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
1. Context, Motivation and Problem
2. Background
3. Our Method
4. Illustration
5. Current status of the work
**Process-based Safety Standards** specify the process to be used for producing/maintaining/changing **Safety-critical Systems**. [Leveson, 2011]

**Criticality Levels**

```
Tasks
Personnel
In/Outputs
Tools
Guidance
```

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]

The properties required by the standard that the task is fulfilling

[Emmerich, et al, 1999]
Normative Requirements

**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.

**Normative Effects**
- Is a type of Obligations & Prohibitions
  - Is a type of Violation
  - Can finish in a
  - Cannot finish in a
  - Could be compensated by a Compensation

**Permission**
- Conditions of the applicability of the norm.

\[
\text{if} \{a_1, \ldots, a_n\} \quad \text{then} \{b\} 
\]

Normative effect.
**Formal Contract Logic (FCL)**

\[ r: \quad a_1, \ldots, a_n \quad \Rightarrow \quad b \quad \otimes \quad c \]

- **Conditions**
- **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:

1. Working time schedule starts at 8:00 a.m.
   - r1: ⇒ [OM] StartWorkingAt8

2. 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

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]

- Method content (Roles, tasks, work products and guidance)
- Processes (its sequential representation)
- Managed content
- Describable element
- Activity Diagram (Dynamic representation of the process)
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

Compliance by design
- Processes Plans
  - Compliance state representation
- Process Modeling and annotation Capabilities
- Safety Standards Requirements
  - Permissible states
- Automatic Compliance Checking
- Normative Representation Capabilities
- Reasoning Capabilities

Process plans with compliant states

Automated Compliance Checking Vision

- EPF Composer
- Formalizes Compliance Rule Base in FCL
- FCL Expert
- Interprets
- Supports models

Standards

- Process Engineer
- SPEM 2.0-like process models

Compliance Checking Customization

Automated Compliance Checking Vision

EPF Composer
- Formalizes Compliance Rule Base in FCL

Transformation Engines
- Rule Set
- Process Description
- Process Structure & Compliance Annotations
- Feedback

Regorous
- Produces Compliance Report

FCL Expert
- Interprets
- Supports
- Models & Annotates
- Analyses & Improves

Process Engineer
- Supports
- Models & Annotates
- Analyses & Improves

SPEM 2.0-like process models
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.

1. Select a recurring structure
2. Describe obligation for compliance
3. Pattern description
4. Define scope
5. Formalize in FCL

ISO 26262
[ISO26262, 2011]

ISO 26262 Compliance Patterns Definition

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

FCL template

\[
\begin{align*}
& r: \{\text{optionalPrerequisites}\} \Rightarrow [O] address\{\text{Phase}\} \\
& r': \text{tailor}\{\text{Phase}\}, \text{rationaleForOmitting}\{\text{Phase}\} \Rightarrow [P] - address\{\text{Phase}\} \\
& r' > r
\end{align*}
\]

Pattern Instantiation

\[
\begin{align*}
& r_{3.5}: \Rightarrow [O] address\{\text{ItemDefinition}\} \\
& r_{3.5t}: \text{tailor}\{\text{ItemDefinition}\}, \text{rationaleForOmitting}\{\text{ItemDefinition}\} \Rightarrow [P] - address\{\text{ItemDefinition}\} \\
& r_{3.5t} > r_{3.5}
\end{align*}
\]
Methodological Guidelines for Formalizing ISO 26262

Formalization oriented pre-processing of ISO 26262

<table>
<thead>
<tr>
<th>1. Scope</th>
<th>From clause 5 = Phases of the safety process</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. N. References</td>
<td>X. Clause Title</td>
</tr>
<tr>
<td>3. Terms</td>
<td>X.1. Objectives</td>
</tr>
<tr>
<td>4. Requirements for compliance</td>
<td>X.2. General</td>
</tr>
<tr>
<td></td>
<td>X.3. Prerequisites</td>
</tr>
<tr>
<td>Tailoring</td>
<td>X.4 Requirements and Recommendation (R&amp;R)</td>
</tr>
<tr>
<td>Tables</td>
<td>X.5. Work Products</td>
</tr>
</tbody>
</table>

Notes

Examples

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

1. Scope

2. N. References

3. Terms

4. Requirements for compliance

Tailoring

Tables

Notes

Examples

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

<table>
<thead>
<tr>
<th>Notation</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural language</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Informal notations</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Semi-formal notations</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Formal notations</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

... 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.

8.5. Work Products
• Software unit design specification...

ISO 26262 [ISO26262, 2011]
**ISO 26262 Formalization**

<table>
<thead>
<tr>
<th>Initiation of a phase</th>
<th>Title</th>
<th>R 6-8. The software unit design phase is an obligatory phase.</th>
<th>Obligation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prerequisites</td>
<td>R 6-8.3.a Providing the software safety requirements is obligatory.</td>
<td>Obligation</td>
<td></td>
</tr>
<tr>
<td>Prerequisites</td>
<td>R 6-8.3.b Providing the software architectural is obligatory.</td>
<td>Obligation</td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td>R 6-8.4.1 Checking if the unit is safety-related is obligatory.</td>
<td>Obligation</td>
<td></td>
</tr>
<tr>
<td>Consecutive entries</td>
<td>R 6-8.4.2.a Highly recommended notations for the ASIL are obligatory.</td>
<td>Obligation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R 6-8.4.2.b Other notations can be applied if rationale exist.</td>
<td>Permission</td>
<td></td>
</tr>
<tr>
<td>Provision of a rationales</td>
<td>R 6-8.4.2.c For a rationale to be valid, it has to be assessed.</td>
<td>Obligation</td>
<td></td>
</tr>
<tr>
<td>Outputs</td>
<td>R 6-8.5 A software unit design specification is an obligatory output.</td>
<td>Obligation</td>
<td></td>
</tr>
<tr>
<td>Work Productss</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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, ProvideRationaleOtherNotations $\implies$ [P]-SelectHighlyRecommendedNotationsASILA
ISO 26262 - Annotation Process

- Select Highly Recommended Notations AS/IL A
- Design SW Unit
- Produce SW Unit Design Specification
- Provide Rational Other Notations (Notations)
- Provide SW Architectural Design
- Provide SW Safety Requirements
- Select Highly Recommended Notations AS/IL A
- Select Other Notations (Notations)
- Start SW Unit Design Implementation Phase
- Verify ASIL Software Unit
- Verify Rationale (Notations)

Method Content
- Content Packages
  - Standards Modeling
  - Roles
  - Tasks
  - Work Products
- Guidance

Guidance
- Provide links to additional information in the form of guidance.

- Provide SW Architectural Design, Standards Modeling/Compliance Information
- Provide SW Safety Requirements, Standards Modeling/Compliance Information
- Start SW Unit Design Implementation Phase, Standards Modeling/Compliance Information

Design software units
- Check units to be design
- Design software units by not selecting highly recommended notations
- Provide rationale
- Verify Rationale
ISO 26262 - Compliance Checking

Compliance Check Results
- Process is non-compliant.

Process Warnings
The warnings below indicate structural issues with the process:

- 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 (_prv4cEGUEemY3qcv1wnc5y)

- Rule Label:
  - r8.4.1

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


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