Lessons Learned while Formalizing ISO 26262 for Compliance Checking

Julieth Patricia Castellanos Ardila and Barbara Gallina[1] and Guido Governatori[2]

[1] Mälardalen University, Sweden
[2] Data61, CSIRO, Australia

Abstract. A confirmation review of the safety plan is required during compliance assessment with ISO 26262. Its production could be facilitated by creating a specification of the standard's requirements in FCL (Formal Contract Logic), which is a language that can be used to automatically checking compliance. However, we have learned, via previous experiences, that interpreting ISO 26262 requirements and specifying them in FCL is complex. Thus, we perform a formalization-oriented pre-processing of ISO 26262 to find effective ways to proceed with this task. In this paper, we present the lessons learned from this pre-processing which includes the identification of the essential normative parts to be formalized, the identification of SCP (Safety Compliance Patterns) and its subsequent documentation as templates, and the definition of a methodological guideline to facilitate the formalization of normative clauses. Finally, we illustrate the defined methodology by formalizing ISO 26262 part 3 and discuss our findings.

Keywords: Compliance checking, standards formalization, Formal Contract Logic.

1 Introduction

A confirmation review of the safety plan is a piece of evidence required for compliance assessment with ISO 26262 [1] in the automotive industry. Producing this evidence is time-consuming since ISO 26262 contains hundreds of requirements that have to be checked based on the information provided by the specification of the development processes used to engineer the safety-critical systems included in cars. To automate this task, requirements should be encoded in formal notations, which can express not only their contradictory, incomplete and inconsistent nature, but also their normative provisions, which are the notions anchored to the structure of regulations [2]. From the compliance perspective, the normative provisions of importance are those related to the obligations, permissions, and prohibitions [3]. Therefore, a promising approach for formalizing requirements could be based on defeasible logic, in which contrary evidence defeats earlier reasoning, supporting the management of inconsistencies [4]. Also, normative provisions should be encoded as implications in which the antecedent is read as a property of a state of affairs, and the conclusion has a deontic nature [5]. Thus, we argue that deontic defeasible reasoning formalisms, such as Formal Contract Logic (FCL) [6], can be used to generate automatic support to reason from standard's requirements and the description of the process they regulate [7, 8].

In our previous work [9], we explored mechanisms to support the formalization work of the process engineers. From this initial attempt, a definition of SCP (Safety Compliance Patterns), as well as a set of ISO 26262-specific FCL-SCP were formulated. We also performed the formalization of standard's requirements into FCL in [7, 8]. Via these experiences, we learnt that interpreting ISO 26262 requirements and specifying them in FCL is a complex task. Therefore, we perform a formalization-oriented pre-processing of ISO 26262 to gain fundamental knowledge about efficient ways to proceed. In this paper, we present the lessons learned resulting form this pre-processing which includes the identification of the essential normative parts to be formalized, the identification of Safety Compliance Patterns (SCP), and its subsequent documentation as templates, and the definition of a methodological guideline, which can be used to facilitate the formalization of normative clauses. We also illustrate the defined methodology by formalizing ISO 26262 part 3 and discuss our findings.

The rest of the paper is structured as follows. In Section 2, we present essential background. In Section 3, we describe the formalization-oriented pre-processing of ISO 26262. In Section 4, we illustrate the methodological guideline derived from the pre-processing of ISO 26262. In Section 5, we discuss our findings. In Section 6, we present related work. Finally, in Section 7, we present conclusions and future work.

2 Background

This section presents the background required in this paper. Section 2.1 recalls essential information related to ISO 26262. Section 2.2 recalls the basis of Formal Contract Logic. Finally, Section 2.3, recalls Safety Compliance Patterns.

2.1 ISO 26262

ISO 26262 [1] is a functional safety standard that regulates all phases of the production process of road vehicles with a gross mass up to 3500 kg. ISO 26262 uses *ASIL (Automotive Safety Integrity Levels)* to specify the safety requirements needed to be fulfilled during the development process of safety-critical systems included in cars, e.g., automobile brake system. ISO 26262 is composed of ten parts. Part 1 specifies the terms, definitions and abbreviated terms for application in all parts of ISO 26262. The remaining nine parts, which are normative, are structured in a similar way, containing, a *foreword*, *introduction*, *bibliography*, *annexes*, and *clauses*. *Clause 1* recalls the general scope of the standard and situates the particular part in this scope. *Clause 2* and *Clause 3*, recalls the normative references indispensable for the adoption of the specific part. *Clause 4*, which is repeated in all parts, contains two general compliance conditions. Item 4.1 relates to the tailoring of the safety activities, which is valid if *"an assessed rationale is available that the non-compliance is acceptable"*. Item 4.2 relates to the interpretation of tables, as follows:

- **Tables with 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.
- **Tables with alternative entries:** An appropriate combination of methods shall be applied in accordance with the ASIL indicated. Methods with the higher recommendation for the ASIL should be preferred. A rationale shall be given that the selected combination of methods complies with the corresponding requirement.

