Supporting Process Design in the Autonomous Era with New Standards and Guidelines

Masao Ito(✉)

NIL Inc., 2-17-7, Kinuta, Setagaya, Tokyo, Japan
nil@nil.co.jp

Abstract. It isn't easy to define a general and clear process in a new system such as an autonomous vehicle. The new technology is complex and lacks experience. In this paper, we reconsider the process used for system development. We assume that the process changes depending on the characteristics of a project [1]. Of course, this is a general agreement. In autonomous vehicles, new technological elements such as AI and ensuring safety are further required. So, it is meaningful to consider how to think about the development process. New standards, guidelines and documents such as UL4600 are emerging for autonomous vehicles. In addition to many standards, dealing with these presents difficulties. We believe that better process design is possible by using the Toulmin model.

Keywords: Process design · Toulmin model · Autonomous vehicle · AI · UL4600 · SCSC-153A

1 Introduction

Currently, many types of research and developments on autonomous vehicles (SEI Level 4 or 5) [2] are underway. Autonomous vehicles do not assume human operations. On the other hand, the standards and guidelines up to now are, of course, premised on human operations. Because of this difference, it is necessary to consider safety differently than before.

In an autonomous vehicle, the machine recognizes the environment and controls it, instead of humans. The main issues regarding safety are as follows.

(A) Is it possible to ensure safety when an autonomous vehicle recognizes the environment and steers itself?
(B) How should we evaluate new technologies such as AI that performs control on behalf of humans?

Several new guidelines and standards have been published or planned to address these challenges. Although not necessarily exhaustive, some are shown in Table 1.

UL4600 [3] is for autonomous vehicles. In UL4600, the regulations mainly use the safety case. It also includes guidelines for AI systems. BSI 1880 [4] is a standard for autonomous vehicle control systems and covers a wide variety of vehicles. SCSC-153A [5] is also targeted at autonomous vehicles. This document has cross-references to UL4600.

M. Yilmaz et al. (Eds.): EuroSPI 2020, CCIS 1251, pp. 525–535, 2020.
https://doi.org/10.1007/978-3-030-56441-4_39

Table 1. Recent standard, gridlines, and documents for AS

ID	No.	Title	Type	Issue date
(1)	UL4600	Standard for Safety for the Evaluation of Autonomous Products	Standard	April/2020
(2)	BSI I880	Guidelines for developing and assessing control systems for automated vehicles	Standard (PAS)	April/2020
(3)	SCSC-I53A	Safety Assurance Objectives for Autonomous Systems	Document	Jan/2020
(4)	MISRA	Guidelines for Automotive Safety Arguments	Guidelines	September/2019
(5)	DIN SPEC 9200I-I	Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part I: Quality Meta Model	Standard (PAS)	April/2019
(6)	ISO/IEC 29II9-II	Testing AI-Based Systems	TR	–

As a document related to the safety case, MISRA gives guidelines on the safety and logical structure of automobiles [6]. It mainly relates to Part 3 of ISO 26262 and gives detailed guidelines for applying the Safety Case. The structure of the safety case is shown here as a template.

The following are the standards/guidelines specialized for AI. The DIN SPEC 92001 series are life cycle processes and quality requirements related to AI [7]. At present, only quality metamodels are published. As for handling AI testing, ISO IEC 29119-11 (Software and systems engineering—Software testing—Part 11: Testing of AI-based systems) is being established, but it is under development [8].

Here, we would like to consider what kind of efforts are made concerning the claims of these documents from the viewpoint of the process, rather than comparing individual standards and guidelines. In Table 1, we use UL4600 in (1) and SCSC-153A in (3), which have descriptions directly related to autonomous vehicles and AI, as examples in this paper.

In the next chapter, we briefly examine UL4600 and SCSC-153A. In Sect. 3, we reconsider the process of ensuring safety through a survey of these standards. In Sect. 4, we propose a method to support process design, using various standards and guidelines. We use the Toulmin model.

