Facilitating Automated Compliance Checking of Processes Against Safety Standards

Julieth Patricia Castellanos Ardila, Barbara Gallina, Faiz Ul Muram
{julieth.castellanos, barbara.gallina, faiz.ul.muram}@mdh.se

This work is supported by:
EU and VINNOVA via the **ECSEL JU project AMASS**
Certifiable Evidences & Justification Engineering-MDH

7th Scandinavian Conference on System & Software Safety (SCSSS)
Stockholm, October 23, 2019

Presentation Outline

1. Context, Motivation and Problem
2. Background
3. Our Method
4. Illustration
5. Current status of the work

MÄLARDALEN UNIVERSITY SWEDEN

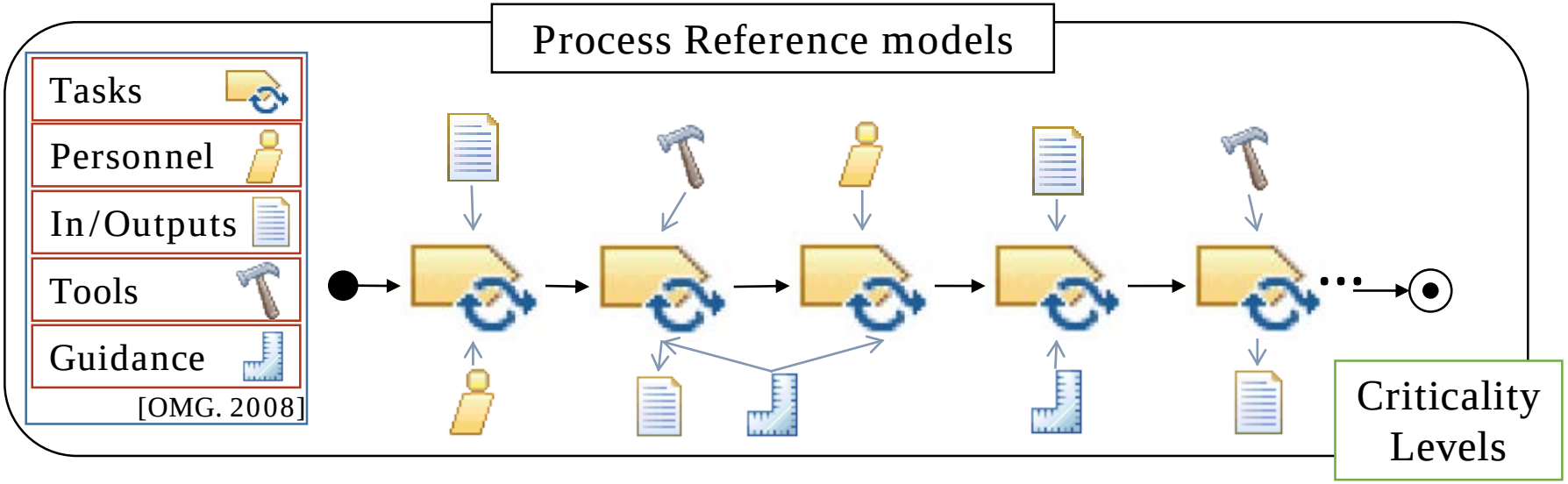
Presentation Outline

1. **Context, Motivation and Problem**
2. Background
3. Our Method
4. Illustration
5. Current status of the work

Context

Process-based Safety Standards specify the process to be used for producing/maintaining/changing **Safety-critical Systems.**

[Leveson, 2011]



Certain elements must be present in the process at specific moments.

Motivation

The Degree of compliance can be defined by checking that process tasks fulfill the properties set down by safety standards at given points.

Compliance checking could be done during process planning to:

- recognize missing characteristics in the process plans,
- prevent uncompliant tasks for being performed at the execution time,
- support the generation of a compliance justification,
- facilitate the creation of compliant process plans.

Problem

Manual compliance checking may be challenging.

1) It demands that the process engineer checks the fulfillment of hundreds of process-based requirements.

2) Companies usually need to check compliance of the specification of several engineering processes against more than one standard.

Automated compliance checking represents an added value for process-based compliance management in the safety-critical context.

Presentation Outline

1. Context, Motivation and Problem
2. Background
3. Our Method
4. Illustration
5. Current status of the work

Compliance by design

[Sadiq, et al, 2007]

Norms

Consistency

Processes

[Emmerich, et al, 1999]

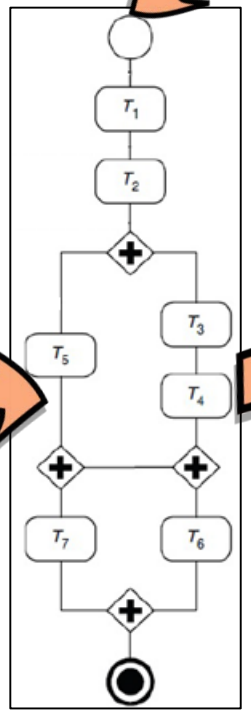
Compliance by design

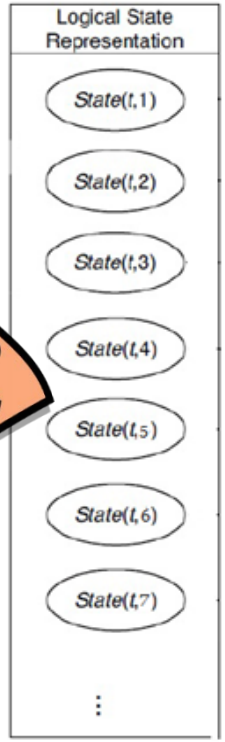
Generic Model

$Rule_1$
$Rule_2$
$Rule_3$
$Rule_4$

Compliance Effects Annotation

The properties required by the standard that the task is fulfilling





Normative Requirements

[Hashmi, 2016]

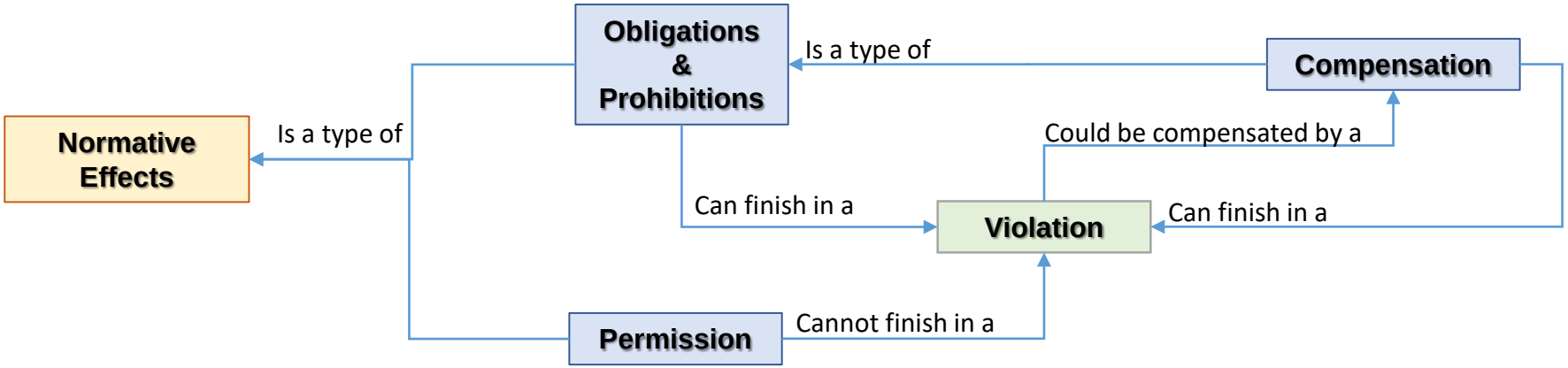
Scope: Regulate behaviors, defining what can and cannot be done.

Norms describe:

- The conditions under which they are applicable.
- Normative effects: Constraints affecting the subjects of the norms.

Deontic Effects

- Obligation.
- Prohibition.
- Permission.

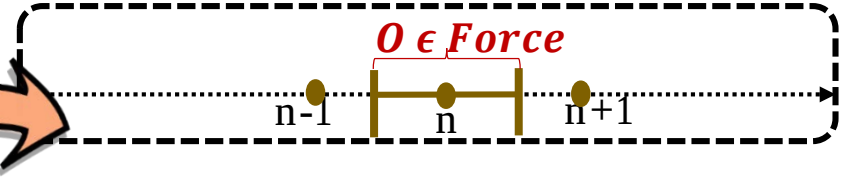


Conditions of the applicability of the norm.	If$\{a_1, \dots, a_n\}$	then{b}	Normative effect.

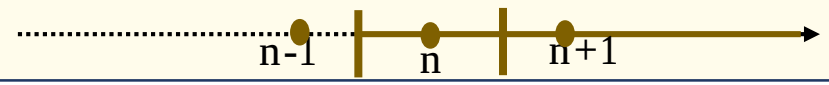
Formal Contract Logic (FCL)

[Governatori, 2005]

r: $\underbrace{a_1, \ldots, a_n}_{\text{Conditions}} \Rightarrow \underbrace{b \otimes c}_{\text{Deontic Effects}}$

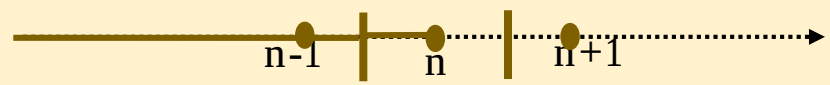


Maintenance Obligation [OM]

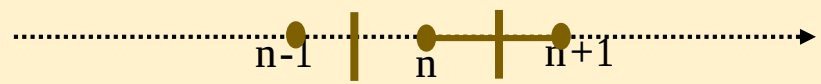


Achievement Obligation [OA_ _]

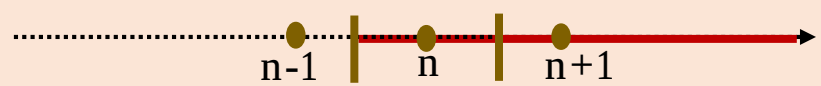
- Preemptive[OAP_]



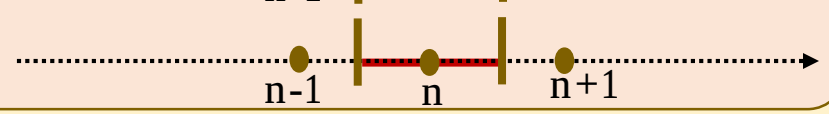
- Non-preemptive[OANP_]



- Perdurant: [OA_P]



- Non-Perdurant:[OA_NP]



Formal Contract Logic (FCL)

[Governatori, 2005]

Examples:

Working time schedule starts at 8:00 a.m.	**r1:** ⇒ **[OM]** StartWorkingAt8
Teleworking modality allows flexible schedule	**r2:** Teleworking ⇒**[P]** -StartWorkingAt8

Superiority relation: **r2>r1** **r2 defeats r1**

Analysis of compliance with FCL rules

REGOROUS [Governatori, 2015]

SPEM 2.0

[OMG. 2008]

Software & Systems Process Engineering Metamodel

Separation of concerns

Method content

Task

Work Product

Role

Managed content

Concept

Reusable asset

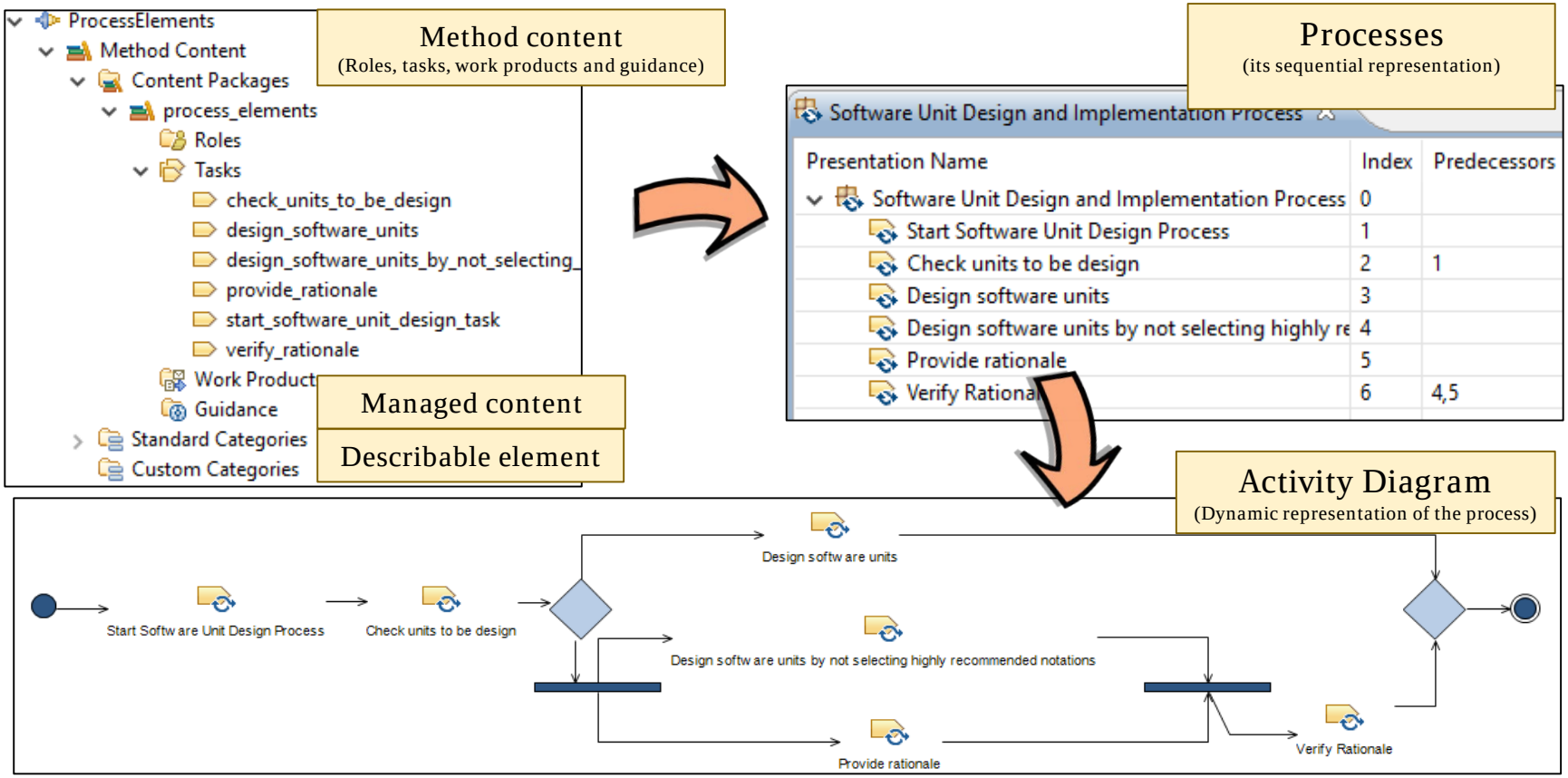
Describable element

Custom category

Processes

SPEM 2.0-like Process Models

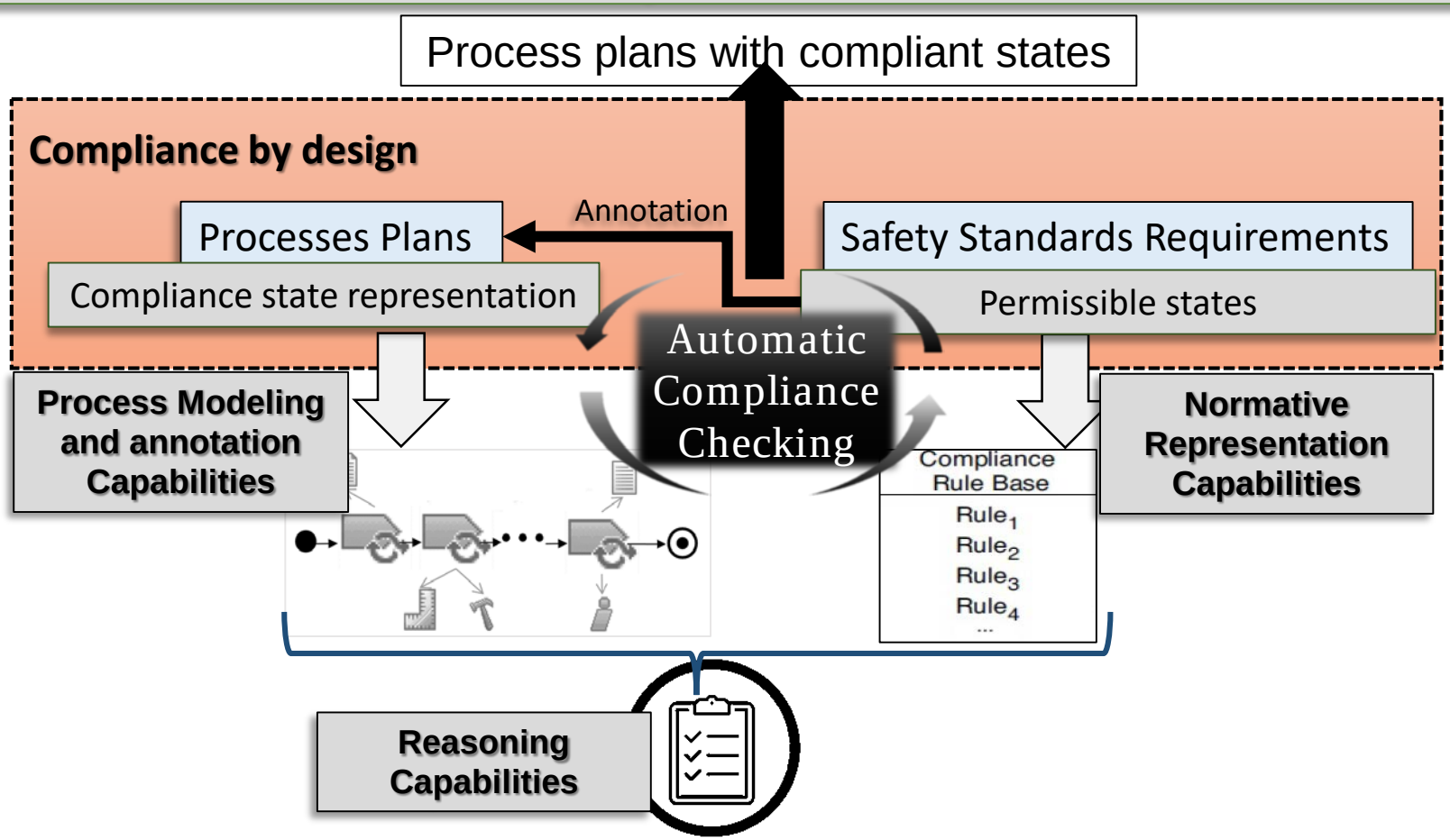
Eclipse Process Framework (EPF) Composer [EPF. 2008]



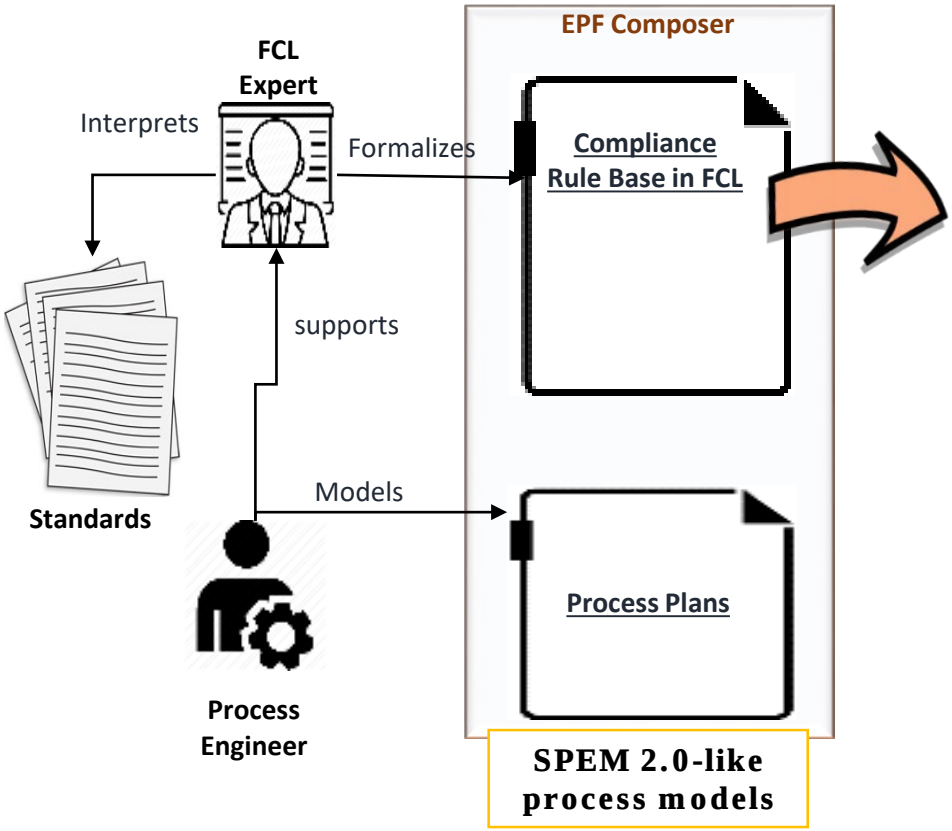
Presentation Outline

1. Context, Motivation and Problem
2. Background
3. Our Method
4. Illustration
5. Current status of the work

Conditions for Automatically Checking Compliance in the Safety-Critical Context

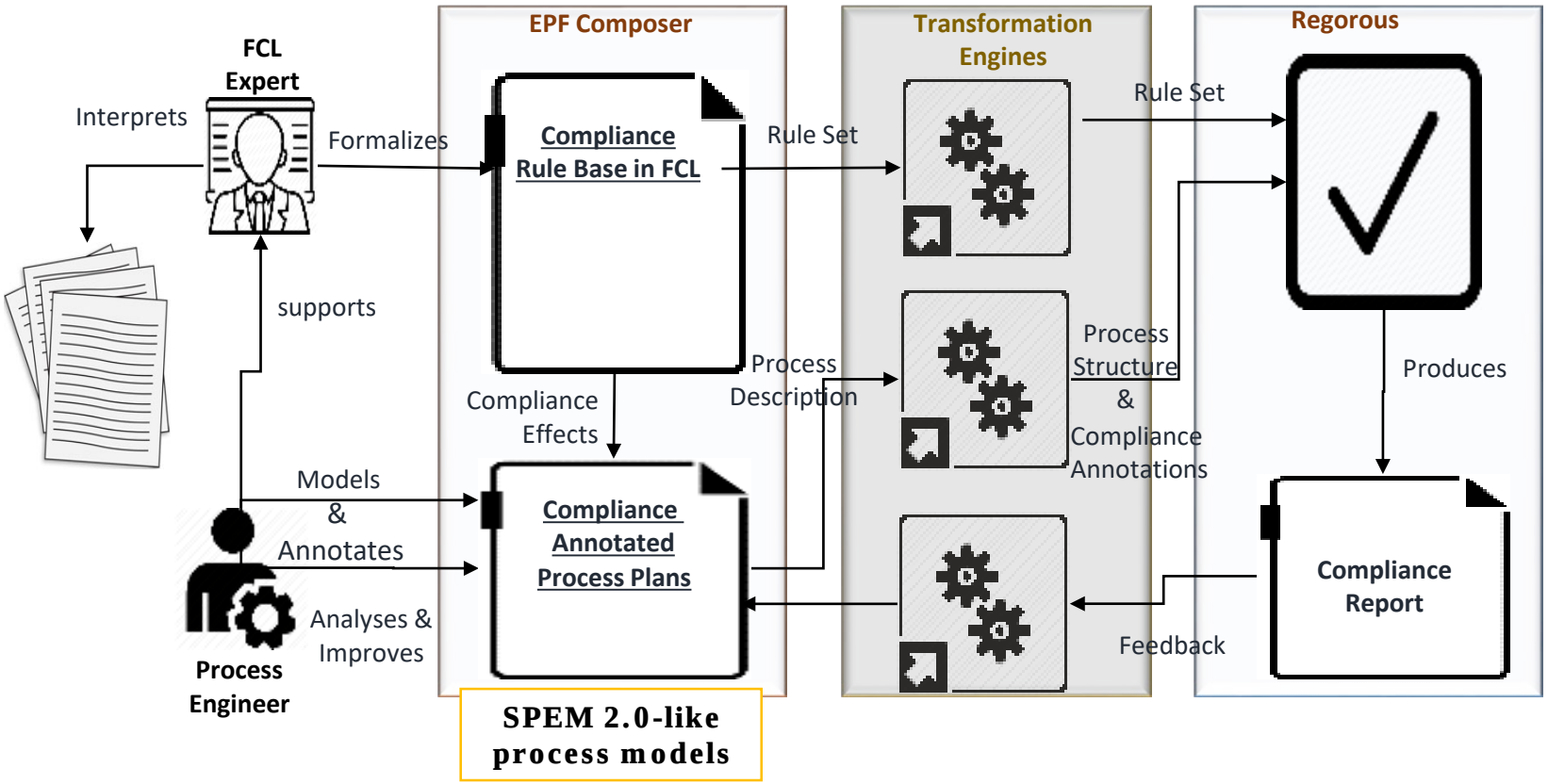


Automated Compliance Checking Vision



Compliance Checking Customization

Automated Compliance Checking Vision



ISO 26262 Compliance Patterns Definition

Normative Representation Capabilities

FCL

Skills which cannot be taken for granted!

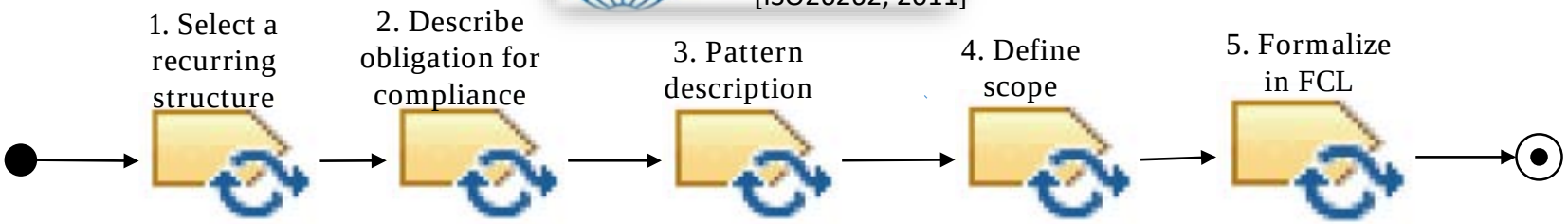
Patterns

Safety Compliance Pattern

Commonly occurring normative safety requirements on the permissible compliant state sequence of a process model.

ISO 26262

[ISO26262, 2011]



ISO 26262 Compliance Patterns Definition

Pattern	Intiation of a Phase
Structure	Phase
Obligation	Every phase proposed by the safety standard should be addressed, unless proper and demonstrated tailoring process is carry out.
Description	A phase must occur throughout a scope. Not addressing the phase requires its tailoring and the provision of a rationale
Scope	Global -> Maintenance Obligation

FCL template

$$\begin{cases} r: \{optionalPrerequirements\} \Longrightarrow [O]address\{Phase\} \\ r': tailor\{Phase\}, rationaleForOmmiting\{Phase\} \Longrightarrow [P] - address\{Phase\} \\ r' > r \end{cases}$$

Pattern Instantiation

$r_{3.5}: \Longrightarrow [O]addressItemDefinition$

$r_{3.5t}: tailorItemDefinition, rationaleForOmmitingItemDefinition \Longrightarrow [P] - addressItemDefinition$

$r_{3.5t} > r_{3.5}$

MÄLARDALEN UNIVERSITY SWEDEN

Methodological Guidelines for Formalizing ISO 26262

Formalization oriented pre-processing of ISO 26262

1. Scope

2. N. References

3. Terms

4. Requirements for compliance

Tailoring | Tables

From clause 5 = Phases of the safety process

X. Clause Title

X.1. Objectives

X.2. General

X.3. Prerequisites

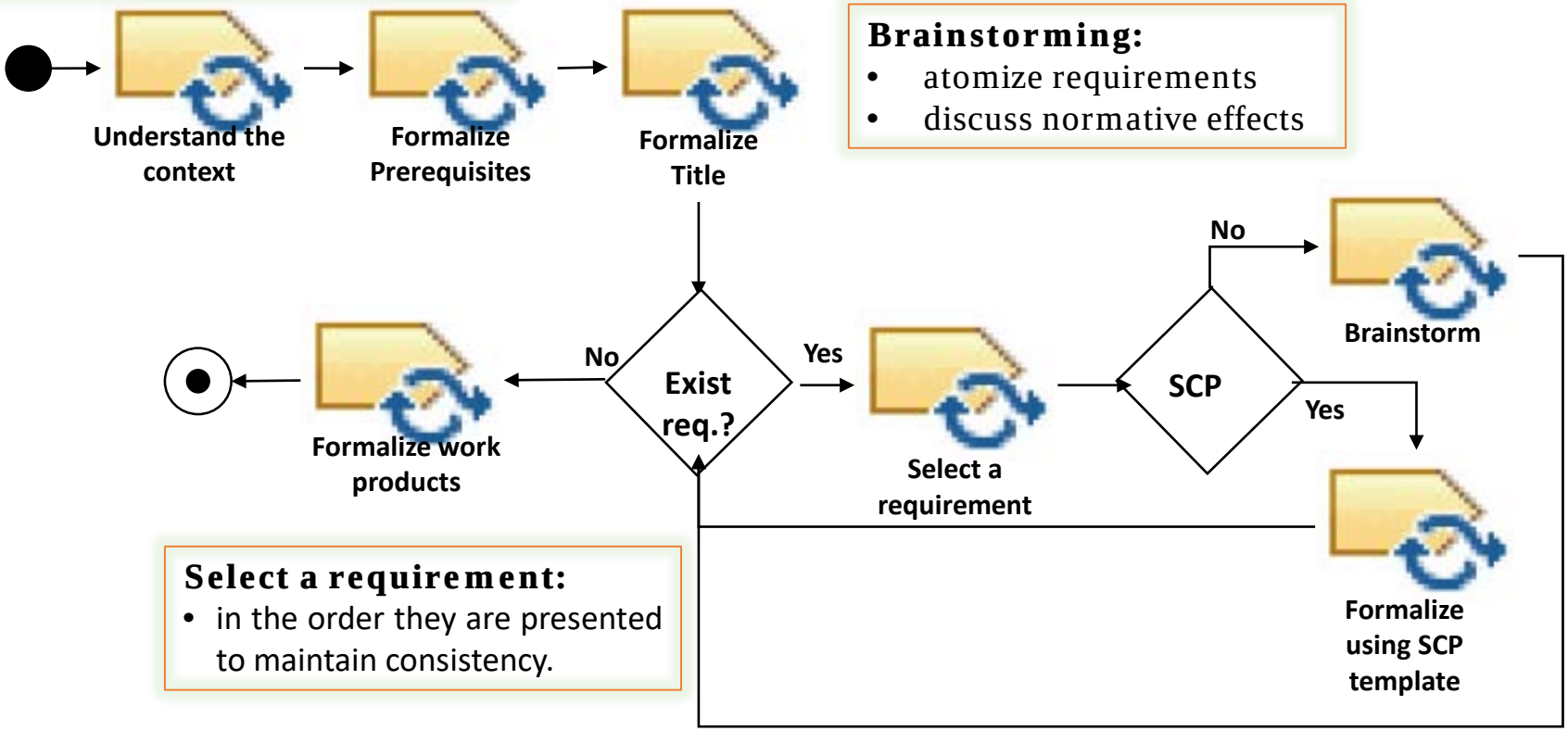
X.4 Requirements and Recommendation (R&R)

X.5. Work Products

Notes | Examples

Methodological Guidelines for Formalizing ISO 26262

Understand the context: reading and analysis of the objectives, and the main general clause



Brainstorming:
- atomize requirements
- discuss normative effects

Select a requirement:
- in the order they are presented to maintain consistency.

Presentation Outline

1. Context, Motivation and Problem
2. Background
3. Our Method
4. Illustration
5. Current status of the work

ISO 26262

[ISO26262, 2011]

8. Software unit design and implementation

8.1. Objective

The first objective is…

The second objective is …

8.2. General

Based on the software architectural design…

8.3. Prerequisites

- Software architectural design
- Software safety requirements

8.4 Requirements

8.4.1. The requirements of this subclause shall be complied with if the software unit is safety-related.

8.4.2. …the software unit design shall be described using the notations listed in the table below:

Notation	A	B	C	D
Natural language	++	++	++	++
Informal notations	++	++	+	+
Semi-formal notations	+	++	++	++
Formal notations	+	+	+	+

…

8.5. Work Products

- Software unit design specification…

General Requirements (from clause 4)

- In tables con consecutive entries all methods shall be applied as recommended in accordance with the ASIL. If methods other than those listed are to be applied, a rationale shall be given that these fulfil the corresponding requirement.
- Available rationale has to be assessed.

Note: "Safety related" means that the unit implements safety requirements.

ISO 26262 Formalization

Initiation of a phase		
Title	**R 6-8.** The software unit design phase is an obligatory phase.	Obligation
Prerequisites	**R 6-8.3.a** Providing the software safety requirements is obligatory.	Obligation
Prerequisites	**R 6-8.3.b** Providing the software architectural is obligatory.	Obligation
Requirements	**R 6-8.4.1** Checking if the unit is safety-related is obligatory.	Obligation
Consecutive entries	**R 6-8.4.2.a** Highly recommended notations for the ASIL are obligatory.	Obligation
	R 6-8.4.2.b Other notations can be applied if rationale exist.	Permission
Provision of a rationales	**R 6-8.4.2.c** For a rationale to be valid, it has to be assessed.	Obligation
Outputs	**R 6-8.5** A software unit design specification is an obligatory output.	Obligation
Work Productss		

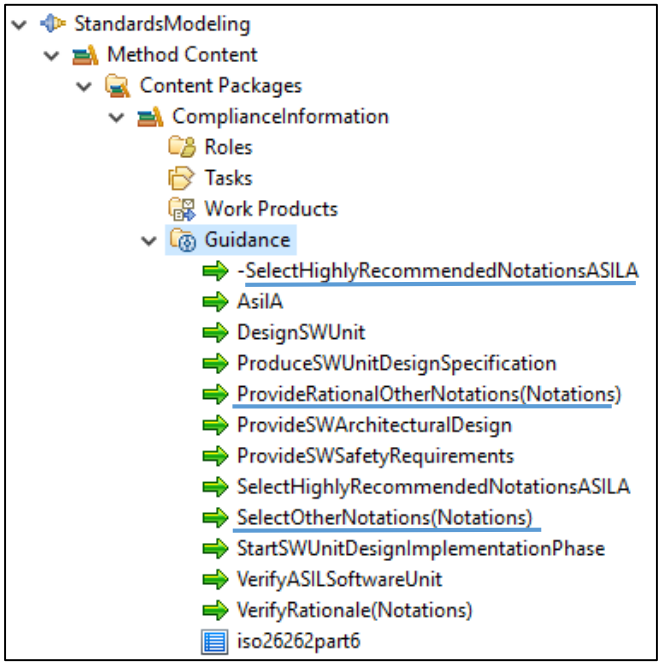
MÄLARDALEN UNIVERSITY SWEDEN

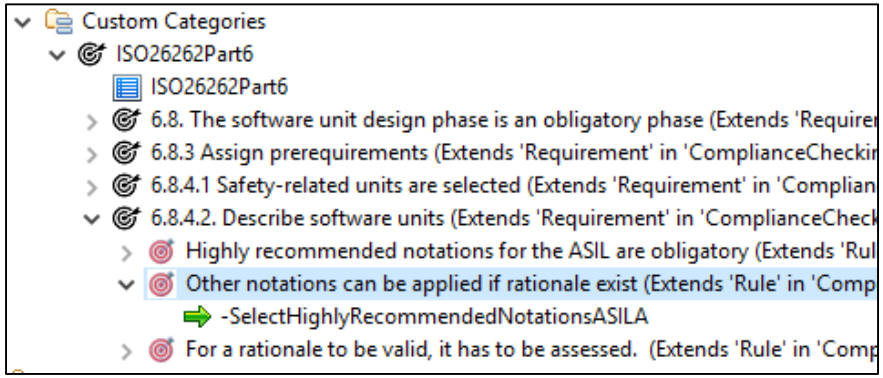
ISO 26262 - RuleSet Modeling

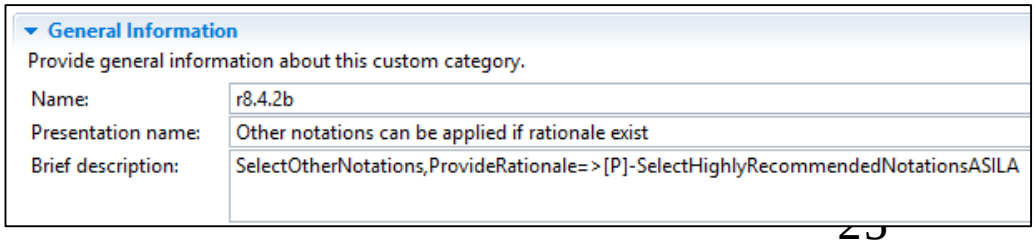
R 6-8.4.2 The software unit shall be described using notations according to ASIL and recommendation levels. Otherwise a rationale must be provided.

r6-8.4.2.b **Other notations can be applied if rationale exist.**

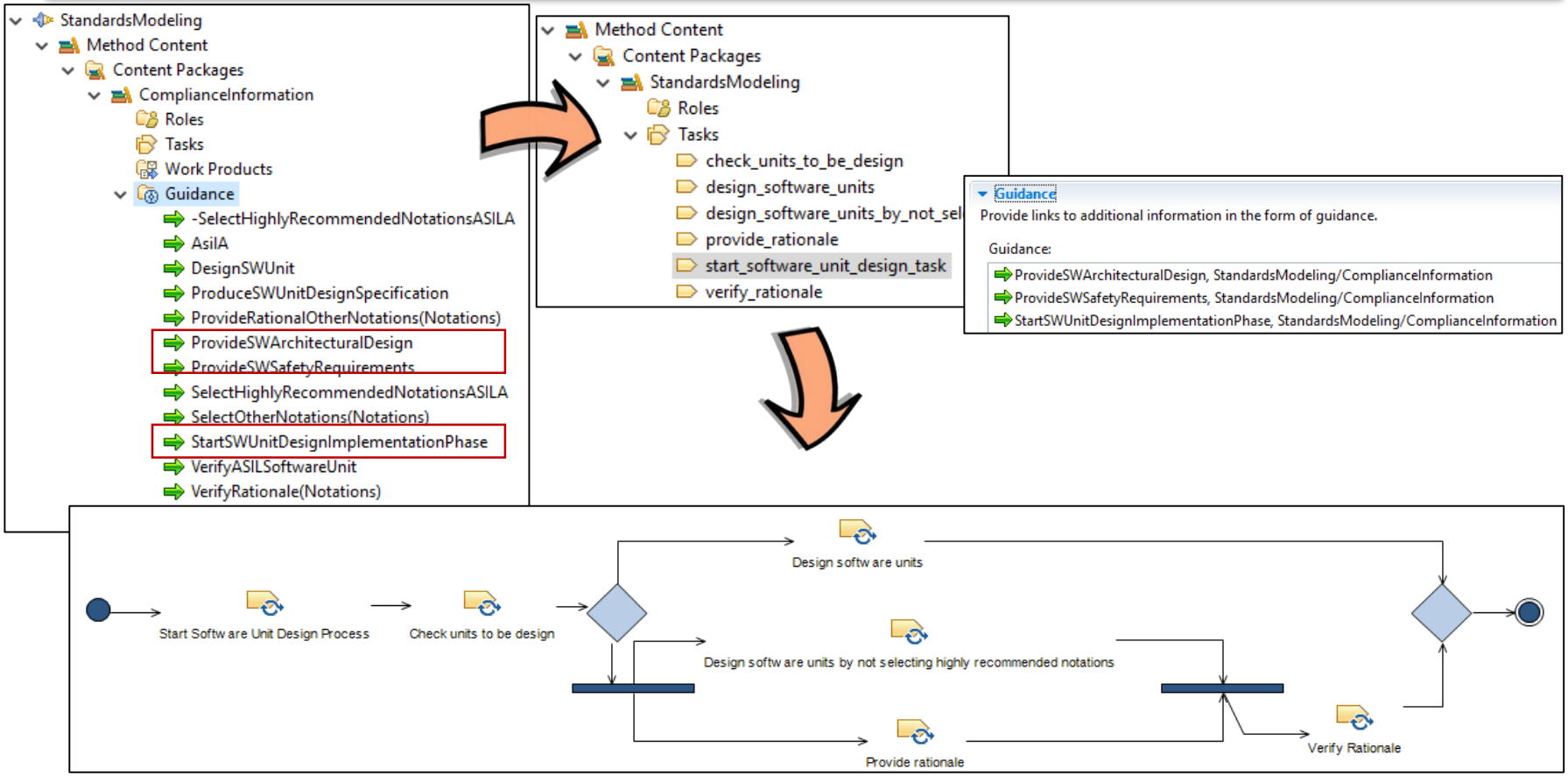
r6−8.4.2.*b* *SelectOtherNotations*, Provide*RationalOtherNotations* $\Longrightarrow$ $[P]$-*SelectHighlyRecommendedNotations*ASILA







ISO 26262 - Annotation Process



ISO 26262 - Compliance Checking

Compliance Check Results

Process is non-compliant.

Process Warnings

The warnings below indicate structural issues with the

Description

Information only (1 items)

Rule 'r8.4.2a' was not invoked

Non-compliant Execution Paths

Non-compliant execution paths and the cause of non-compliance are listed below.

[Start,Start Software Unit Design Process,Check units to be design,Design Software Unit,End]

Unfulfilled obligation to 'VerifyASIL' (Achievement, pre-emptive, persistent)

Compliance Issue Details

Execution Path: [Start,Start Software Unit Design Process,Check units to be design,Design Software Unit,End]

Description: Unfulfilled obligation to 'VerifyASIL' (Achievement, pre-emptive, persistent)

Element name/Id: Start Software Unit Design Process (_prt4oEGUEemY3qcv1wmqsg)

Rule Label: r8.4.1

Possible resolutions: 1. Prevent violation by performing 'VerifyASIL' at any step in the process

Custom Categories

ISO26262Part6

ISO26262Part6

6.8. The software unit

6.8.3 Assign prerequir

6.8.4.1 Safety-related

Verify that the soft

VerifyASIL

Presentation Outline

1. Context, Motivation and Problem
2. Background
3. Our Method
4. Illustration
5. Current status of the work

The current status of the Work

Manual annotations of compliance effects	Methodologies for process annotations
Analysis of compliance in sequences of task	Analysis of compliance in process elements beyond tasks, i.e., roles, tools and guidance
Analysis of patterns/guidelines only for ISO 26262	• Comparative studies between standards • Definition of generalized patterns • Definition of standard-specific patterns
Work evaluated with academic examples	• Further validation of the approach with more complex cases. • Validation with experts.
Preliminary result regarding reusability of proofs	Conditions that are required for compositionality of proofs of compliance

Thank you for your attention!

MÄLARDALEN UNIVERSITY SWEDEN

References

[Leveson, 2011] Leveson, N., '**The Use of Safety Cases in Certification and Regulation**', *Massachusetts Institute of Technology. Engineering Systems Division*, 2011

[OMG. 2008] Object Management Group Inc., '**Software & Systems Process Engineering Meta-Model Specification. Version 2.0.**', *OMG Std., Rev*, 2008, 236

[Sadiq, et al, 2007] Sadiq, Shazia, Guido Governatori, and Kioumars Namiri, '**Modeling Control Objectives for Business Process Compliance**', in *International Conference on Business Process Management*, 2007, pp. 149–64

[Hashmi, et al. 2016] Hashmi, M., G. Governatori, and M. T. Wynn, '**Normative Requirements for Regulatory Compliance: An Abstract Formal Framework**', *Information Systems Frontiers*, 2016, 429–55

[Governatori, 2005] Governatori, G., '**Representing Business Contracts in RuleML**', *International Journal of Cooperative Information Systems.*, 2005, 181–216

[Gallina, et al, 2018] Gallina, B., F. Ul Muram, and J.P. Castellanos Ardila, '**Compliance of Agilized (Software) Development Processes with Safety Standards: A Vision**', in *4th International Workshop on Agile Development of Safety-Critical Software (ASCS)*, 2018

[EPF. 2008] Eclipse Foundation, '**Eclipse Composer Framework**' <https://www.eclipse.org/epf/>

[Javed, et al. 2018] Javed, M., and B. Gallina, '**Safety-Oriented Process Line Engineering via Seamless Integration between EPF Composer and BVR Tool**', in *22nd International Systems and Software Product Line Conference* (ACM, 2018), pp. 23–28

[ISO26262, 2011] ISO 26262, '**Road Vehicles-Functional Safety. International Standard**', 2011

Facilitating Automated Compliance Checking of Processes Against Safety Standards

Julieth Patricia Castellanos Ardila, Barbara Gallina, Faiz Ul Muram
{julieth.castellanos, barbara.gallina, faiz.ul.muram}@mdh.se

This work is supported by:
EU and VINNOVA via the **ECSEL JU project AMASS**
Certifiable Evidences & Justification Engineering-MDH

7th Scandinavian Conference on System & Software Safety (SCSSS)
Stockholm, October 23, 2019