The description of the phases of the safety process is distributed in clauses included in the nine normative parts starting from Clause 5. Each of these clauses states its *Objectives*, which describe the generic goals of the clause, *General information*, which gives an overall explanation of the clause, *Inputs of the clause*, i.e., prerequisites, *Requirements and Recommendations (R&R)*, which describe the specific conditions that the process should fulfill, and the *Work products*, which are the mandatory deliverables. *Notes* and *Examples* are expected to help the applicant in interpreting the requirements. We focus on a subset of requirements from ISO 26262 part 3 presented in Table 1, which specifies the requirements for the concept phase for automotive applications.

Table 1. ISO 26262:2011 part 3 (Adapted from [1])

5 Item definition - *5.3 Inputs of this clause:* None.

5.4 Requirements and recommendations

5.4.1 Functional and non-functional requirements shall be made available, including: a) functional concept and b) operational constraints.

5.5 Work products: Item definition resulting from the requirements of 5.4.

6 Initiation of the safety lifecycle - *6.3 Inputs of this clause:* item definition in accordance with 5.5.

7 Hazard analysis and risk assessment

7.4.3 The severity shall be assigned to one of the severity classes S0, S1, S2 or S3 in accordance with:

Severity	S0	S1	S2	S3
Category	No Injury	Moderate	Severe	Life-threatining

7.4.4 Determination of ASIL and safety goals - The safety requirements shall be specified by an appropriate combination of the methods as presented in the table (H means *Highly* and R means *Recommended*).

Notation		**A**	**B**	**C**	**D**
1a	Informal	HR	HR	R	R
1b	Semi-formal	R	R	HR	HR
1c	Formal	R	R	R	R

2.2 Formal Contract Logic

Formal Contract Logic (FCL) [6] is a defeasible deontic logic created for checking the compliance of business contracts, and modelling normative requirements. An FCL rule has the form:

$r : a_1, ..., a_n \Rightarrow c$, where:

$a_1, ..., a_n =$ Conditions of the applicability of the norm.

$c =$ Normative effect.

Normative effects trigger deontic notions, namely, *Obligations*, *Prohibitions*, and *Permissions*. An *Obligation* is a statement describing a mandatory situation. In FCL, an obligation is represented by the operator $[O]$ plus a proposition, which corresponds to the content of the obligation. FCL is equipped with different kind of obligations, which depend on the timing of the application of the normative provision and their persistence after a violation (see [6]). A *Prohibition* is a forbidden situation. In FCL a prohibition is represented as the negation of the content of an obligation. A *Permission* is an allowed situation. Exceptions for the rules can be formalized by using permissions, taking into account the premise *"if something is permitted the obligation to the contrary does not hold"* [3]. Permissions in FLC are represented with the operator $[P]$. The formalization

of normative systems sometimes may contain conflicting normative effects, such as the obligation of performing an action but also its prohibition. In FCL, these conflicts can be solved by defining a superiority relation between rules ($>$).

2.3 Safety Compliance Patterns

Safety compliance patterns (SCP) [9] describe commonly occurring normative safety requirements on the permissible state sequence of a finite state model of a process. An SCP has a general formalization structure (which in our case is defined in FCL), which is derived from the interpretation of a recurring structure described in the text of the standard. Currently, a list of SCP is defined in [9].

