Joint functional safety ISO 26262 and cybersecurity STRIDE/HEAVENS assessment by developers within MBSE SPES framework using extended SysML diagrams and minor automations

PSAM16, 27.-31.06.2022, Honolulu, O'ahu, Hawaii
Session Model-Based System Engineering,
Thursday, 30.6.2022, 15:30-17:00

Ivo Häring [a], Vivek Sudheendran [b], Roman Sankin [c], Stefan Hiermaier [d]

[a] Fraunhofer Institute for High-Speed Dynamics, Ernst-Mach-Institut, EMI, Efringen-Kirchen, Germany, ivo.haering@emi.fraunhofer.de,

[b] Work done at Bosch Engineering GmbH, Current Affiliation: Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany, vivek.sudheendran@desy.de

[c] Bosch Engineering GmbH, Abstatt, Germany, roman.sankin@de.bosch.com

[d] Department of Sustainable Systems Engineering, INATECH, University of Freiburg, Germany, stefan.hiermaier@inatech.uni-freiburg.de

Agenda

- Motivation and gaps
- Research questions
- Basic theory: SysML, SPES, ISO26262 and HEAVENS
- Concept as metamodels
- Concept validation
- Prototype: Malfunction Indicator Lamp
- Validation and results
- Conclusion and summary

1. Motivation and gaps

Motivation

- Increasing complexity of modern automotive systems need MBSE
- Handling of product quality: functional safety and cybersecurity (e.g. ISO26262, SOTIF)
- Suitable modeling approaches need to be selected
- SysML standard systems modeling language used by OEMs, Tier 1 and Tier 2 companies

Research gaps

- Cybersecurity standard for automotive under development
- Integrate safety, security and systems engineering
- Integration of models are challenging

2. Research questions

- How to integrate functional safety and cybersecurity analysis to the Model Based Engineering approaches like SPES?

- How to reduce the efforts of manual approaches to system development with respect to **Completeness, Traceability, Automation** etc.?

- How do model based approaches help in effective management of complex development lifecycles?

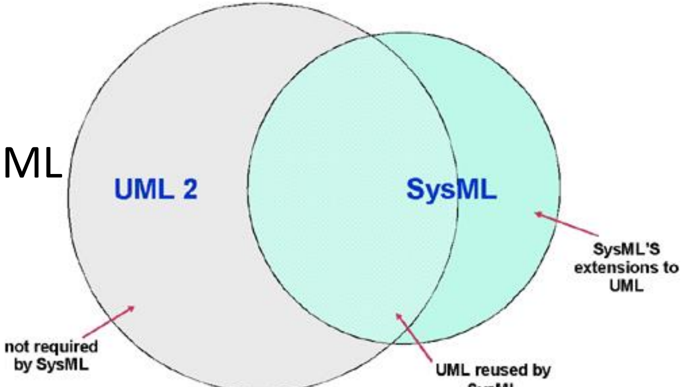
- How to support the implementation of functional safety and cybersecurity standards in MBE approaches using semiformal models?

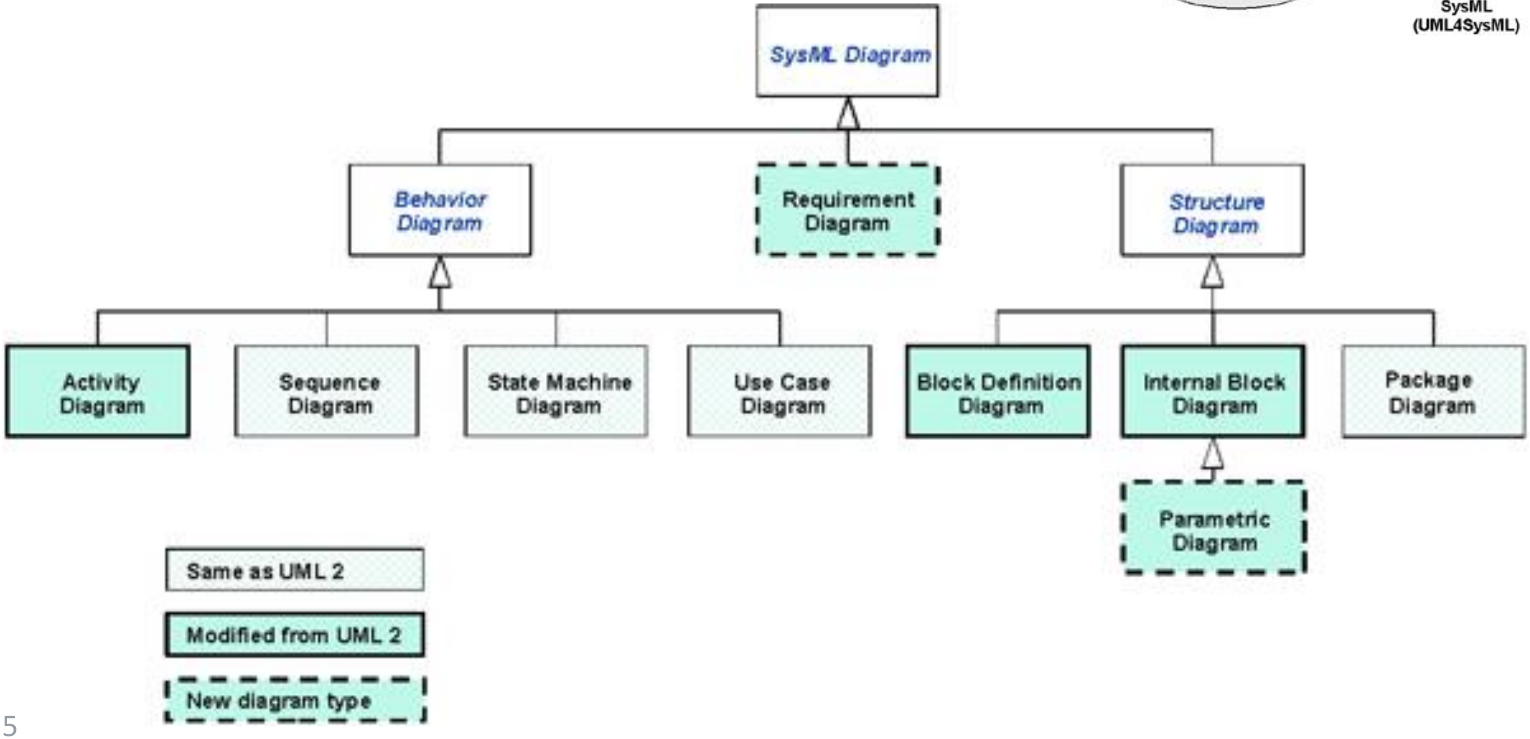
- How to validate the approach?

3. Basic theory

SysML

- Semi formal modeling language derived from UML
- Enhancements for requirements engineering
- SysML is not software centric



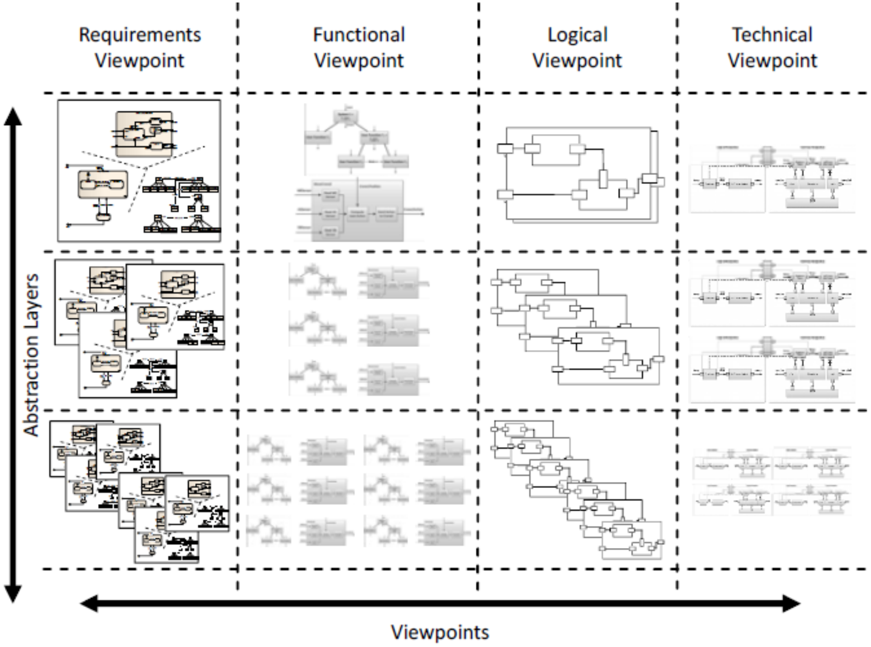


https://www.omgsysml.org/what-is-sysml.htm

3. Basic theory

Software Platform Embedded Systems (SPES): MBSE methodology

- Separate problem and solution
- Consider system decomposition
- Seamless model-based engineering
- Differentiate between logical and technical solution
- Continuous development of cross-cutting product properties