2 UL4600 and SCSC-153A

2.1 Characteristics

UL4600 aims to "*support(s) state-of-the-art safety case approaches, which permit standardizing an approach to safety while at the same time enabling the use of rapidly evolving technology, tools, and methods. It is both technology neutral and development process agnostic)*" [9]. That is, the objective is to react quickly as the autonomous

vehicle evolves. Also, this standard has many prompt lists, predetermined lists of risk categories. These lists assist us in creating a safety case.

The UL 4600 chaptering is very interesting (Fig. 1). The first half of the standard (up to Chapter 8) is probably the one that the standard setters consider important, namely the safety case (Chapter 5), Risk Assessment (Chapter 6), and autonomy (Chapter 8). It has become. The second half is a relatively traditional title, but it's still a bit different from the general one; Software and system engineering process (Chapter 9), dependability (Chapter 10), data and network (Chapter 11) and V & V and testing (Chapter 12).

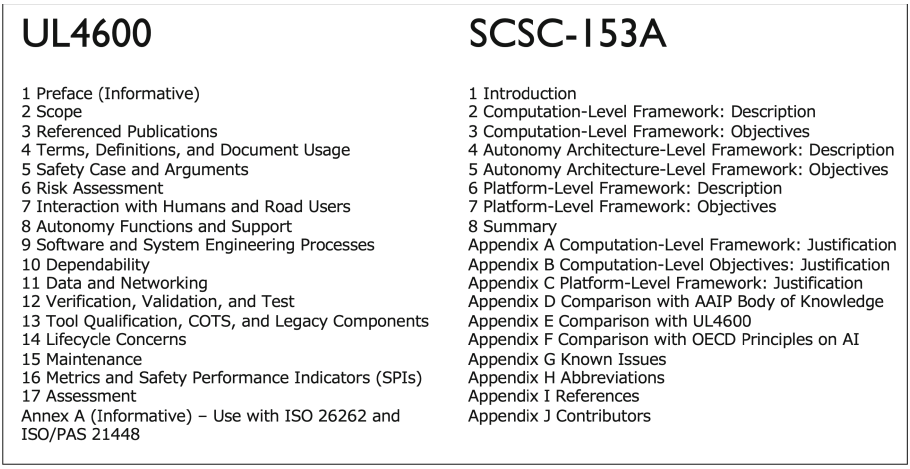

Fig. 1. Structure of UL4600 and SCSC-153A chapters

The SCSC-153A aims to focus "*on aspects directly related to autonomy, and enabling technologies such as AI and ML, rather than more general safety engineering or system engineering, where it is assumed that relevant general standards, guidelines and best practice will be applied*".

The whole is divided into three levels: compute level, autonomous architecture level, and platform level. Each level has a projection. At the computational level, there are five projections: Experience, Task, Algorithm, Software, and Hardware. It has the objective for each projection.: "*Each objective is accompanied by a discussion that illustrates how the objective contributes to AS safety. This is followed by examples of approaches that could be taken to satisfy, or partially satisfy, the objective*."

2.2 Processes

In the introduction to UL4600, there is the following statement: *"Traditional safety standards are prescriptive"*. And these traditional standards provide *"how to do safety (process, work products)"*. So, UL4600 says it provides a goal. By the way, the traditional standards are ISO 26262 [10], ISO/PAS 21448 [11], IEC 61508, MIL-STD

882 and so forth. If we use the term process in a sense that is more like a procedure, we think there is probably a misunderstanding here. Indeed, ISO 26262 defines the work products, but it doesn't have the procedure.

What is the process is a delicate matter, but it is described as follows in the standard ISO 12207 [12] for software life cycle processes. *"... this document does not prescribe any particular sequence of processes within the life cycle model. The sequence of the processes is determined by project objectives and by selection of the life cycle model"*. ISO 26262 conforms to ISO 12207 Just because you specify a process does not mean that you specify a chronological order. I will discuss this point in the next chapter.