3 Formalization-oriented Pre-processing of ISO 26262

Our initial efforts to formalize ISO 26262 into FCL (see our previous work [9, 8]), gave us some insights about the complexity that this task entails. As recalled in Section 2.1, ISO 26262 is structured in a specific way, i.e., it is composed of parts, which are subdivided into very structured clauses. We encounter that not all the structures are required to be formalized. We also find that some structures are repetitive, and can be represented as SCP. Therefore, to be able to formalize effectively, we consider that doing a pre-processing of ISO 26262 was necessary. The pre-processing, which is depicted in Figure 1, includes three tasks. Initially, we identify the essential normative structures (see Section 3.1), namely those structures that define the safety process to be adopted for developing the car's safety-critical systems. Then, we identified the repetitive structures of the standard that can be considered SCP (see Section 3.2). With the identified SCP, we create templates to consolidate a reusable knowledge base for future formalization jobs (see Section 3.3). Finally, the knowledge gathered in the pre-processing is used to define a methodological guideline for facilitating the formalization of normative clauses in ISO 26262 (see Section 3.4). The pre-processing tasks were performed in the form of intensive group brainstorming sessions (in total ten 5-hour sessions). The group included three participants. The first participant has expertise in formal approaches (particularly FCL) applied to legal informatics. The second participant has expertise in certification (particularly in the safety-critical context). The third participant is a Ph.D. student whose research work is focused on the compliance checking of safety processes against safety standards (particularly ISO 26262).

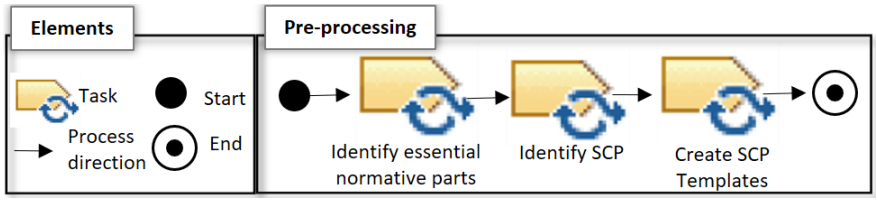

Fig. 1. Methodological Guidelines to formalize ISO 26262 into FCL.

3.1 Identify essential normative parts in ISO 26262

As presented in Section 2.1, ISO 26262 has nine normative parts. In each of the normative parts, Clauses 1, 2 and 3 are not required to be formalized since the text does not represent a constraint to the development process. Clause 4 is subdivided into two items. Item 4.1, which title is *General requirements*, describes the tailoring of safety activities, namely its application in a way different to the indicated by the standard. Item 4.2, which title is *Interpretation of tables*, illustrate the way in which normative methods listed in tables (with consecutive and alternative entries) should be applied. By themselves, these two requirements are not directly constraining the process. However, they shape the way in which other requirements should be applied. Therefore, Clause 4 is an essential structure that gives important elements for the formalization process. The specific normative clauses that describe the phases of the safety process are documented from Clause 5 in each of the nine normative parts. In those clauses, the title represents the initiation of the phase. Therefore, the title is part of the formalization process. However, the formalization of the title must be preceded by the formalization of the *Prerequisites*, since they represent the preconditions constraining the initiation of the particular phase. *Prerequisites* as well as *Work products* are essential since they are part of the description of the safety process which is expected to be represented via a model embracing input/output elements. The presence of the verbs *shall* and *must* in the section *R&R* is an explicit indication of a normative provision that constrains the breakdown of the work as well as for guidance on how it should be planned and executed. Information under the titles *Objectives*, *General*, *Notes* and *Examples* are not formalized since they do not prescribe the process to be adopted. However, these elements can be used to provide context for the application of the requirements.

3.2 Identify SCP

Within the essential normative parts of ISO 26262, seven SCP are identified. In Clause 4, the *provision of a rationale* is done in the same way whenever a safety activity is tailored. Similarly, the applicable methods that are described in *tables with alternative entries* and *tables with consecutive entries*. Therefore, Clause 4 describes three repetitive structures. Within the description of the phases of the safety process, represented from clause 5 in each of the normative parts, other three repetitive structures are easily recognizable. The first one is the *Initiation of a phase*, which is recognized in the title of every clause. The second repetitive structure corresponds to the *Prerequisites*, which describe the preconditions of the phase. Similarly, the *Work products*, which are defined as the result of safety activities, represent a third repetitive structure. Since *R&R* contains many requirements, and each one describes a different structure, we cannot consider this structure as repetitive itself (even though we can find the title *R&R* in all the normative parts). However, inside the *R&R* one repetitive structure, called *Guidance*, is recognized. A requirement, which we call the main normative effect, contains guidance when it is accompanied by a list of descriptive items (a, b,..., n). Together, those items provide additional normative descriptions about the main normative effect, and therefore, they also become mandatory requirements.

3.3 Create SPC templates

For each SCP, we provide a general formalization structure in FCL which is derived from the interpretation of the repetitive structure it represents (as described in Section 2.3). The rules in the template contains the symbol # between brackets ({ }), which should be replaced with the identification of the requirement that the rule is representing. The space between the brackets in the rule statement ({ }) is a placeholder for the particular instantiation of the template.

