ECSEL
Joint Undertaking

European
Commission

AMASS

Architecture-driven, Multi-concern and Seamless Assurance and
Certification of Cyber-Physical Systems

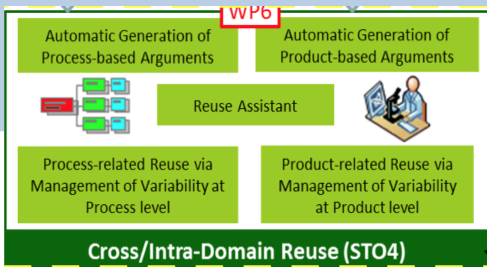
# AMASS Usage Scenario 2: Process & Product Configuration and Compliance Management

Second EAB Workshop
Västerås, Sept 17, 2018

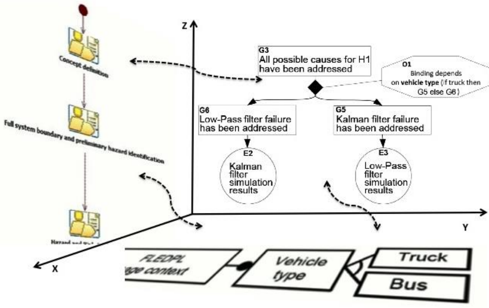
Barbara Gallina, Ph.D.
WP6 Leader, T6.1-2 Leader, TM

MÄLARDALEN UNIVERSITY
SWEDEN

# Intro: Cross-&-Intra Domain Reuse -AREAS

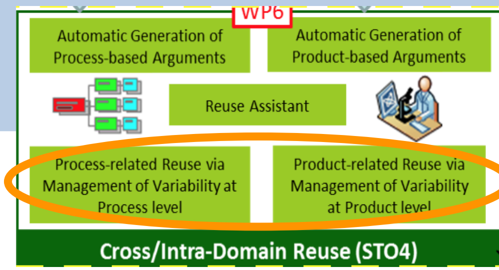

WP6
Automatic Generation of Process-based Arguments
Automatic Generation of Product-based Arguments
Reuse Assistant
Process-related Reuse via Management of Variability at Process level
Product-related Reuse via Management of Variability at Product level
Cross/Intra-Domain Reuse (STO4)

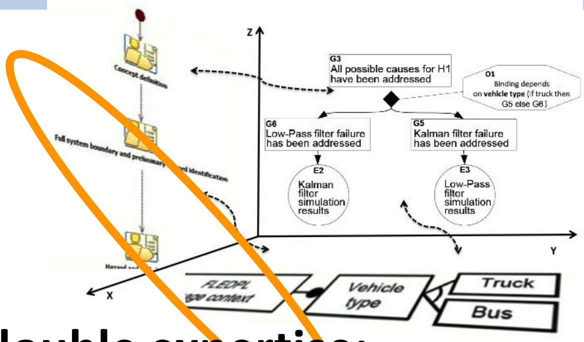
- Process/Product/Assurance Case Line Specification
  - Variability Management for Cross and Intra Domain reuse
    - Process (P1) –families/line of processes
    - Product (P1, partly) –families/line of products
    - Argumentation (P2) –families/line of arguments


Concept definition
Full system boundary and preliminary hazard identification
Hazard and ...
Z
Y
X
G3
All possible causes for H1 have been addressed
O1
Binding depends on vehicle type (if truck then G5 else G6)
G6
Low-Pass filter failure has been addressed
G5
Kalman filter failure has been addressed
E2
Kalman filter simulation results
E3
Low-Pass filter simulation results
FLEDPL
usage context
Vehicle type
Truck
Bus

- Measurement framework for Safety-oriented Process Line Engineering (SoPLE)
- Compliance management: further developed vision
- Argument fragments generation (Process and Product-based)
  - Towards fallacy-free process-based argumentation generation (P2)
- Semantics-based equivalent standards mapping
- Reuse assistant
  - Syntax-based Reuse Interface
  - Semantics-based Reuse Interface (P2)

# Assumptions

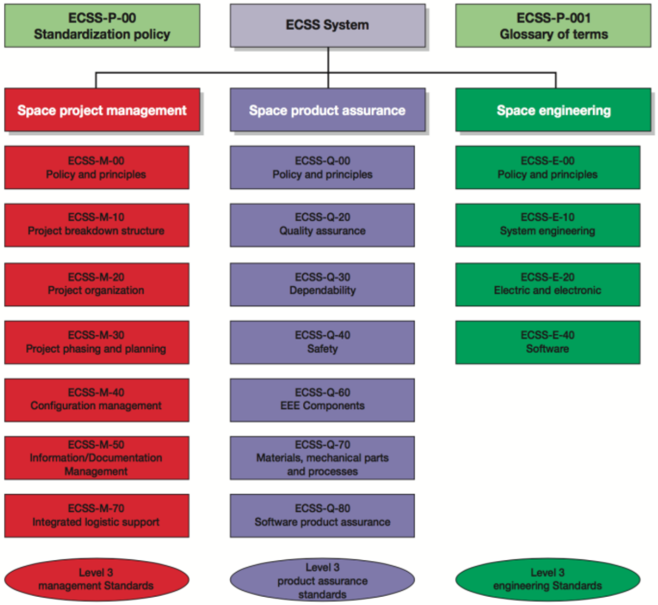

WP6
Automatic Generation of Process-based Arguments
Automatic Generation of Product-based Arguments
Reuse Assistant
Process-related Reuse via Management of Variability at Process level
Product-related Reuse via Management of Variability at Product level
Cross/Intra-Domain Reuse (STO4)


Z
Concept definition
Full system boundary and preliminary hazard identification
Hazard
G3
All possible causes for H1 have been addressed
O1
Binding depends on vehicle type (if truck then G5 else G6)
G6
Low-Pass filter failure has been addressed
G5
Kalman filter failure has been addressed
E2
Kalman filter simulation results
E3
Low-Pass filter simulation results
Y
X
FLEDPL age context
Vehicle type
Truck
Bus

- **Who am I in this scenario?**
  - **A technical engineer having double expertise:**
    - **process engineering (process engineer)**
    - **product engineering (better a designer)**
  - **Domain of expertise: space-related development processes and systems design**
  - **Standards: ECSS**
  - **My company produces families of systems**

**->systematic reuse can be beneficial**

  - **How I decide?**
    - **First, I embrace a measurement framework**
    - **Then, if positive, I adopt the approach**



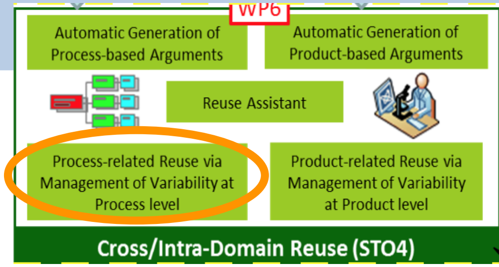
ECSS-P-00 Standardization policy

ECSS System

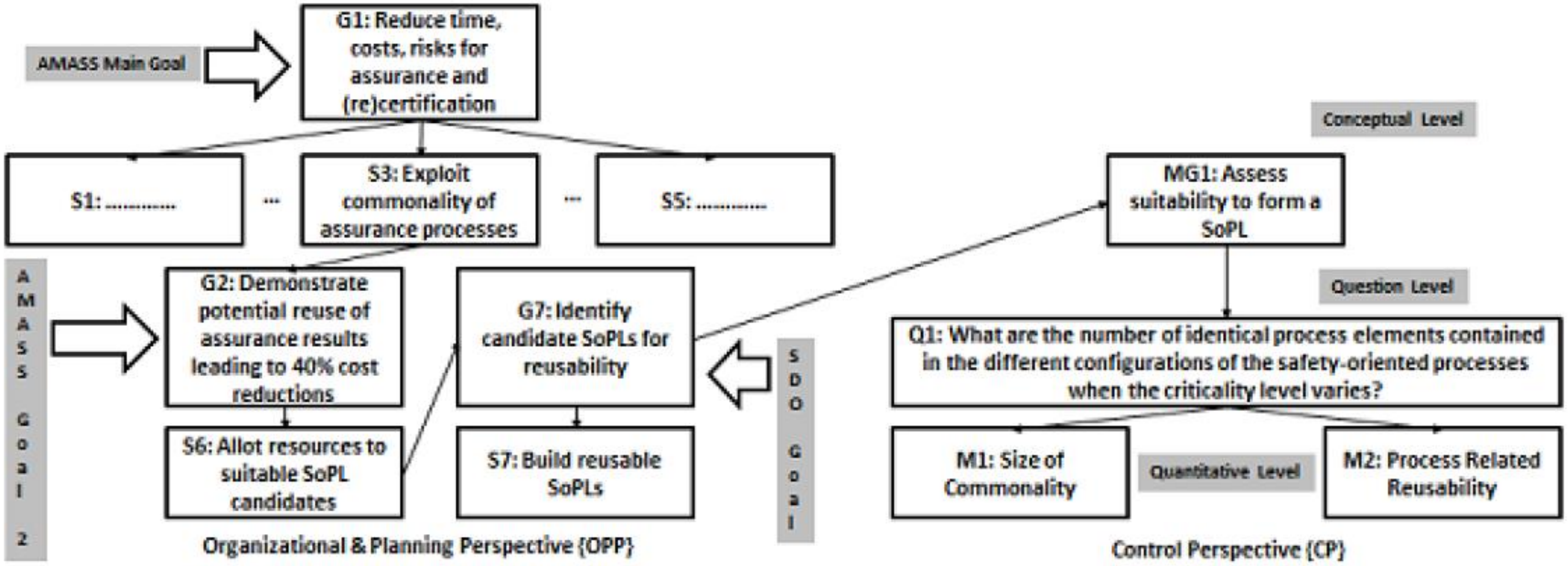
ECSS-P-001 Glossary of terms

| Space project management | Space product assurance | Space engineering |
|---|---|---|
| ECSS-M-00<br>Policy and principles | ECSS-Q-00<br>Policy and principles | ECSS-E-00<br>Policy and principles |
| ECSS-M-10<br>Project breakdown structure | ECSS-Q-20<br>Quality assurance | ECSS-E-10<br>System engineering |
| ECSS-M-20<br>Project organization | ECSS-Q-30<br>Dependability | ECSS-E-20<br>Electric and electronic |
| ECSS-M-30<br>Project phasing and planning | ECSS-Q-40<br>Safety | ECSS-E-40<br>Software |
| ECSS-M-40<br>Configuration management | ECSS-Q-60<br>EEE Components | |
| ECSS-M-50<br>Information/Documentation Management | ECSS-Q-70<br>Materials, mechanical parts and processes | |
| ECSS-M-70<br>Integrated logistic support | ECSS-Q-80<br>Software product assurance | |
| Level 3<br>management Standards | Level 3<br>product assurance standards | Level 3<br>engineering Standards |

# AMASS Goal: G1 & G2


WP6
Automatic Generation of Process-based Arguments
Automatic Generation of Product-based Arguments
Reuse Assistant
Process-related Reuse via Management of Variability at Process level
Product-related Reuse via Management of Variability at Product level
Cross/Intra-Domain Reuse (STO4)

**O3:** consolidate a ***cross-domain and intra-domain assurance*** reuse approach to improve mutual recognition agreement of compliance approvals and to help assess the return of investment of reuse decisions.


AMASS Main Goal

G1: Reduce time, costs, risks for assurance and (re)certification

S1: ............

...

S3: Exploit commonality of assurance processes

...

S5: ............

AMASS Goal 2

G2: Demonstrate potential reuse of assurance results leading to 40% cost reductions

G7: Identify candidate SoPLs for reusability

S6: Allot resources to suitable SoPL candidates

S7: Build reusable SoPLs

SDO Goal

Organizational & Planning Perspective (OPP)

Conceptual Level

MG1: Assess suitability to form a SoPL

Question Level

Q1: What are the number of identical process elements contained in the different configurations of the safety-oriented processes when the criticality level varies?

M1: Size of Commonality

Quantitative Level

M2: Process Related Reusability

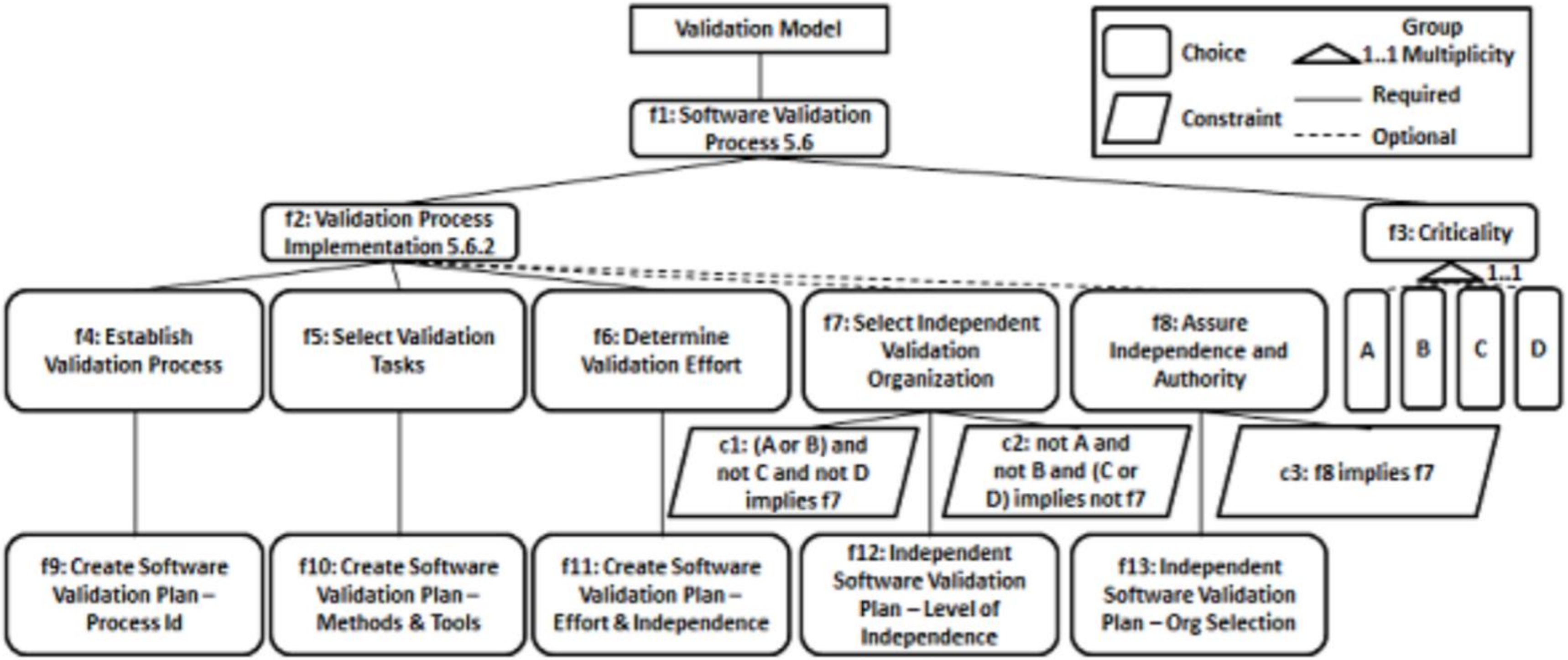
Control Perspective (CP)

