In Pursuit of Robust FMEA in the Design Phase

Capella Days 2023 – Session 3

Alice Cellamare (p2m berlin) and Steven Huang (ManTech)

Introduction

Steven Huang (ESEP)

- ManTech International
- Engineering Fellow, Intelligent Systems Engineering

Alice Cellamare

- p2m berlin
- Process and site engineer, systems modeller

Cooperation & Collaboration as part of the INCOSE Mentoring Program

Outline

- Safety assessment methods
- Known tools for Capella
- Project scope and workflow
- Developed methods
- Example
- Conclusions

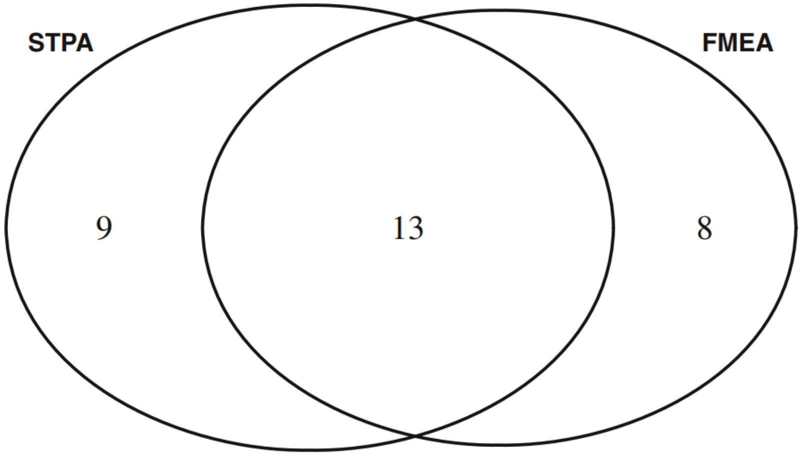
This symbol
means a
poll is coming!

✹

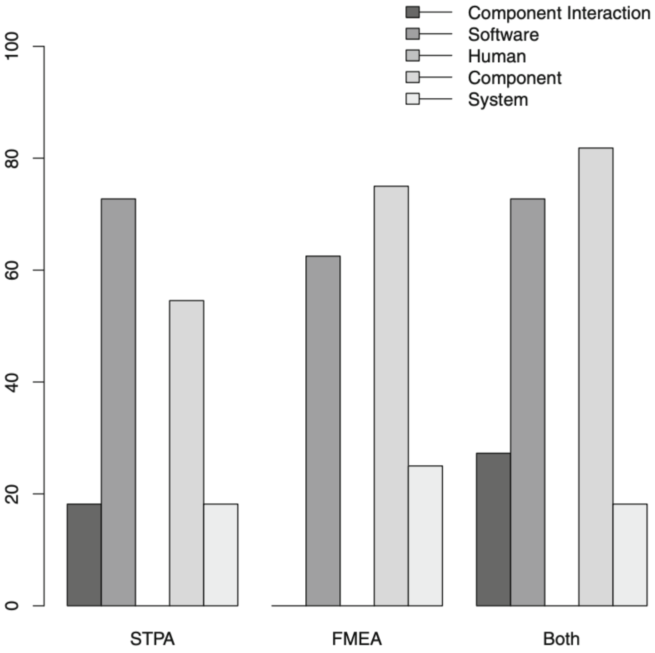
Safety Assessment Methods

Method	Description	Modeling Elements considered	Approach
FHA (Fault Hazard Analysis)	Evaluates functions to identify and classify potential failures	Functions	top-down
FTA (Fault Tree Analysis)	Deductive analysis focusing on causal relationships, using Boolean logic	Functions and their relationships	top-down
STPA (Systems Theoretic Process Analysis)	A systems approach focusing on unsafe control actions	Control actions	top-down
FMEA (Failure Mode and Effects Analysis)	A multi-perspective approach with focus	Components, subsystems, functions, processes	bottom-up

What about completeness?



Number of common and distinct hazards identified by FMEA and STPA



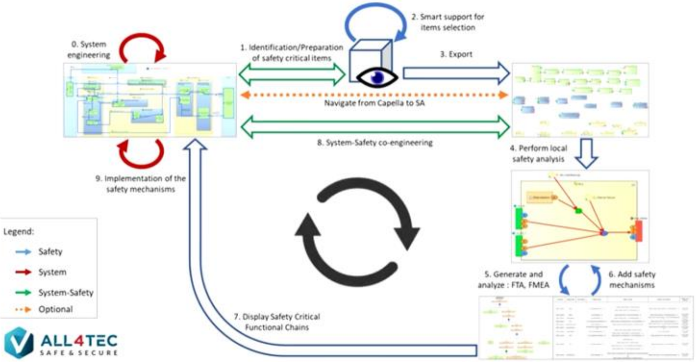
Classification of the identified hazards

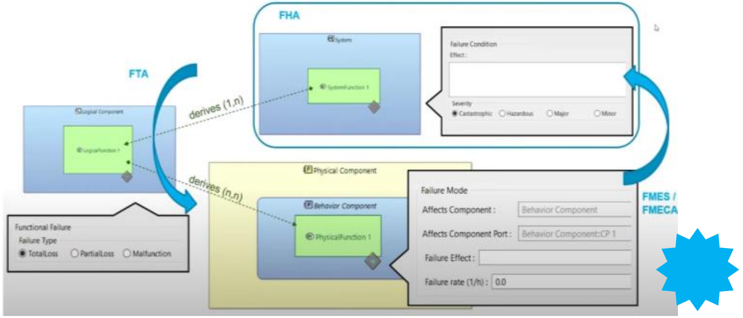
Reference [3]

Known tools for Capella

ManTech p2mberlin

Name	Implemented methods	ARCADIA diagrams
Safety Architect	FHA FTA FMEA	[SAB] →“SFBD“ [PAB] [PAB]
ATICA4 CAPELLA	FHA FTA FMEA	[SFBD], [SAB] [LAB] [PAB]


0. System engineering
1. Identification/Preparation of safety critical items
2. Smart support for items selection
3. Export
Navigate from Capella to SA
8. System-Safety co-engineering
4. Perform local safety analysis
9. Implementation of the safety mechanisms
Legend:
Safety
System
System-Safety
Optional
5. Generate and analyze : FTA, FMEA
6. Add safety mechanisms
7. Display Safety Critical Functional Chains
ALL4TEC
SAFE & SECURE



References [8] and [9]

Project scope

- FAILURE IDENTIFICATION (as opposed to EVALUATION) as priority: this method does not aim at substituting any safety analysis method, but rather at providing the modeller with a model-generated failure list at the start

- „QUANTITY APPROACH“: no qualitative criteria is applied in the failure identification phase. All failure which „the model can imagine“ are listed

- IGNORING CAUSES at first: this approach does not consider possible failure causes. These will be considered in the evaluation phase.