Provision of a rationale: A rationale implies *compliance with conditions*. For being valid, it should be always verified by an expert. Therefore, when a rationale is attached to a safety process, its verification is obligated (see Template 1).

$$\mathbf{r}\{\mathbf{\#:}\}attach\{Rationale\} \Rightarrow [O]verifyByExpert\{Rationale\} \tag{1}$$

Alternative entries: The normative provision for tables with alternative entries obliges the verification of the ASIL, the provision of a combination of the listed methods and a the provision of a weak rationale. The combination of these methods also obliges the inclusion of those marked with the highest recommendation level. The provision of other methods is also possible if a strong rationale is provided. There is an inconsistency between rules $\mathbf{r}\{\mathbf{\#.a:}\}$and $\mathbf{r}\{\mathbf{\#.g:}\}$. Thus, a superiority relation that gives priority to $\mathbf{r}\{\mathbf{\#.g:}\}$ (which describes an exception for the requirement) is provided (see template 2).

$$\begin{aligned}
\mathbf{r}\{\mathbf{\#.a:}\}verify\{ASIL\} &\Rightarrow [O]provideCombinationOfListedMethods\\
\mathbf{r}\{\mathbf{\#.b:}\}provideCombinationOfListedMethods &\Rightarrow [O]include\{HigherRecommendedNotationsForASIL\}\\
\mathbf{r}\{\mathbf{\#.c:}\}include\{HigherRecommendedNotationsForASIL\} &\Rightarrow [O]attachWeakRationaleSupportingListedMethods\\
\mathbf{r}\{\mathbf{\#.d:}\}attachWeakRationaleSupportingMethods &\Rightarrow [O]VerifyWeakRationaleByDomainExpert\\
\mathbf{r}\{\mathbf{\#.e:}\}includeNotListedMethods\{OtherMethods\} &\Rightarrow [O]atachStrongRationaleSupporting\{OtherMethods\}\\
\mathbf{r}\{\mathbf{\#.f:}\}atachStrongRationaleSupporting\{OtherMethods\} &\Rightarrow [O]verifyStrongRationaleByDomainExpert\{OtherMethods\}\\
\mathbf{r}\{\mathbf{\#.g:}\}includeNotListedMethods\{OtherMethods\}, &[O]atachStrongRationaleSupporting\{OtherMethods\},\\
verifyStrongRationaleByDomainExpert\{OtherMethods\} &\Rightarrow [P]-provideCombinationOfTheListedMethods\\
r\{\#.g\} > r\{\#.a\}&
\end{aligned} \tag{2}$$

Consecutive entries: The normative provision for tables with consecutive entries obliges the verification of the ASIL and the utilization of all listed methods. The combination of methods beyond the ones listed in the table is also possible if a strong rationale is provided (see Template 3). There is an inconsistency between rules $\mathbf{r}\{\mathbf{\#.a:}\}$and $\mathbf{r}\{\mathbf{\#.d:}\}$. Thus, a superiority relation that gives priority to $\mathbf{r}\{\mathbf{\#.d:}\}$ (which describes an exception for the requirement) is provided (see template 2).

$$\begin{aligned}
\mathbf{r}\{\mathbf{\#.a:}\}verify\{ASIL\} &\Rightarrow [O]provideAllfListedMethods\\
\mathbf{r}\{\mathbf{\#.b:}\}includeNotListedMethods\{otherMethods\} &\Rightarrow [O]atachStrongRationaleSupporting\{otherMethods\}\\
\mathbf{r}\{\mathbf{\#.c:}\}attachStrongRationaleSupporting\{otherMethods\} &\Rightarrow [O]verifyStrongRationaleByDomainExpert\{otherMethods\}\\
\mathbf{r}\{\mathbf{\#.d:}\}includeNotListedMethods\{otherMethods\}, &[O]atachStrongRationaleSupporting\{otherMethods\},\\
verifyStrongRationaleByDomainExpert\{OtherMethods\} &\Rightarrow [P]-provideCombinationOfTheListedMethods\\
r\{\#.d\} > r\{\#.a\}&
\end{aligned} \tag{3}$$

Prerequisites: The antecedents of the initiation of a phase are the prerequisites. Therefore, they are obliged before the phase is initiated (see Template 4).

$$\begin{aligned} \mathbf{r\{\#.a\}:} &\Rightarrow [O]provide\{prerequisiteA\} \\ &\ldots \\ \mathbf{r\{\#.n\}:} &\Rightarrow [O]provide\{prerequisiteN\} \end{aligned} \tag{4}$$