**Measurement framework for SoPLE-SoPLE-targeted GQM Plus Strategies model**

$$(a)\ SoC = \left|\bigcap_{1}^{n} C_{\mathrm{pi}}\right| \qquad (b)\ P_{\mathrm{r}}R_{\mathrm{i}} = \frac{SoC}{|C_{\mathrm{pi}}|}$$

**B. Gallina and S. Iyer. Towards Quantitative Evaluation of Reuse within Safety-oriented Process Lines. 25th European & Asian Systems, Software & Service Process Improvement & Innovation (EuroSPI), Communications in Computer and Information Science, Springer, pp. 162-174, Bilbao, Spain, 5.-7. Sept. 2018.**

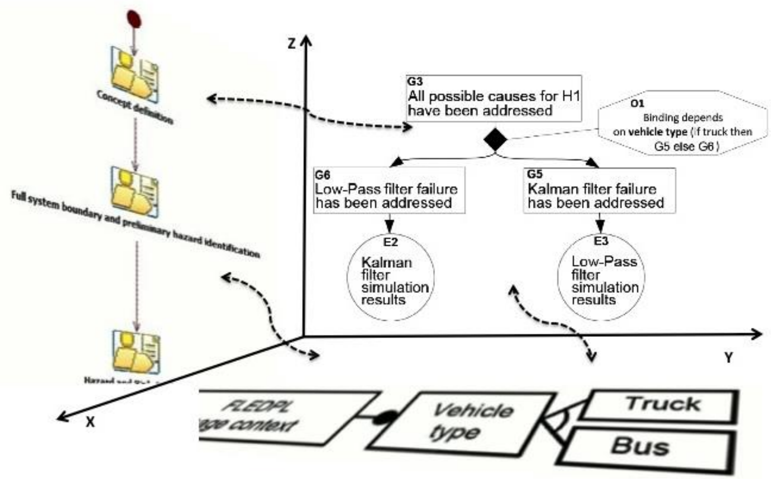
# Applying SoPLE-targeted GQM Plus Strategies model


Validation Model
f1: Software Validation Process 5.6
f2: Validation Process Implementation 5.6.2
f3: Criticality
1..1
f4: Establish Validation Process
f5: Select Validation Tasks
f6: Determine Validation Effort
f7: Select Independent Validation Organization
f8: Assure Independence and Authority
A
B
C
D
c1: (A or B) and not C and not D implies f7
c2: not A and not B and (C or D) implies not f7
c3: f8 implies f7
f9: Create Software Validation Plan – Process Id
f10: Create Software Validation Plan – Methods & Tools
f11: Create Software Validation Plan – Effort & Independence
f12: Independent Software Validation Plan – Level of Independence
f13: Independent Software Validation Plan – Org Selection
Choice
Constraint
Group 1..1 Multiplicity
Required
Optional

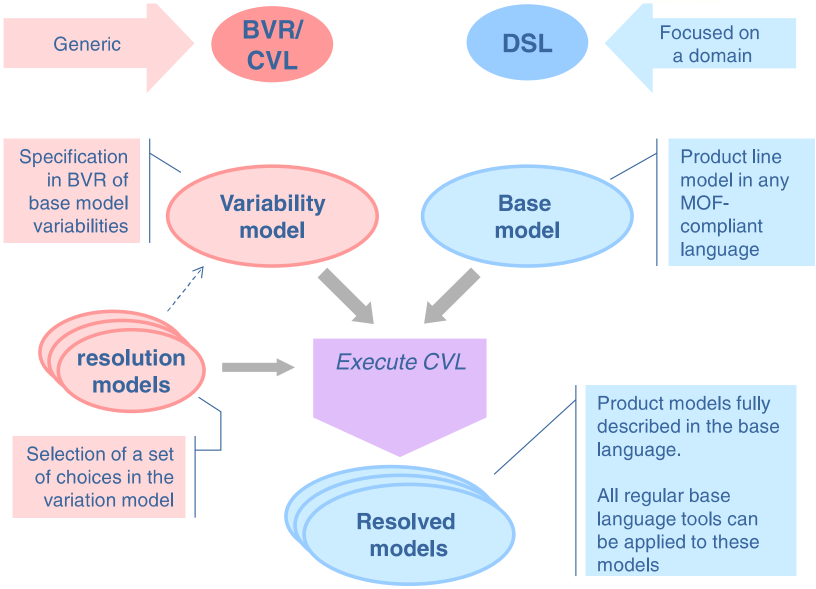
As we had 6 common elements, **SoC computes to 6**. We have at least 4 single processes.
The number of elements in the single processes for criticality levels A, B, C and D are 10 (f4 through f13), 10 (f4 through f13), 6 (f4, f5, f6, f9, f10 and f11) and 6 (f4, f5, f6, f9, f10 and f11) respectively.
Thus, **PrR's for single processes A, B, C and D are computed as 0.6, 0.6, 1 and 1 respectively**.

**B. Gallina and S. Iyer. Towards Quantitative Evaluation of Reuse within Safety-oriented Process Lines. 25th European & Asian Systems, Software & Service Process Improvement & Innovation (EuroSPI), Communications in Computer and Information Science, Springer, pp. 162-174, Bilbao, Spain, 5.-7. Sept. 2018.**

# Let’s go


Z
Concept definition
Full system boundary and preliminary hazard identification
Hazard and risk
G3
All possible causes for H1 have been addressed
O1
Binding depends on vehicle type (if truck then G5 else G6)
G6
Low-Pass filter failure has been addressed
G5
Kalman filter failure has been addressed
E2
Kalman filter simulation results
E3
Low-Pass filter simulation results
Y
X
FLEDPL age context
Vehicle type
Truck
Bus

- OpenCert
- PolarSys CHESS
- eclipse PROCESS FRAMEWORK


Generic

BVR/ CVL

DSL

Focused on a domain

Specification in BVR of base model variabilities

Variability model

Base model

Product line model in any MOF-compliant language

resolution models

Execute CVL

Selection of a set of choices in the variation model

Resolved models

Product models fully described in the base language.

All regular base language tools can be applied to these models

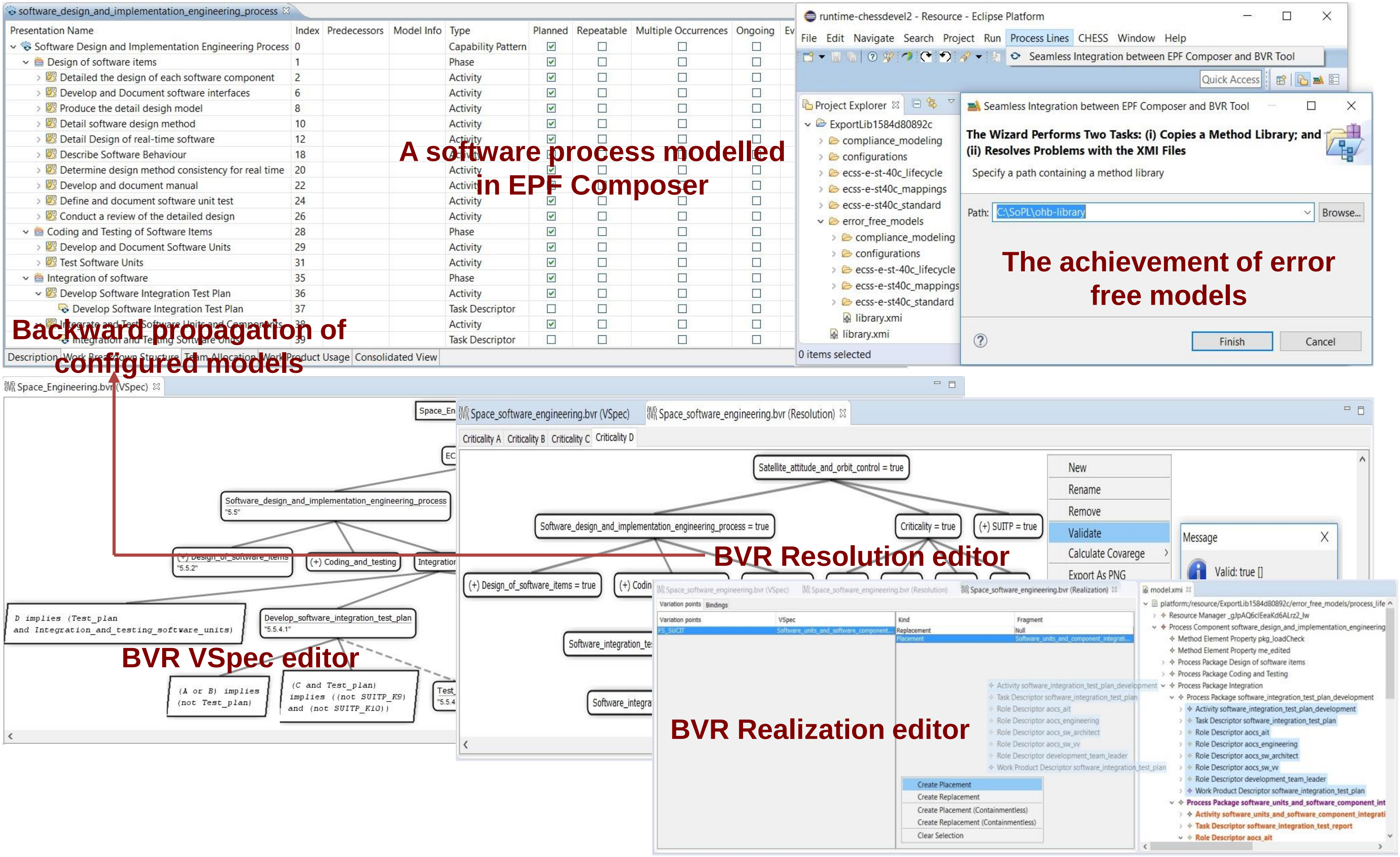
**Our DSL= X {UMA, CHESSML, CACM-arg}**

**Base model = X-compliant model**

**Resolved model = X-compliant model**

# Orthogonal variability management-P1

## Exemplification at Process Level –Paper presented at SPLC-2018/Tool track



software_design_and_implementation_engineering_process

| Presentation Name | Index | Predecessors | Model Info | Type | Planned | Repeatable | Multiple Occurrences | Ongoing | Ev |
|---|---|---|---|---|---|---|---|---|---|
| Software Design and Implementation Engineering Process | 0 | | | Capability Pattern | ☑ | ☐ | ☐ | ☐ | |
| Design of software items | 1 | | | Phase | ☑ | ☐ | ☐ | ☐ | |
| Detailed the design of each software component | 2 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Develop and Document software interfaces | 6 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Produce the detail design model | 8 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Detail software design method | 10 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Detail Design of real-time software | 12 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Describe Software Behaviour | 18 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Determine design method consistency for real time | 20 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Develop and document manual | 22 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Define and document software unit test | 24 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Conduct a review of the detailed design | 26 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Coding and Testing of Software Items | 28 | | | Phase | ☑ | ☐ | ☐ | ☐ | |
| Develop and Document Software Units | 29 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Test Software Units | 31 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Integration of software | 35 | | | Phase | ☑ | ☐ | ☐ | ☐ | |
| Develop Software Integration Test Plan | 36 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Develop Software Integration Test Plan | 37 | | | Task Descriptor | ☐ | ☐ | ☐ | ☐ | |
| Integrate and Test Software Units and Components | 38 | | | Activity | ☑ | ☐ | ☐ | ☐ | |
| Integration and Testing Software Units | 39 | | | Task Descriptor | ☐ | ☐ | ☐ | ☐ | |

A software process modelled in EPF Composer

runtime-chessdevel2 - Resource - Eclipse Platform

- ExportLib1584d80892c
  - compliance_modeling
  - configurations
  - ecss-e-st-40c_lifecycle
  - ecss-e-st40c_mappings
  - ecss-e-st40c_standard
  - error_free_models
    - compliance_modeling
    - configurations
    - ecss-e-st-40c_lifecycle
    - ecss-e-st40c_mappings
    - ecss-e-st40c_standard
    - library.xmi
  - library.xmi

0 items selected

Seamless Integration between EPF Composer and BVR Tool

**The Wizard Performs Two Tasks: (i) Copies a Method Library; and (ii) Resolves Problems with the XMI Files**

Specify a path containing a method library

Path: C:\SoPL\ohb-library

The achievement of error free models

Backward propagation of configured models

Space_Engineering.bvr (VSpec)

Space_En
EC
Software_design_and_implementation_engineering_process
"5.5"
(+) Design_of_software_items
"5.5.2"
(+) Coding_and_testing
Integration
D implies (Test_plan and Integration_and_testing_software_units)
Develop_software_integration_test_plan
"5.5.4.1"
(A or B) implies (not Test_plan)
(C and Test_plan) implies ((not SUITP_K9) and (not SUITP_K10))
Test_
"5.5.4

BVR VSpec editor

Space_software_engineering.bvr (VSpec)
Space_software_engineering.bvr (Resolution)

Message

Valid: true []

BVR Resolution editor

Space_software_engineering.bvr (VSpec)
Space_software_engineering.bvr (Resolution)
Space_software_engineering.bvr (Realization)

| Variation points | VSpec |
|---|---|
| FS_SUCIT | Software_units_and_software_component... |

| Kind | Fragment |
|---|---|
| Replacement | Null |
| Placement | Software_units_and_component_integrati... |

- Activity software_integration_test_plan_development
- Task Descriptor software_integration_test_plan
- Role Descriptor aocs_ait
- Role Descriptor aocs_engineering
- Role Descriptor aocs_sw_architect
- Role Descriptor aocs_sw_vv
- Role Descriptor development_team_leader
- Work Product Descriptor software_integration_test_plan

Create Placement
Create Replacement
Create Placement (Containmentless)
Create Replacement (Containmentless)
Clear Selection