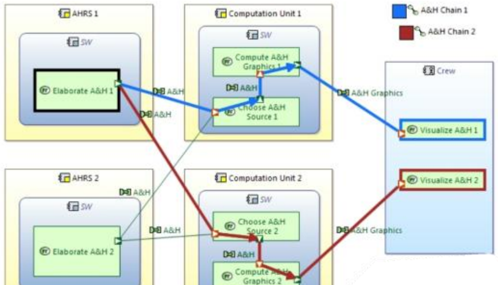
- Tailoring to PHYSICAL ARCHITECTURE: only this layer was considered

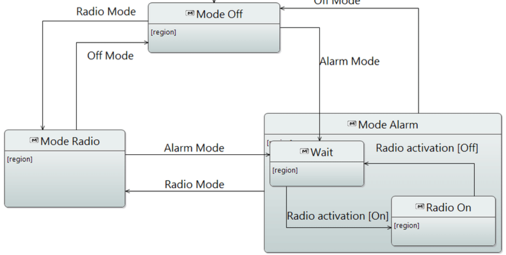
Relevant model elements

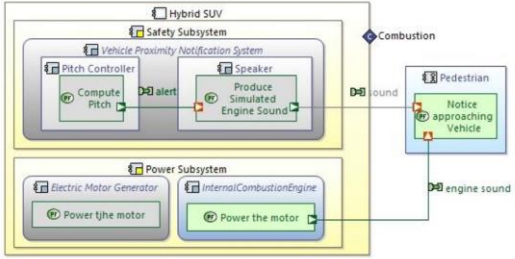
- Functional exchanges, functional chains, exchange scenarios

- Modes / States with triggers and entry/do/exit functions

- Configurations / Situations





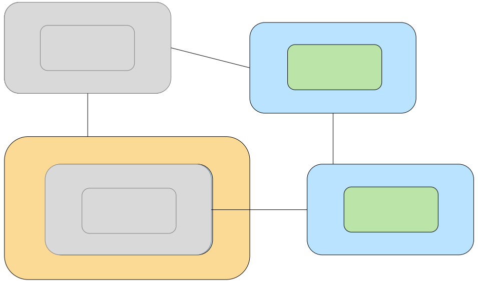


References [4], [5] and [6]

The VPMS Viewpoint

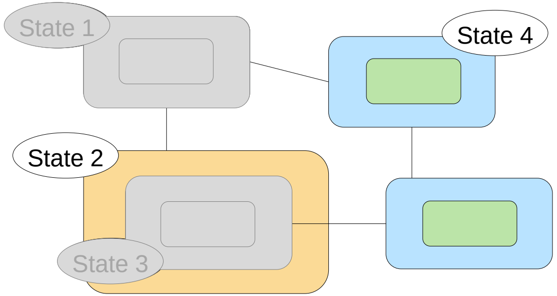
ManTech p2mberlin

Configuration



= subset of the system

Situation



= collection of states

Project activities

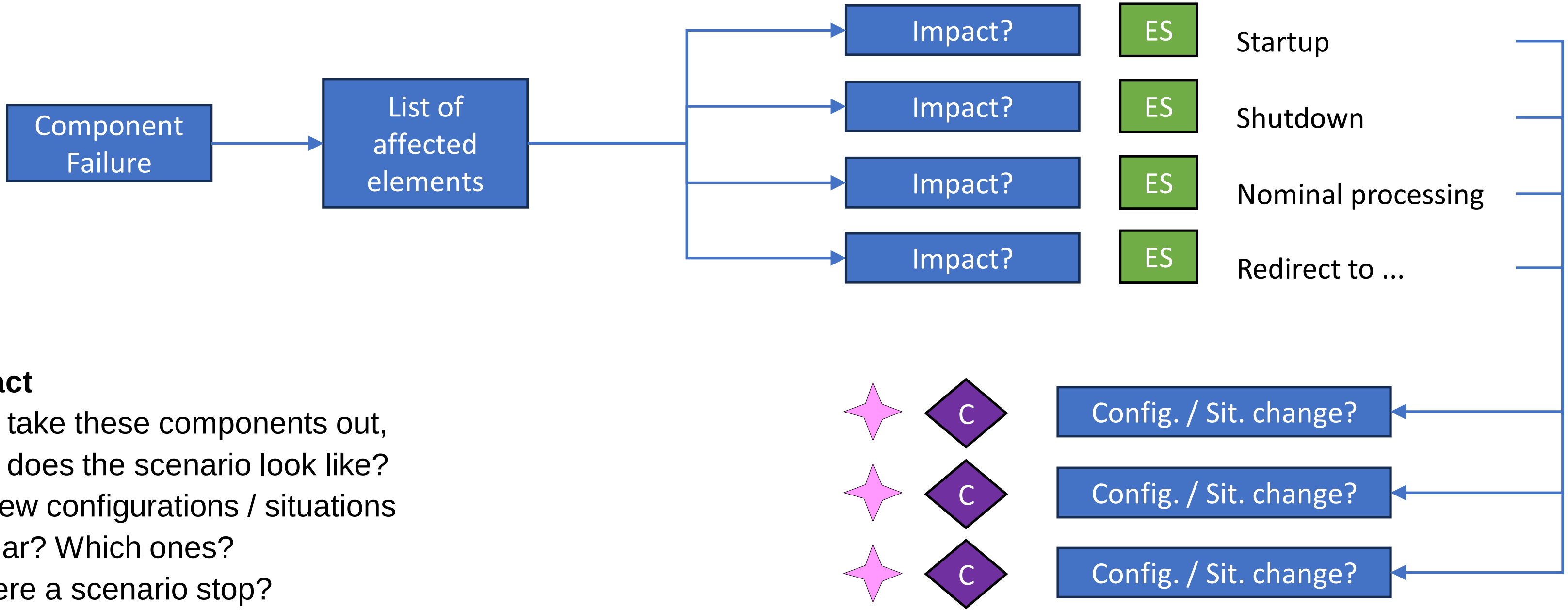
- Determine failure types
- Define ways of identifying failures (depending on type)
- Define ways of assessing the consequences of each failure
- Define ways of rendering the analysis results to a human evaluator
- Code this procedure in python (py-capellambse)
- Test the results on a company model

Failure Types

Name	Description	ARCADIA „translation“	Reference diagram (PA)
Component failure	A component becomes unavailable	A component, a function, a links or an exchange are not available	[PAB]
Function or process failure	A function or process becomes unavailable	A function or state/mode is realized at an unwanted time; An expected function or state/mode realization does not occur	[ES], [PFCD]
Content failure	A message, result or parameter becomes wrong	A component attribute is set to an unexpected value; An exchange item attribute is set to an unexpected value	[ES],