SCSC-153A, like its title, is a description of objectives regarding safety assurance. Therefore, there is not much description of the process. From the few descriptions, there is the following description regarding security as an example.

We can't get the same assurance evidence as to the normal development process.: "*..., from a practical perspective, most ML pipelines make extensive use of open source frameworks and tools which, generally speaking, do not provide the same type of assurance evidence. as is delivered by development processes for critical software*".

In Table 10 of SCSC-153A, Mapping Projections to Typical Software Development, there is a correspondence between standard life cycle and Computation level projection. For example, Experience projection is related to design and implementation. Task projection is related to everything from Plan to Test.

3 Rethink the Process

UL4600 claims that existing safety standards are prescriptive (c.f. Sect. 2.2). To be sure, there is such an aspect, but I think that it may be said to be the influence of the times when the standard was established. At the initial examination stage of ISO 26262, I believe that passenger cars of Level 3 and above were not the centre of interest. Generally, in the area where the change is drastic, and the technical accumulation is small when we try to make some rule, it becomes the goal-based writing method. Also, the explanations are exemplary or an exhaustive list. Here, we will reconsider the process and consider a way that is effective in both formats.

3.1 Is the Process a Kind of Recipe?

Here, we will reconsider the process. First of all, think about whether a process is a time-based definition of activities like a recipe.

Refer to the definition of ISO 12207: the process is a *"set of interrelated or interacting activities that transforms inputs into outputs"*. Of course, I'm not saying here a sequence of activities. More aggressively:

This document does not prescribe any particular life cycle model. Instead it defines a set of processes, termed life cycle processes, which can be used in the definition of the system's life cycle. Also, this document does not prescribe any particular sequence of processes within the life cycle model.

However, the input and output may implicitly determine the order. Now, if activity A outputs a work product and activity B takes that the one as input, then activity A must

precede activity B. it may be possible to say that the order of execution of activities is determined by specifying input and output without defining the order explicitly.

In ISO 26262-3, we need to define items and create work products called item definitions. To enact the Hazard Analysis and Risk Assessment (HARA) activity, we need the item definition as an input.

```
5.5 Work products
5.5.1 Item definition resulting from requirements in 5.4.
...
6 Hazard analysis and risk assessment
6.3 Inputs to this clause
6.3.1 Prerequisites
The following information shall be available: — item definition
in accordance with 5.5.1.
```

There is an order relation between the item definition and HARA via the work product called the item definition. But this doesn't seem to have any further meaning. It's just like you can't test that code without coding it. The actual working process is not that simple.

In Agile development, we write the code for testing as preparation for unit testing before implementation. Of course, we do test after coding, but we can design test cases and then write the test code. The test-first approach might have the potential to produce good quality code with validation in mind. You may also notice design mistakes in the process of creating test cases.

We will consider the above in detail. Let's say design is D and coding is C. The test is T. Abstractly, we can think the following order.

$$D \prec C \prec T \tag{1}$$

The symbol '$\prec$ ' indicates the execution order here.
Now, decompose T1 as follows.

$$T = T1 \prec T2 \prec T3 \tag{2}$$

Here, T1 is test case creation, T2 is test code creation, and T3 is test execution.
Simply, (1) becomes

$$D \prec C \prec (T1 \prec T2 \prec T3) \tag{3}$$

However, if you think about test first, you can also:

$$D \prec T1 \prec T2 \prec C \prec T3 \tag{4}$$

Alternatively,

$$T1 \prec D \prec T2 \prec C \prec T3 \tag{5}$$

The input and output of each process element do not change in either case. Only the choice is different.

Next, consider the process instance. Now assume that {di1, di2, …, din} is an instance of process element D.

di1, …, d1k: DP1 is a critical element, and I want to confirm its feasibility early. On the other hand, d1l, …, d1n: DP2 is an easy element so that we can design it later.