BVR Realization editor

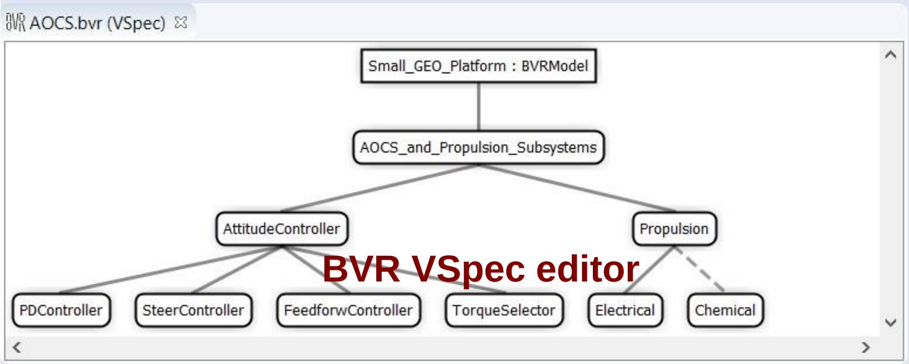
model.xmi

- platform:/resource/ExportLib1584d80892c/error_free_models/process_life
  - Resource Manager _gJpAQ6clEeaKd6ALrz2_lw
  - Process Component software_design_and_implementation_engineering
    - Method Element Property pkg_loadCheck
    - Method Element Property me_edited
    - Process Package Design of software items
    - Process Package Coding and Testing
    - Process Package Integration
      - Process Package software_integration_test_plan_development
        - Activity software_integration_test_plan_development
        - Task Descriptor software_integration_test_plan
        - Role Descriptor aocs_ait
        - Role Descriptor aocs_engineering
        - Role Descriptor aocs_sw_architect
        - Role Descriptor aocs_sw_vv
        - Role Descriptor development_team_leader
        - Work Product Descriptor software_integration_test_plan
      - Process Package software_units_and_software_component_int
        - Activity software_units_and_software_component_integrati
        - Task Descriptor software_integration_test_report
        - Role Descriptor aocs_ait

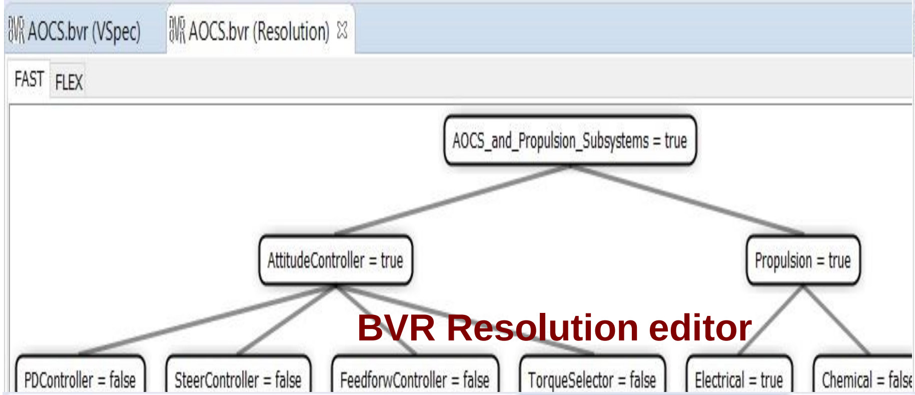
# Orthogonal variability management-P1

## Exemplification at Product Level

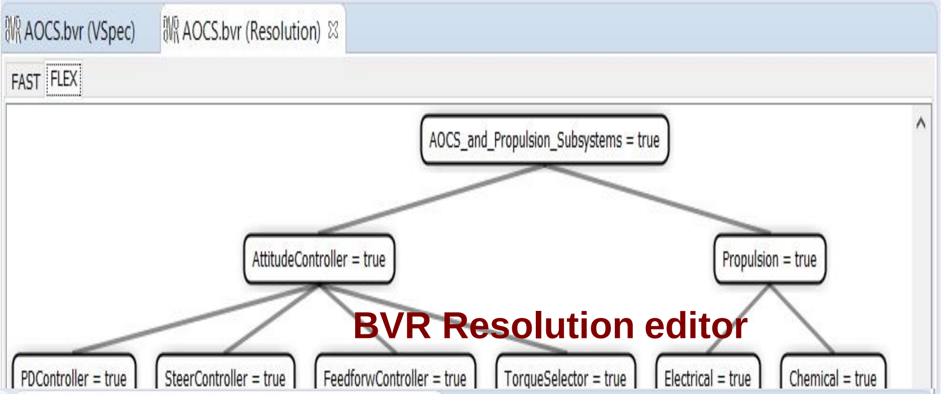
- Small GEO product line has two main configurations:
  - FAST with a combination of chemical and electrical propulsion
  - FLEX based on only electrical propulsion for both orbit transfer and station-keeping



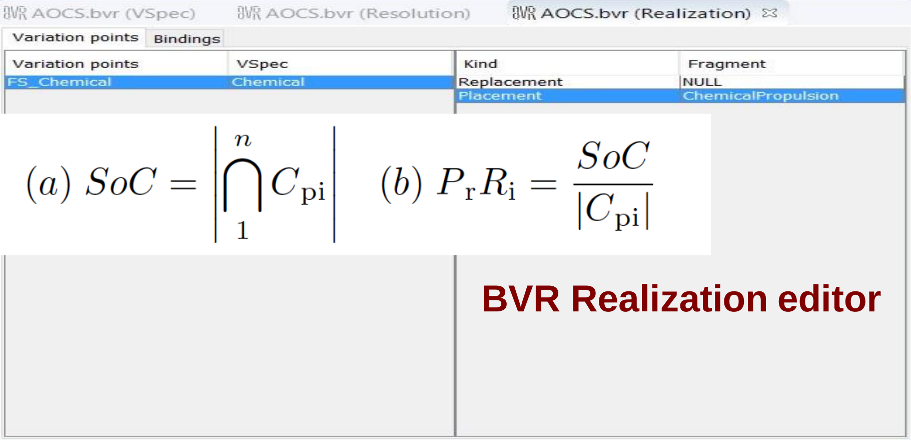
BVR VSpec editor



BVR Resolution editor



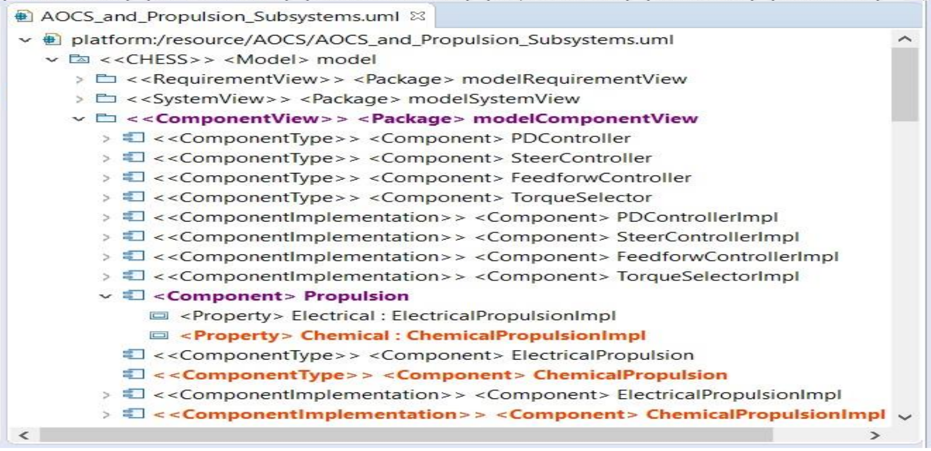
BVR Resolution editor



| Variation points | VSpec | Kind | Fragment |
|---|---|---|---|
| FS_Chemical | Chemical | Replacement | NULL |
| | | Placement | ChemicalPropulsion |

$$(a)\ SoC = \left|\bigcap_{1}^{n} C_{\mathrm{pi}}\right| \qquad (b)\ P_{\mathrm{r}}R_{\mathrm{i}} = \frac{SoC}{|C_{\mathrm{pi}}|}$$

BVR Realization editor

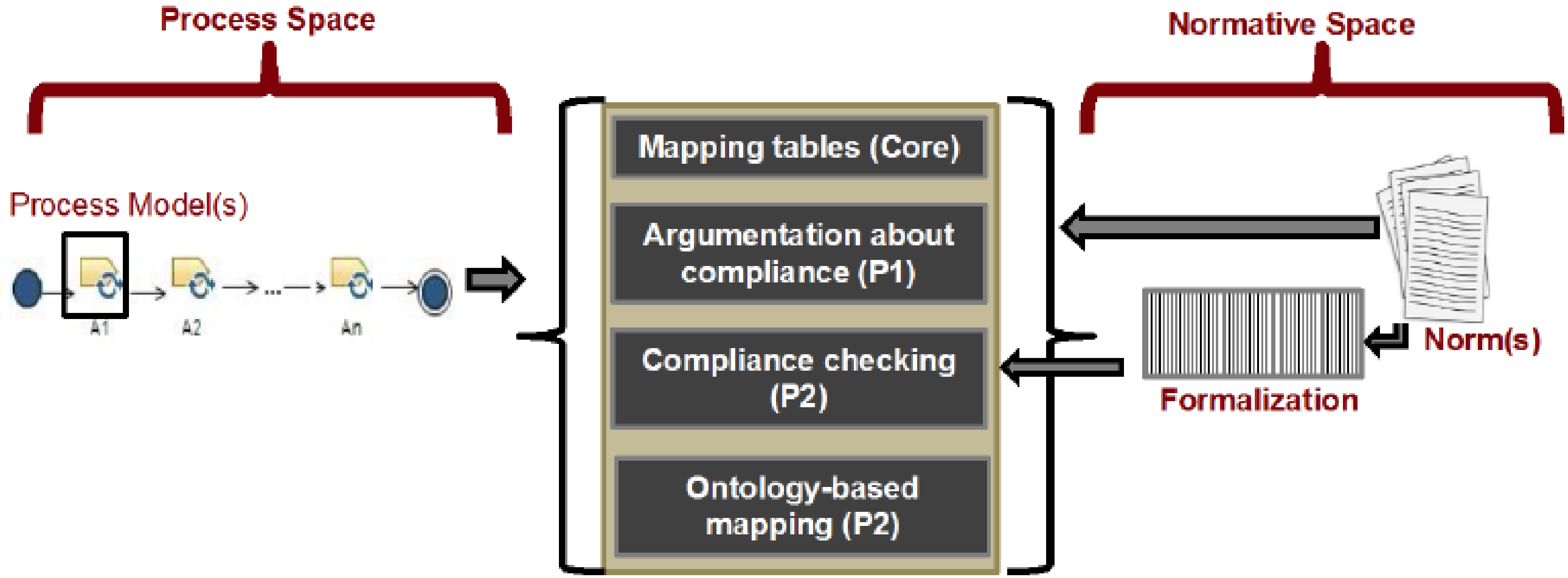

AOCS_and_Propulsion_Subsystems.uml

```
platform:/resource/AOCS/AOCS_and_Propulsion_Subsystems.uml
  <<CHESS>> <Model> model
    <<RequirementView>> <Package> modelRequirementView
    <<SystemView>> <Package> modelSystemView
    <<ComponentView>> <Package> modelComponentView
      <<ComponentType>> <Component> PDController
      <<ComponentType>> <Component> SteerController
      <<ComponentType>> <Component> FeedforwController
      <<ComponentType>> <Component> TorqueSelector
      <<ComponentImplementation>> <Component> PDControllerImpl
      <<ComponentImplementation>> <Component> SteerControllerImpl
      <<ComponentImplementation>> <Component> FeedforwControllerImpl
      <<ComponentImplementation>> <Component> TorqueSelectorImpl
      <Component> Propulsion
        <Property> Electrical : ElectricalPropulsionImpl
        <Property> Chemical : ChemicalPropulsionImpl
      <<ComponentType>> <Component> ElectricalPropulsion
      <<ComponentType>> <Component> ChemicalPropulsion
      <<ComponentImplementation>> <Component> ElectricalPropulsionImpl
      <<ComponentImplementation>> <Component> ChemicalPropulsionImpl
```

# Discussion

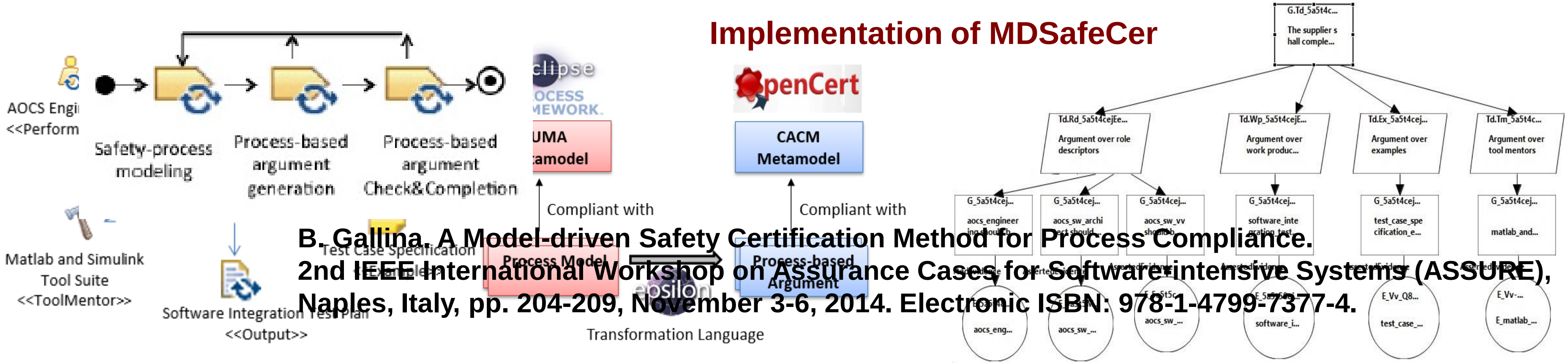
# Compliance management:

## vision and current development –vision presented at ASCS-2018



B. Gallina, F. Ul Muram, and J. P. Castellanos Ardila. Compliance of Agilized (Software) Development Processes with Safety. Proceedings of the 4th international workshop on agile development of safety-critical software (ASCS), co-located with XP 2018, May 21st, Porto, Portugal, 2018.