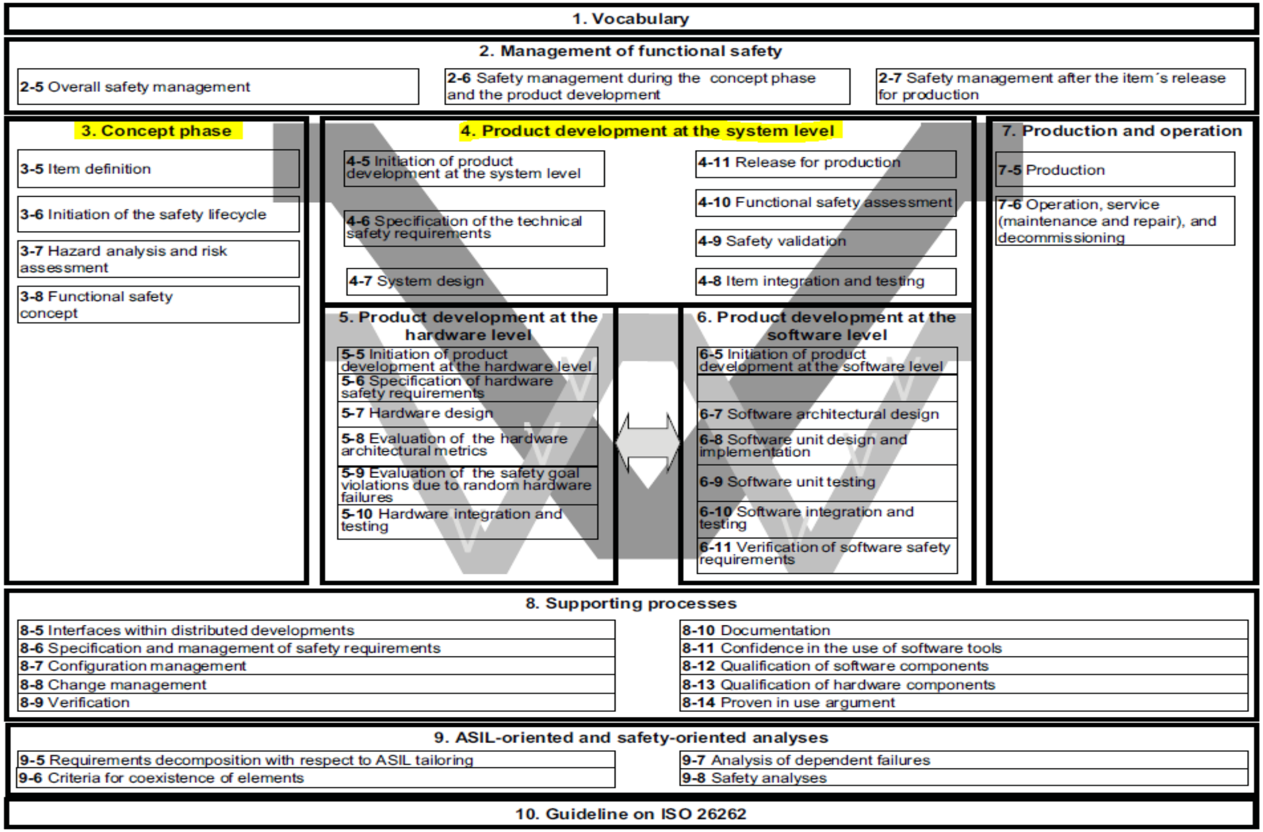


(Manfred et al. 2012)

3. Basic theory

ISO26262: Road vehicles functional safety

- ISO 26262 is an adaptation of the Functional Safety standard IEC 61508
- Automotive Electric/Electronic Systems



1. Vocabulary

2. Management of functional safety

- 2-5 Overall safety management
- 2-6 Safety management during the concept phase and the product development
- 2-7 Safety management after the item´s release for production

3. Concept phase

- 3-5 Item definition
- 3-6 Initiation of the safety lifecycle
- 3-7 Hazard analysis and risk assessment
- 3-8 Functional safety concept

4. Product development at the system level

- 4-5 Initiation of product development at the system level
- 4-6 Specification of the technical safety requirements
- 4-7 System design
- 4-11 Release for production
- 4-10 Functional safety assessment
- 4-9 Safety validation
- 4-8 Item integration and testing

5. Product development at the hardware level

- 5-5 Initiation of product development at the hardware level
- 5-6 Specification of hardware safety requirements
- 5-7 Hardware design
- 5-8 Evaluation of the hardware architectural metrics
- 5-9 Evaluation of the safety goal violations due to random hardware failures
- 5-10 Hardware integration and testing

6. Product development at the software level

- 6-5 Initiation of product development at the software level
- 6-7 Software architectural design
- 6-8 Software unit design and implementation
- 6-9 Software unit testing
- 6-10 Software integration and testing
- 6-11 Verification of software safety requirements

7. Production and operation

- 7-5 Production
- 7-6 Operation, service (maintenance and repair), and decommissioning

8. Supporting processes

- 8-5 Interfaces within distributed developments
- 8-6 Specification and management of safety requirements
- 8-7 Configuration management
- 8-8 Change management
- 8-9 Verification
- 8-10 Documentation
- 8-11 Confidence in the use of software tools
- 8-12 Qualification of software components
- 8-13 Qualification of hardware components
- 8-14 Proven in use argument

9. ASIL-oriented and safety-oriented analyses

- 9-5 Requirements decomposition with respect to ASIL tailoring
- 9-6 Criteria for coexistence of elements
- 9-7 Analysis of dependent failures
- 9-8 Safety analyses

10. Guideline on ISO 26262

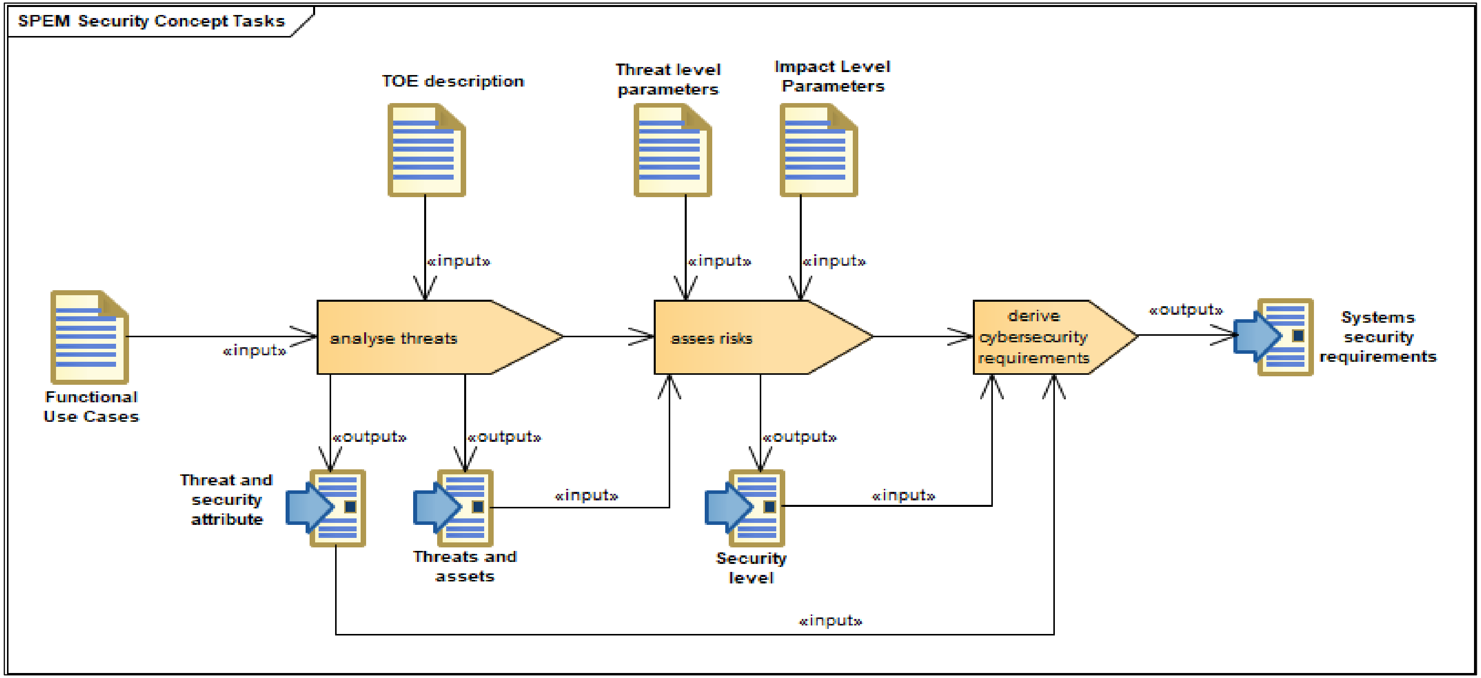
https://www.iso.org/standard/68383.html

3. Basic theory

HEAling Vulnerabilities to ENhance Software Security and Safety(HEAVENS)