Component failure analysis

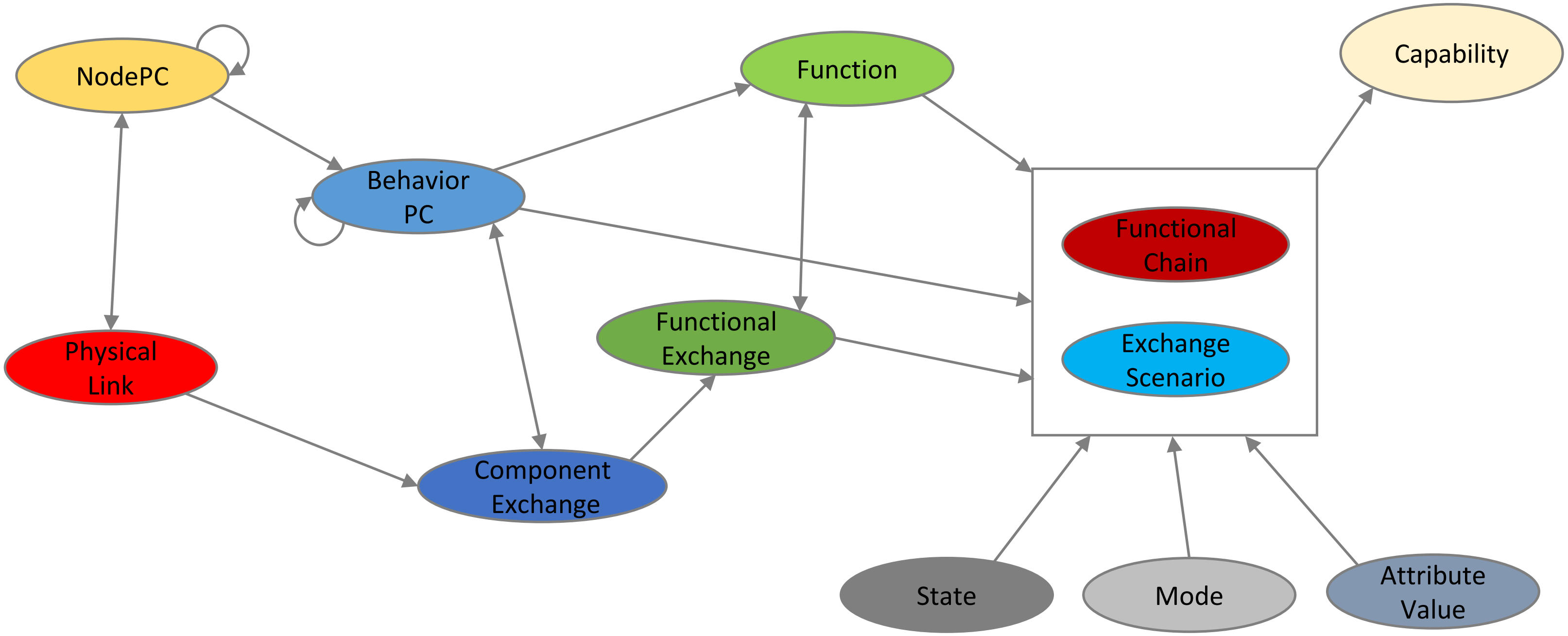
ManTech p2mberlin



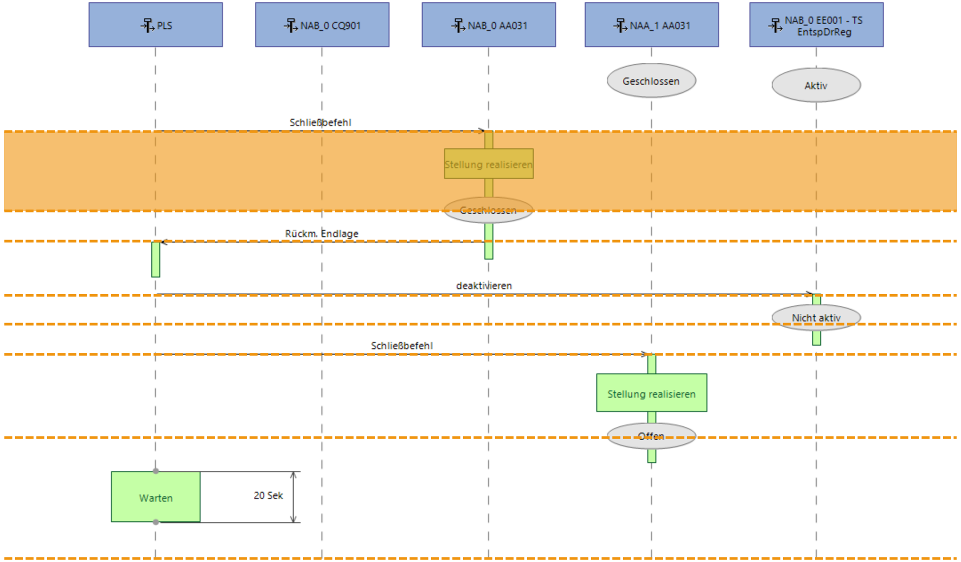
Impact
If we take these components out, what does the scenario look like? Do new configurations / situations appear? Which ones?
Is there a scenario stop?

Analysis of affected elements

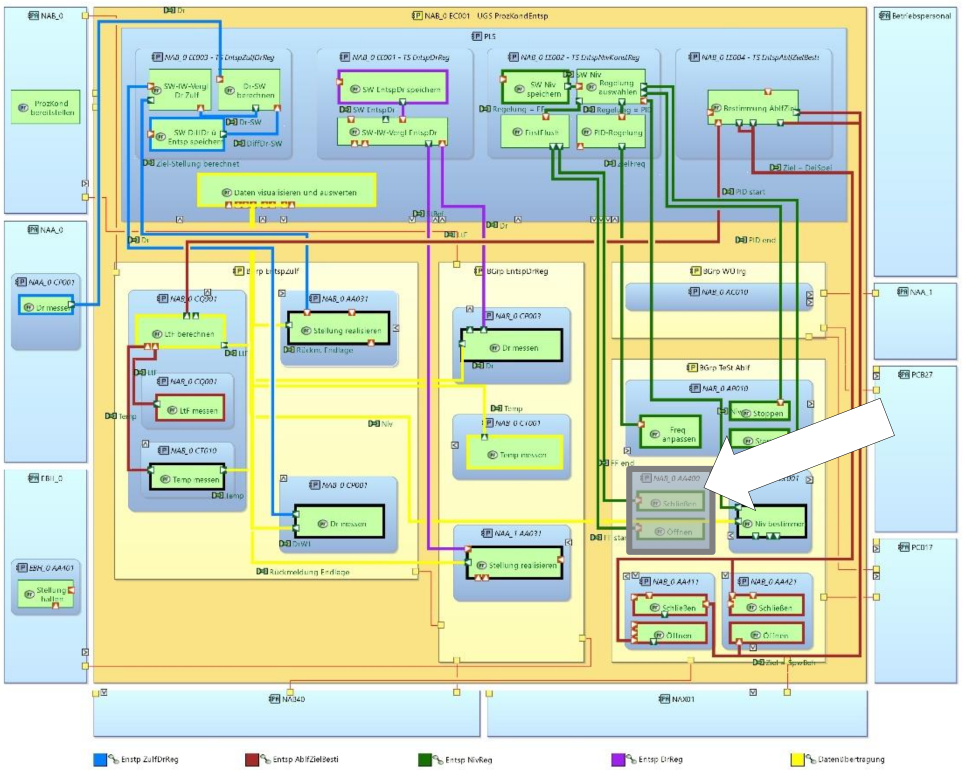
ManTech p2mberlin



Scenario breakdown



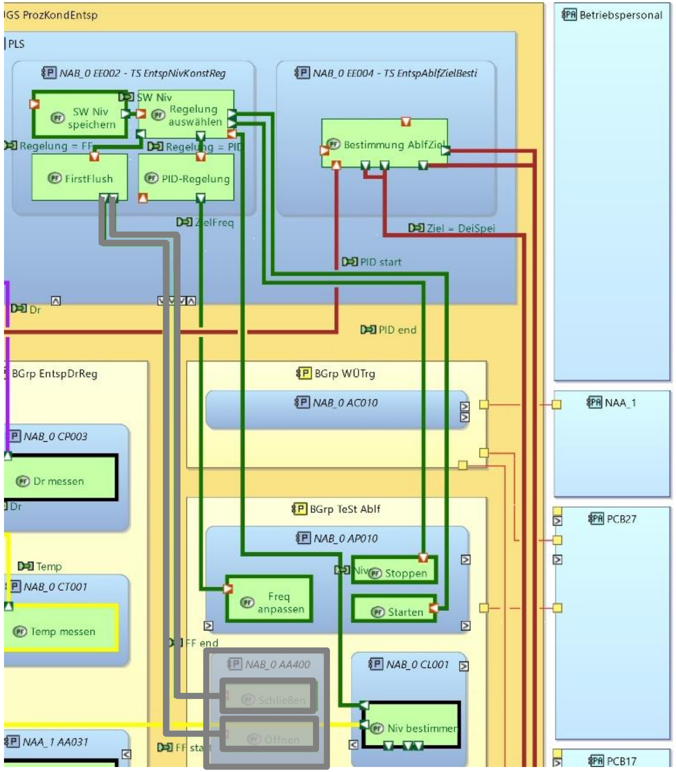
Example: component failure

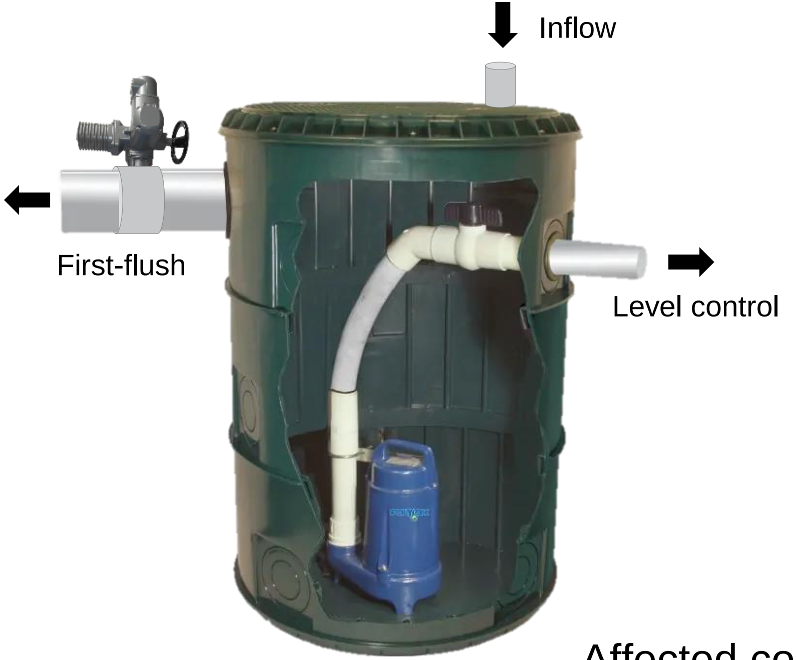


Failure identification:

„NAB_0 AA400„ is not available

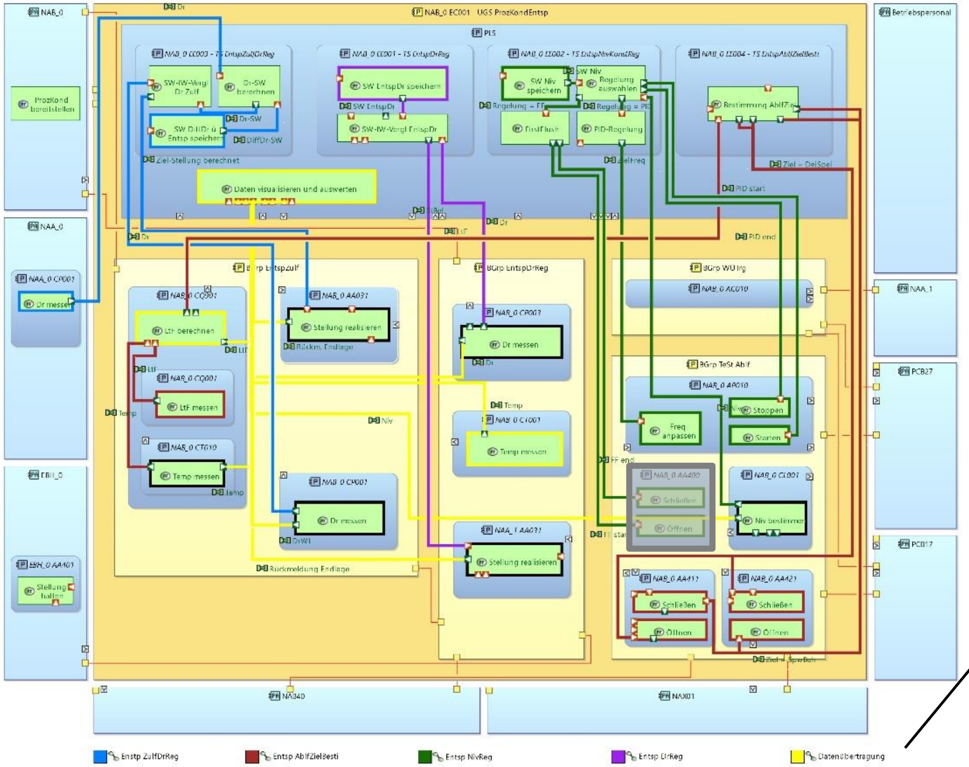
Example: component failure





Affected components identification

Example: component failure



🟦 Enstp ZulfDrReg

🟥 Entsp AblfZielBesti

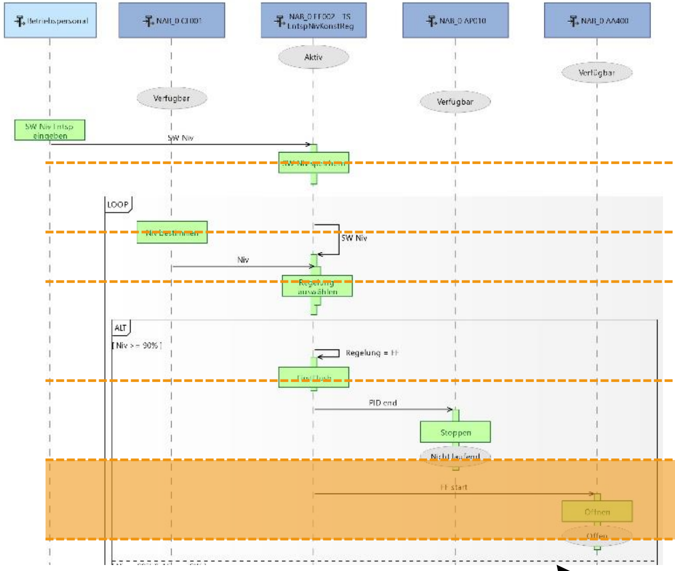
🟩 Entsp NivReg

🟪 Entsp DrReg

🟨 Datenübertragung

Identification of affected Exchange Scenarios

Example: component failure



NAB_0 CL001	NAB_0 EE002 - TS EntspNivKonstReg	NAB_0 AP010		NAB_0 AA400		
SM0	**SM1**	**SM0**	**SM1**	**SM0**	**SM1**	1.1
available	active	available	?	available	?	1.2
available	active	available	?	available	?	1.3
available	active	available	?	available	?	1.4
available	active	available	?	available	?	1.5
available	active	available	not running	Not available	?	1.6
available	active	available	not running	Not available	?	1.7

Control action: activate first-flush when >90% full

Entsp NivReg

[ES] analysis

Example: component failure

NAB_0 CL001	NAB_0 EE002 - TS EntspNivKonstReg	NAB_0 AP010		NAB_0 AA400		
SM0	**SM1**	**SM0**	**SM1**	**SM0**	**SM1**	1.1
available	active	available	?	available	?	1.2
available	active	available	?	available	?	1.3
available	active	available	?	available	?	1.4
available	active	available	?	available	?	1.5
available	active	available	not running	not available	?	1.6
available	active	available	not running	not available	?	1.7

NAB_0 CL001	NAB_0 EE002 - TS EntspNivKonstReg	NAB_0 AP010		NAB_0 AA400		
available	active	available	not running	not available	closed	1.6
available	active	available	not running	not available	closed	1.7

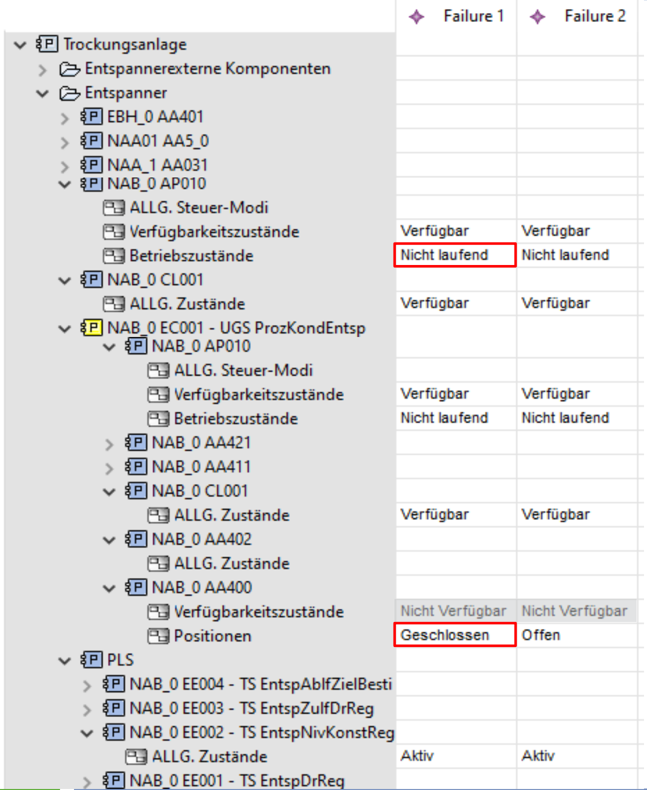
Situation: Failure 1

NAB_0 CL001	NAB_0 EE002 - TS EntspNivKonstReg	NAB_0 AP010		NAB_0 AA400		
available	active	available	not running	not available	open	1.6
available	active	available	not running	not available	open	1.7

Situation: Failure 2

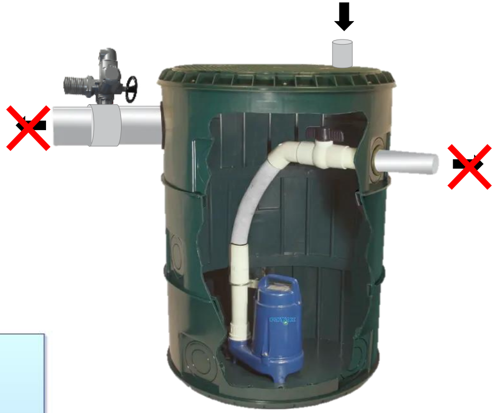
[ES] analysis

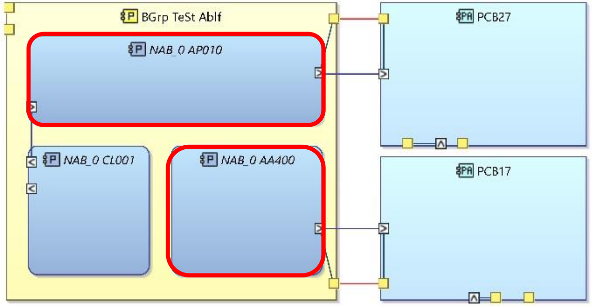
Example: component failure



	✦ Failure 1	✦ Failure 2
Trockungsanlage		
Entspannerexterne Komponenten		
Entspanner		
EBH_0 AA401		
NAA01 AA5_0		
NAA_1 AA031		
NAB_0 AP010		
ALLG. Steuer-Modi		
Verfügbarkeitszustände	Verfügbar	Verfügbar
Betriebszustände	Nicht laufend	Nicht laufend
NAB_0 CL001		
ALLG. Zustände	Verfügbar	Verfügbar
NAB_0 EC001 - UGS ProzKondEntsp		
NAB_0 AP010		
ALLG. Steuer-Modi		
Verfügbarkeitszustände	Verfügbar	Verfügbar
Betriebszustände	Nicht laufend	Nicht laufend
NAB_0 AA421		
NAB_0 AA411		
NAB_0 CL001		
ALLG. Zustände	Verfügbar	Verfügbar
NAB_0 AA402		
ALLG. Zustände		
NAB_0 AA400		
Verfügbarkeitszustände	Nicht Verfügbar	Nicht Verfügbar
Positionen	Geschlossen	Offen
PLS		
NAB_0 EE004 - TS EntspAblfZielBesti		
NAB_0 EE003 - TS EntspZulfDrReg		
NAB_0 EE002 - TS EntspNivKonstReg		
ALLG. Zustände	Aktiv	Aktiv
NAB_0 EE001 - TS EntspDrReg		

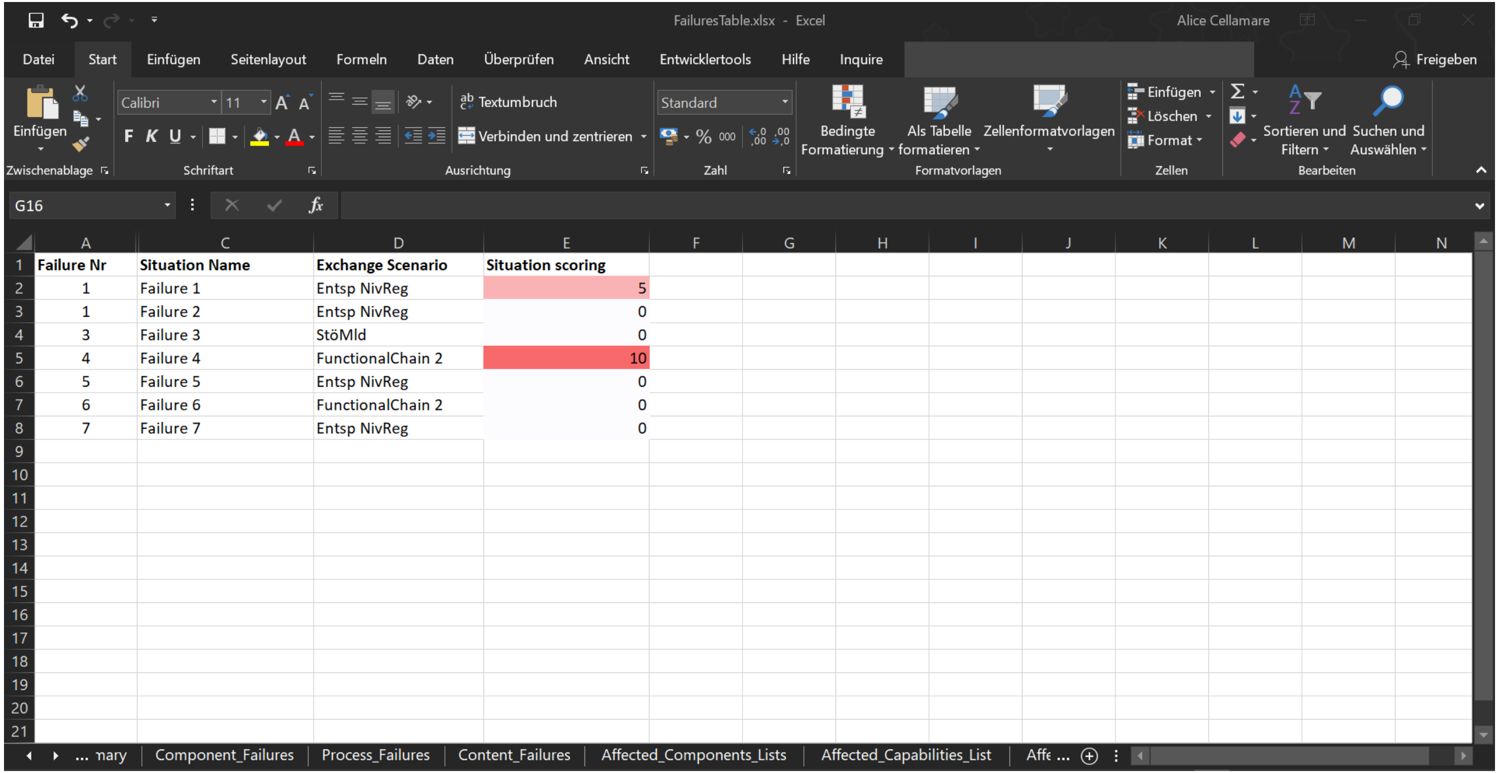
✦ Failure 1





Consequences as Situations

Example: component failure

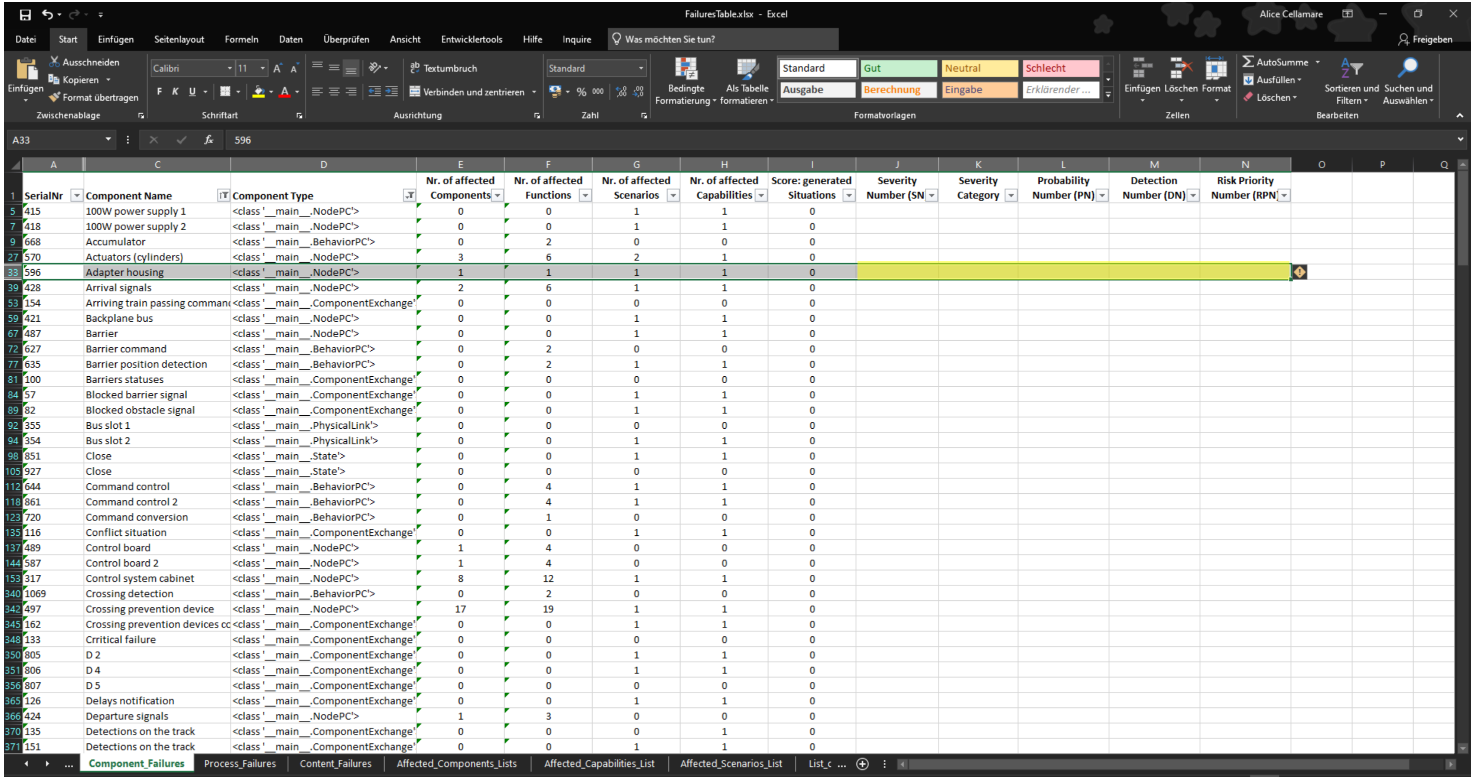


Failure Nr	Situation Name	Exchange Scenario	Situation scoring
1	Failure 1	Entsp NivReg	5
1	Failure 2	Entsp NivReg	0
3	Failure 3	StöMld	0
4	Failure 4	FunctionalChain 2	10
5	Failure 5	Entsp NivReg	0
6	Failure 6	FunctionalChain 2	0
7	Failure 7	Entsp NivReg	0

Situation scoring

Example: component failure

Failure Evaluation



	SerialNr	Component Name	Component Type	Nr. of affected Components	Nr. of affected Functions	Nr. of affected Scenarios	Nr. of affected Capabilities	Score: generated Situations	Severity Number (SN	Severity Category	Probability Number (PN)	Detection Number (DN)	Risk Priority Number (RPN
5	415	100W power supply 1	<class '__main__.NodePC'>	0	0	1	1	0					
7	418	100W power supply 2	<class '__main__.NodePC'>	0	0	1	1	0					
9	668	Accumulator	<class '__main__.BehaviorPC'>	0	2	0	0	0					
27	570	Actuators (cylinders)	<class '__main__.NodePC'>	3	6	2	1	0					
33	596	Adapter housing	<class '__main__.NodePC'>	1	1	1	1	0					
39	428	Arrival signals	<class '__main__.NodePC'>	2	6	1	1	0					
53	154	Arriving train passing comman	<class '__main__.ComponentExchange'	0	0	0	0	0					
59	421	Backplane bus	<class '__main__.NodePC'>	0	0	1	1	0					
67	487	Barrier	<class '__main__.NodePC'>	0	0	1	1	0					
72	627	Barrier command	<class '__main__.BehaviorPC'>	0	2	0	0	0					
77	635	Barrier position detection	<class '__main__.BehaviorPC'>	0	2	1	1	0					
81	100	Barriers statuses	<class '__main__.ComponentExchange'	0	0	0	0	0					
84	57	Blocked barrier signal	<class '__main__.ComponentExchange'	0	0	1	1	0					
89	82	Blocked obstacle signal	<class '__main__.ComponentExchange'	0	0	1	1	0					
92	355	Bus slot 1	<class '__main__.PhysicalLink'>	0	0	0	0	0					
94	354	Bus slot 2	<class '__main__.PhysicalLink'>	0	0	1	1	0					
98	851	Close	<class '__main__.State'>	0	0	1	1	0					
105	927	Close	<class '__main__.State'>	0	0	0	0	0					
112	644	Command control	<class '__main__.BehaviorPC'>	0	4	1	1	0					
118	861	Command control 2	<class '__main__.BehaviorPC'>	0	4	1	1	0					
123	720	Command conversion	<class '__main__.BehaviorPC'>	0	1	0	0	0					
135	116	Conflict situation	<class '__main__.ComponentExchange'	0	0	1	1	0					
137	489	Control board	<class '__main__.NodePC'>	1	4	0	0	0					
144	587	Control board 2	<class '__main__.NodePC'>	1	4	0	0	0					
153	317	Control system cabinet	<class '__main__.NodePC'>	8	12	1	1	0					
340	1069	Crossing detection	<class '__main__.BehaviorPC'>	0	2	0	0	0					
342	497	Crossing prevention device	<class '__main__.NodePC'>	17	19	1	1	0					
345	162	Crossing prevention devices cc	<class '__main__.ComponentExchange'	0	0	1	1	0					
348	133	Crritical failure	<class '__main__.ComponentExchange'	0	0	0	0	0					
350	805	D 2	<class '__main__.ComponentExchange'	0	0	1	1	0					
351	806	D 4	<class '__main__.ComponentExchange'	0	0	1	1	0					
356	807	D 5	<class '__main__.ComponentExchange'	0	0	0	0	0					
365	126	Delays notification	<class '__main__.ComponentExchange'	0	0	1	1	0					
366	424	Departure signals	<class '__main__.NodePC'>	1	3	0	0	0					
370	135	Detections on the track	<class '__main__.ComponentExchange'	0	0	0	1	0					
371	151	Detections on the track	<class '__main__.ComponentExchange'	0	0	1	1	0					

Conclusions

- Overall project impression
- Feasibility
- Critique

We welcome your feedback to improve our evolving approach for holistic fault analysis for Arcadia users

Stay tuned for session 4! →

Efficient and Comprehensive FMECAs: Harnessing the Power of MBSE Models in Capella

Thank you for your participation!

If you have any questions or would like to connect, please contact:

steven.huang@mantech.com alice.cellamare@p2mberlin.de

References

1. Baklouti, Anis & Nguyen, Nga & Mhenni, Faïda & Choley, Jean-Yves & Mlika, A.. (2019). Improved Safety Analysis Integration in a Systems Engineering Approach.pplied Sciences. 9. 10.3390/app9061246

2. Lai, K, Robert, T, Shindman, D & Olechowski, A 2021, 'Integrating Safety Analysis into Model-Based Systems Engineering for Aircraft Systems: A Literature Review and Methodology Proposal', INCOSE International Symposium, Vol. 31, No. 1, pp. 988–1003

3. Sulaman, S., Beer, A., Felderer, M. et al. Comparison of the FMEA and STPA safety analysis methods–a case study. Software Qual J 27, 349–387 (2019)

4. Voirin, Jean-Luc & Bonnet, Stéphane & Exertier, Daniel & Normand, Véronique. (2016). Simplifying (and enriching) SysML to perform functional analysis and model instances. INCOSE International Symposium. 26. 253-268. 10.1002/j.2334-5837.2016.00158.x

5. Duhil, Christophe & Babau, Jean-Philippe & Lepicier, Eric & Voirin, Jean-Luc & Navas, Juan. (2020). Chaining Model Transformations for System Model Verification: Application to Verify Capella Model with Simulink. 279-286. 10.5220/0008902302790286

References

6. Modeling states and modes with Arcadia and Capella: method and tool perspectives | Webinar Capella: https://www.youtube.com/watch?v=74eKWrSs8hI

7. https://github.com/eclipse/capella-vpms/

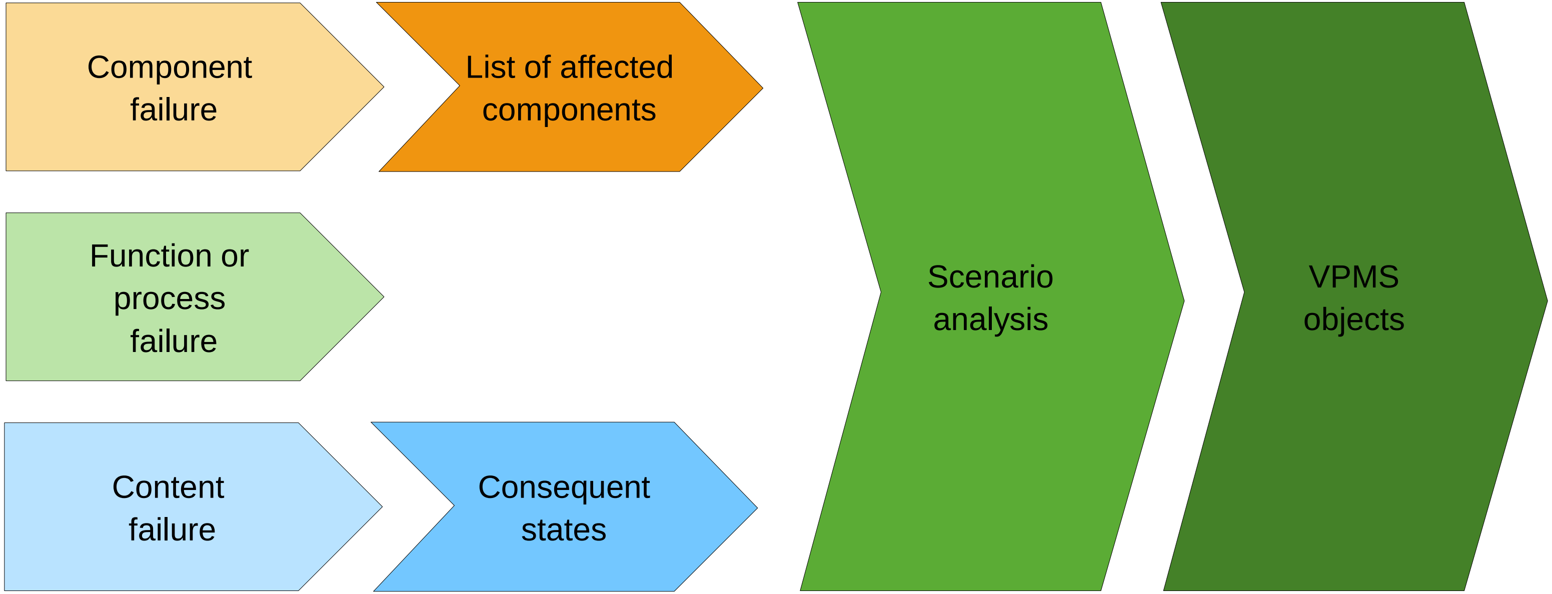
8. https://www.anzenengineering.com/mbse-mbsa/

9. https://www.all4tec.com/en/safety-architect-fmeca-fta-sofware/

EXTRA SLIDES

Analysis output

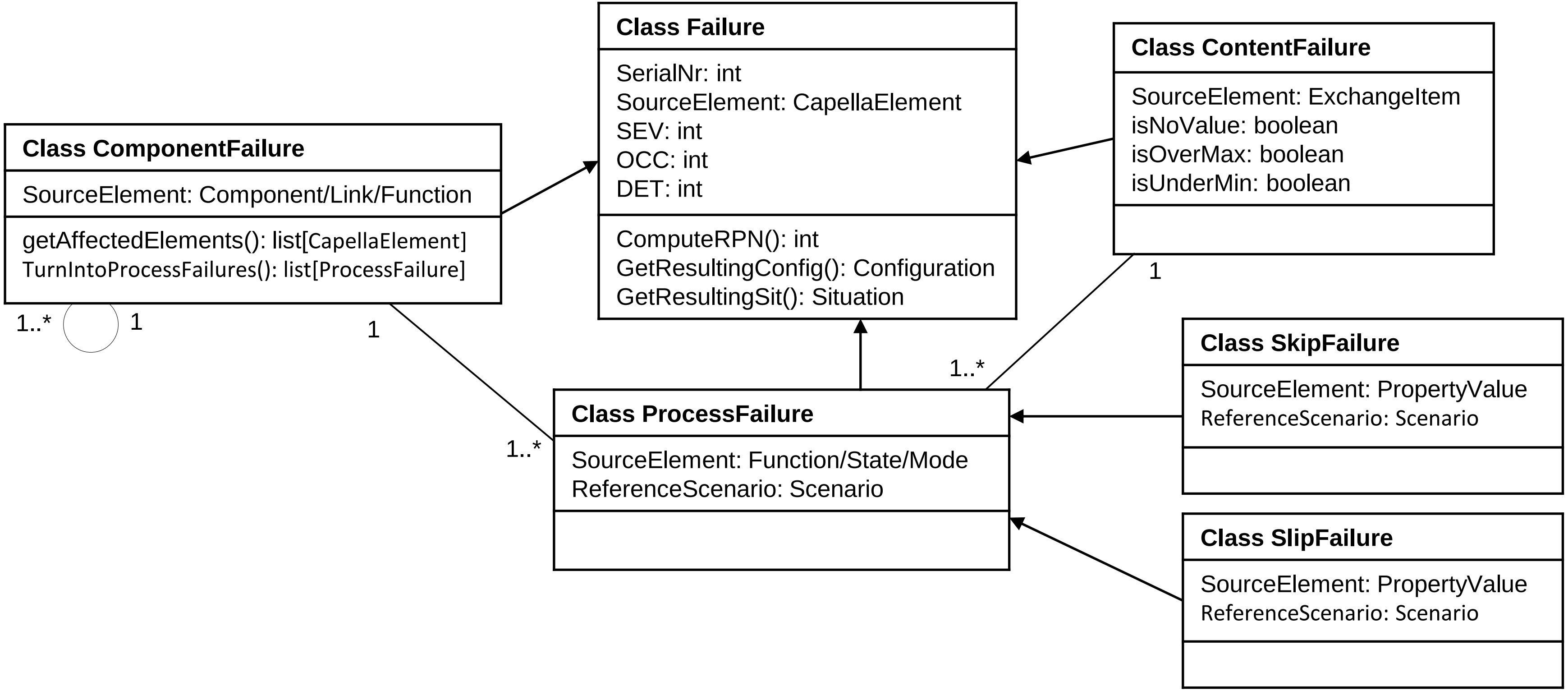
ManTech p2mberlin



Points to expand

- Failure combinations
- Situation diagrams?
- Limits of the approach: any failure modes that would only come up through, for example, physical modelling?
- Other

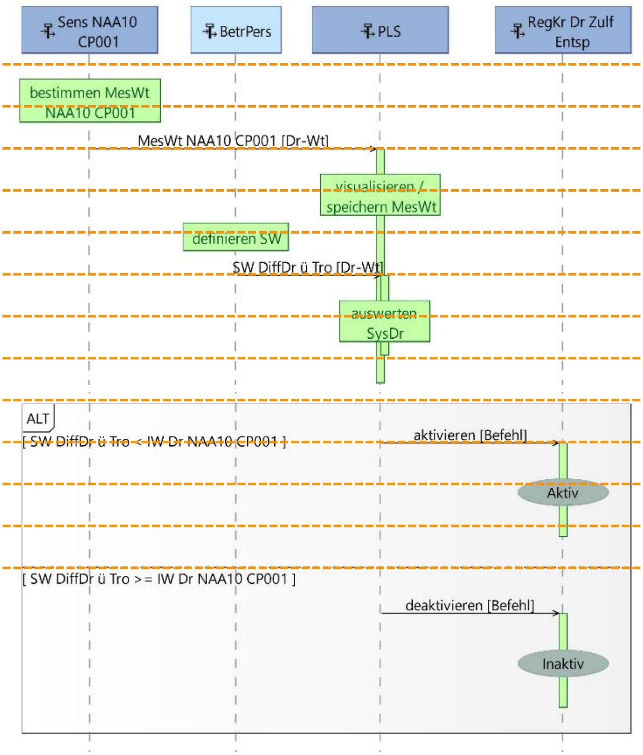
Failure classes


Class Failure
SerialNr: int
SourceElement: CapellaElement
SEV: int
OCC: int
DET: int
ComputeRPN(): int
GetResultingConfig(): Configuration
GetResultingSit(): Situation
Class ContentFailure
SourceElement: ExchangeItem
isNoValue: boolean
isOverMax: boolean
isUnderMin: boolean
Class ComponentFailure
SourceElement: Component/Link/Function
getAffectedElements(): list[CapellaElement]
TurnIntoProcessFailures(): list[ProcessFailure]
1..*
1
1
1
1..*
Class SkipFailure
SourceElement: PropertyValue
ReferenceScenario: Scenario
Class ProcessFailure
SourceElement: Function/State/Mode
ReferenceScenario: Scenario
1..*
Class SlipFailure
SourceElement: PropertyValue
ReferenceScenario: Scenario

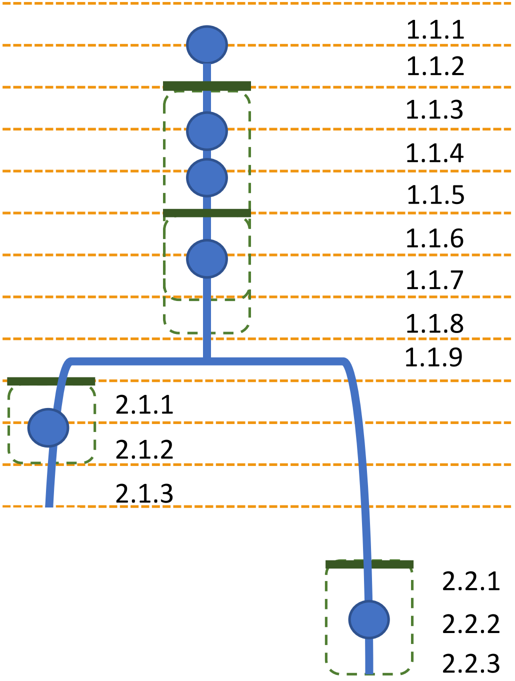
Scenario breakdown w. ALT

ManTech p2mberlin

	Sens NAA10 CP001	BetrPers	PLS	RegKr Dr Zulf Entsp
1.1.1	available		available	?
1.1.2	available		available	?
1.1.3	available		available	?
1.1.4	available		available	?
1.1.5	available		available	?
1.1.6	available		available	?
1.1.7	available		available	?
1.1.8	available		available	?
1.1.9	available		available	?
2.1.1	available		available	?
2.1.2	available		available	Active
2.1.3	available		available	Active



Sens NAA10 CP001
BetrPers
PLS
RegKr Dr Zulf Entsp
bestimmen MesWt NAA10 CP001
MesWt NAA10 CP001 [Dr-Wt]
visualisieren / speichern MesWt
definieren SW
SW DiffDr ü Tro [Dr-Wt]
auswerten SysDr
ALT
[SW DiffDr ü Tro < IW Dr NAA10 CP001]
aktivieren [Befehl]
Aktiv
[SW DiffDr ü Tro >= IW Dr NAA10 CP001]
deaktivieren [Befehl]
Inaktiv



1.1.1
1.1.2
1.1.3
1.1.4
1.1.5
1.1.6
1.1.7
1.1.8
1.1.9
2.1.1
2.1.2
2.1.3
2.2.1
2.2.2
2.2.3