How to Reach Complete Safety Requirement Refinement for Autonomous Vehicles

Carl Bergenhem, Rolf Johansson, Andreas Söderberg, Jonas Nilsson, Jörgen Tryggvesson, Martin Törngren, Stig Ursing

How to Reach Complete Safety Requirement Refinement for Autonomous Vehicles

Carl Bergenhem
Qamcom Research and Technology

Rolf Johansson, Andreas Söderberg
SP Research Institute of Sweden

Jonas Nilsson
Volvo Car Corporation

Jörgen Tryggvesson
Comentor

Martin Törngren
KTH

Stig Ursing
Semcon

***Abstract* — The introduction of highly automated driving and autonomous road vehicles will imply new functional safety challenges. The higher complexity and the partly implicit definition of the tasks for the E/E systems will make it harder to argue completeness and correctness of the safety requirements in each phase of the ISO 26262 lifecycle. This paper discusses the new situation in terms of an increasing semantic gap, and recommends to perform more safety refinement steps. As a consequence, ISO 26262 should be amended with activities prescribing new refinements levels.**

Keywords — ISO 26262, autonomous vehicles, safety requirements, refinement verification, satisfaction argument.

I. INTRODUCTION

In a near future it is likely to find more or less autonomous (e.g. self-driving) road vehicles for public traffic in series production. For example, Volvo Car Group's project 'Drive Me' aims (2017) to featuring 100 self-driving cars on public roads in everyday driving conditions. The first test cars are already rolling around the Swedish city of Gothenburg [1]. Both Mahindra and Baidu are developing technology for autonomous vehicles [2, 3]. Mercedes has shown its concept for autonomous trucks [4].

A general question is to what extent autonomous systems will put new requirements on the international standard on functional safety for road vehicles, ISO 26262 [5]. A fundamental principle in this standard is that the scope for functional safety is confined to one 'item' at the time. This implies that there is no explicit requirement to show that the complete vehicle is functionally safe. The vehicle manufacturer (OEM) is free to declare the functionality that is realised by electronic/electrical (E/E) systems and organize this into a set of items. It is further assumed that the vehicle driver can perform the necessary actions to stay safe, either by this functionality or by other functionality that is implemented by other technologies than E/E system.

When autonomous vehicles are put in series production, the E/E system will need to provide all functionality to assure that the vehicle always behaves safely on the road, also without any involvement of a driver. Instead of a manual driver taking care of all unplanned and unforeseen situations, it is expected that the autonomous vehicle itself (by means of the E/E systems) can play this role. This implies that the set of items must be capable of all functionalities according to the declared scope of such an autonomous vehicle in order to behave safely on the road.

The current ISO 26262 requires that the documented and analysed list of hazardous events (HE) for each item is extensive enough, so that the derived Safety Goals (SG) will cover *all* possible hazardous events for that item. According to the ISO 26262 life cycle, there is an explicit requirement to verify this. The verification is documented in the work product: "Verification review report of the hazard analysis and risk assessment and the safety goals".

When autonomous vehicles are introduced, the completeness of the SGs needs to be argued not only w.r.t. each item, but also regarding all required functionality to handle all foreseeable situations in a safe way. If hazardous events exist that are not addressed by any defined item, then the set of items is incomplete and needs to be extended. There is no manual driver to act and taking care of the situations not covered by the existing items. One trivial way to handle this is to introduce a very general item - autonomous driver (AD); that takes care of everything not addressed anywhere else and coordinates the use of other items. Even if this may help us to show the completeness of the items, the problem of showing the completeness of all SGs remains. This problem may be very hard because it shall be shown that all possible scenarios and environments have been considered.

After the SGs have been found, and completeness of them have been showed (assuming that there is a way to show this), the reference life cycle states that a Functional Safety Concept (FSC) shall be defined. The FSC contains a set of functional safety requirements (FSR) that together imply all the SGs. After this follows the technical safety concept (TSC), in which a set of technical safety requirements (TSR) shall be shown how they imply the FSC. Each safety concept needs to be detailed enough to enable allocating responsibilities among the parts of concern, i.e. architectural elements.