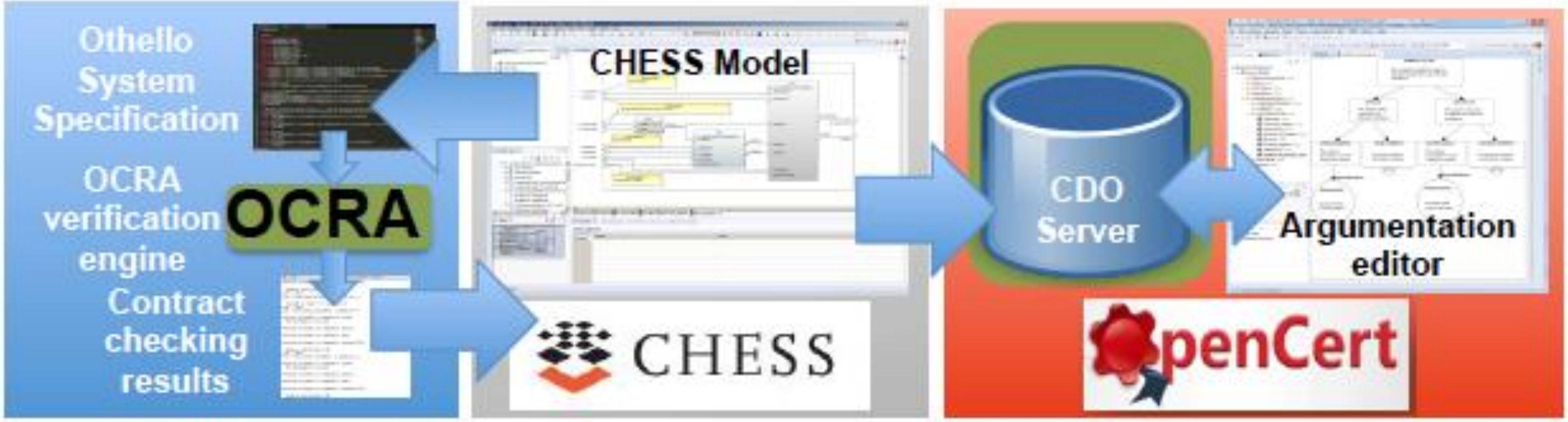
# Process&Product-based argument fragment generation –P1



## Implementation of MDSafeCer

**B. Gallina. A Model-driven Safety Certification Method for Process Compliance. 2nd IEEE International Workshop on Assurance Cases for Software-intensive Systems (ASSURE), Naples, Italy, pp. 204-209, November 3-6, 2014. Electronic ISBN: 978-1-4799-7377-4.**

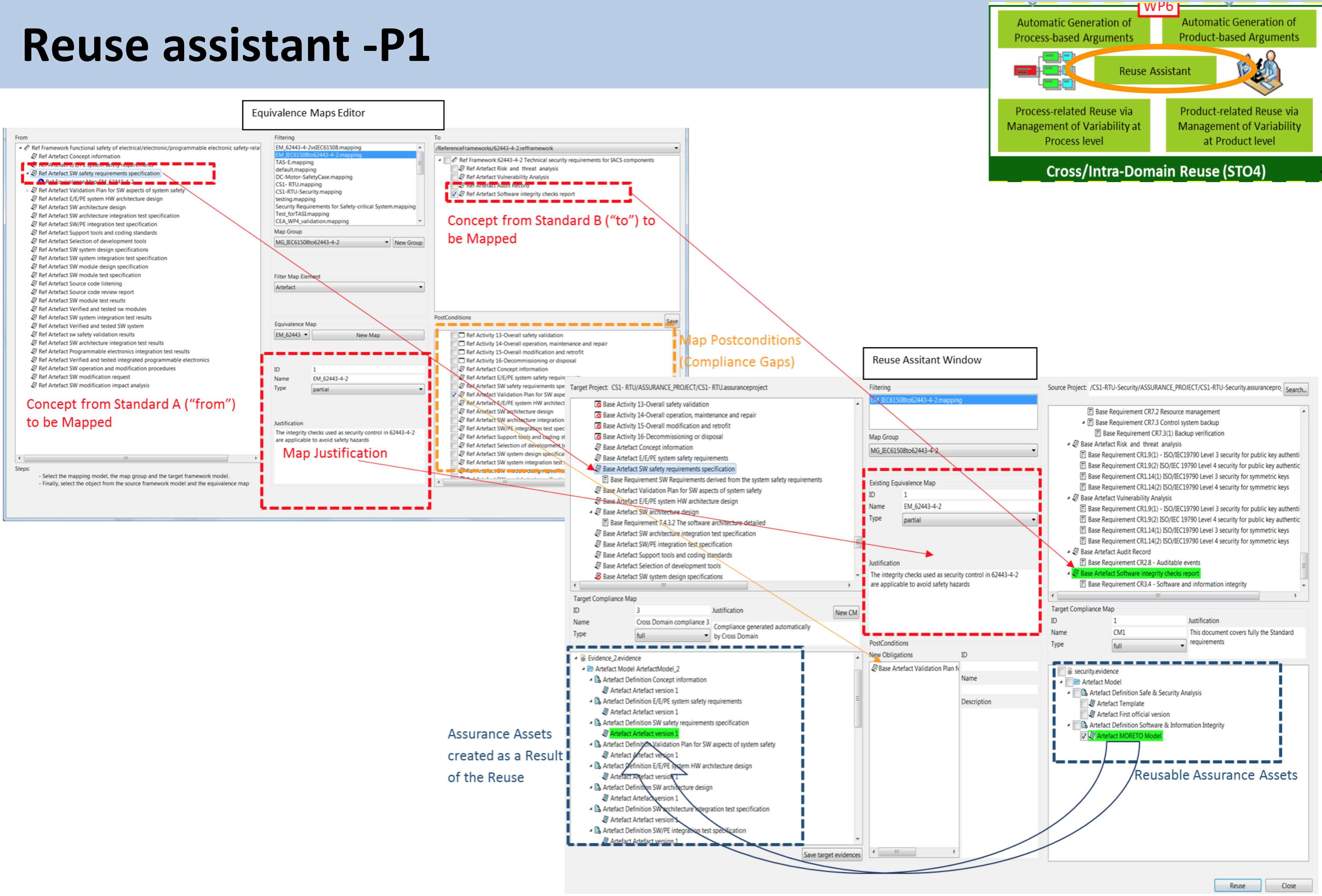
**F. UL Muram, B. Gallina and L. Gomez Rodriguez. Preventing Omission of Key Evidence Fallacy in Process-based Argumentations. 11th International Conference on the Quality of Information and Communications Technology (QUATIC), in press, Coimbra, Portugal, September 4-7, 2018**



**I. Sljivo, B. Gallina, J. Carlson, H. Hansson, S. Puri. Tool-Supported Safety-Relevant Component Reuse: From Specification to Argumentation. 23rd International Conference on Reliable Software Technologies (Ada-Europe), Lisbon, Portugal, June 18-22, 2018.**

# Second prototype (P1)
# Other functionalities

# Reuse assistant -P1



WP6
Automatic Generation of Process-based Arguments
Automatic Generation of Product-based Arguments
Reuse Assistant
Process-related Reuse via Management of Variability at Process level
Product-related Reuse via Management of Variability at Product level
Cross/Intra-Domain Reuse (STO4)

Equivalence Maps Editor

From

- Ref Framework Functional safety of electrical/electronic/programmable electronic safety-rela
  - Ref Artefact Concept information
  - Ref Artefact E/E/PE system safety requirements
  - Ref Artefact SW safety requirements specification
    - Ref Equivalence Map EM_62443-4-2
  - Ref Artefact Validation Plan for SW aspects of system safety
  - Ref Artefact E/E/PE system HW architecture design
  - Ref Artefact SW architecture design
  - Ref Artefact SW architecture integration test specification
  - Ref Artefact SW/PE integration test specification
  - Ref Artefact Support tools and coding standards
  - Ref Artefact Selection of development tools
  - Ref Artefact SW system design specifications
  - Ref Artefact SW system integration test specification
  - Ref Artefact SW module design specification
  - Ref Artefact SW module test specification
  - Ref Artefact Source code listening
  - Ref Artefact Source code review report
  - Ref Artefact SW module test results
  - Ref Artefact Verified and tested sw modules
  - Ref Artefact SW system integration test results
  - Ref Artefact Verified and tested SW system
  - Ref Artefact sw safety validation results
  - Ref Artefact SW architecture integration test results
  - Ref Artefact Programmable electronics integration test results
  - Ref Artefact Verified and tested integrated programmable electronics
  - Ref Artefact SW operation and modification procedures
  - Ref Artefact SW modification request
  - Ref Artefact SW modification impact analysis

Concept from Standard A ("from") to be Mapped

Steps:

- Select the mapping model, the map group and the target framework model.
- Finally, select the object from the source framework model and the equivalence map

Filtering

EM_62443-4-2vsIEC61508.mapping
EM_IEC61508to62443-4-2.mapping
TAS-E.mapping
default.mapping
DC-Motor-SafetyCase.mapping
CS1- RTU.mapping
CS1-RTU-Security.mapping
testing.mapping
Security Requirements for Safety-critical System.mapping
Test_forTASI.mapping
CEA_WP4_validation.mapping

Map Group: MG_IEC61508to62443-4-2 — New Group

Filter Map Element: Artefact

Equivalence Map: EM_62443 — New Map

ID: 1
Name: EM_62443-4-2
Type: partial

Justification: The integrity checks used as security control in 62443-4-2 are applicable to avoid safety hazards

Map Justification

To: /ReferenceFrameworks/62443-4-2.refframework

- □ Ref Framework 62443-4-2 Technical security requirements for IACS components
  - □ Ref Artefact Risk and threat analysis
  - □ Ref Artefact Vulnerability Analysis
  - □ Ref Artefact Audit Record
  - ☑ Ref Artefact Software integrity checks report

Concept from Standard B ("to") to be Mapped

PostConditions — Save

- □ Ref Activity 13-Overall safety validation
- □ Ref Activity 14-Overall operation, maintenance and repair
- □ Ref Activity 15-Overall modification and retrofit
- □ Ref Activity 16-Decommissioning or disposal
- □ Ref Artefact Concept information
- □ Ref Artefact E/E/PE system safety require
- □ Ref Artefact SW safety requirements spe
- ☑ Ref Artefact Validation Plan for SW aspe
- □ Ref Artefact E/E/PE system HW architect
- □ Ref Artefact SW architecture design
- □ Ref Artefact SW architecture integration
- □ Ref Artefact SW/PE integration test spec
- □ Ref Artefact Support tools and coding st
- □ Ref Artefact Selection of development t
- □ Ref Artefact SW system design specificat
- □ Ref Artefact SW system integration test
- □ Ref Artefact SW module design specifica

Map Postconditions (Compliance Gaps)

Reuse Assitant Window

Target Project: CS1- RTU/ASSURANCE_PROJECT/CS1- RTU.assuranceproject

- Base Activity 13-Overall safety validation
- Base Activity 14-Overall operation, maintenance and repair
- Base Activity 15-Overall modification and retrofit
- Base Activity 16-Decommissioning or disposal
- Base Artefact Concept information
- Base Artefact E/E/PE system safety requirements
- Base Artefact SW safety requirements specification
  - Base Requirement SW Requirements derived from the system safety requirements
- Base Artefact Validation Plan for SW aspects of system safety
- Base Artefact E/E/PE system HW architecture design
- Base Artefact SW architecture design
  - Base Requirement 7.4.3.2 The software architecture detailed
- Base Artefact SW architecture integration test specification
- Base Artefact SW/PE integration test specification
- Base Artefact Support tools and coding standards
- Base Artefact Selection of development tools
- Base Artefact SW system design specifications

Target Compliance Map — New CM

ID: 3
Name: Cross Domain compliance 3
Type: full
Justification: Compliance generated automatically by Cross Domain

- Evidence_2.evidence
  - Artefact Model ArtefactModel_2
    - Artefact Definition Concept information
      - Artefact Artefact version 1
    - Artefact Definition E/E/PE system safety requirements
      - Artefact Artefact version 1
    - Artefact Definition SW safety requirements specification
      - Artefact Artefact version 1
    - Artefact Definition Validation Plan for SW aspects of system safety
      - Artefact Artefact version 1
    - Artefact Definition E/E/PE system HW architecture design
      - Artefact Artefact version 1
    - Artefact Definition SW architecture design
      - Artefact Artefact version 1
    - Artefact Definition SW architecture integration test specification
      - Artefact Artefact version 1
    - Artefact Definition SW/PE integration test specification
      - Artefact Artefact version 1

Save target evidences

Assurance Assets created as a Result of the Reuse

Filtering: EM_IEC61508to62443-4-2.mapping

Map Group: MG_IEC61508to62443-4-2

Existing Equivalence Map
ID: 1
Name: EM_62443-4-2
Type: partial

Justification: The integrity checks used as security control in 62443-4-2 are applicable to avoid safety hazards

PostConditions

New Obligations: Base Artefact Validation Plan f

ID

Name

Description

Source Project: /CS1-RTU-Security/ASSURANCE_PROJECT/CS1-RTU-Security.assurancepro — Search...

- Base Requirement CR7.2 Resource management
- Base Requirement CR7.3 Control system backup
  - Base Requirement CR7.3(1) Backup verification
- Base Artefact Risk and threat analysis
  - Base Requirement CR1.9(1) - ISO/IEC19790 Level 3 security for public key authenti
  - Base Requirement CR1.9(2) ISO/IEC 19790 Level 4 security for public key authentic
  - Base Requirement CR1.14(1) ISO/IEC19790 Level 3 security for symmetric keys
  - Base Requirement CR1.14(2) ISO/IEC19790 Level 4 security for symmetric keys
- Base Artefact Vulnerability Analysis
  - Base Requirement CR1.9(1) - ISO/IEC19790 Level 3 security for public key authenti
  - Base Requirement CR1.9(2) ISO/IEC 19790 Level 4 security for public key authentic
  - Base Requirement CR1.14(1) ISO/IEC19790 Level 3 security for symmetric keys
  - Base Requirement CR1.14(2) ISO/IEC19790 Level 4 security for symmetric keys
- Base Artefact Audit Record
  - Base Requirement CR2.8 - Auditable events
- Base Artefact Software integrity checks report
  - Base Requirement CR3.4 - Software and information integrity

Target Compliance Map

ID: 1
Name: CM1
Type: full
Justification: This document covers fully the Standard requirements

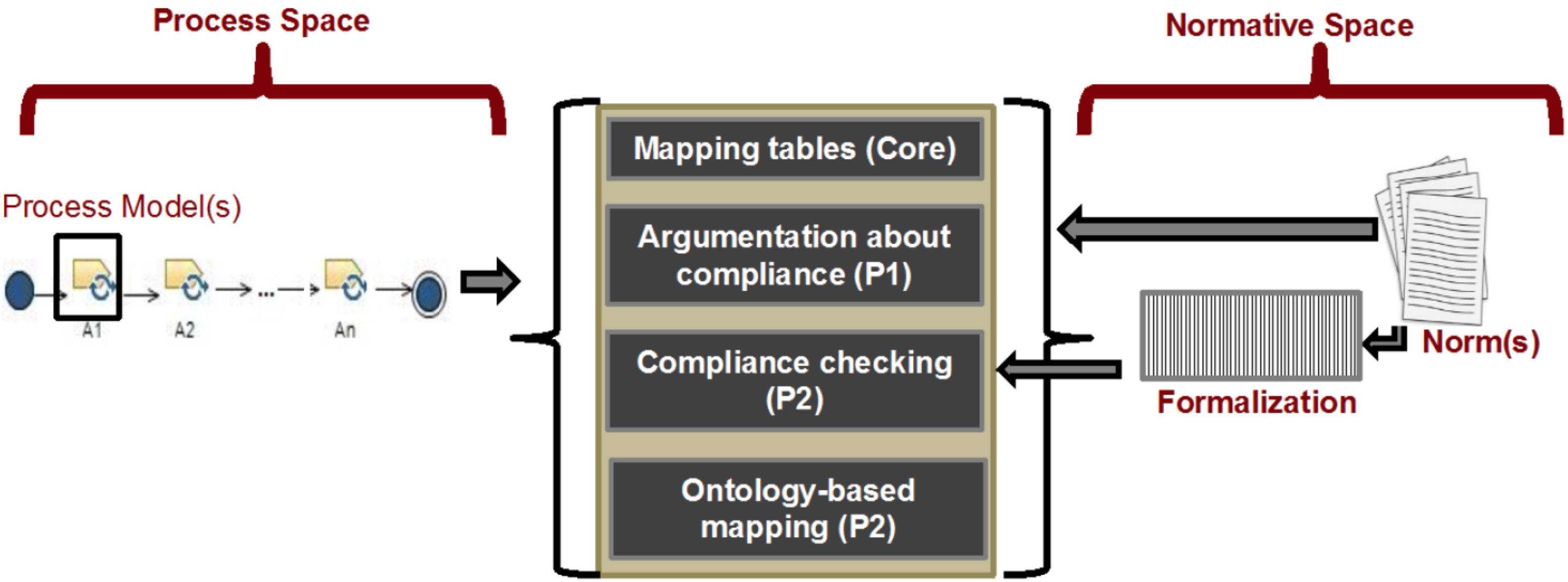
- □ security.evidence
  - □ Artefact Model
    - □ Artefact Definition Safe & Security Analysis
      - □ Artefact Template
      - □ Artefact First official version
    - □ Artefact Definition Software & Information Integrity
      - ☑ Artefact MORETO Model

Reusable Assurance Assets

Reuse — Close

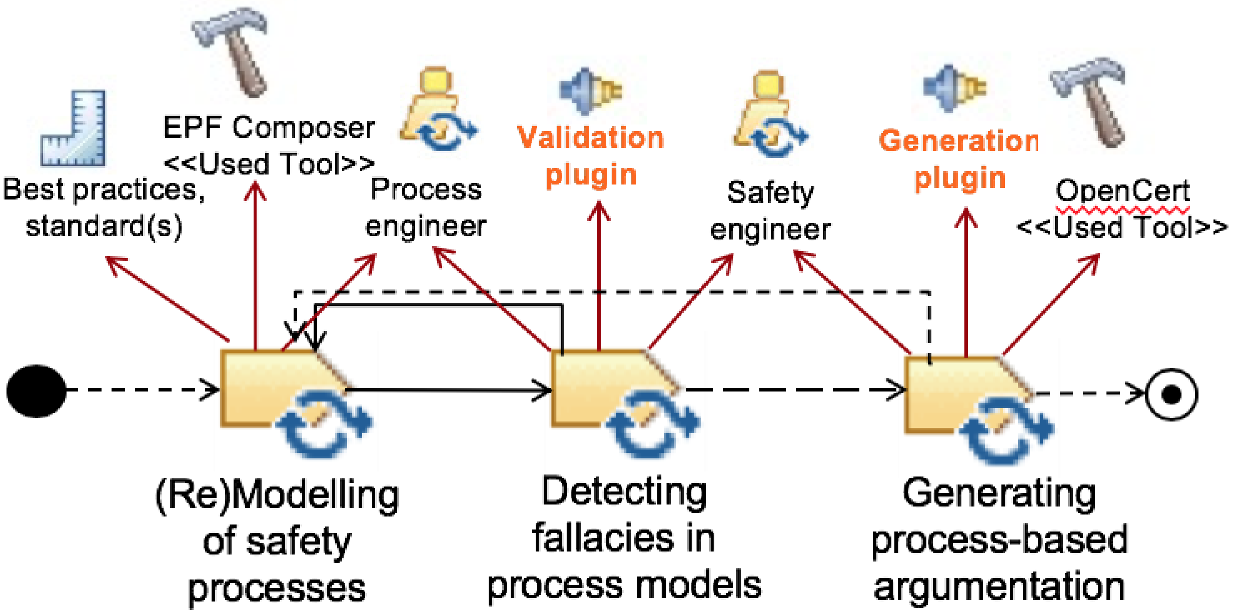
# Third prototype (P2)

# Process&Product-based argument fragment generation



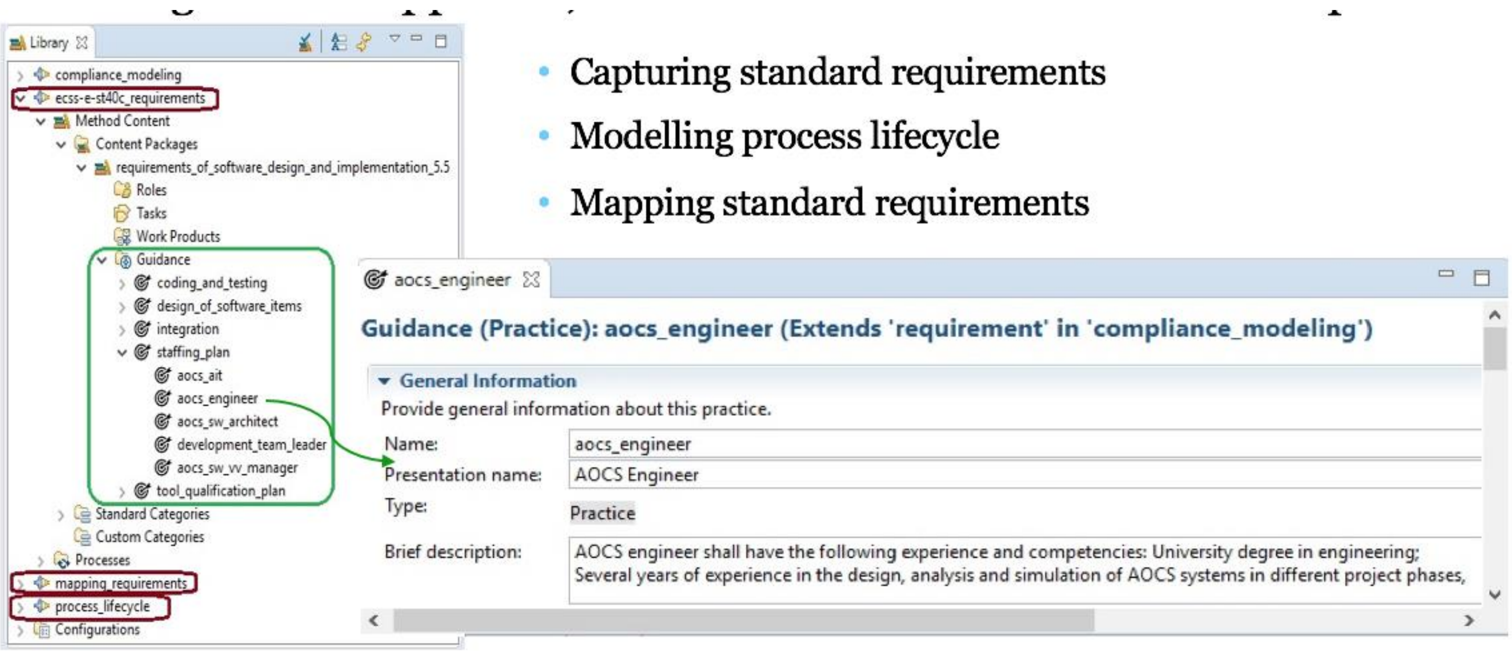
# Fallacy detection and process-based argument generation

**Extending/Empowering MDSafeCer**



F. UL Muram, B. Gallina and L. Gomez Rodriguez. Preventing Omission of Key Evidence Fallacy in Process-based Argumentations. 11th International Conference on the Quality of Information and Communications Technology (QUATIC), in press, Coimbra, Portugal, September 4-7, 2018

# Fallacy detection and process-based argument generation



- Capturing standard requirements
- Modelling process lifecycle
- Mapping standard requirements

**Guidance (Practice): aocs_engineer (Extends 'requirement' in 'compliance_modeling')**

**General Information**

Provide general information about this practice.

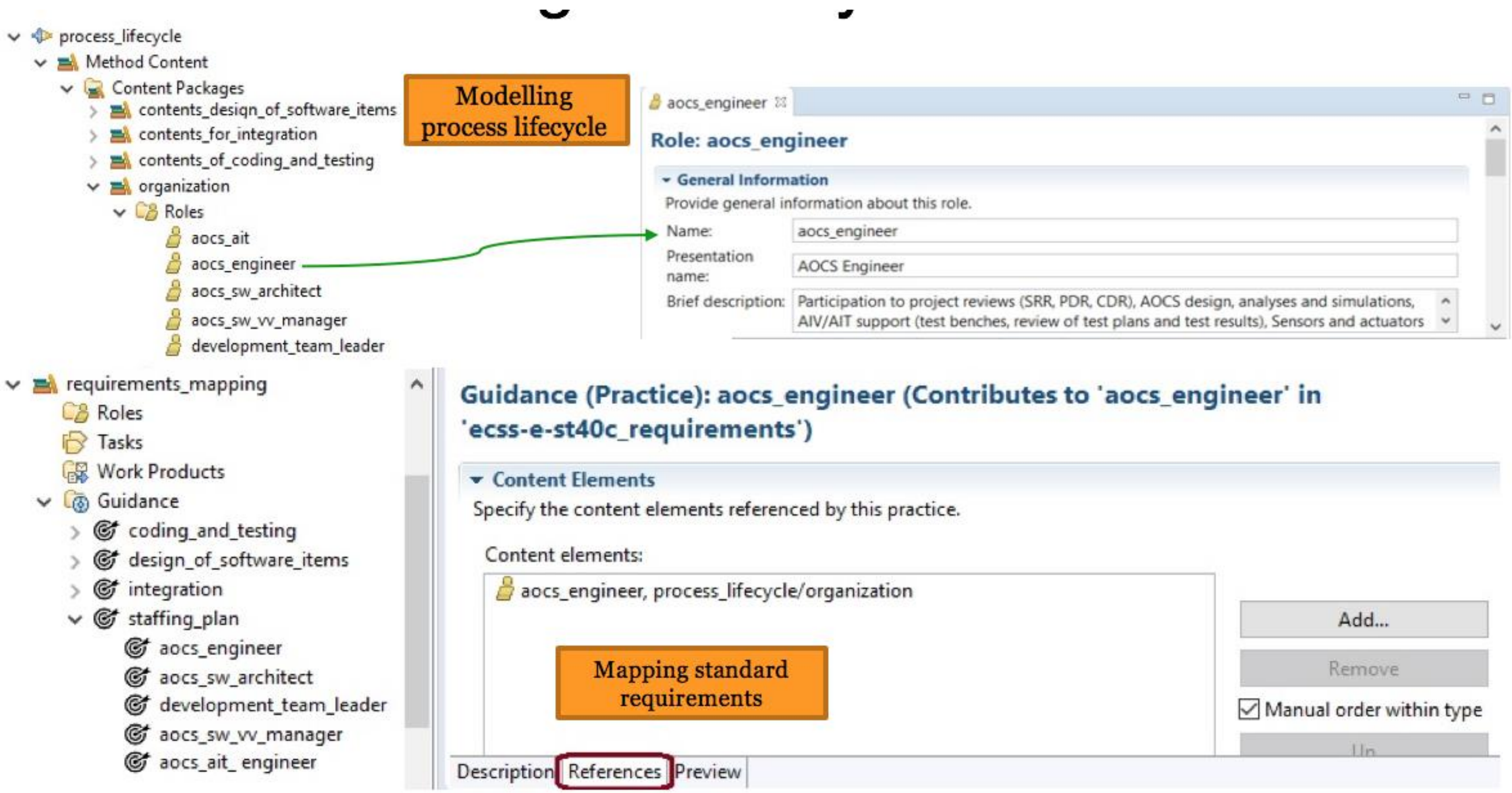
Name: aocs_engineer

Presentation name: AOCS Engineer

Type: Practice

Brief description: AOCS engineer shall have the following experience and competencies: University degree in engineering; Several years of experience in the design, analysis and simulation of AOCS systems in different project phases,

# Fallacy detection and process-based argument generation



- process_lifecycle
  - Method Content
    - Content Packages
      - contents_design_of_software_items
      - contents_for_integration
      - contents_of_coding_and_testing
      - organization
        - Roles
          - aocs_ait
          - aocs_engineer
          - aocs_sw_architect
          - aocs_sw_vv_manager
          - development_team_leader

Modelling process lifecycle

aocs_engineer

Role: aocs_engineer

General Information

Provide general information about this role.

Name: aocs_engineer

Presentation name: AOCS Engineer

Brief description: Participation to project reviews (SRR, PDR, CDR), AOCS design, analyses and simulations, AIV/AIT support (test benches, review of test plans and test results), Sensors and actuators

- requirements_mapping
  - Roles
  - Tasks
  - Work Products
  - Guidance
    - coding_and_testing
    - design_of_software_items
    - integration
    - staffing_plan
      - aocs_engineer
      - aocs_sw_architect
      - development_team_leader
      - aocs_sw_vv_manager
      - aocs_ait_ engineer

Guidance (Practice): aocs_engineer (Contributes to 'aocs_engineer' in 'ecss-e-st40c_requirements')

Content Elements

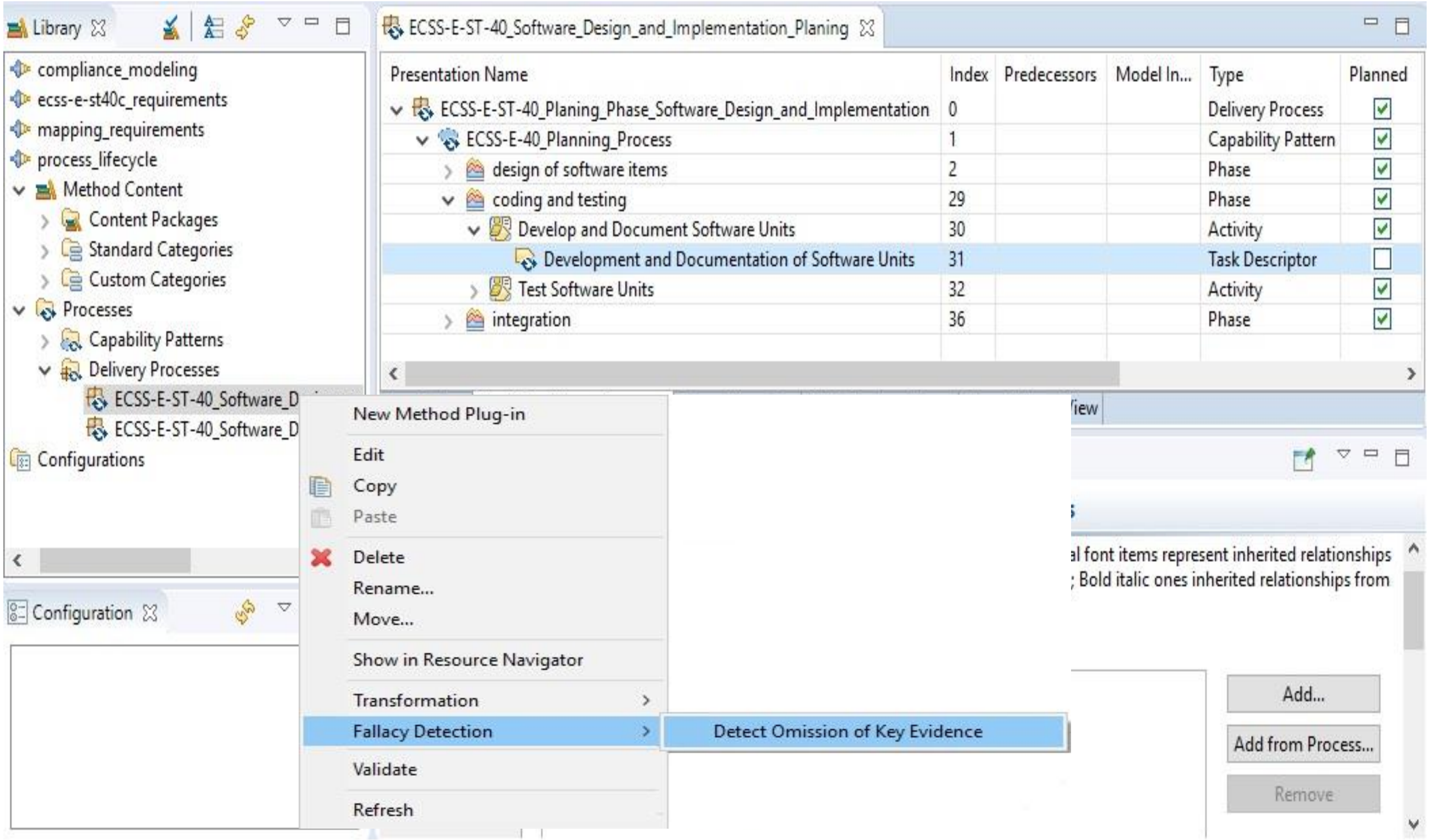
Specify the content elements referenced by this practice.

Content elements:

aocs_engineer, process_lifecycle/organization

Mapping standard requirements

# Fallacy detection and process-based argument generation



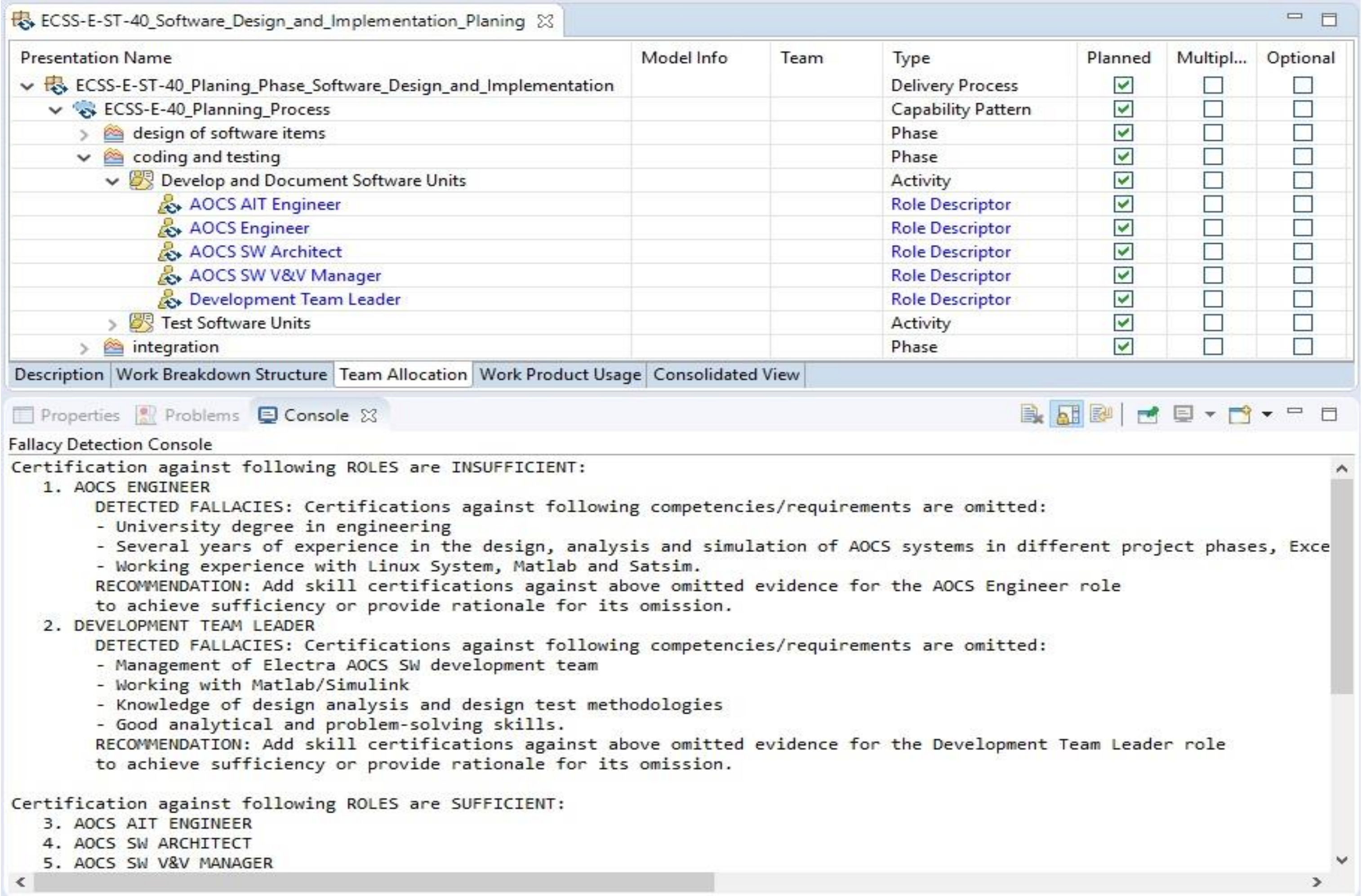
ECSS-E-ST-40_Software_Design_and_Implementation_Planing

| Presentation Name | Index | Predecessors | Model In... | Type | Planned |
|---|---|---|---|---|---|
| ECSS-E-ST-40_Planing_Phase_Software_Design_and_Implementation | 0 | | | Delivery Process | ☑ |
| ECSS-E-40_Planning_Process | 1 | | | Capability Pattern | ☑ |
| design of software items | 2 | | | Phase | ☑ |
| coding and testing | 29 | | | Phase | ☑ |
| Develop and Document Software Units | 30 | | | Activity | ☑ |
| Development and Documentation of Software Units | 31 | | | Task Descriptor | □ |
| Test Software Units | 32 | | | Activity | ☑ |
| integration | 36 | | | Phase | ☑ |

al font items represent inherited relationships

; Bold italic ones inherited relationships from

# Fallacy detection and process-based argument generation



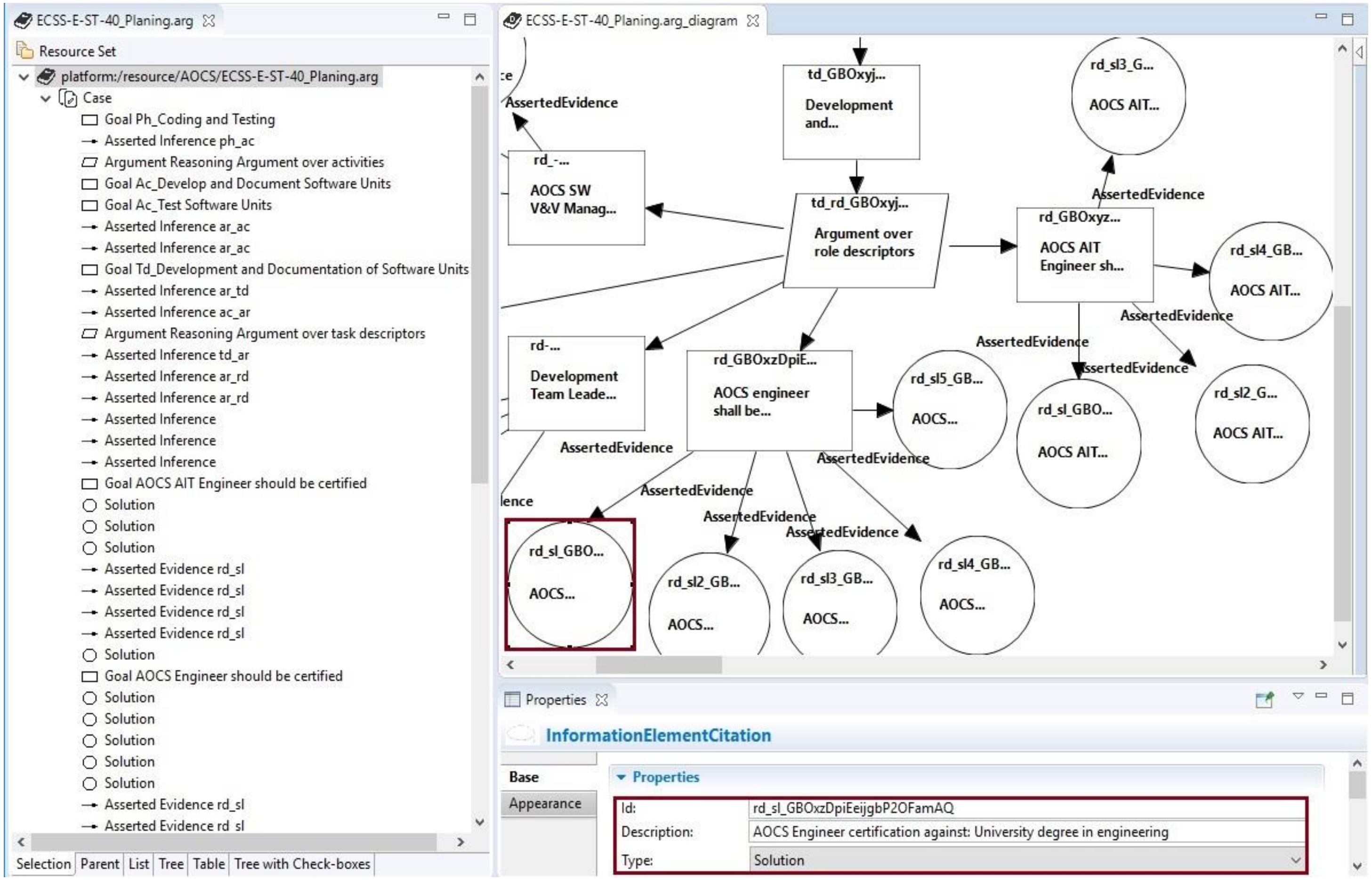
| Presentation Name | Model Info | Team | Type | Planned | Multipl... | Optional |
|---|---|---|---|---|---|---|
| ECSS-E-ST-40_Planing_Phase_Software_Design_and_Implementation | | | Delivery Process | ☑ | □ | □ |
| ECSS-E-40_Planning_Process | | | Capability Pattern | ☑ | □ | □ |
| design of software items | | | Phase | ☑ | □ | □ |
| coding and testing | | | Phase | ☑ | □ | □ |
| Develop and Document Software Units | | | Activity | ☑ | □ | □ |
| AOCS AIT Engineer | | | Role Descriptor | ☑ | □ | □ |
| AOCS Engineer | | | Role Descriptor | ☑ | □ | □ |
| AOCS SW Architect | | | Role Descriptor | ☑ | □ | □ |
| AOCS SW V&V Manager | | | Role Descriptor | ☑ | □ | □ |
| Development Team Leader | | | Role Descriptor | ☑ | □ | □ |
| Test Software Units | | | Activity | ☑ | □ | □ |
| integration | | | Phase | ☑ | □ | □ |

Fallacy Detection Console

```
Certification against following ROLES are INSUFFICIENT:
   1. AOCS ENGINEER
        DETECTED FALLACIES: Certifications against following competencies/requirements are omitted:
        - University degree in engineering
        - Several years of experience in the design, analysis and simulation of AOCS systems in different project phases, Exce
        - Working experience with Linux System, Matlab and Satsim.
        RECOMMENDATION: Add skill certifications against above omitted evidence for the AOCS Engineer role
        to achieve sufficiency or provide rationale for its omission.
   2. DEVELOPMENT TEAM LEADER
        DETECTED FALLACIES: Certifications against following competencies/requirements are omitted:
        - Management of Electra AOCS SW development team
        - Working with Matlab/Simulink
        - Knowledge of design analysis and design test methodologies
        - Good analytical and problem-solving skills.
        RECOMMENDATION: Add skill certifications against above omitted evidence for the Development Team Leader role
        to achieve sufficiency or provide rationale for its omission.

Certification against following ROLES are SUFFICIENT:
   3. AOCS AIT ENGINEER
   4. AOCS SW ARCHITECT
   5. AOCS SW V&V MANAGER
```

# Fallacy detection and process-based argument generation



platform:/resource/AOCS/ECSS-E-ST-40_Planing.arg

- Case
  - □ Goal Ph_Coding and Testing
  - → Asserted Inference ph_ac
  - ▱ Argument Reasoning Argument over activities
  - □ Goal Ac_Develop and Document Software Units
  - □ Goal Ac_Test Software Units
  - → Asserted Inference ar_ac
  - → Asserted Inference ar_ac
  - □ Goal Td_Development and Documentation of Software Units
  - → Asserted Inference ar_td
  - → Asserted Inference ac_ar
  - ▱ Argument Reasoning Argument over task descriptors
  - → Asserted Inference td_ar
  - → Asserted Inference ar_rd
  - → Asserted Inference ar_rd
  - → Asserted Inference
  - → Asserted Inference
  - → Asserted Inference
  - □ Goal AOCS AIT Engineer should be certified
  - ○ Solution
  - ○ Solution
  - ○ Solution
  - → Asserted Evidence rd_sl
  - → Asserted Evidence rd_sl
  - → Asserted Evidence rd_sl
  - → Asserted Evidence rd_sl
  - ○ Solution
  - □ Goal AOCS Engineer should be certified
  - ○ Solution
  - ○ Solution
  - ○ Solution
  - ○ Solution
  - ○ Solution
  - → Asserted Evidence rd_sl
  - → Asserted Evidence rd_sl

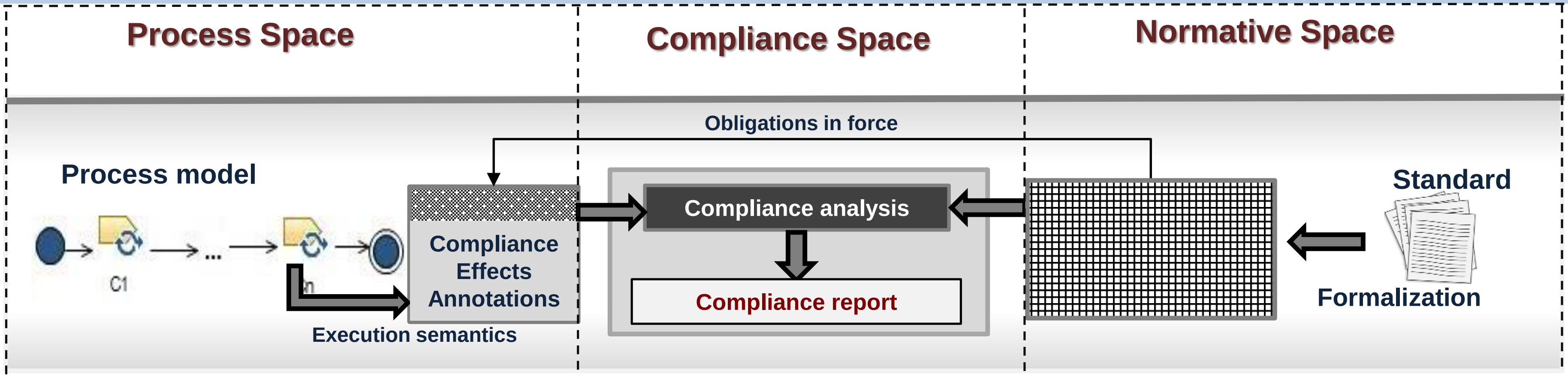
InformationElementCitation

Properties

| | |
|---|---|
| Id: | rd_sl_GBOxzDpiEeijgbP2OFamAQ |
| Description: | AOCS Engineer certification against: University degree in engineering |
| Type: | Solution |

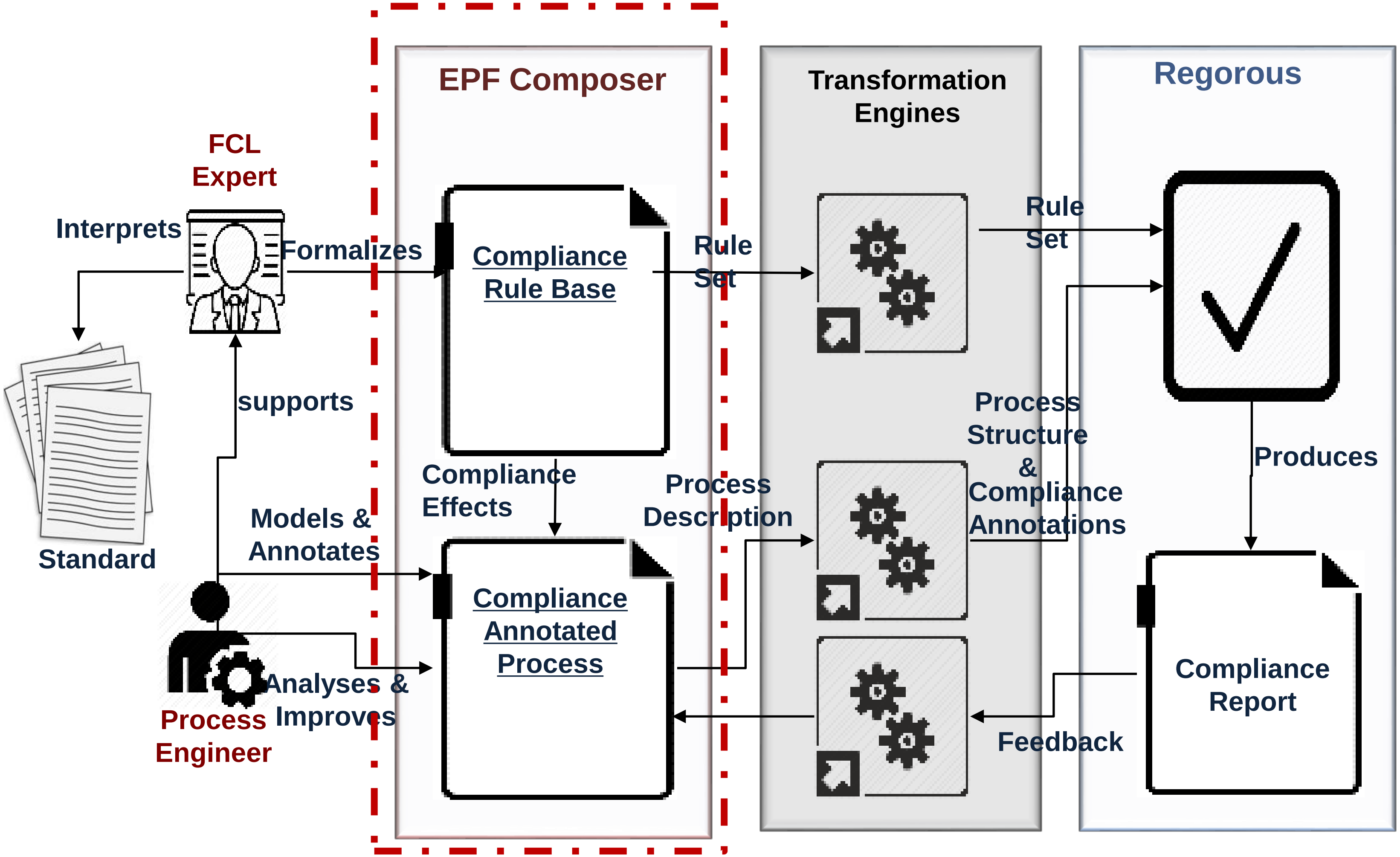
# Discussion

# Compliance checking + patterns



J. P. Castellanos Ardila and B. Gallina. Formal Contract Logic Based Patterns for Facilitating Compliance Checking against ISO 26262. Proceedings of the 1st Workshop on Technologies for Regulatory Compliance co-located with the 30th International Conference on Legal Knowledge and Information Systems (JURIX 2017), CEUR Workshop Proceedings, Vol-2049, pp. 65-72, Luxembourg, Luxembourg, 13 of December, 2017.

# Automated Compliance Checking Vision

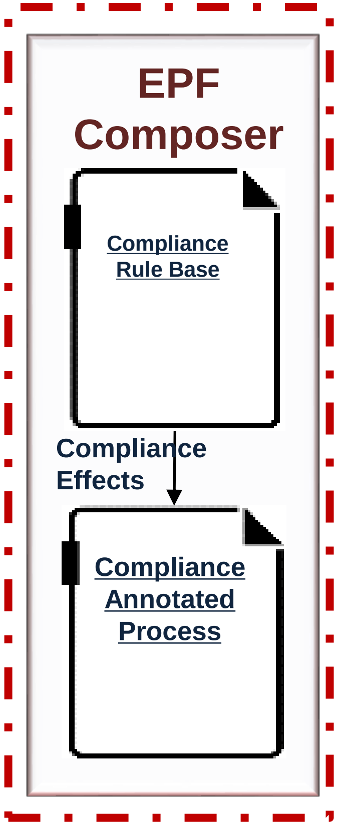


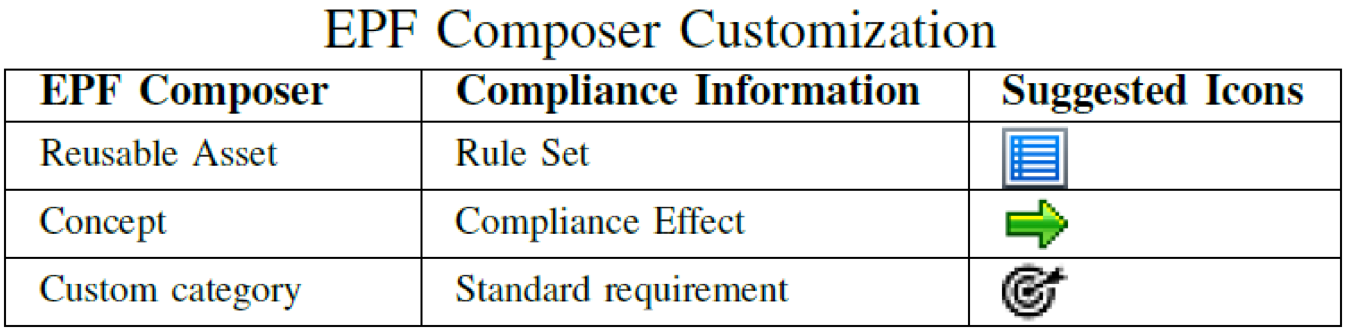
J. P. Castellanos Ardila and B. Gallina and F. Ul Muram. Enabling Compliance Checking against Safety Standards from SPEM 2.0 Process Models. 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Prague, Czech republic, 29-31 August, 2018.

J. P. Castellanos Ardila and B. Gallina and F. Ul Muram. Transforming SPEM 2.0-compatible Process Models into Models Checkable for Compliance. 18th International SPICE Conference (SPICE), Thessaloniki, Greece, October 9-10, 2018.

# Automated Compliance Checking Vision

## EPF Composer Modeling Capabilities





EPF Composer Customization

| EPF Composer | Compliance Information | Suggested Icons |
|---|---|---|
| Reusable Asset | Rule Set | ☰ |
| Concept | Compliance Effect | ➡ |
| Custom category | Standard requirement | 🎯 |

## AMASS Version of the Standards Mapping Method

**Standards requirements**

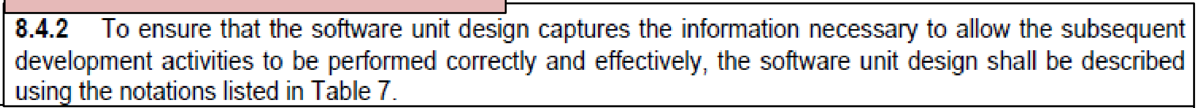
**Lifecycle elements**

**Annotated Process**

# Automated Compliance Checking Vision

## Plugin modeling

**Rules formalization** [Castellanos2017]



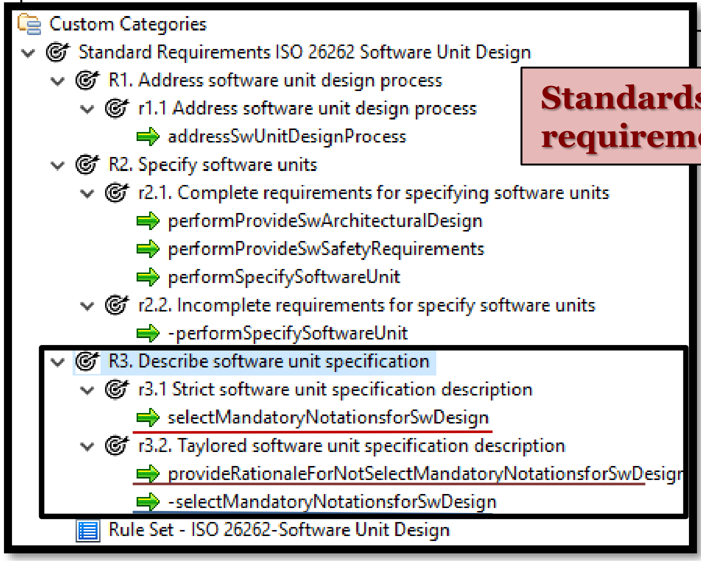
8.4.2 To ensure that the software unit design captures the information necessary to allow the subsequent development activities to be performed correctly and effectively, the software unit design shall be described using the notations listed in Table 7.

$$r_3 : performSpecifySwUnit \Rightarrow [O]\, selectMandatoryNotationsforSwDesign$$

$$r'_3 : provideRationaleForNotSelectMandatoryNotationsforSwDesign \Rightarrow [P] - selectMandatoryNotationsforSwDesign$$

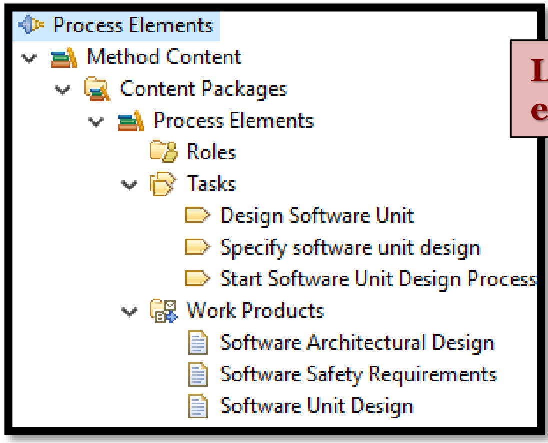
$$r'_3 > r_3$$

**Standards requirements**



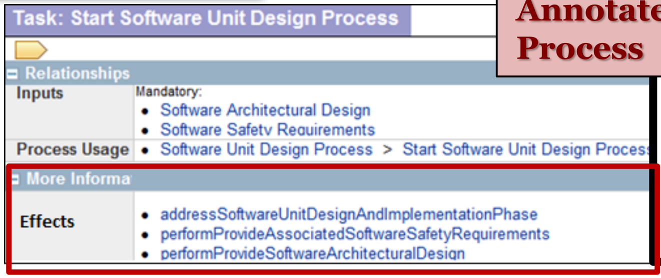
- Custom Categories
  - Standard Requirements ISO 26262 Software Unit Design
    - R1. Address software unit design process
      - r1.1 Address software unit design process
        - addressSwUnitDesignProcess
    - R2. Specify software units
      - r2.1. Complete requirements for specifying software units
        - performProvideSwArchitecturalDesign
        - performProvideSwSafetyRequirements
        - performSpecifySoftwareUnit
      - r2.2. Incomplete requirements for specify software units
        - -performSpecifySoftwareUnit
    - R3. Describe software unit specification
      - r3.1 Strict software unit specification description
        - selectMandatoryNotationsforSwDesign
      - r3.2. Taylored software unit specification description
        - provideRationaleForNotSelectMandatoryNotationsforSwDesign
        - -selectMandatoryNotationsforSwDesign
  - Rule Set - ISO 26262-Software Unit Design

**Lifecycle elements**



- Process Elements
  - Method Content
    - Content Packages
      - Process Elements
        - Roles
        - Tasks
          - Design Software Unit
          - Specify software unit design
          - Start Software Unit Design Process
        - Work Products
          - Software Architectural Design
          - Software Safety Requirements
          - Software Unit Design

**Annotated Process**



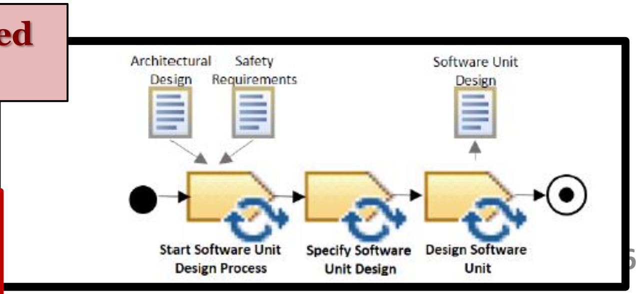
Task: Start Software Unit Design Process

Relationships

| | |
|---|---|
| Inputs | Mandatory:<br>• Software Architectural Design<br>• Software Safety Requirements |
| Process Usage | • Software Unit Design Process > Start Software Unit Design Process |