(Lautenbach et al. 2016)

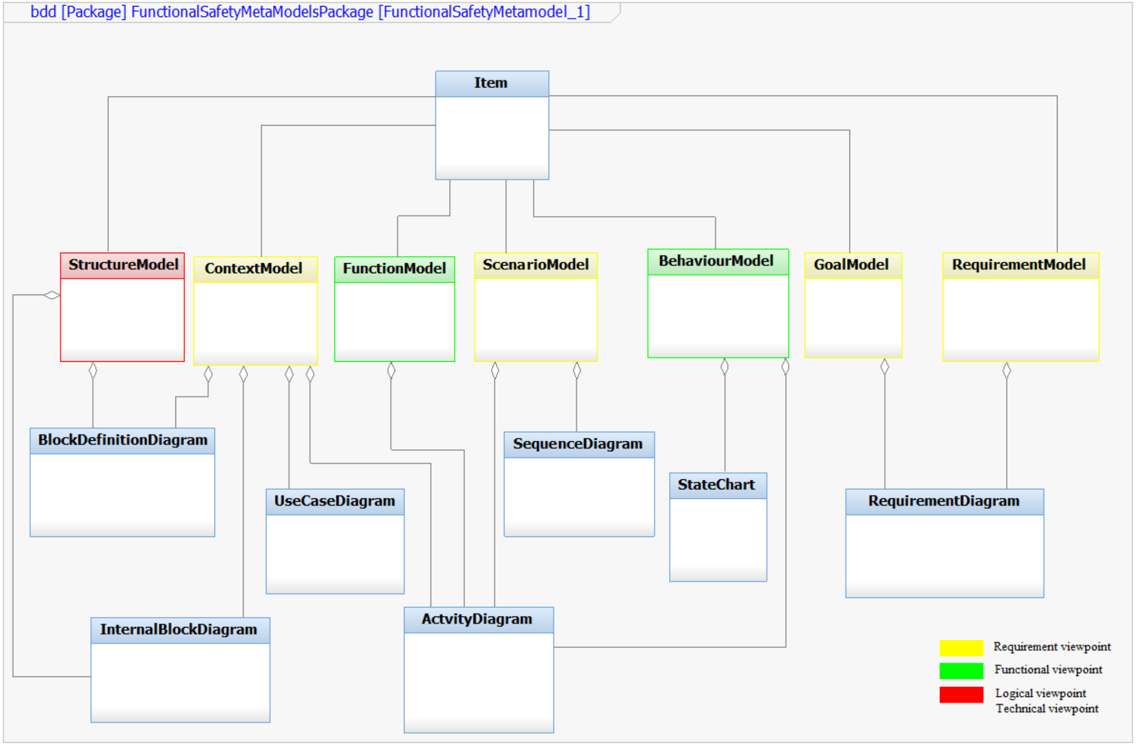
- HEAVENS is a security model developed for automotive domain
- Considers other existing cybersecurity models (STRIDE, CC, ETSI, SECTRA) to derive security requirements



4. Concepts as metamodels

Model based documentation of concept

- Developed for functional safety and cybersecurity
- Ensure seamless model based engineering
- Help to extend the current SPES profile with safety and security extensions
- Sample metamodel represents item definition



5. Concept validation

Criteria based concept validation (metamodels of the concept are developed and validated based on these criteria)

Criteria	Description
Completeness	• coverage for each phase in safety and security • elimination of incompleteness with respect to the attributes and parameter values
Traceability	• model elements can be inter-reachable and obtainable from high level to low level
Automation	• reduction in manual effort to handle document based assessment methods

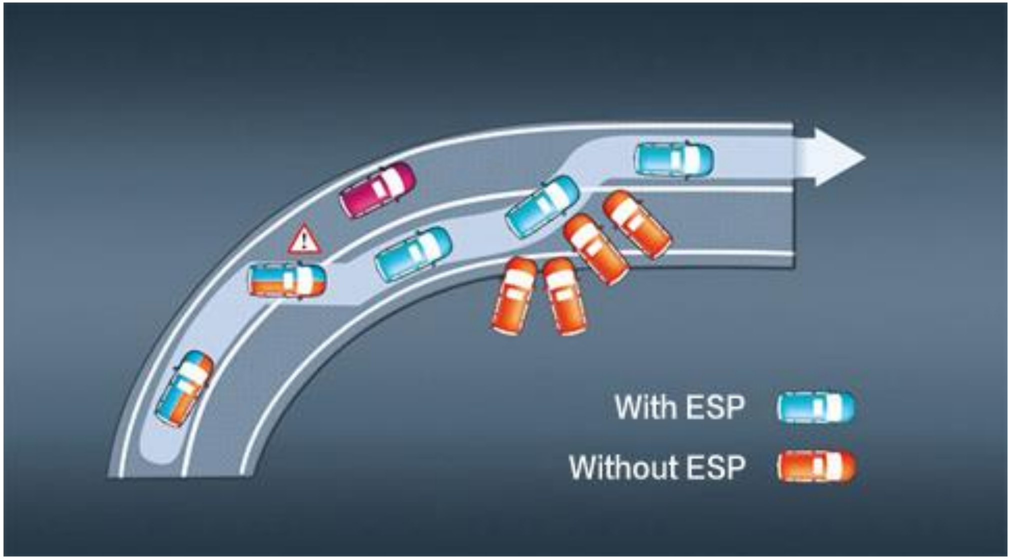
6. Prototype Malfunction Indicator Lamp

Malfunction Indicator Lamp (MIL): Electronic Stability Program (ESP) breakdown

- MIL shows vehicle issues in instrument cluster
- Focus on ESP breakdown

MIL

ESP

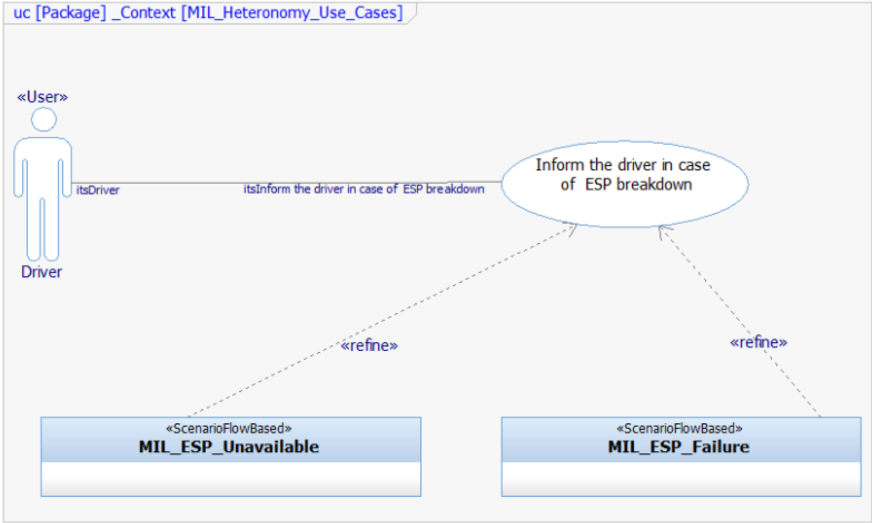


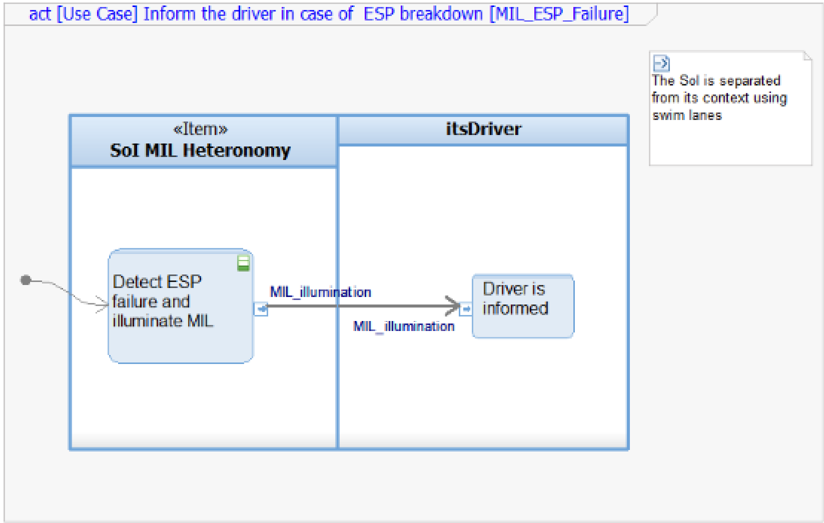
http://www.haval-global.com/havalh6.html