When following the reference life cycle of ISO 26262, there might be a hand-over between customer and supplier at any stage, and for the same item this refinement of requirements may form a tree structure where the supplier handover may occur at different stages for different branches of the tree. Furthermore, it is possible that a supplier develops a Safety-Element-out-of-context (SEooC), where

assumptions on e.g. SGs and FSC are made as a starting point. On the other hand, it is also possible that the OEM (that develops in-context) defines a TSC that is detailed enough for the OEM perspective.

In ISO 26262 there are general requirements that any lower level safety concept needs to be verified and shown to be complete w.r.t. the higher level safety concept. Having a number of 'partial' safety concepts spread among a number of companies, this verification task is certainly challenging. This is especially true for autonomous vehicles where the complexity of the functionalities is significantly higher than for manually driven vehicles.

This paper proposes means for mastering the general problem of verifying that all steps of the safety requirements refinement are complete and correct. This is illustrated by an example as given in Section II. Section III defines the semantic gap. Section IV discusses solutions. Section V gives comment to "Safety of The Intended Function". Finally Section VI concludes the paper.

II. Example

To illustrate the reasoning in the rest of the paper a highly automated driving function is used. The item is defined as:

Item definition: *The function will offer in-lane autonomous driving, allowing the driver to engage in secondary activities, without the need for a lead vehicle at speeds up to 70 km/h.*

This is a very general definition, and it might be hard to identify all hazards and situations relevant for the hazard analysis and risk assessment (HA&RA). However, corresponding SGs might be:

SG A: *The host vehicle shall keep an absolute safe distance to other objects. An absolute safe distance is a distance such that the host vehicle shall always avoid a collision. ASIL B.*
SG B: *The host vehicle shall always stay on the road when the host vehicle speed is higher than 7 km/h. ASIL D.*

This list of SGs is clearly not complete, e.g. does not address user interaction with function, but as discussed in the coming sections it might be hard to show when the set of SG is complete w.r.t. all possible hazardous event in the item of concern. The next step for the safety requirement refinement is to define a FSC. In this example the functional architecture, is given in Fig. 1. For the refinement of SG A, FSRs are proposed:

FSR 1/2/3/4: *Sensing (allocated on each of camera/radar/infrared/sensor fusion) shall output the classification and position of appropriate Objects in front of the host vehicle, present in the current lane, with a tolerance in the distance of 10%. ASIL B.*
FSR 5: *Decision&Control shall output an Acceleration Request sufficient to avoid a collision with any Object. ASIL B.*
FSR 6: *Speed Control shall always control the Vehicle Acceleration according to the Acceleration Request with a maximum deviation of 2%. ASIL B.*

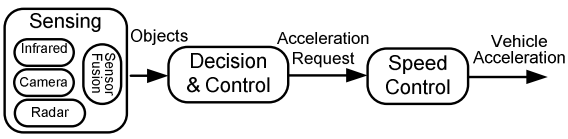

Fig. 1. Functional architecture

These requirements constitute a part of the FSC and may then be further broken down into a TSC. As with the example set of SGs, the list of FSRs for SG A is clearly not complete. Even with added FSRs, it is still a challenge to argue that an FSC fulfils the corresponding SG.

III. The Semantic Gap

Defining an autonomous function without arriving at an abstract description is a challenge. The Item definition in the example, Section II, leaves much room for added definition, e.g. concerning the driving environment. The HA&RA may be done at this high level of abstraction, but will inevitably assume implicit definitions in the item w.r.t. functionality: "everything shall work safely always". The outcome of HA&RA is hence SGs that tend to be abstract and will be at best challenging to synthesise into e.g. an FSC. The abstract nature of the (high-level) SG makes it difficult to determine whether it is fulfilled or not, i.e. correct and complete, by the composition. Further, it may be challenging to make a correct refinement of the abstract SG, i.e. to structure a solution.

In ISO 26262, every safety requirement refinement needs to be proven complete and correct by means of a verification activity e.g. ISO 26262-3:8.4.5.1. This implies that for the item it shall be verified that the set of SGs are complete, for each SG it shall be verified that a certain set of FSRs are complete, for each FSR it shall be verified that a certain set of TSR are complete, etc. In Section II, SGs are exemplified and it can be easily argued that the FSRs are not complete w.r.t. SG A. However, it would be equally hard to show completeness even after adding a few more FSR on the same level of abstraction. The "distance" between the SG and FSRs (or any other adjacent requirements levels) is denoted the Semantic Gap, see Fig. 2.

The concept phase of ISO 26262 (part 3) describes how SGs are determined from the results of the HA&RA. SGs are refined into FSRs, which implies that the SGs can be interpreted as top-level safety requirements in a layered requirement hierarchy. An observation is that the SG is a high-level description of an objective on vehicle level, and

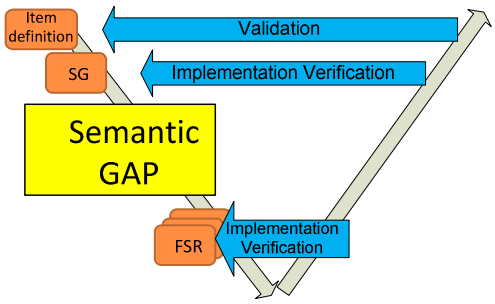

Fig. 2. The semantic gap

the refinement of the SG to reach FSC may need a substantial amount of assumptions, domain knowledge or other input. If no or only weak arguments for the refinement of SG to FSC exist, then verification to argue correctness and completeness is at best difficult.

Lamsweerde and Darimont [6] discuss elaboration of requirements and especially how to perform refinement. One of the observations from looking at use cases was that the result of refinement of requirements often was incomplete w.r.t. fulfilling the upper level requirement. The gap was found when attempting to formally prove the correctness and completeness of the composition. This is an indicator of the importance of the mandatory life cycle activities of safety requirement refinement verification, and that it is far from enough just to have a feeling that a set of refined requirements are complete without proving this.

IV. Solutions

A. *Refinement Verification*

A requirement (the upper level of two adjacent requirement levels) is refined into a composition of lower level requirements and rationale, known as satisfaction arguments [7, 8], shall be collected for the composition, see Fig. 4. This bridge of information should "fill" the semantic gap. Satisfaction arguments may be e.g. assumptions, domain knowledge, design patterns. This is essential in almost every non-trivial refinement. The rationale justifies the "refinement path taken" through the semantic gap and improves to traceability.

Satisfaction arguments are used in refinement verification to prove completeness and correctness. The result is a refinement verification report that give proof that the composition of refined requirements fulfil the requirement (upper level). We wish to distinguish refinement verification from implementation verification. The latter concerns ensuring that requirements are correctly implemented, see Fig. 4. Cleary the two activities have different goals and may utilise different tools.

The larger the semantic gap is, the harder is the refinement verification, and the more the completeness proof will rely on the satisfaction arguments. But encapsulating too much information in the satisfaction arguments, may lead to proofs that are less convincing as they rely on the same conclusions of the impact of (very much) domain knowledge. In order to construct convincing refinement verifications, we claim in this position paper that the corresponding semantic gaps should be as small as possible.

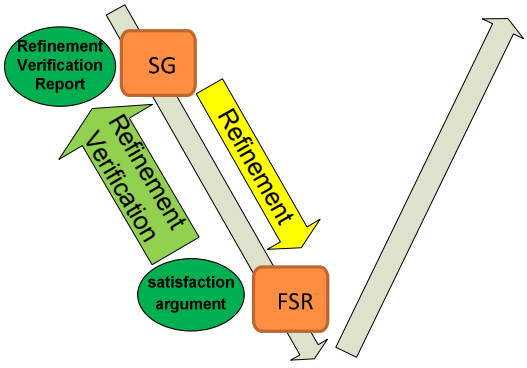

Fig. 4. Activities between adjacent requirement levels.

B. *Multi-level requirements hierarchy*

As discussed in the previous section, too large semantic gaps are problematic when developing convincing refinement verification. To improve the situation, we propose that refinements within same abstraction level are introduced in the ISO 26262 reference lifecycle. Note that the current version lacks refinement activities where the input and the output are of the same kind (SG-SG, FSC-FSC, TSC-TSC, etc.). Further, there are no defined activities which can host the refinement verification between safety requirements of the same kind. Adding requirement levels contributes to bridging the semantic by enabling manageable complexity of argumentation. This is a way of handling the scaling of complexity coming from highly autonomous functions.

Fig. 3. outlines such an extended reference life cycle. Satisfaction arguments and verification for each refinement are accumulated in the safety case. Following this pattern, the SGs in the example could be further refined. The more abstract SGs enable verification of the completeness towards the Item, and the refined SGs are to simplify performing the refinement verification of the FSR. For example, SG A, which is seen as a high level SG, can be refined into lower level SGs.

> **SG A.1/2:** *If a stationary {vehicle or obstacle} is present in the host vehicle's lane, the function shall brake the vehicle hard enough to completely avoid a collision. ASIL B.*
> **SG A.3/4:** *If a {moving vehicle or large animal} is present in or enters the host vehicle's lane, the function shall brake the vehicle hard enough to completely avoid a collision. ASIL B.*
> ***Satisfaction argument**: (simplified) All objects are either of: stationary objects, moving vehicles or stationary vehicles.*

It is assumed that there are no moving obstacles (that are not vehicles) that are hazardous to the host vehicle. This

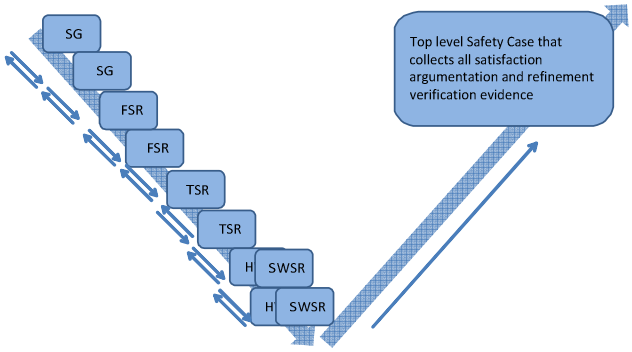

Fig. 3. A multi-level requirement hierarchy

structure of refinement to more detailed and differentiated SGs, enables having different strategies when refining to FSRs for identifying and classifying vehicles that are moving and stationary, respectively. FSRs are allocated with different implications on the different sensor blocks. In the sensor fusion algorithm different sensor inputs can have higher weighting according to the object, for example: Radar for moving vehicles; Camera for stationary vehicles and obstacles, and; Infrared for large animals.

The introduction of multi-level refinement is motivated as follows: 1) Analysis is simplified by making every step manageable; enabled by the reduced semantic gap. Refinement verification correspondingly simplified, including the satisfaction argumentation. In the above example of refining SG A, the relation to the FSC is clearer, i.e. the semantic gap is made smaller. 2) Multiple levels fit the current way that customer/supplier relation are organised and communicate. For example, both may have FSC or TSC, i.e. a concept at the same level. However they potentially have different scope and detail, e.g. the supplier TSC can be more detailed but have reduced scope than the customer´s.

V. Comments to Safety of The Intended Function

In the current work with revision of ISO 26262, there is a sub-working group denoted 'safety of the intended function' (SoTIF). The problem addressed, is that today there are examples of items that shall take care of complex environment sensing as e.g. part of an ADAS function. It is difficult to build a sensing system that is able to take care of *all* possible situations. Given that nominal performance limits of the sensing system are accepted, there is still the risk of (very rare) situations leading to violations of a SG, without any fault in the sensing system itself. The cause might be that the processing algorithm takes a hazardous decision about the environment. The SoTIF initiative aims at providing guidance to manage such a violation of a SG.

We claim that the SoTIF discussion is a consequence of improper safety requirement refinement and/or of improper item definition, i.e. the initial requirement statement. If the intended function is potentially hazardous in some situation, then the item is simply not well defined. We claim that every SG violation will manifest as a violation of the underlying safety requirement structure. This means that either at least one of underlying FSR is violated or is the verification of the completeness of the FSC not performed properly. Similarly, the violation of such a FSR implies that either at least one technical safety requirement is violated, or is the verification of the completeness of the TSC incorrect.

It is our understanding that the SoTIF discussion has evolved from a problem of handling implicit complex items. The SoTIF perspective becomes redundant once a methodology is applied in which the item definition is checked to reflect the responsibility of the E/E system and where the set of SGs can be shown to be complete w.r.t. all hazardous events possible according to the item definition. The complexity problem will then be possible to master by introducing safety requirement in so many steps that each step can be verified w.r.t. correctness and completeness.

VI. Conclusions and Future Work

In this position paper we claim that for autonomous vehicles, the safety requirement hierarchy of ISO 26262 implies sematic gaps that may be hard to verify w.r.t. correctness and completeness. Our position is that ISO 26262 needs to be complemented to explicitly prescribe activities, e.g. refinement verification, and corresponding work products for refinement, on every existing level of the reference life cycle. In current version there are requirement for correctness and completes when moving between levels of different kind, e.g. SG-FSC and FSC-TSC. However there is no support or corresponding requirements for making refinement to a new iterations of the same level, e.g. SG-SG and FSC-FSC. Consequently there is no prescribed work product in which e.g. a refined FSC is verified to be correct and complete w.r.t. to another higher level FSC.

We argue the importance of having a strong evidence e.g. proof in a formal syntax for the correctness and completeness of every refinement verification, and that the domain knowledge acting as satisfaction arguments needs to be explicit enough to serve as a part of this formalism. Large semantic gaps imply complex satisfaction arguments, which are hard to use in a convincing proof.

The introduction of iterative refinement both solves the problem of supplier and customer being active in the same activities/phase of ISO 26262 and decreasing each sematic gap to a size where evidence, including satisfaction arguments, can be formulated for each refinement verification.

Acknowledgment

This work has been financed by the Swedish government agency for innovation (VINNOVA) in the FUSE project (ref 2013-02650). The participating companies are: Volvo Car Corporation, SP, Semcon, Qamcom, KTH and Comentor.

References

[1] M. Persson, "Volvo Car Group's first self-driving Autopilot cars test on public roads around Gothenburg," ed: Volvo Car Group Media Relations, 2014-04-29.

[2] V. Aggarwal. (2015-03-05) Mahindra experimenting with driverless cars; developing software to control car's movement in India. *Economic Times*.

[3] Bloomberg, "CEO Says: Baidu May Introduce Autonomous Car This Year," ed, 2015-03-10

[4] P. E. Ross. (2014-09-29) Mercedes Shows Off Self-Driving "Future Truck 2025". *IEEE Spectrum*.

[5] ISO, "International Standard 26262 Road vehicles -- Functional safety," ed, 2011.

[6] A. Van Lamsweerde, R. Darimont, and P. Massonet, "Goal-directed elaboration of requirements for a meeting scheduler: Problems and lessons learnt," in *Proc. IEEE Requirements Engineering*, 1995, pp. 194-203.

[7] K. Attwood, T. Kelly, and J. McDermid, "The use of satisfaction arguments for traceability in requirements reuse for system families: Position paper," in *Proc. Workshop on Requirements Reuse in System Family Engineering, Madrid Spain*, 2004, pp. 18-21.

[8] E. Hull, K. Jackson, and J. Dick, *Requirements engineering*: Springer Science & Business Media, 2010.