At this time, if you take a strategy to tackle difficult issues first, we can get the sequence showing below:

$$DP1 \prec CP1 \prec TP1 \prec DP2 \prec CP2 \prec TP2 \tag{6}$$

This is a kind of incremental approach. It is easy to find for us by using parenthesis.

$$(DP1 \prec CP1 \prec TP1) \prec (DP2 \prec CP2 \prec CP2) \prec \ldots \tag{7}$$

The work products are the same in both cases. The only difference is the choice of the process designer.

That is, the work product definition of each activity does not uniquely determine the order in which the activities are enacted.

3.2 How We Define a Recipe for a Project?

As mentioned earlier, each project will determine the appropriate process based on the given conditions. At this time, how will the process be decided? Usually, you will make a trade-off of QCD based on your development experience. If you are trying a new process model for the first time, you are going to make a trial and use that experience. Also, we design a process. In this section, we consider ways to support process design.

The method we propose uses the Toulmin model [13]. The Toulmin model is the idea behind the GSN to express the safety case. Initially, in an attempt to clarify the structure of everyday discussions, the figures were written for the explanation, and there was no precise definition. Various people are expanding, but here we consider the original expression (Fig. 2).

- Data: The facts or evidence used to prove the argument.
- Claim: The statement being argued.
- Warrants: The general, hypothetical (and often implicit) logical statements that serve as bridges between the claim and the data.
- Qualifiers: Statements that limit the strength of the argument or statements that propose the conditions under which the argument is true.
- Rebuttals: Counter-arguments or statements indicating circumstances when the general argument does not hold true.
- Backing: Statements that serve to support the warrants.

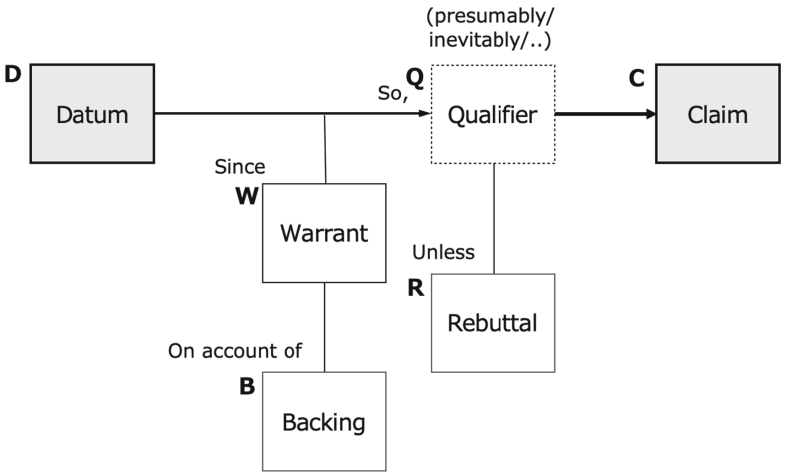

Fig. 2. Toulmin model

By reversing this structure in time, we can see the similarity with GSN. However, in general, there is no equivalent to Qualifier, which means the confidence factor of the claim. The standard GSN does not have an equivalent for this Qualifier.

Let me give you an example. Now, let's say that we have to carry out development using technology elements that we have never used (D). Since it is unclear whether the technical element can be used, it may be possible to choose to proceed with development (W) while checking. Generally, this method is known as incremental development (B). From now on, perhaps (Q), select the development process in incremental development (C). However, it may become difficult if there are restrictions on costs and delivery (R).

In this way, it is possible to evaluate if an argument is defined. Of course, since there is a modality there, it cannot be said to be entirely correct, and in some cases, Claim may not be established. It is possible to evaluate including that.

4 Toulmin Model and New Technology Elements

At present, it is not easy to cover all aspects of how to ensure safety in the new field of autonomous vehicles and the new technological field of AI. It is also possible that the goals for ensuring safety will change as the technology evolves. Therefore, developers need to follow the revisions of many existing standards/guidelines/documents. Furthermore, the number of target documents is expected to increase in the future. For example, there is an IEEE P7000 series on ethics that is not included this time.