Initiation of a phase: The template considers the prerequisites (formalization presented in Template 4) as the conditions of the applicability of the rule which normative conclusion is the initiation of the phase (see Template 5).

$$\begin{aligned} \mathbf{r\{\#:\}}provide\{prerequisiteA\}\ldots, provide\{prerequisiteN\} \\ \Rightarrow [O]initiate\{TitleClause\} \end{aligned} \tag{5}$$

Guidance: Guidance components are the conditions that obliges the provision of the element guided (see formula 6).

$$\begin{aligned} \mathbf{r\{\#.a:\}}\{TriggeringObligation\} &\Rightarrow [O]provide\{FirstGuidanceElement\} \\ &\ldots \\ \mathbf{r\{\#.n:\}}\{TriggeringObligation\} &\Rightarrow [O]provide\{LastGuidanceElement\} \\ \mathbf{r\{\#:\}}provide\{FirstGuidanceElement\}, \ldots, provide\{LastGuidanceElement\} &\Rightarrow [O]provide\{GuidedElement\} \end{aligned} \tag{6}$$

Work Product: Work products are the result of certain requirements. Therefore, these requirements are presented as antecedents that obliged the provision of the related work product (see Template 7).

$$\mathbf{r\#:}provide\{PreviousObligations\} \Rightarrow [O]produce\{WorkProduct\} \tag{7}$$

3.4 Methodological guideline for formalizing ISO 26262

From the pre-processing tasks described above, we got an understanding of what to formalize and how we could proceed in the formalization process. The parts to formalized are those that determine the safety lifecycle, meaning those clauses that start from Clause 5. To formalize these clauses, we have described a methodological guideline, which we depict in Figure 2.

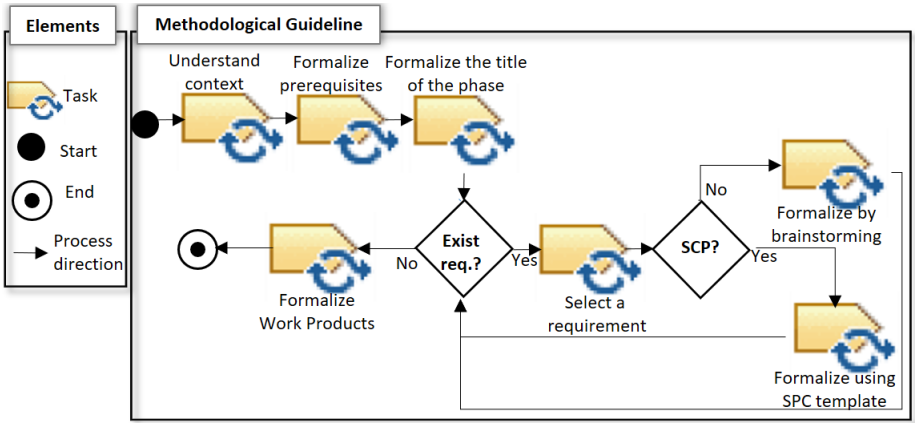

Fig. 2. Methodological Guidelines to formalize ISO 26262 into FCL.

Initially, the context of the phase should be understood. For this, the reading and analysis of the objectives and the main general information of the clause to be formalized is required. Then, the formalization process initiates with the prerequisites and followed by the title. These two formalizations can be done by following the SCPs called *Prerequisites* and *Initiation of a phase*. After, one requirement is selected from the list of *R&R*. We suggest that the requirements are selected in the order they are presented and that the rules are named following the requirement numeration to ensure consistency and traceability. For instance, if a textual requirement is marked with the label *5.1*, the corresponding rule should be called *r.5.1*. During the formalization of the requirements, SCP templates could be used to facilitate this task. However, if there are no templates, brainstorming sessions are required. The brainstorming session can be carried out in different ways, but the most relevant is that the group takes one requirement at the time, discuss its importance in the compliance process (e.g., related requirements or permits for tailoring), divide the requirement into smaller sentences that have only one idea, and discuss every sentence. If the requirement has to be divided into several rules, the name of the rule has to be named with the number that accompanies the requirement plus a letter, i.e., *r.5.1.a*, *r.5.1.b*. Finally, when all requirements available in *R&R* are covered, the work products can be formalized by using the SCP template called *Work Product*. The generated rule set should be verified to avoid inconsistencies and typos in the rules since Regorous do not recognize incorrectly formed rule sets.