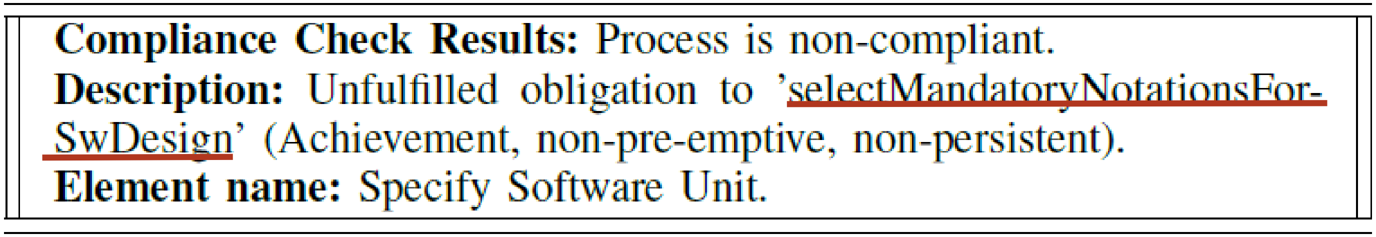
More Informa

| | |
|---|---|
| Effects | • addressSoftwareUnitDesignAndImplementationPhase<br>• performProvideAssociatedSoftwareSafetyRequirements<br>• performProvideSoftwareArchitecturalDesign |

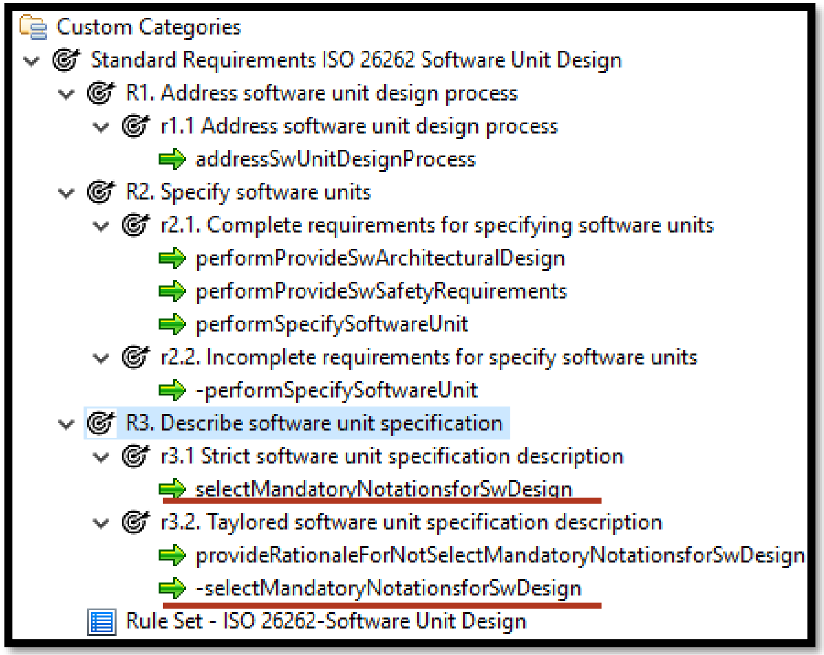


# Automated Compliance Checking Vision

## Regorous report



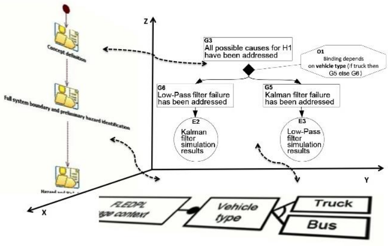
**Compliance Check Results:** Process is non-compliant.
**Description:** Unfulfilled obligation to ’selectMandatoryNotationsForSwDesign’ (Achievement, non-pre-emptive, non-persistent).
**Element name:** Specify Software Unit.



- Custom Categories
- Standard Requirements ISO 26262 Software Unit Design
  - R1. Address software unit design process
    - r1.1 Address software unit design process
      - addressSwUnitDesignProcess
  - R2. Specify software units
    - r2.1. Complete requirements for specifying software units
      - performProvideSwArchitecturalDesign
      - performProvideSwSafetyRequirements
      - performSpecifySoftwareUnit
    - r2.2. Incomplete requirements for specify software units
      - -performSpecifySoftwareUnit
  - R3. Describe software unit specification
    - r3.1 Strict software unit specification description
      - selectMandatoryNotationsforSwDesign
    - r3.2. Taylored software unit specification description
      - provideRationaleForNotSelectMandatoryNotationsforSwDesign
      - -selectMandatoryNotationsforSwDesign
  - Rule Set - ISO 26262-Software Unit Design

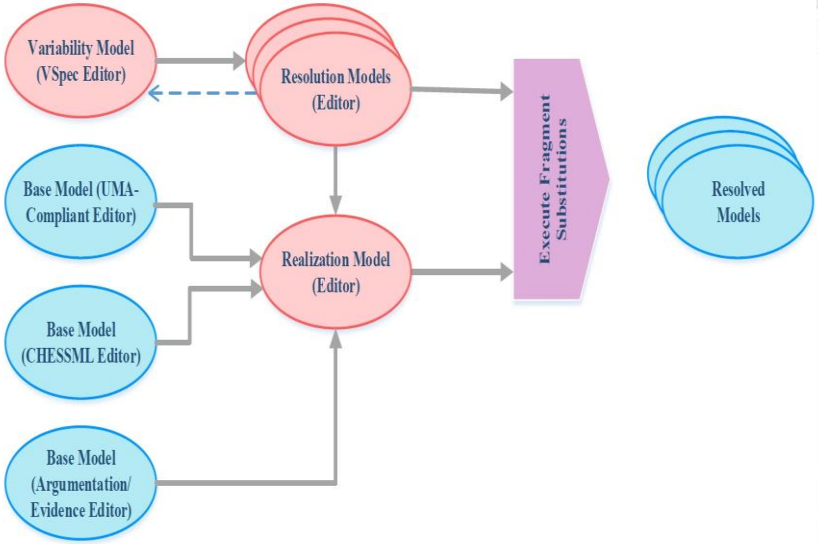
# Discussion

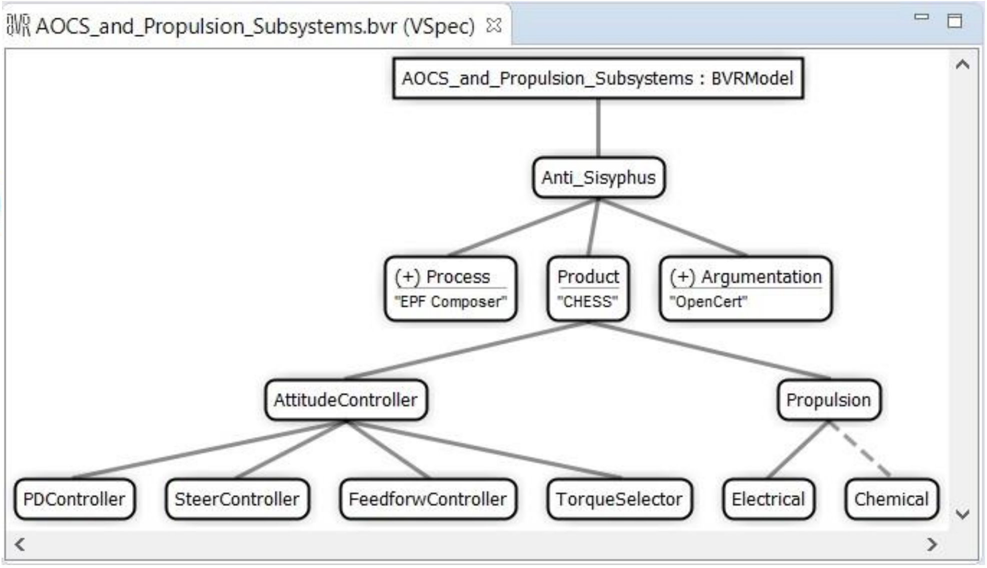
# Anti-Sisyphus via UMA, CHESSML, CACM and BVR


Concept definition
Full system boundary and preliminary hazard identification
Hazard and ...
Z
G3
All possible causes for H1 have been addressed
O1
Binding depends on vehicle type (if truck then G5 else G6)
G6
Low-Pass filter failure has been addressed
G5
Kalman filter failure has been addressed
E2
Kalman filter simulation results
E3
Low-Pass filter simulation results
Y
X
FLEDPL ...ge context
Vehicle type
Truck
Bus

B. Gallina. 2015. Towards Enabling Reuse in the Context of Safety-Critical Product Lines. In 5th IEEE/ACM International Workshop on Product Line Approaches in Software Engineering, PLEASE 2015, Florence, Italy, 15–18 May 19, 2015

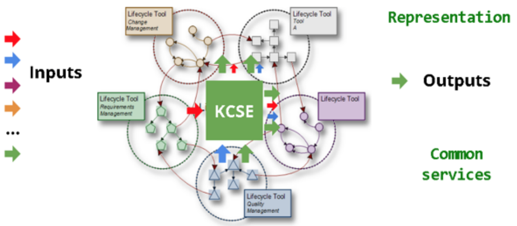
- The process, product and argumentation models can be linked to enable
  - impact analysis
  - process engineers, designers and assurance managers to work separately


Variability Model (VSpec Editor)
Resolution Models (Editor)
Base Model (UMA-Compliant Editor)
Realization Model (Editor)
Base Model (CHESSML Editor)
Base Model (Argumentation/ Evidence Editor)
Execute Fragment Substitutions
Resolved Models



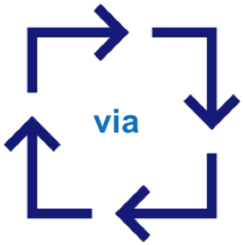
# Reuse discovery

## IoT: Internet Of Tools



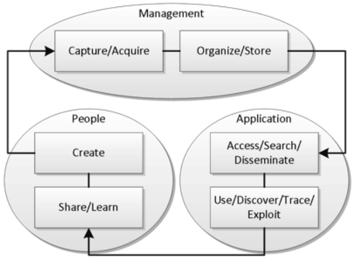
Knowledge-centric Systems Engineering

…to support…



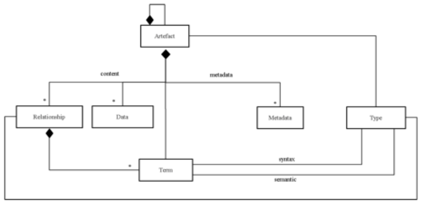
…interoperability…

## Knowledge Management processes



## Reuse discovery and selection

A shape: System Representation language



A set of **reuse operations** on top of the industrial knowledge graph…

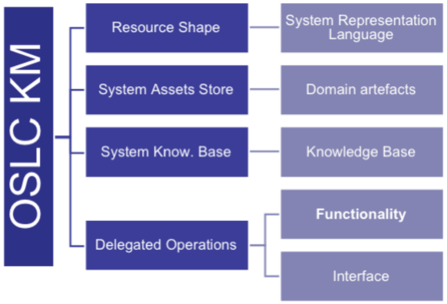
- Index
- Mapping & Merge
- Search
- Filter
- Access & Share

OSLC KM:
Represent any type of system artefact
&
Access any (delegated) operation

Metadata

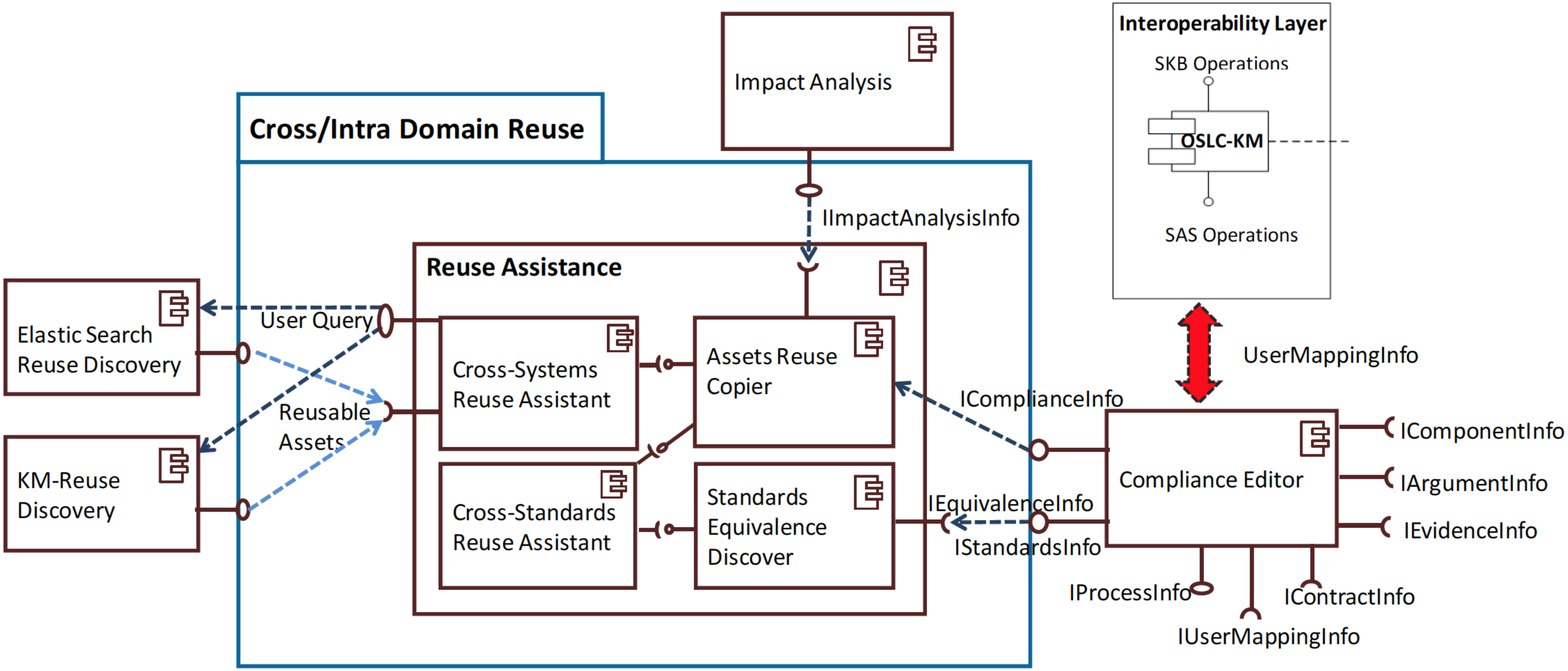
Contents

Services & Operations



| Operation | URI Template |
|---|---|
| Base URI/prefix | `http://www.reusecompany.com/oslc/km/operations` |
| **Search artifact** | `<base_uri>/sas/search`<br>`Query params: query={text}`<br>`Body params: srl={srl content}` |
| **Filter** | `<base_uri>/sas/filter`<br><br>`-Similar to query capabilities`<br>`-Similar to Linkedin: {(key=value,)+}` |

# Interplay of the functionalities -Reuse assistant perspective



# Thank you for your attention!

Any questions ?