7. Validation and results

Item definition and Target of Evaluation (TOE) description prototypes

- Starting point for item and TOE description: use case analysis
- Use case refined to 2 scenarios
- Derivation of user functions using scenarios
- Diagrams shown in chapter 7 are SPES models mapped to ALs and VPs


uc [Package] _Context [MIL_Heteronomy_Use_Cases]
«User»
Driver
itsDriver
itsInform the driver in case of ESP breakdown
Inform the driver in case of ESP breakdown
«refine»
«refine»
«ScenarioFlowBased»
MIL_ESP_Unavailable
«ScenarioFlowBased»
MIL_ESP_Failure

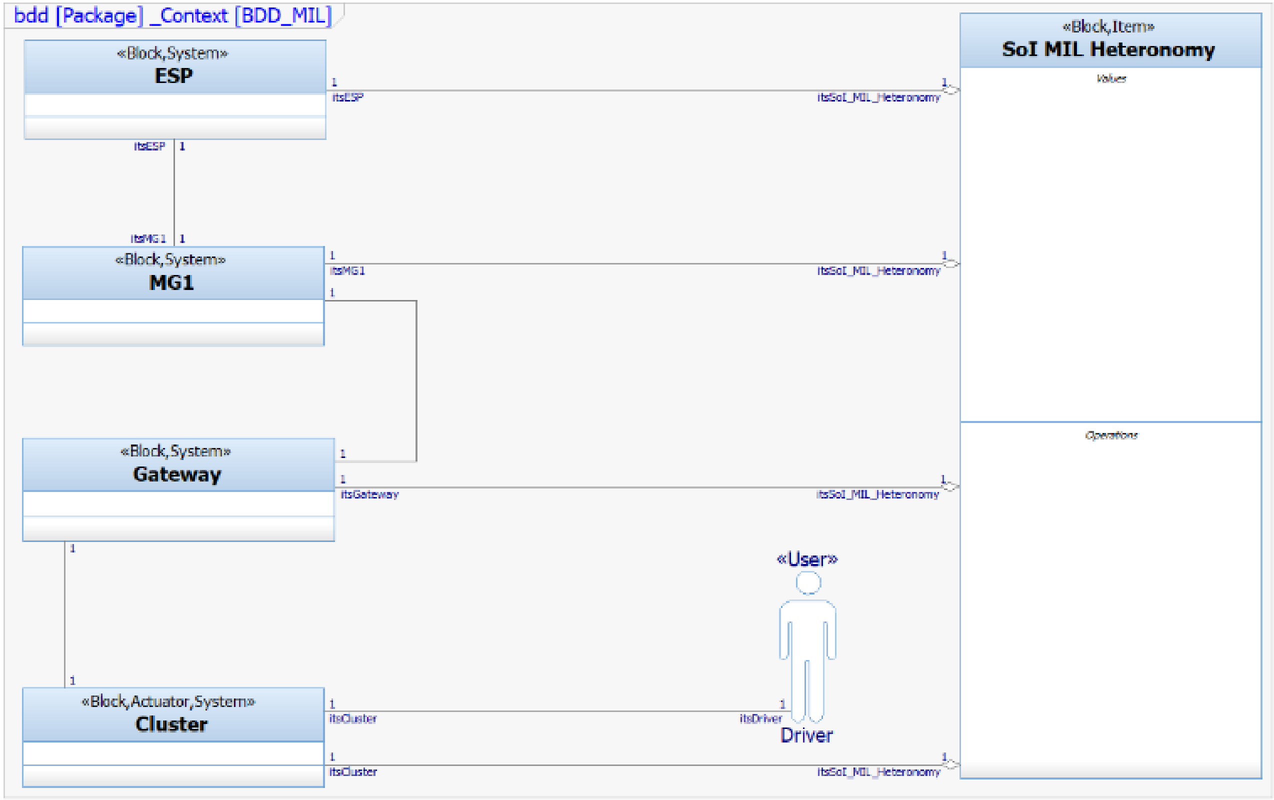

act [Use Case] Inform the driver in case of ESP breakdown [MIL_ESP_Failure]
«Item»
SoI MIL Heteronomy
itsDriver
Detect ESP failure and illuminate MIL
MIL_illumination
MIL_illumination
Driver is informed

The SoI is separated from its context using swim lanes

7. Validation and results

Context architecture for item or TOE

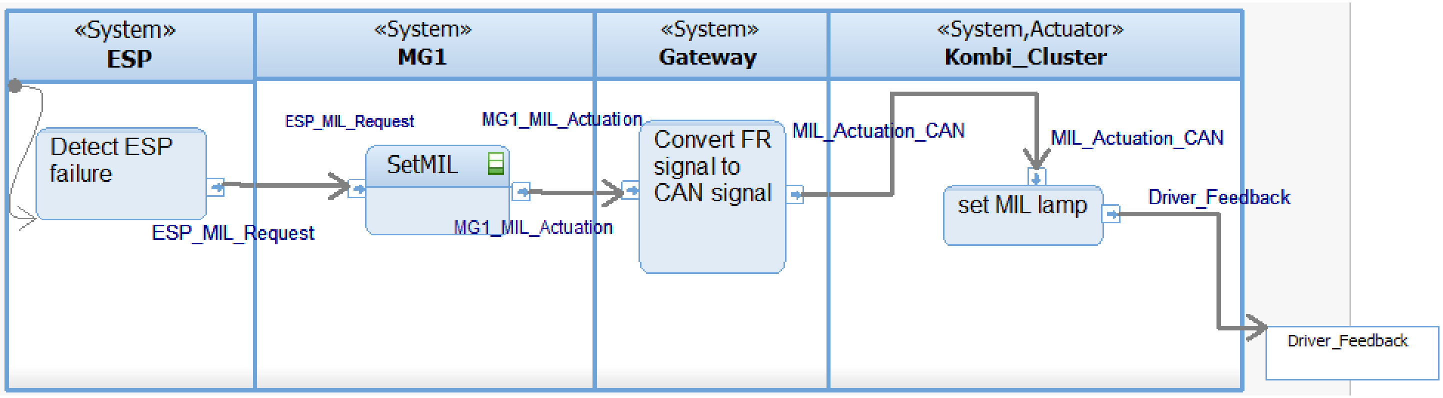
- Context architecture visualised in block definition diagram
- Internal connections shown in internal block diagram

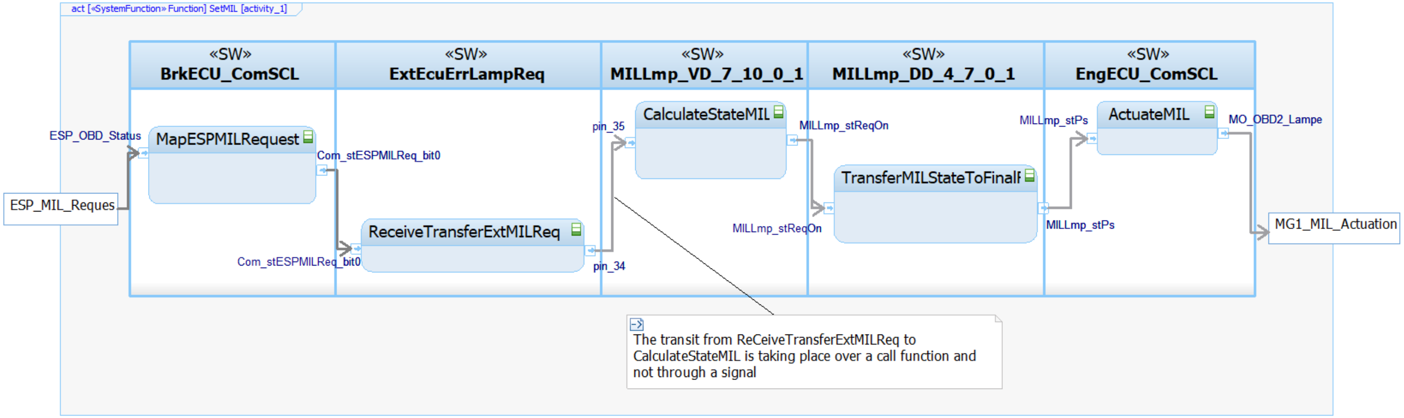


7. Validation and results

Functional decomposition: SysML activity diagram

- Description of functional chains
- Starting point of functional anaylsis



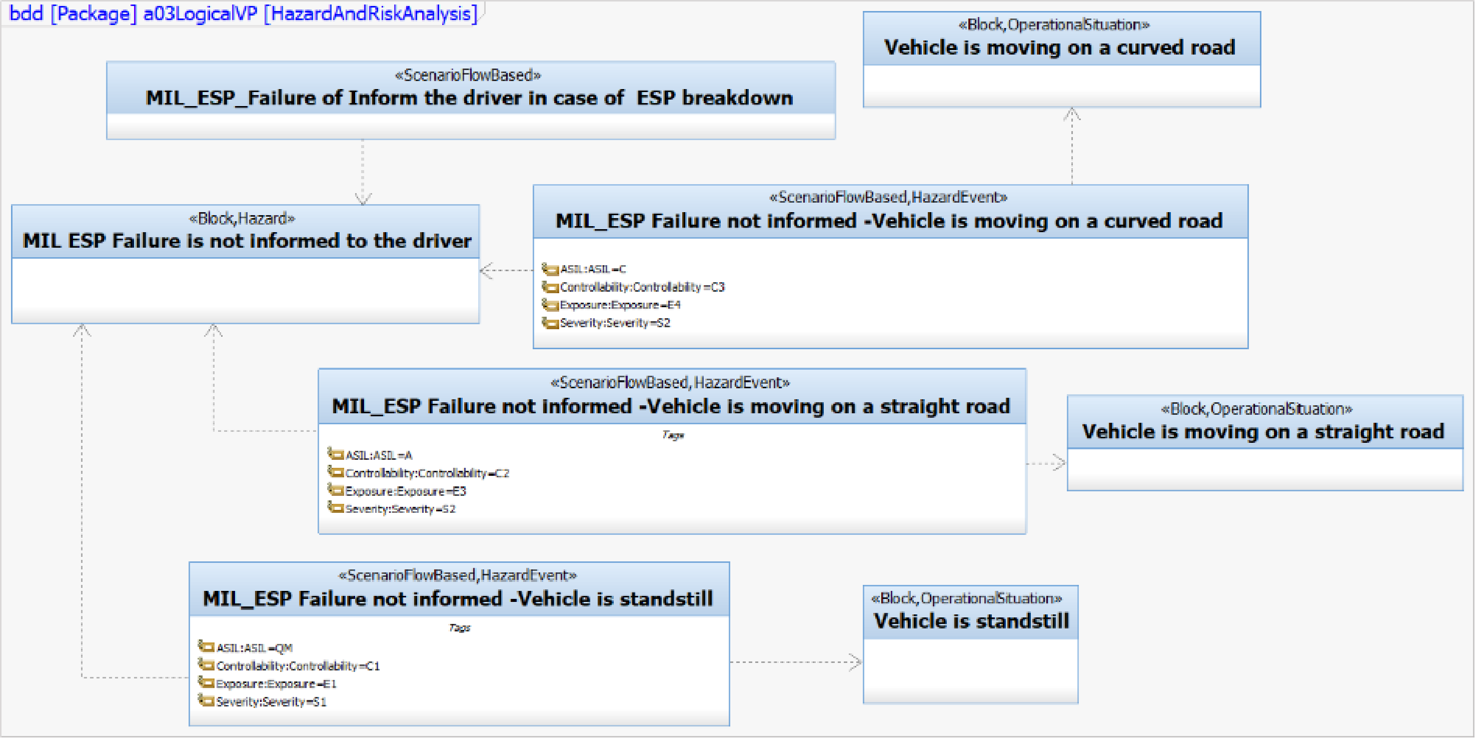


The transit from ReCeiveTransferExtMILReq to CalculateStateMIL is taking place over a call function and not through a signal

7. Validation and results

Functional safety extensions: Risk analysis support

- User configures controllability, severity and exposure for hazard events
- ASIL determination using automation scripts
- Support available in block definition diagram and also in table view



bdd [Package] a03LogicalVP [HazardAndRiskAnalysis]

«Block,OperationalSituation»
Vehicle is moving on a curved road

«ScenarioFlowBased»
MIL_ESP_Failure of Inform the driver in case of ESP breakdown

«Block,Hazard»
MIL ESP Failure is not informed to the driver

«ScenarioFlowBased,HazardEvent»
MIL_ESP Failure not informed -Vehicle is moving on a curved road

ASIL:ASIL=C
Controllability:Controllability=C3
Exposure:Exposure=E4
Severity:Severity=S2

«ScenarioFlowBased,HazardEvent»
MIL_ESP Failure not informed -Vehicle is moving on a straight road

Tags

ASIL:ASIL=A
Controllability:Controllability=C2
Exposure:Exposure=E3
Severity:Severity=S2

«Block,OperationalSituation»
Vehicle is moving on a straight road

«ScenarioFlowBased,HazardEvent»
MIL_ESP Failure not informed -Vehicle is standstill

Tags

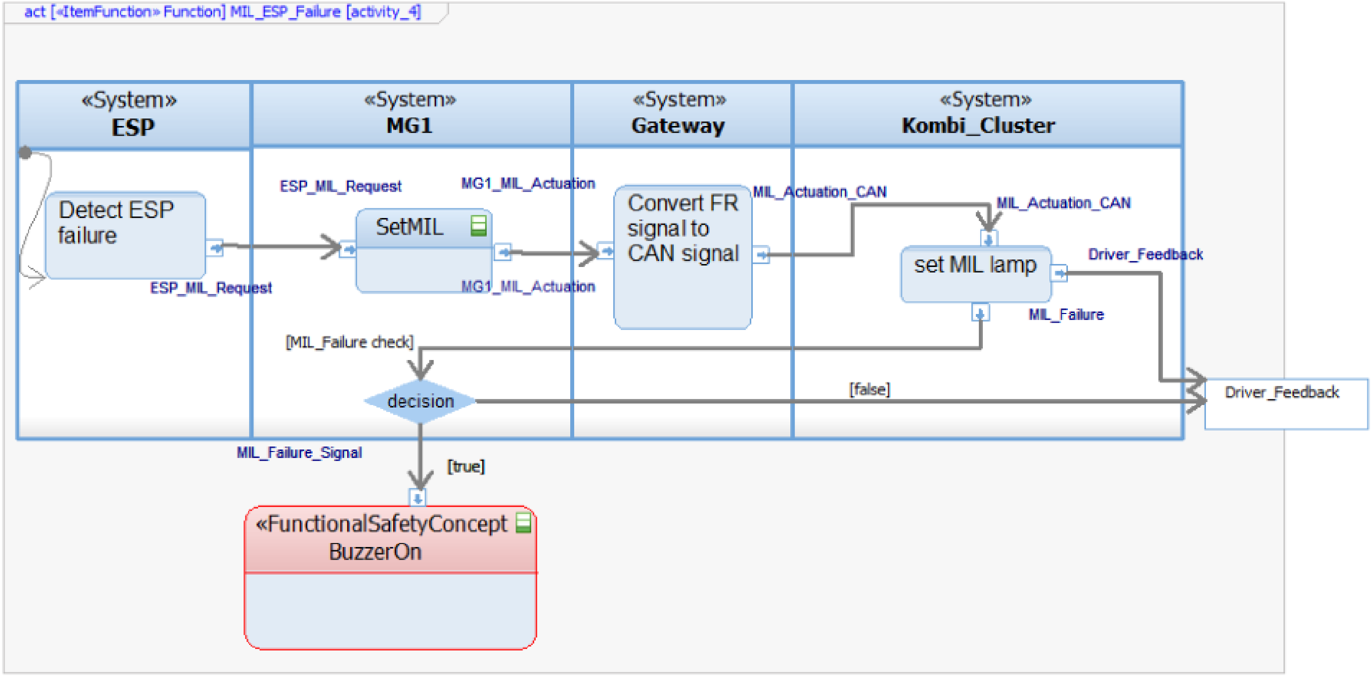
ASIL:ASIL=QM
Controllability:Controllability=C1
Exposure:Exposure=E1
Severity:Severity=S1

«Block,OperationalSituation»
Vehicle is standstill

7. Validation and results

Functional safety concept (FSC): BuzzerOn

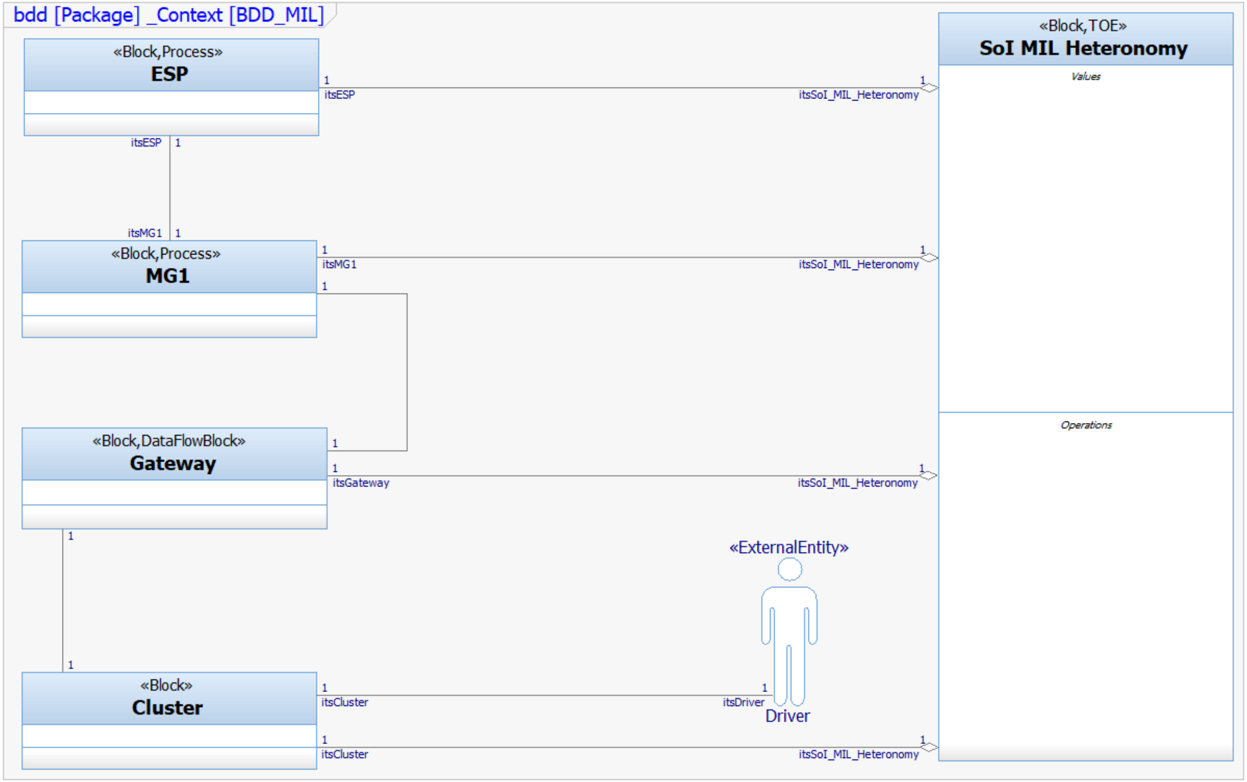
- The FSC added to item function, architecture is also extended with safety relevant blocks
- FSC monitors item function
- Similarly, technical safety function monitors system function



7. Validation and results

Cybersecurity extensions: Threat analysis

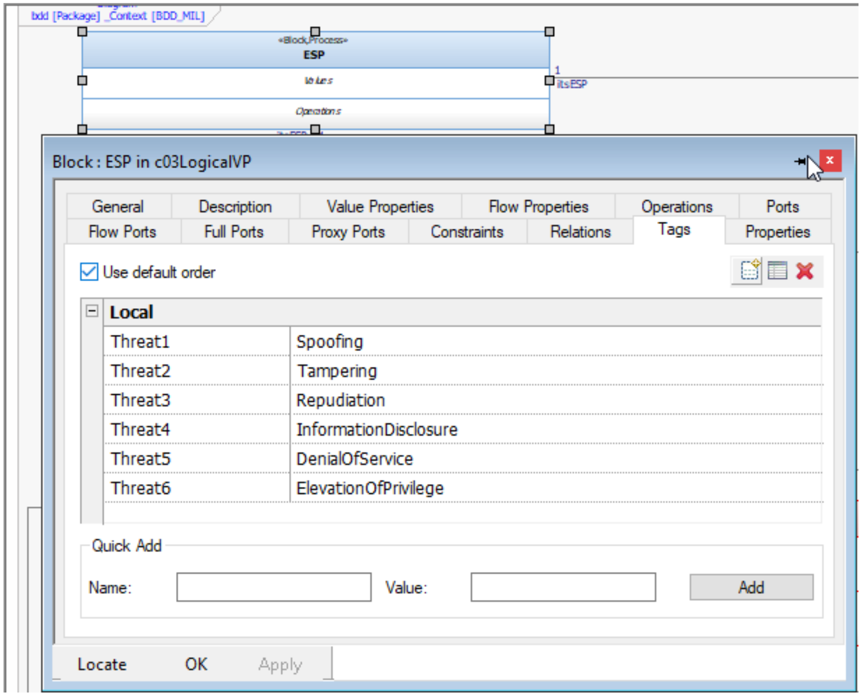
- Assets are classified based on stereotypes: Process, DataFlow, DataFlowBlock, DataStore, ExternalEntity
- For instance, ESP and MG1 are assets classified as <<Process>>



7. Validation and results

Cybersecurity extensions: Risk assessment

- STRIDE threats are generated inside model elements
- Threat level, Impact level and Security level automatically determined for each threat



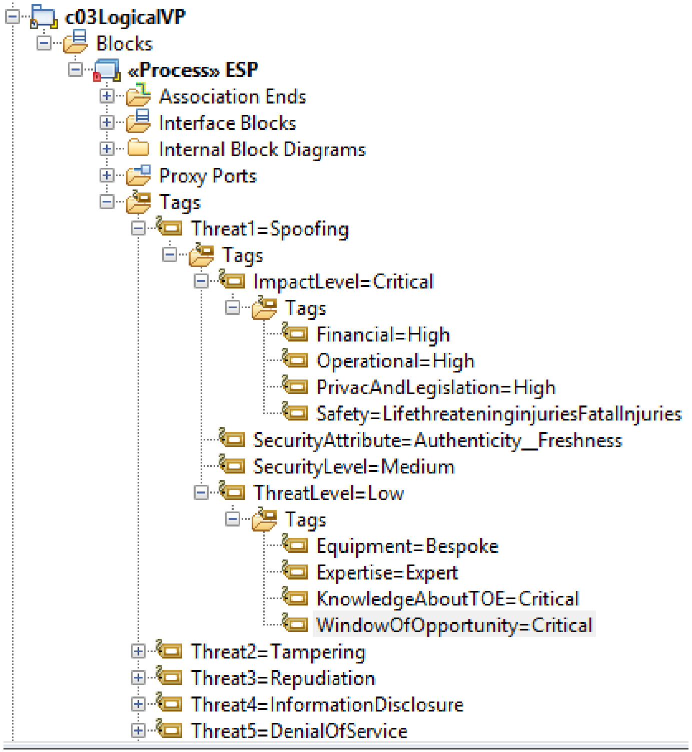
Block : ESP in c03LogicalVP

☑ Use default order

Local	
Threat1	Spoofing
Threat2	Tampering
Threat3	Repudiation
Threat4	InformationDisclosure
Threat5	DenialOfService
Threat6	ElevationOfPrivilege

Name:

Value:

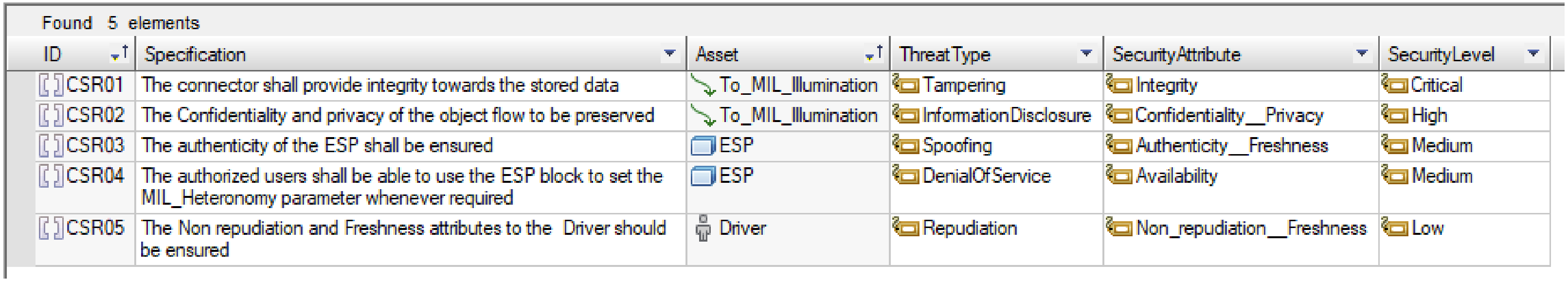


- c03LogicalVP
  - Blocks
    - «Process» ESP
      - Association Ends
      - Interface Blocks
      - Internal Block Diagrams
      - Proxy Ports
      - Tags
        - Threat1=Spoofing
          - Tags
            - ImpactLevel=Critical
              - Tags
                - Financial=High
                - Operational=High
                - PrivacAndLegislation=High
                - Safety=LifethreateninginjuriesFatalInjuries
            - SecurityAttribute=Authenticity__Freshness
            - SecurityLevel=Medium
            - ThreatLevel=Low
              - Tags
                - Equipment=Bespoke
                - Expertise=Expert
                - KnowledgeAboutTOE=Critical
                - WindowOfOpportunity=Critical
        - Threat2=Tampering
        - Threat3=Repudiation
        - Threat4=InformationDisclosure
        - Threat5=DenialOfService

7. Validation and results

Traceability: cybersecurity and functional safety table views

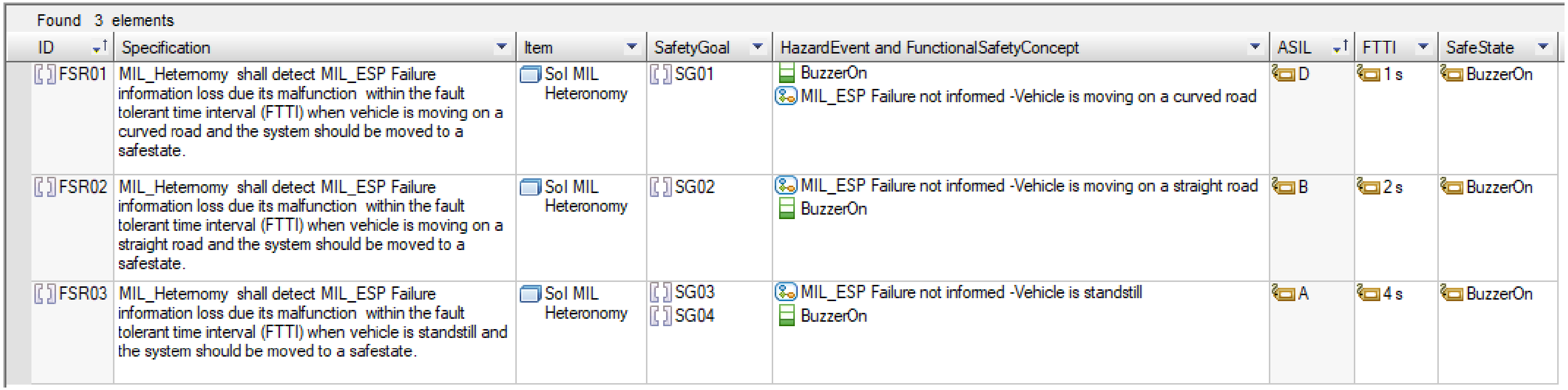
Cybersecurity requirements table



Found 5 elements

ID	Specification	Asset	ThreatType	SecurityAttribute	SecurityLevel
CSR01	The connector shall provide integrity towards the stored data	To_MIL_Illumination	Tampering	Integrity	Critical
CSR02	The Confidentiality and privacy of the object flow to be preserved	To_MIL_Illumination	InformationDisclosure	Confidentiality__Privacy	High
CSR03	The authenticity of the ESP shall be ensured	ESP	Spoofing	Authenticity__Freshness	Medium
CSR04	The authorized users shall be able to use the ESP block to set the MIL_Heteronomy parameter whenever required	ESP	DenialOfService	Availability	Medium
CSR05	The Non repudiation and Freshness attributes to the Driver should be ensured	Driver	Repudiation	Non_repudiation__Freshness	Low

Functional safety requirements table

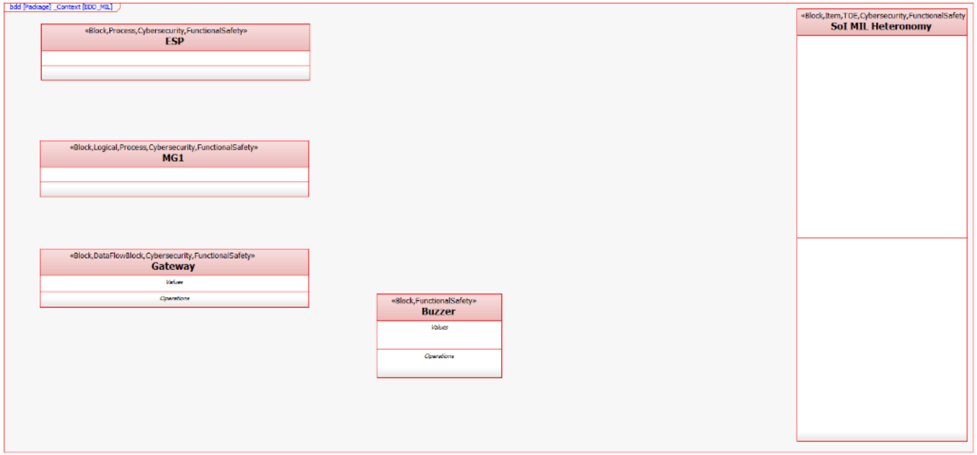


Found 3 elements

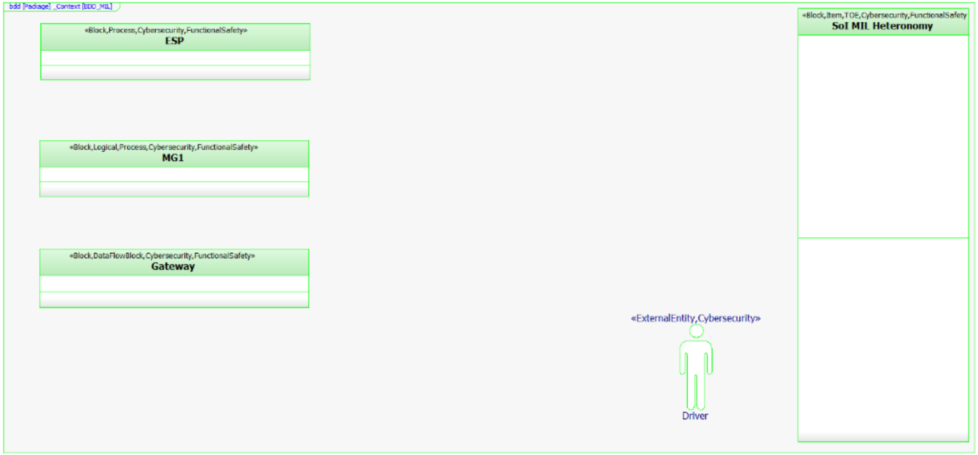
ID	Specification	Item	SafetyGoal	HazardEvent and FunctionalSafetyConcept	ASIL	FTTI	SafeState
FSR01	MIL_Heternomy shall detect MIL_ESP Failure information loss due its malfunction within the fault tolerant time interval (FTTI) when vehicle is moving on a curved road and the system should be moved to a safestate.	Sol MIL Heteronomy	SG01	BuzzerOn MIL_ESP Failure not informed -Vehicle is moving on a curved road	D	1 s	BuzzerOn
FSR02	MIL_Heternomy shall detect MIL_ESP Failure information loss due its malfunction within the fault tolerant time interval (FTTI) when vehicle is moving on a straight road and the system should be moved to a safestate.	Sol MIL Heteronomy	SG02	MIL_ESP Failure not informed -Vehicle is moving on a straight road BuzzerOn	B	2 s	BuzzerOn
FSR03	MIL_Heternomy shall detect MIL_ESP Failure information loss due its malfunction within the fault tolerant time interval (FTTI) when vehicle is standstill and the system should be moved to a safestate.	Sol MIL Heteronomy	SG03 SG04	MIL_ESP Failure not informed -Vehicle is standstill BuzzerOn	A	4 s	BuzzerOn

7. Validation and results

Diagram Views: filter safety and security relevant elements from BDD,IBD,UCD and AD


bdd [Package] _Context [BDD_MIL]
«Block,Process,Cybersecurity,FunctionalSafety»
ESP
«Block,Logical,Process,Cybersecurity,FunctionalSafety»
MG1
«Block,DataFlowBlock,Cybersecurity,FunctionalSafety»
Gateway
Values
Operations
«Block,FunctionalSafety»
Buzzer
Values
Operations
«Block,Item,TOE,Cybersecurity,FunctionalSafety
SoI MIL Heteronomy

Functional safety view based on <<FunctionalSafety>>

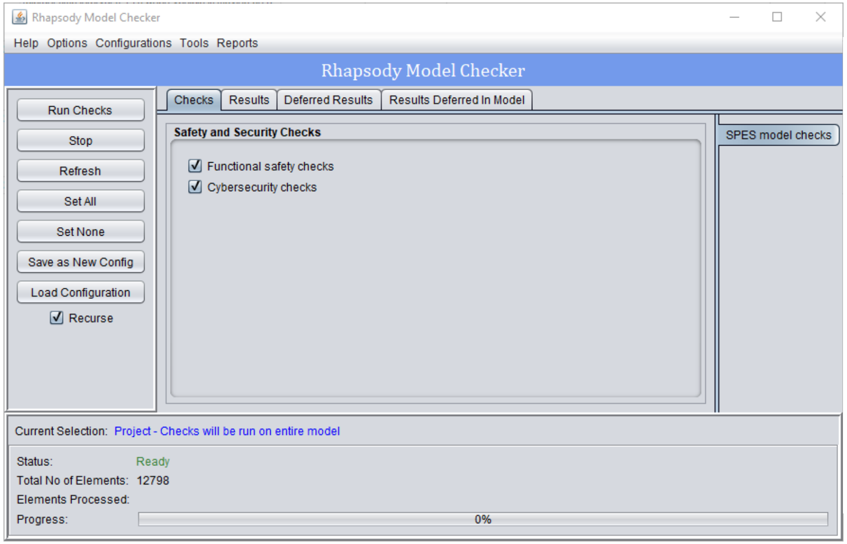

bdd [Package] _Context [BDD_MIL]
«Block,Process,Cybersecurity,FunctionalSafety»
ESP
«Block,Logical,Process,Cybersecurity,FunctionalSafety»
MG1
«Block,DataFlowBlock,Cybersecurity,FunctionalSafety»
Gateway
«ExternalEntity,Cybersecurity»
Driver
«Block,Item,TOE,Cybersecurity,FunctionalSafety
SoI MIL Heteronomy

Cybersecurity relevant view based on <<Cybersecurity>>

7. Validation and results

Model checker: safety and security relevant checks

- Safety checks for hazard events, safety goals and functional safety requirements
- Security checks for cybersecurity requirements



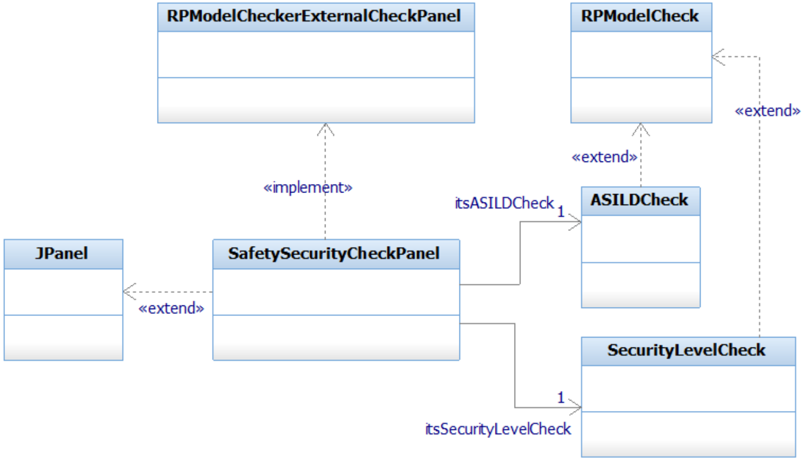
☑ Recurse

Safety and Security Checks

☑ Functional safety checks
☑ Cybersecurity checks

Current Selection: Project - Checks will be run on entire model

Status: Ready
Total No of Elements: 12798
Elements Processed:
Progress: 0%



7.Validation and results

Discussion: Realistic assessment

Sl no.	SysML diagram or model elements	Count
1	Block definition diagram	3
2	Internal block diagram	3
3	Use case diagram	1
4	Activity diagram	12
5	Requirement diagram	2
6	Diagram view	2
7	Table view	8
8	Stereotypes	32
9	Query	4

- Complexity of models: total diagram counts ≈ 30 pages document
- Descriptive information managed in the model as tags, attributes, comments
- Summarizing tables can be generated

7.Validation and results

Discussion: Realistic assessment, BOSCH expert feedbacks

Cybersecurity

Positives:

- Ideas for threat analysis and risk assessment tool
- Approach relatable with the current non MBSE practice at Bosch
- Integrating functional safety and cybersecurity within MBSE is commendable

Negatives:

- Replacing the current practices still challenging due to dependency with specific open source tools

Functional safety

Positives:

- MBSE with automations improve efficiency
- Integration of functional safety with systems engineering reduces overheads

Negatives:

- Choice of prototype is not much of interest for functional safety user world
- A parallel light architecture for MIL better than safety concept to make buzzer sound

7.Validation and results

Discussions

Advantages

1. Seamless workflow: **models, automations** and **model checker**

2. The **completeness** and **traceability** criteria are achieved

3. The **model checker** and **automations** enhance the **usability** of the models and increase **efficiency**

4. Method implementation is only once and it provides **reusability**

5. The **system development**, **functional safety** and **cybersecurity** assessment go in parallel using single model source

Disadvantages

1. Danger: formal assessment by **just clicking** at threats

2. The **user** needs to **remove non-significant threats**

3. It demands at least an **intermediate level knowledge in SysML** for functional safety and cybersecurity experts

4. All **the SPES models not used** for prototyping

8.Conclusion

Summary

Integrated MBSE assessment approach for functional safety, cybersecurity and systems engineering

Concept is validated based on criteria: completeness, traceability and automation

Guidelines of functional safety standard ISO26262 and Microsoft STRIDE based HEAVENS model are incorporated

Approach is validated in a real project scenario with MIL and ESP

8.Conclusion

Outlook: some feasible future works

Technical cybersecurity concept creation

Automatic document generation from SysML models

Integrate model based attack tree

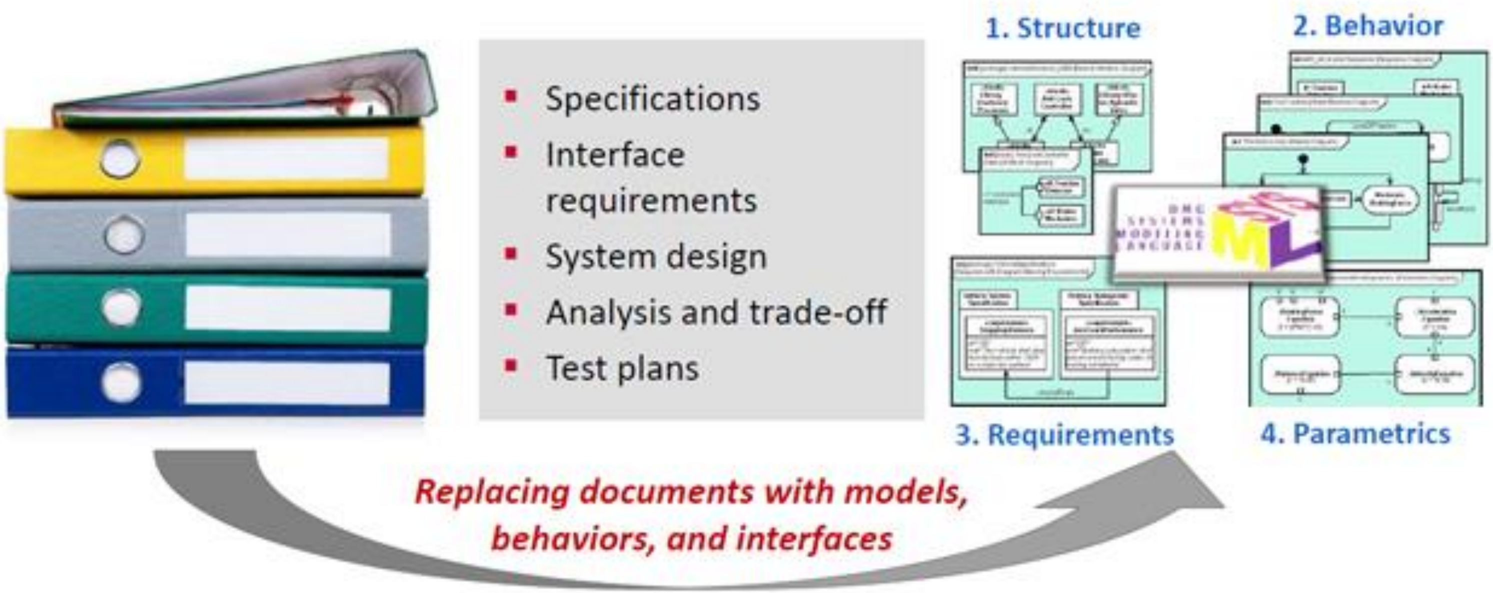
Increase the coverage of ISO26262 phases

Semi automated model based FMEA, RBD or FTA

https://press.zf.com/press/en/releases/release_8195.html

https://www.cisomag.com/upstream-security-partners-with-microsoft-to-defend-against-automotive-cyber-threats/

THANK YOU!!!!



- Specifications
- Interface requirements
- System design
- Analysis and trade-off
- Test plans

Replacing documents with models, behaviors, and interfaces

https://www.aras.com/de-de/resources/all/mbse-business-of-engineering-aras-plm

References

[1] Broy, Manfred; Damm, Werner; Henkler, Stefan; Pohl, Klaus; Vogelsang, Andreas; Weyer, Thorsten. Introduction to the SPES Modeling Framework. 31–49. 10.1007/978-3-642-34614-9_3.

[2] Aljoscha Lautenbach; Mafijul Islam. HEAVENS – HEAling Vulnerabilities to ENhance Software Security and Safety. Security Models. 2.0. The HEAVENS Consortium. Volvo Technology AB - BF 40700 Electrical and embedded systems; Fordonsutveckling/Vehicle Development (Research Program - Vinnova/FFI). March 18. 2016. https://www.vinnova.se/en/p/heavens-healing-vulnerabilities-to-enhance-software-security-and-safety/.