4 Formalizing ISO 26262 part 3

In this section, we illustrate the methodological guideline depicted in Figure 2 by formalizing the requirements provided in Table 1. The formalization starts from clause 5 (see Table 1) providing requirements for the *definition and description of the item*. There are no prerequisites in this clause. Thus, we continue with the formalization of the title of the phase, which provides the obligation to initiate item definition (see rule *r*5). Then, we continue with the formalization of requirement 5.4.1, that defines guidance to provide the functional and non-functional requirements for the item definition. This requirements is formalized by using the SCP *Guidance* in which initially, the components of the guidance should be provided (see rules *r*5.4.1.*a* and *r*5.4.1.*b*), and then, they integrate the main normative provision (see *r*5.4.1). When all the requirements are formalized, we proceed with the work products, which are defined in clause 5.5. To formalize a work product, the requirements (in this case specified by clause 5.4) are presented as antecedents, and the work product itself is the normative provision (see *r*.5.5).

$$\textbf{r5:} \Rightarrow [O]initiateItemDefinition$$
$$\textbf{r5.4.1.a:} performItemDefinition \Rightarrow [O]provideFunctionalConcept$$
$$\textbf{r5.4.1.b:} performItemDefinition \Rightarrow [O]provideOperationalConstraints$$
$$\textbf{r5.4.1:} provideFunctionalConcept, provideOperationalConstraints \Rightarrow [O]provideFunctionalAndNonFunctionalRequirements$$
$$\textbf{r.5.5:} provideFunctionalAndNonFunctionalRequirements \Rightarrow [O]produceItemDefinition$$

Clause 6 in Table 1 presents the title of the clause and its inputs. Thus, we only applied the steps related to the formalization of the prerequisites and the title of the clause. The

prerequisites were already formalized in clause 5, (see Table 1, *Inputs of this clause: item definition in accordance with 5.5.*). Therefore, it is only needed the formalization of the title which defines the obligation of performing the phase *Initiation of the safety lifecycle* after the normative provision *produceItemDefinition* (see rule $r6$).

$$\textbf{r6:} produceItemDefinition \Rightarrow [O]initiateInitiationSafetyLifeCycle$$

Clause 7, which is related to *Hazard analysis and risk assessment*, does not mention any inputs for the clause and it is formalized as rule $r5$ (see rule $r7$). Two requirements are described in tables. The first requirement refers to a table which has constitutive information. The formalization of this table is done by taking the description of the entries as the antecedent of the rule and its category as the normative provision (see rules $r7.4.3.a$ to $7.4.3.d$).

$$\begin{aligned}
\textbf{r7:} &\Rightarrow [O]performHazardAnalysisAndRiskAssessment\\
\textbf{r7.4.3.a:} &descriptionSeverityS0 \Rightarrow [O]CatergoryNoInjuries\\
\textbf{r7.4.3.b:} &descriptionSeverityS1 \Rightarrow [O]CatergoryLightAndModerateInjuries\\
\textbf{r7.4.3.c:} &descriptionSeverityS2 \Rightarrow [O]CatergorySevereAndlifeThreateningInjuries\\
\textbf{r7.4.3.d:} &descriptionSeverityS2 \Rightarrow [O]CatergoryFatalInjuries
\end{aligned}$$

Clause 7.4.4 is presented in a table with alternative entries. We take into account the selection of methods for ASIL A. As recalled in Section 2.1 for alternative entries the normative provision suggest the obligation to provide a combination of the methods listed in the table (see rule $r7.4.4.a$), which at the same time obliges the selection of those with higher recommendation level for the ASIL, in this case, *Informal Notations* (see rule$r7.4.4.b$). Also, a rationale shall be given that the selected methods comply with the corresponding requirements (see rule $r7.4.4.c$). If the highest recommended is selected, only a weak rationale (i.e., a less stringent explanation of the selection) must be provided. However, if the highest recommended is not selected, a more elaborated rationale (called strong) should be provided (see rule $r7.4.4.e$). A domain expert should verify the rationales (strong and weak) (see rule $r7.4.4.d$ and $r7.4.4.f$). Providing the strong rationale, its verification and the methods selected, grant the permit of not providing the combinations of the recommended methods (see rule $r7.4.4.g$).