In this paper, we propose the following method. We are organizing the claims for safety in each document using the Toulmin model. It gives users a centralized way to access what a document requires. Also, you can easily change or add.

On the other hand, there are difficulties. For example, terms are not always consistent across standards. For example, the SCSC-153A has 'Experience' projection at

the computation level. This refers to the dataset used for training. However, similar things may not be called Experience in other documents. We allow replacement using terminology dictionaries, but not a complete solution. When you use Experience literally, it spoils its meaning by replacement. The final solution can only be obtained by referring to and understanding the original document. The main purpose of this scheme is to get to the relevant part of the required document without leakage.

Also, multiple W/B/R may be required for a particular Claim. For the graphical description, you can represent everything as nodes, but I don't think it is a proper method. This is because the structure that is too complicated impairs the intelligibility of the diagram. We express that there is another W/B/R in Qualifier as a link.

Methods of integrating knowledge are often difficult to use continuously in real problems (e.g. Unified Process). We would like to use Toulmin's model as a way to reach all the relevant parts of a proper document without leaking them. We are not trying to integrate everything.

For SCSC-153A, take an example from COM1-1.

Belonging to the computational experience are the following four Objectives.

COM1-1: Data is acquired and controlled appropriately.
COM1-2: Pre-processing methods do not introduce errors.
COM1-3: Data captures the required algorithm behaviour.
COM1-4: Adverse effects arising from distribution shift are protected against.

These are Objectives for the data used in ML. The following sentence can be found in the Example.

If a complete data set is acquired from an external party then care should be taken to ensure that it has not been subject to "Data Poisoning"; for example, the addition of a small number of maliciously crafted samples can create a backdoor" The same techniques used to confirm the authenticity of information downloaded from the Internet (e.g., checksums) may be helpful here.

An example of the representation using the Toulmin model is shown in Fig. 3.

Here, the data happens to be the same as the node name in the Toulmin model. The qualifier is "with other warrants" because there can be other Ws. For example, prevention of semantic errors in data definition is equivalent (whether the vehicle speed is expressed in MKH or MPH in vehicle speed data). This can also be added as W or B.

The left side at the top of the figure shows the target system, the center shows the technology category, and the right side shows the related phases in the life cycle; R: requirement (analysis), D: design, I: implementation, T: test. Figure 3 shows that ML for autonomous vehicles is a factor to be considered in design (D) and implementation (I).

You can find a similar example in UML4600. There are the following rules regarding data for ML.

8.5.3 Machine learning training and V & V shall use acceptable data.
c) Data provenance: historical record of data and its origins
NOTE: This can support better understanding of data, track back sources of errors, and provide auditability and quality trails

Similarly, Fig. 4 shows the thing written by the Toulmin model.

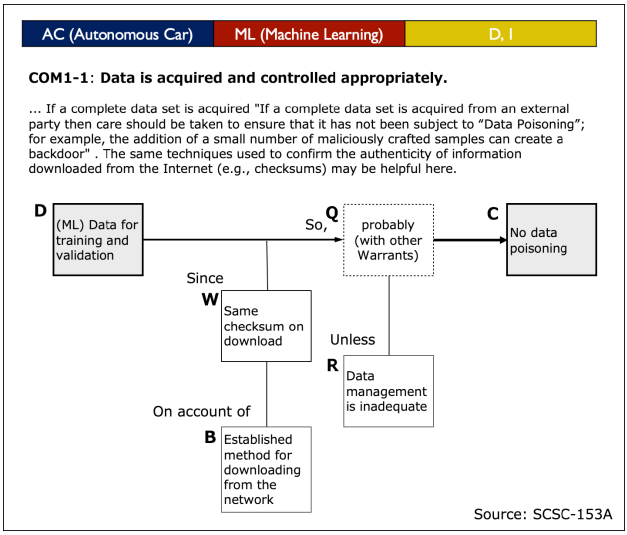

Fig. 3. A Toulmin model from SCSC-153A COM1-1

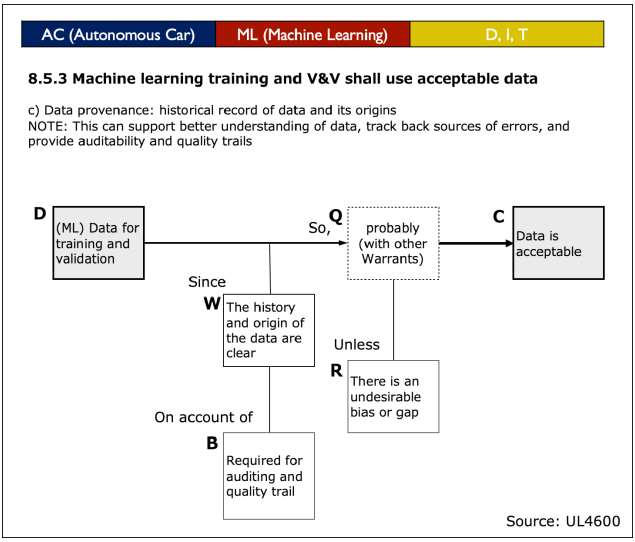

Fig. 4. A Toulmin model from UL4600 8.5.3

5 Summary

As we see, several standards and guidelines have been established or are being developed to ensure safety for the practical application of autonomous vehicles. These documents also reflect the fact that many technologies in the new AI category are used in the new platform of autonomous vehicles. As technology evolves rapidly, so does the way it ensures safety. This document provides an overview of these new documents and highlights their characteristics.

Next, we consider the relationship with the process. The discussion here is traditional. That is, a process does not necessarily describe a temporal sequence. The process designer understands the process elements, considers the process instance, and defines the process for each project based on the trade-off.

Also, these standards/guidelines are less likely to be related to their life cycle than existing standards (e.g. ISO 26262). Both UL4600 and SCSC-153A, which were taken up this time, are so-called goal-based descriptions. Therefore, process designers encounter difficulties when considering what to achieve at what timing when designing a process that considers safety.

In this paper, we have proposed a method for organizing these new documents using Toulmin's model. I don't think of using a graph with a large argument structure. As shown in Sect. 4, for one goal (requirement), one argument structure is made into one card using the Toulmin model. Depending on the stage of development, we will collect the necessary cards and judge whether they are sufficient. Proper maintenance is easy. You can also add new cards as needed.

We believe this method will be useful as we adapt to new standards and guidelines that will continue for the next few years.

References

1. Korsaa, M., et al.: The SPI manifesto and the ECQA SPI manager certification scheme. J. Softw. Evol. Process **24**(5), 525–540 (2012)
2. SAE J3016. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, J3016_201806: SAE International (2018)
3. UL4600. Standard for Safety for the Evaluation of Autonomous Products, UL (2020)
4. BSI PAS 1880:2020. Guidelines for developing and accessing control systems for automated vehicles (2020)
5. SCSC SASWG, SASC-153A. Safety Assurance Objectives for Autonomous Systems (2020)
6. MISRA. Guidelines for Automotive Safety Arguments (2019)
7. DIN SPEC 92001. Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part 1: Quality Meta Model (2019)
8. ISO/IEC 29119-11. Software and systems engineering—Software testing—Part 11: Testing of AI-based systems (under development)

9. https://medium.com/@pr_97195/an-overview-of-draft-ul-4600-standard-for-safety-for-the-evaluation-of-autonomous-products-a50083762591. Accessed 4 2020
10. ISO 26262:2018. Road vehicles – Functional safety (2018)
11. ISO/PAS 21448:2019. Road vehicles – Safety of the intended functionality (2019)
12. ISO/IEC/IEEE 12207:2017. Systems and software engineering—Software life cycle processes (2017)
13. Toulmin, S.E.: The uses of argument updated ed. Cambridge University Press, p. 97 (2003)