complete system	case 1					case 1
1SS	Loss of one subsystem	case 2a	case 2a		case 2a	case 2a	case 2b
1APU-1SS	Loss of one group of redundant APU in one subsystem		case 3a	case 3a			
1VU-1SS	Loss of one group of redundant voters in one subsystem				case 3b	case 3b	
1AF-1SS	Loss of one function in all divisions of one subsystem		case 4a	case 4a	case 4b	case 4b	
1AF-1D-1SS	Loss of one function in one division of one subsystem		case 4c	case 4c			

Fatal failure
Fatal or Non-fatal

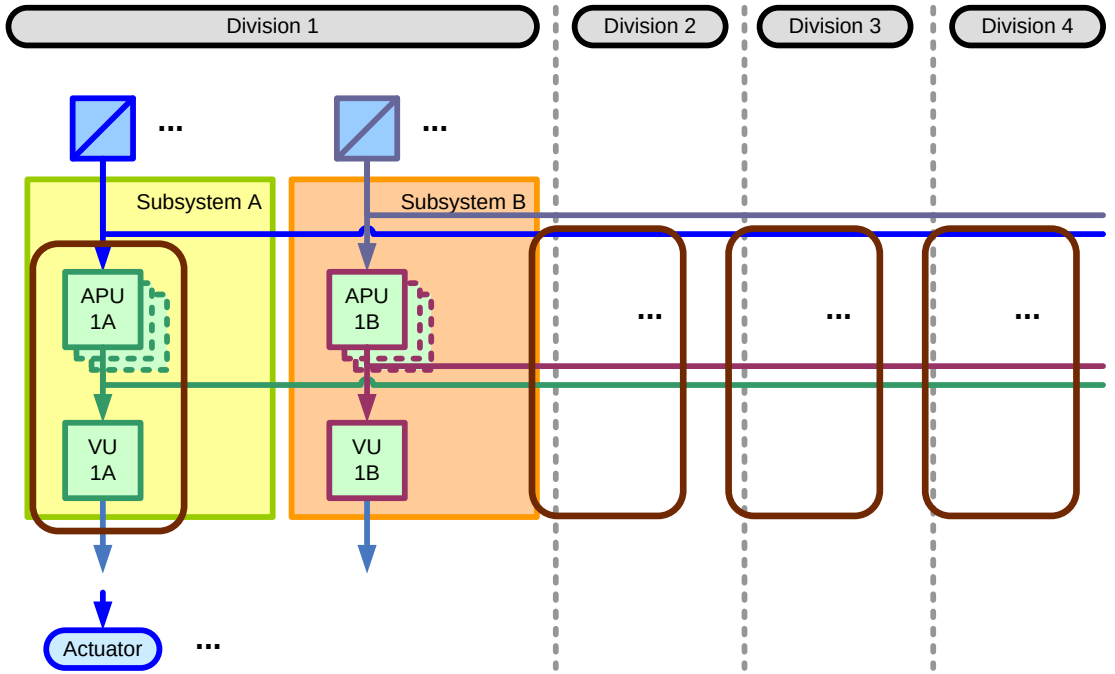
Software fault cases, case 1

Effects	Software fault location					
	SyS	APU-FRS	APU-AS	VU-FRS	VU-AS[7]	DCS
SYSTEM	case 1					case 1
1SS	case 2a	case 2a		case 2a	case 2a	case 2b
1APU-1SS		case 3a	case 3a			
1VU-1SS				case 3b	case 3b	
1AF-1SS		case 4a	case 4a	case 4b	case 4b	
1AF-1D-1SS		case 4c	case 4c			



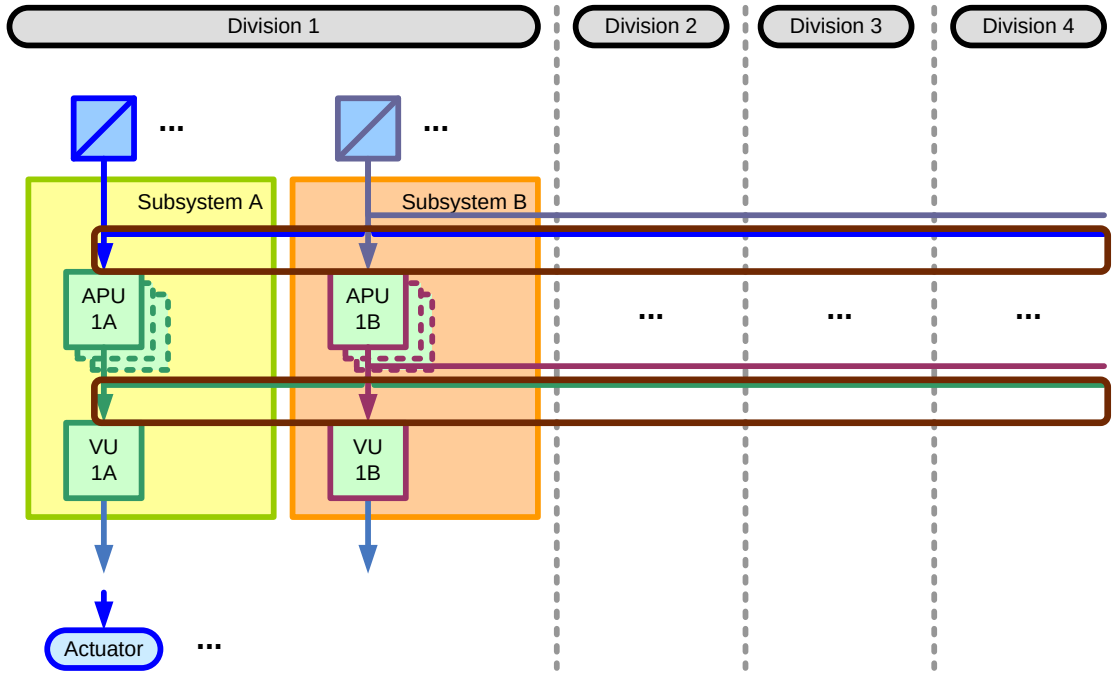
Software fault cases, case 2a

Effects	Software fault location					
	SyS	APU-FRS	APU-AS	VU-FRS	VU-AS[7]	DCS
SYSTEM	case 1					case 1
1SS	case 2a	case 2a		case 2a	case 2a	case 2b
1APU-1SS		case 3a	case 3a			
1VU-1SS				case 3b	case 3b	
1AF-1SS		case 4a	case 4a	case 4b	case 4b	
1AF-1D-1SS		case 4c	case 4c			



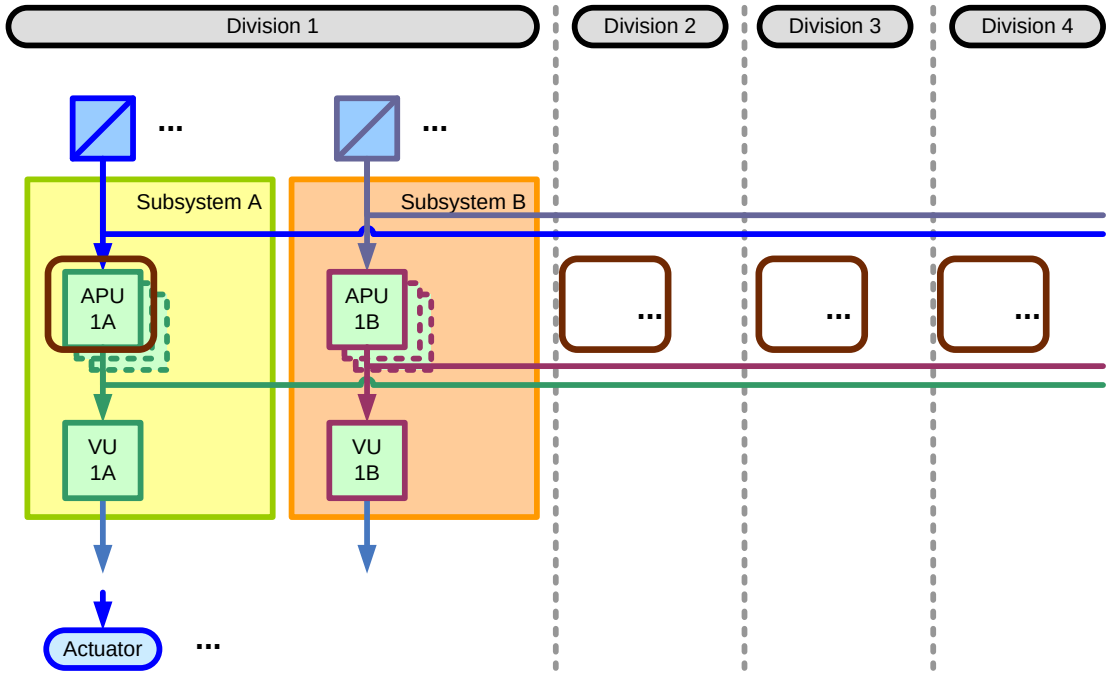
Software fault cases, case 2b

Effects	Software fault location					
	SyS	APU-FRS	APU-AS	VU-FRS	VU-AS[7]	DCS
SYSTEM	case 1					case 1
1SS	case 2a	case 2a		case 2a	case 2a	case 2b
1APU-1SS		case 3a	case 3a			
1VU-1SS				case 3b	case 3b	
1AF-1SS		case 4a	case 4a	case 4b	case 4b	
1AF-1D-1SS		case 4c	case 4c			



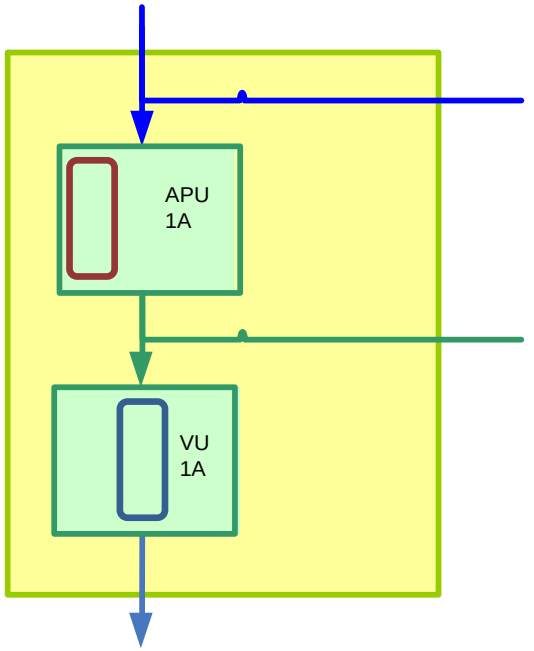
Software fault cases, case 3a (similar for 3b)

Effects	Software fault location					
	SyS	APU-FRS	APU-AS	VU-FRS	VU-AS[7]	DCS
SYSTEM	case 1					case 1
1SS	case 2a	case 2a		case 2a	case 2a	case 2b
1APU-1SS		case 3a	case 3a			
1VU-1SS				case 3b	case 3b	
1AF-1SS		case 4a	case 4a	case 4b	case 4b	
1AF-1D-1SS		case 4c	case 4c			



Software fault cases, case 4a and 4b (corresponding for 4c)

Effects	Software fault location					
	SyS	APU-FRS	APU-AS	VU-FRS	VU-AS[7]	DCS
SYSTEM	case 1					case 1
1SS	case 2a	case 2a		case 2a	case 2a	case 2b
1APU-1SS		case 3a	case 3a			
1VU-1SS				case 3b	case 3b	
1AF-1SS		case 4a	case 4a	case 4b	case 4b	
1AF-1D-1SS		case 4c	case 4c			



Outline of method

- **SyS, System software** (case 1 and 2a)
  - Fatal failures assumed
  - SyS should be preferably be evaluated based on operational experience.

- **DCS, DLC** (case 2b)
  - Needs to be treated separately, since the effect of failure in communication may be unique.
  - Preferably estimated based on operating experience.

- **AS, Application software** (case 3 and 4)
  - Fatal and non-fatal failures
  - Matrix of complexity and Verification & Validation used to adjust failure probability

- **EF, Elementary functions**
  - Considered covered by AS (for non-fatal failures) and SyS (for fatal failures)
  - Not studied separately further

- **Proprietary SW**
  - Is part of hardware units and are considered part of the HW faults.

SyS failure data estimated using operating experience

- Example, based on TXS experience

CCF triggering mechanism	Latent fault location			Failures in operation	Accumulated operation time [h]	Failure rate [1/h]	Event duration [h]	Failure probability(2)
	SyS	AS	DCS					
Temporal effects	x			0	6.5E+6	7.8E-8	-	1.9E-6
Faulty telegrams	x		x	3	6.5E+6	5.4E-7	0.25	1.3E-5
Same signal trajectory	x	x		0	6.4E+7	7.8E-9	-	1.9e-7

- **P(1SS-SyS fatal, case 2a)** = { Temporal effects } =
  P(case 2a) ~ $\lambda_{2a} \cdot T_m$ ~ **2E-6**

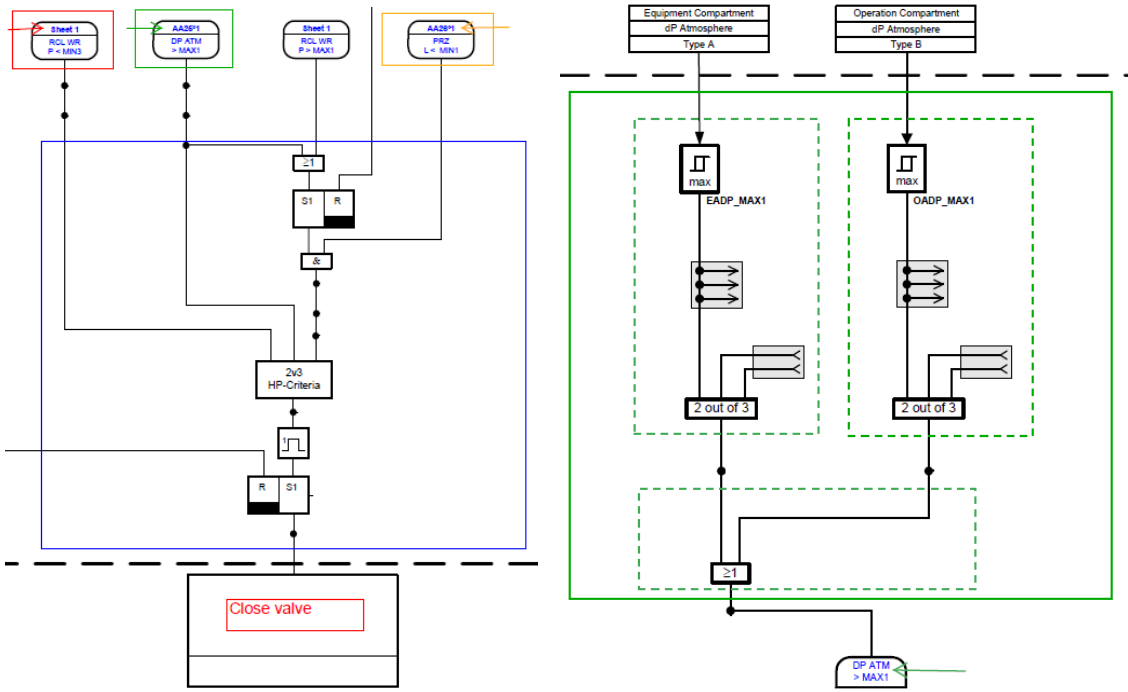
- **P(1SS-DCU fatal, case 2b)** = { Faulty telegrams } =
  P(case 2b) ~ $\lambda_{2b} \cdot T_m$ ~ **1E-5**

- **P(System SyS fatal, case 1)** = { Same signal trajectories, and CCF assumption } =
  P(case 3) $\cdot \beta = \lambda_3 \cdot T_m \cdot \beta$ ~ **2E-9**

  - If sufficient diversity of the application software is ensured in the design, then the correlation of both subsystems is very weak and can be modelled by a small correlation factor (e.g. 1%)

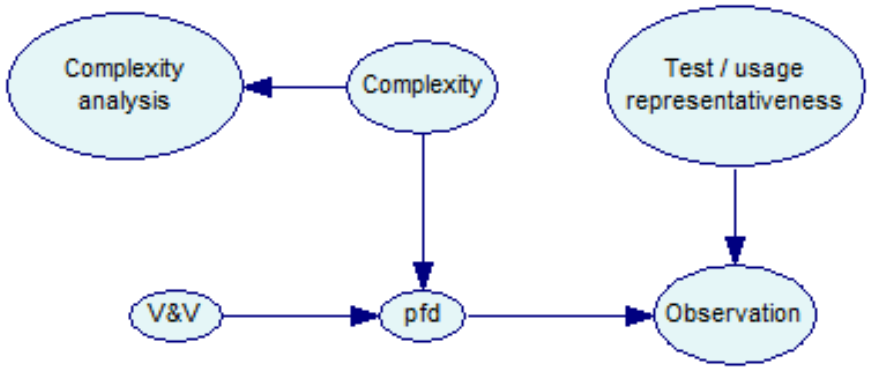
Application Software (AS) module vs Application Software

- An Application Software module is defined as:
  - An AS module corresponds to a function dedicated to a specific task. Depending on the specific case the application software can be represented by one or more AS modules
- The rule is that an AS module shall be used in its entirety.
  - If parts of the Logic diagram is used as inputs to several functions, then the AS module shall be split
- AS modules are defined at least per APU/VU



Application software module reliability

- The software failure probability for an application software module is hard to estimate.
- The proposed method is based on a BBN approach



- It is far from straightforward how the update based on operational data should be done.

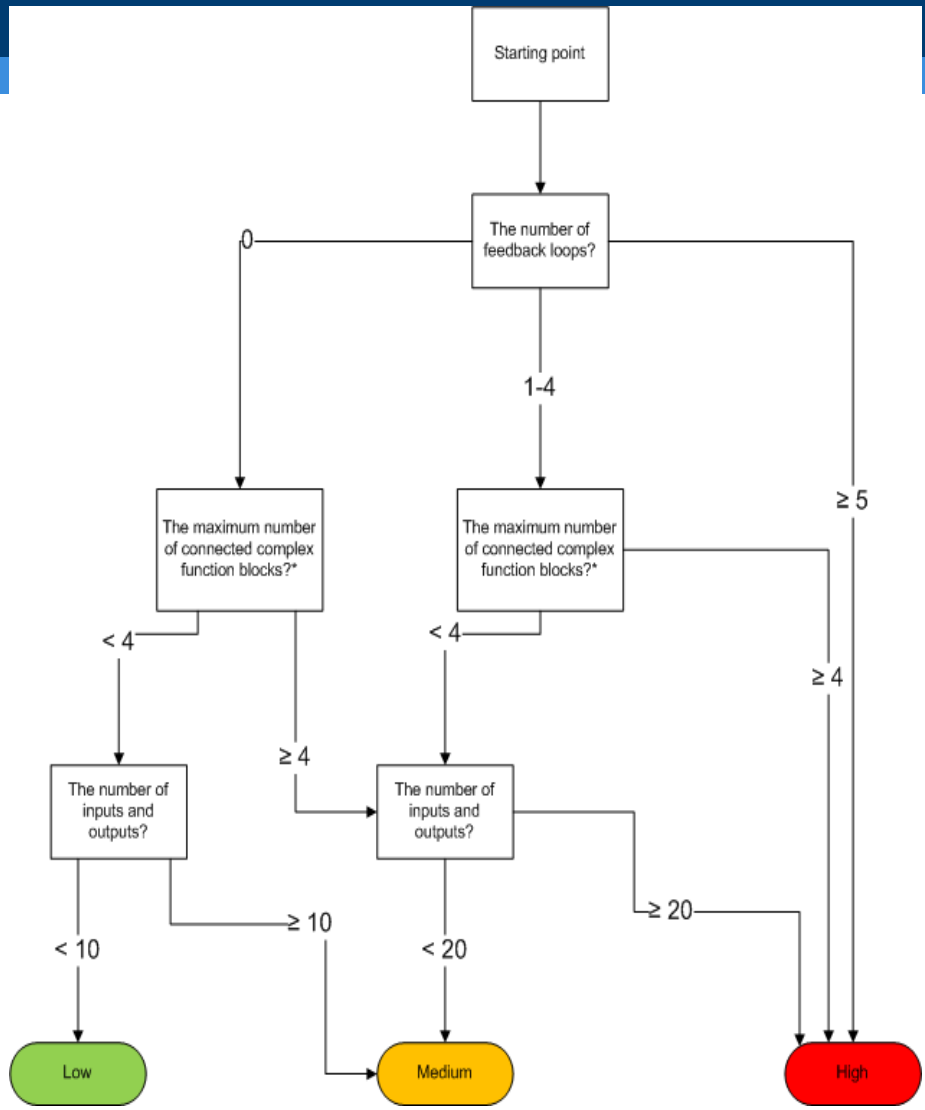
Application software module

- Estimation of application software reliability
  - Based on a matrix of Complexity and Verification and Validation
  - Based on ”application software modules”
  - Prior based on
    - E[PNSAl F] = 1E-6 * F

		Complexity		
		High	Medium	Low
V&V	0	10000	1000	100
	1	1000	100	10
	2	100	10	1
	3	10	1	0.1
	4	1	0.1	0.01

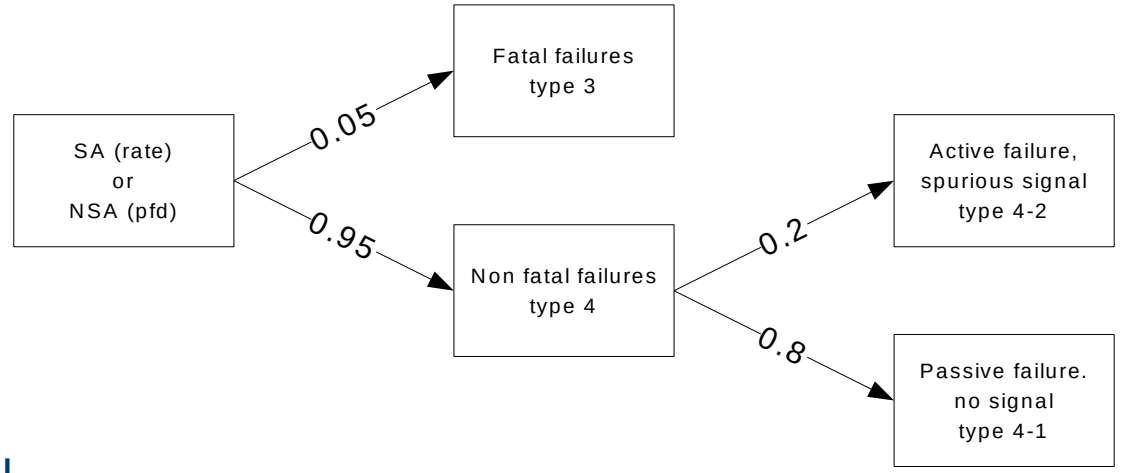
Complexity analysis of application software

- No unambiguously correct way to define complexity
- Classification to low, medium and high
  - The lines between categories not clear
- ISTec's method
  - Complexity metric calculated based on 9 indicators
  - Complicated to calculate by hand - automated on TXS
- SICA
  - Decision rules to categorise modules
  - Visual assessment
- More research is still needed before we can confidently claim what is the best way to define the complexity

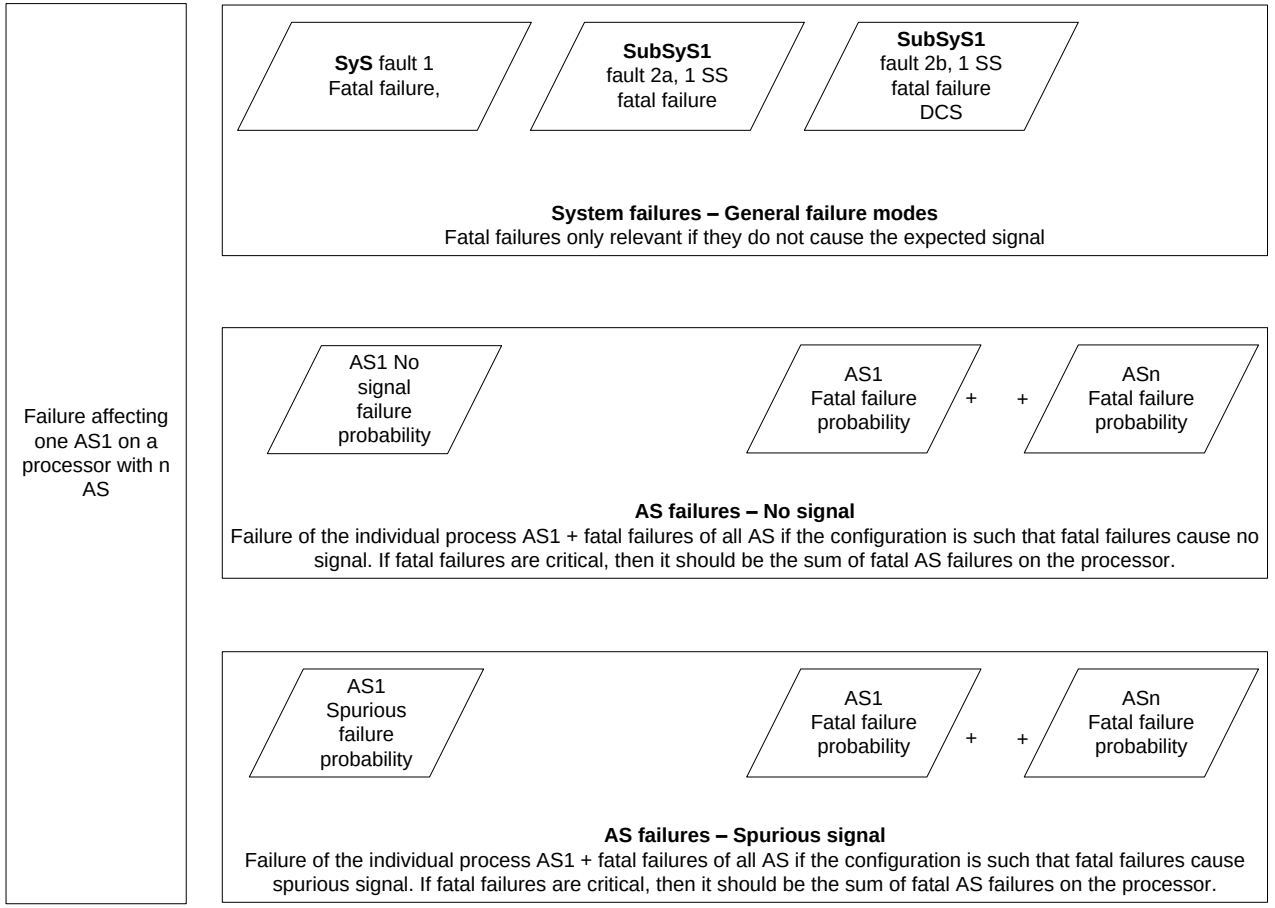


AS estimation methods

- Estimate based on failure detection method
  - Estimate of AS module failures
  - Based on tentative fractions estimated by Areva
- Estimate based on failure mode
  - Fatal failures estimated on processor level
  - Non fatal failures estimated as above



Overview of the potential software failures relevant to represent



Software reliability: Lessons learnt

- It is reasonable to define different types and failure modes representing software failures
- Separation of failure modes makes it possible to discuss CCF effects in a reasonable way
- There is operational experience available
- Failure data for system CCF failure modes should be straight forward to estimate
- Estimation of failure data for application software modules
  - Requires engineering judgement (prior)
  - Pooling of operational experience for software modules is not straight forward
  - Bayesian update is possible, but will require substantial amount of operational experience to have impact

Lessons learnt & way forward — Failure data

- Digital I&C vendors are a good data source
  - Many plants needed in order to get enough data
  - However, there may be issues with confidentiality and conflicts of interest
- For application software with low demand frequency (such as RPS), there was not enough data to justify low failure frequencies, unless pooling data with high demand frequency systems
- A new activity proposal for OECD/NEA WGRISK on Digital I&C has been prepared by the project group and GRS (Germany)
  - Focus: Diversity, CCF, collection and application of operating experience data

Conclusions

- An approach to analyse the reliability of digital I&C
  - Hardware failures
    - Taxonomy
    - Suggested level of detail
  - Software failures
    - Taxonomy
    - Suggested failure effects
    - Suggested approach to estimate data

- Important to continue international collaboration to reach consensus
- Joint workshop of NKS projects MODIG and PLANS September 29-30, 2015 at VTT, Espoo

Ola Bäckström
LR Consulting Energy – Sweden AB
T +46 70 742 13 93 E ola.backstrom@lr.org

Lloyd's Register Consulting
www.lr.org/consulting, www.riskspectrum.com

Lloyd's Register Consulting

Working together
for a safer world