$$\begin{aligned}
\textbf{r7.4.4.a:} &verifyASILA \Rightarrow [O]provideCombinationOfListedMethods\\
\textbf{r7.4.4.b:} &provideCombinationOfListedMethods \Rightarrow [O]includeInformalNotations\\
\textbf{r7.4.4.c:} &includeInformalNotations \Rightarrow [O]attachWeakRationaleSupportingListedMethods\\
\textbf{r7.4.4.d:} &attachWeakRationaleSupportingMethods \Rightarrow [O]VerifyWeakRationaleByDomainExpert\\
\textbf{r7.4.4.e:} &includeNotListedMethods \Rightarrow [O]atachStrongRationaleSupportingNotListedMethods\\
\textbf{r7.4.4.f:} &attachStrongRationaleSupportingNotListedMethods \Rightarrow [O]verifyStrongRationaleByDomainExpert\\
\textbf{r7.4.4.g:} &includeNotListedMethods, atachStrongRationaleSupportingNotListedMethods,\\
&verifyStrongRationaleByDomainExpert \Rightarrow [P] - provideCombinationOfTheListedMethods\\
&r7.4.4.g > r7.4.4.a
\end{aligned}$$

5 Discussion

Interpreting and specifying ISO 26262 requirements in FCL can be time-consuming and error-prone. One reason is that ISO 26262 is a large document with hundreds of

requirements, which like many other standards and regulations, are difficult to interpret. The other reason is that FCL is not yet enough known and existing examples (mostly from the business and legal domains) are insufficient to guide the formalization of the normative notions that constrain the processes used in safety-critical development projects. However, the effort invested in the production of formal specifications of safety standards is compensated by several advantages, i.e., deep understanding of its requirements, practical application in development projects, and the provision of an essential input for facilitating automated compliance checking. Therefore, we consider that discovering efficient ways to proceed with the formalization work can boost the usage of this formal language in the compliance tasks in the safety-critical context. In the remainder, we discuss some aspects that were observed during the performing of the formalization-oriented pre-processing of ISO 26262 and the formalization of its part 3.

Useful formalization path: As recalled in Section 2.1, ISO 26262 is structured in a specific way. However, not all the structures should be used to obtain the formal specification in FCL. Therefore, performing a pre-processing of ISO 26262 provides us an useful formalization path to follow, i.e., a methodological guideline and SCP templates. Process engineers in the automotive context, who are interested in starting their formalization work with FCL, may find useful this formalization path since it permits to focus on specific tasks and skip some others that may be not relevant in the formalization process. Performing a similar pre-processing of safety standards beyond ISO 26262 may also be useful for increasing and spreading the use of FCL and its potential benefits.

Related skills, competencies and responsibilities: In our experience, group brainstorming sessions have facilitated the production of the FCL specifications. In particular, the participation of different kind of experts has provided different views that were important in the discussions performed during the formalization process. We highlight the fact that brainstorming sessions were mainly guided by the certification expert, whose knowledge provided specific details that are of importance for the safety auditor during the safety assessment. The opportunity to have an FCL expert speeded up the formalization and the creation of templates for reuse. However, FCL is not a very known language. Thus, there are not many FCL experts available. Therefore, it is necessary to provide training courses for teaching FCL. The target group for the training may be mainly conformed by process engineers who already have expertise in compliance management.

Tooling: In our current work, the rules were written manually, introducing the possibility of typos in the syntax of the rules and inconsistencies in the rules statements. Therefore, the production of our initial FCL specifications resulted in incoherent files that were not understood by the compliance checker. To solved this issue, edition and syntactic correctness of the FCL specifications were ensured manually. However, manual corrections are long and tedious activities. For this reason, we consider that the provision of tools for supporting the process of writing and verifying rules, as well as the creation and instantiation of SCP templates should be developed. Part of our work should also be the provision of these tools.

6 Related Work

The collection of experiences distilled from formalization projects is an advisable way to save time and avoid mistakes in future projects. We can find lessons learned in the formalization of software engineering standards in [10]. In a similar way, we have collected the specific lessons learned in a methodological guideline, aiming at facilitating the formalization of the ISO 26262 clauses. Guidelines are also widely used to spread the used of novel methods in engineering tasks. One example is the Oracle Policy Modeling best practices guidelines [11] whose aim is helping analysts to describe the way in which different types of business rules can be modeled. Similarly, a methodology to guide companies to establish Cyber-Physical Social System data subjects consent and data usage policies (described in OWL) is presented in [12]. Guidelines for supporting the formal representation of safety regulatory requirements (using Z) are introduced in [13]. The use of tabular expressions in [14] can also be seen as a guideline to generate formal models of system requirements. The authors of FCL have also published explicative examples of the modeling of FCL rules within the business context, e.g., [15, 16], which can be used as a guideline for learning the language.

The use of FCL for supporting compliance management tasks in automotive is a novelty. We did not find yet specific examples or guidelines that apply to the domain. Therefore, the results of the formalization-oriented pre-processing of ISO 26262 documented in this paper may be of interest for process engineers involved in the cars manufacturing. Additionally, we consider that this work can be used as a starting point to derive domain-specific guidance applicable to process-based safety standards beyond ISO 26262.

7 Conclusions and Future Work

In this paper, we presented the lessons learned from performing a formalization-oriented pre-processing of ISO 26262. Initially, we identify the essential normative structures, namely those structures that define the safety process to be adopted for developing the car's safety-critical systems. Then, we identified the repetitive structures of the standard that can be considered SCP. With the identified SCP, we create templates to consolidate a reusable knowledge base for future formalization jobs. The knowledge gathered in the pre-processing is used to define a methodological guideline for facilitating the formalization of normative clauses in ISO 26262. We also illustrate the defined methodology by formalizing ISO 26262 part 3 and discussed our findings.

From the discussion presented in Section 5, we consider that one important part of the future work is the training of process engineers in FCL. Therefore, a course called *Quality assurance - Certification of safety-critical (software) systems*[3], which is under construction, will consider *an overview of compliance checking* and *the formalization of compliance rules with FCL*. We also need to optimize the creation and verification of rule sets. Thus, we are considering the design and development of a pattern-based rule editor to facilitate rules creation, and rule sets verification. We consider that methodological guidelines are also needed in other safety-critical domains. Thus, we aim at

[3] http://www.promptedu.se/quality-assurance-certification-of-safety-critical-software-systems/

studying other safety standards and adapt any required step resulting from their specificities. This work is also expected to be combined with previously achieved results [7, 8] regarding the provision of automated compliance checking vision for the safety-critical context.

Acknowledgments. This work is supported by the EU and VINNOVA via the ECSEL JU project AMASS (No. 692474) [17].

References

1. ISO 26262: Road Vehicles-Functional Safety. International Standard (2011)
2. Francesconi, E.: Semantic model for legal resources: Annotation and reasoning over normative provisions. Semantic Web **7**(3) (2016) 255–265
3. Hashmi, M., Governatori, G., Wynn, M.T.: Normative requirements for regulatory compliance: An abstract formal framework. Information Systems Frontiers **18**(3) (2016) 429–455
4. Nute, D.: Defeasible Logic. In: International Conference on Applications of Prolog, Springer (2001) 151–169
5. Alberti, M., Gavanelli, M., Lamma, E., Riguzzi, F., Zese, R.: Dischargeable Obligations in Abductive Logic Programming. In Springer, ed.: International Joint Conference on Rules and Reasoning. (2017) 7–21
6. Governatori, G.: Representing business contracts in RuleML. International Journal of Cooperative Information Systems. (2005) 181–216
7. Castellanos Ardila, J.P., Gallina, B., Ul Muram, F.: Enabling Compliance Checking against Safety Standards from SPEM 2.0 Process Models. In: Euromicro Conference on Software Engineering and Advanced Applications (SEAA). (2018)
8. Castellanos Ardila, J.P., Gallina, B., UL Muram, F.: Transforming SPEM 2.0-compatible Process Models into Models Checkable for Compliance. In: 18th International SPICE Conference. (2018)
9. Castellanos Ardila, J.P., Gallina, B.: Formal Contract Logic Based Patterns for Facilitating Compliance Checking against ISO 26262. In: 1st Workshop on Technologies for Regulatory Compliance (TeReCom). (2017) 65–72
10. Verlage, M., Munch, J.: Formalizing software engineering standards. In: Third IEEE International Software Engineering Standards Symposium and Forum. (1997) 196–206
11. Lee, J.: Oracle Policy Automation (OPA). Best Practice Guide for policy Modelers. (2018)
12. Fernandez, J.: Deliverable 6.1: Privacy policy formalization (v. 1) (2018)
13. Vilkomir, S., Bowen, J., Ghose, A.: Formalization and assessment of regulatory requirements for safety-critical software. Innovations in Systems and Software Engineering **2**(3-4) (2006) 165–178
14. Singh, N., Lawford, M., Maibaum, T., Wassyng, A.: Use of Tabular Expressions for Refinement Automation. In: International Conference on Model and Data Engineering. (2017) 167–182
15. Governatori, G.: Practical normative reasoning with defeasible deontic logic. In: Reasoning Web International Summer School. (2018) 1–25
16. Governatori, G.: The regorous approach to process compliance. In: IEEE 19th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations. (2015) 33–40
17. AMASS.: Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems