Defining Autonomous Functions Using Iterative Hazard Analysis and Requirements Refinement

Fredrik Warg[1], Martin Gassilewski[2], Jörgen Tryggvesson[3],
Viacheslav Izosimov[4], Anders Werneman[5], and Rolf Johansson[1]

[1]SP Technical Research Institute of Sweden, Borås, Sweden
{fredrik.warg, rolf.johansson@sp.se}
[2]Volvo Cars, Göteborg, Sweden
martin.gassilewski@volvocars.com
[3]Comentor AB, Göteborg, Sweden
jorgen.tryggvesson@comentor.se
[4]KTH Royal Institute of Technology, Stockholm, Sweden
izosimov@kth.se
[5]Qamcom AB, Göteborg, Sweden
anders.werneman@qamcom.se

Published in:

Proceedings of SAFECOMP 2016 Workshops, ASSURE, DECSoS, SASSUR, and TIPS, Trondheim, Norway, September 20, 2016, pp 286-297, Springer International Publishing.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-45480-1_23

Defining Autonomous Functions Using Iterative Hazard Analysis and Requirements Refinement

Fredrik Warg[1], Martin Gassilewski[2], Jörgen Tryggvesson[3], Viacheslav Izosimov[4], Anders Werneman[5], and Rolf Johansson[1]

[1]SP Technical Research Institute of Sweden, Borås, Sweden
{fredrik.warg, rolf.johansson@sp.se}
[2]Volvo Cars, Göteborg, Sweden
martin.gassilewski@volvocars.com
[3]Comentor AB, Göteborg, Sweden
jorgen.tryggvesson@comentor.se
[4]KTH Royal Institute of Technology, Stockholm, Sweden
izosimov@kth.se
[5]Qamcom AB, Göteborg, Sweden
anders.werneman@qamcom.se

Abstract. Autonomous vehicles are predicted to have a large impact on the field of transportation and bring substantial benefits, but they present new challenges when it comes to ensuring safety. Today the standard ISO 26262:2011 treats each defined function, or item, as a complete scope for functional safety; the driver is responsible for anything that falls outside the items. With autonomous driving, it becomes necessary to ensure safety at all times when the vehicle is operating by itself. Therefore, we argue that the hazard analysis should have the wider scope of making sure the vehicle's functions together fulfill its specifications for autonomous operation. The paper proposes a new iterative work process where the item definition is a product of hazard analysis and risk assessment rather than an input. Generic operational situation and hazard trees are used as a tool to widen the scope of the hazard analysis, and a method to classify hazardous events is used to find dimensioning cases among a potentially long list of candidates. The goal is to avoid dangerous failures for autonomous driving due to the specification of the nominal function being too narrow.

Keywords: ISO 26262 · functional safety · autonomous vehicles · hazard analysis · safety goals · item definition

1 Introduction

Fully autonomous cars are expected to bring substantial benefits both to society at large and the individual car users. Examples are fewer accidents due to elimination of human driving errors, better traffic flow management leading to increased road capacity and reduced pollution, and relieving the drivers from the task of driving.

However, automation also introduces new sources of error, including insufficient understanding of the environment and failure to take proper action in all situations that can arise. In the case of manual driving, and even with advanced driver assistance systems (ADAS), the human driver is always responsible for controlling the vehicle, which means, for instance, that omission errors for an ADAS function can be acceptable. In this paper we consider autonomous driving (AD) with a level of automation where the system has full responsibility for the driving task in at least some situations (level 3 or above in the NHTSA definition [11]). This level of automation has a profound effect on how to ensure functional safety, as the human driver (HD) no longer can be expected to take over in an emergency situation. It is not only very difficult to maintain the required level of vigilance as a passive overseer [10], but such a requirement would be contradictory to one of the basic promises of AD: to free up the drivers' time. As a consequence, it becomes necessary to make sure an AD vehicle can reach a safe state on its own in all relevant situations when AD is activated.

In the functional safety standard for road vehicles, ISO 26262:2011 [7], the scope and requirements of an electrical/electronic (E/E) function are parts of the *item definition,* which is an input to the functional safety process and the hazard analysis. This means only situations and hazards that affect the already defined function need to be taken into account. In this work we argue that, in the context of AD functions, the hazard analysis should have the wider scope of making sure the proposed function itself is adequately specified so that it, possibly in conjunction with other functions contributing to the AD functionality, covers all hazardous events (HEs) relevant to the goal of autonomous operation. The result of this extended analysis may, in addition to safety goals, be necessary changes to the scope and requirements of the proposed function. While there is ongoing work on how to manage violations of safety goals due to limitations in nominal functionality called '*Safety of the Intended Functionality*' (SotIF), we claim that the problem is rather a consequence of improper item definition and/or safety requirement refinement (see also discussion in [4]).

In this paper, we propose a new structured iterative work process and analysis techniques where the hazard analysis is used not only to ensure functional safety, but also as an aid when defining the scope of the function. Working iteratively is a natural choice since the scope of the function affects the hazard analysis and vice versa in this process. Input is an initial description of the goals and benefits of the proposed function. We call it the *preliminary feature description* in order to make it clear that it is not a final work product, but rather a starting point that will be refined and improved upon. The item definition, which describes the final scope and requirements of the function, is an output of the process, together with the item's safety goals. The main goal of the approach is to avoid dangerous failures due to the specification of the nominal function being too narrow. To that end, generic situation and hazard trees are used as a tool to widen the scope of the analysis beyond the limits of the initially proposed feature. The process also integrates rules from [3] which are used to classify hazardous events and can find omissions or overlaps in the set of HEs; i.e. finding the dimensioning HEs that will result in unique safety goals.

Section 2 of the paper discusses related work. Section 3 describes the proposed process in detail, with an integrated example. Finally, Section 4 concludes the paper.

2 Related Work

There are a number of techniques that address the problem of identifying candidate hazards. Methods such as Preliminary Hazard Analysis, HAZOP and FMEA are often recommended. Other methods include one described by Jesty et al. in [8]; this method is based on a state machine model of the transitions between a failure occurring in a system and a hazardous event. While useful, these techniques are not aimed at aiding the process of defining the scope of a function, which is our goal in this work.

One technique in our process is the use of generic situation and hazard trees to help the safety engineers find all relevant situations and hazards. There is some previous work on classifying generic situations. Jang et al. [1] decompose a situation into hierarchical categories of properties where the top level properties *vehicle*, *road*, and *environment* are followed by three levels of sub-factors. Situations are constructed by selecting properties from the last level, called state. The purpose is to make situation analysis more efficient, but how the classification is used is only discussed superficially. However, the way situations are constructed from our situation tree is similar. Another effort, even more related to our use of trees, is the situation classification described by Kemmann [5] in the context of the SAHARA framework for structured hazard analysis and risk assessment (note that this is not the same method as the safety/security-oriented SAHARA described by Maher et al. [12]). The SAHARA framework is ontology-based, and the situation analysis part, OASIS, organizes situation properties in trees which can be reused and extended with increasing experience. The situations created by combining the properties are meant for reuse between projects, together with their attached exposure class which therefore only have to be assessed once. While we have opted for a less formal, more light-weight way to construct and use situation trees in this work, the option to integrate the SAHARA approach in our iterative process would certainly be feasible, and could bring additional benefits for reuse and automation. The aim of OASIS, to create a knowledge base and make sure important situations are not missed in the situation analysis, is the same as ours. Neither the approach of Kemmann nor that of Jang has the purpose of being part of the function definition process however. The German organization VDA has created a standardized list of situations with the purpose of harmonizing the use of exposure factors [2]. However, this list is not structured into properties or useful for determining completeness of analysis like a situation tree.

Agile and lean development methods have gained in popularity during the last decade. Although nothing in ISO 26262 explicitly forbids agile methods it can be difficult to understand how to apply them, as the standard uses a conceptual V-model more similar to traditional sequential development processes. Stålhane et al. [6], propose *SafeScrum* to combine agile software development and safety according to IEC 61508. SafeScrum is based on the assumption that safety requirements are rather static, and refers to traditional processes for the safety lifecycle outside the agile software development. Another investigation into how to combine agile methods with safety standards has been done by Vuori [9]. While our approach to conduct hazard analysis and requirements elicitation in a semiformal and iterative process is partly inspired by agile ideas and concepts, its usefulness is not limited to agile organizations.

3 Iterative Hazard Analysis and Function Refinement Process

The upper part of **Fig. 1** illustrates the proposed work process. It starts with a preliminary feature description from which hazard analysis and risk assessment (HA&RA) and function definition are iteratively refined (solid arrows). These steps are repeated until they are mature enough to create both item definition and safety goals. This differs somewhat from ISO 26262 (lower part of the figure), where the item definition is an input to HA&RA, and the safety goals are the output. Even if not shown in the figure, it is implicit that hazard analysis might be revisited and updated later in the design process due to changes to function requirements or discovery of additional hazards. The following subsections describe each of the steps in more detail.

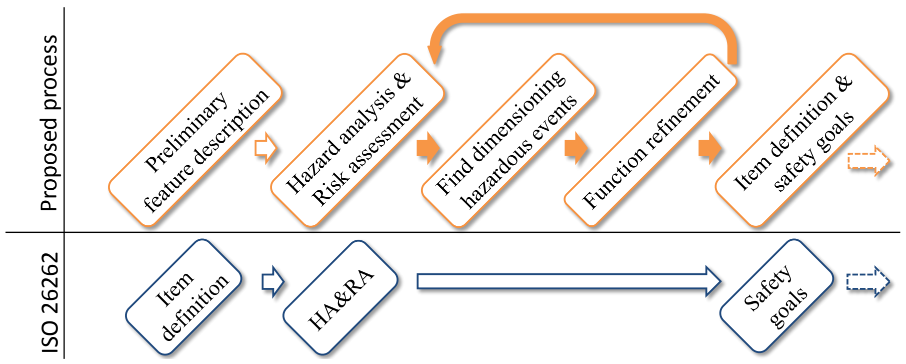
Proposed process
Preliminary feature description
Hazard analysis & Risk assessment
Find dimensioning hazardous events
Function refinement
Item definition & safety goals
ISO 26262
Item definition
HA&RA
Safety goals

Fig. 1. Proposed hazard analysis and function refinement process compared to ISO 26262.

3.1 Preliminary Feature Description

The aim of the preliminary feature description is to describe the expected (end customer) benefits and define the initial scope of the proposed feature. The format for such input (market research, input from previous projects, use cases, initial requirements etc.) can be freely selected to fit the organization. For our example we borrow the concept of user stories common in agile methodologies; a user story describes benefit from an end user perspective, but is usually not written by end users. It should be noted that these user stories will be at a higher level of abstraction than for their typical use in software projects. The format of a story is: *"As a <role>, I want <goal/desire> so that <benefit>"*. A full feature is described as a collection of user stories (sometimes called a theme). This simple format allows for easy modification and expansion of the theme as needed, and can be easily understood by all stakeholders, not only safety engineers.

Throughout Sec. 3, we will consider an example where an automated emergency braking (AEB) feature for an autonomous vehicle is analyzed. This envisioned feature has the purpose of avoiding accidents in the face of obstacles that appear suddenly and therefore are not part of the tactical plan of a likewise envisioned AD function handling normal autonomous operation. Keep in mind that the example is simplified to fit into this paper, and therefore covers a much smaller number of situations than a real such feature would likely do. **Table 1** shows the initial user stories, which describe some conditions when the emergency brake must work. The level of detail for the initial input is not critical since the subsequent steps have the purpose of bringing up questions regarding the scope of the function. However, it is supposed to guide the hazard analysis as to which situations and hazards are relevant for the feature. Very general stories are less likely to be useful in that respect, making the process more difficult than it needs to be. For instance, a single story saying simply *"I want emergency braking"* would be more difficult to work with than the stories in **Table 1**.

Table 1. AEB example: Initial user stories.

Automated Emergency Brake
As a driver, I want AEB for crossing animals so that my automated car doesn't run into animals.
As a driver, I want AEB for other motor vehicles so that my automated car doesn't hit vehicles making unexpected maneuvers.
As a driver, I want AEB for tricycles so that my automated car doesn't hit children on tricycles.

The preliminary feature description may also include other relevant information, for instance limitations imposed if the feature includes or expands upon an already existing component, or if part of the solution for some other external reason is given.

3.2 Hazard Analysis and Risk Assessment

The objective of the hazard analysis is to identify hazards that may occur in the E/E function and operational situations where the occurrence of these hazards may be dangerous; operational situations and hazards are combined to form hazardous events. In order to reduce the risk of missing relevant operational situations and hazards in the analysis we use tree structures representing a knowledge base of potential situations and hazards to investigate. By systematically going through the trees, one can avoid omissions resulting from the anchoring bias that is likely to occur based on the initial idea of the scope of the function. Furthermore, the trees both act as a tool to keep track of covered aspects when performing hazard analysis for a new function, and make sure gained experience is preserved for future projects.

Situation Analysis. An operational situation tree is shown in **Fig. 2**. The root of the tree is the operational situation. Each successive level breaks down the situation in different aspects of increasing detail, where the goal is that siblings on any given level are mutually exclusive and collectively exhaustive with respect to their parent. The tree in the figure does not fulfill the exhaustive property for space reasons; only selected aspects are shown to illustrate the principle. Even given unlimited space, however, a tree is unlikely to ever be perfect, instead it is meant to be continuously updated to reflect current best understanding of potentially relevant situations.

Operational situations for use in the hazard analysis are composed by selecting and combining leaves (properties) from the tree; the combination is what defines the unique operational situation. Leaves sharing the same parent are of the same kind (aspect) but describe mutually exclusive properties of a situation. If no leaf at all is selected from a particular aspect, the situation is considered to be valid for all properties of that aspect. For instance, the situation *"automatic drive on highway or rural road with animal obstacle"*, using the highlighted properties in **Fig. 2**, would implicitly include *"in all physical environments, road layouts, and speeds"*. In this way, we have a semi-formal description of the situation that can be used to classify situations and find gaps in the analysis, i.e. combinations that have not yet been considered.

It should be noted that any reference to "completeness of analysis" means with respect to the currently used situation tree. Since the tree constitutes the known universe of situations for the analysis, making sure the tree contains an adequate representation of reality is also imperative. This is discussed further at the end of this section.

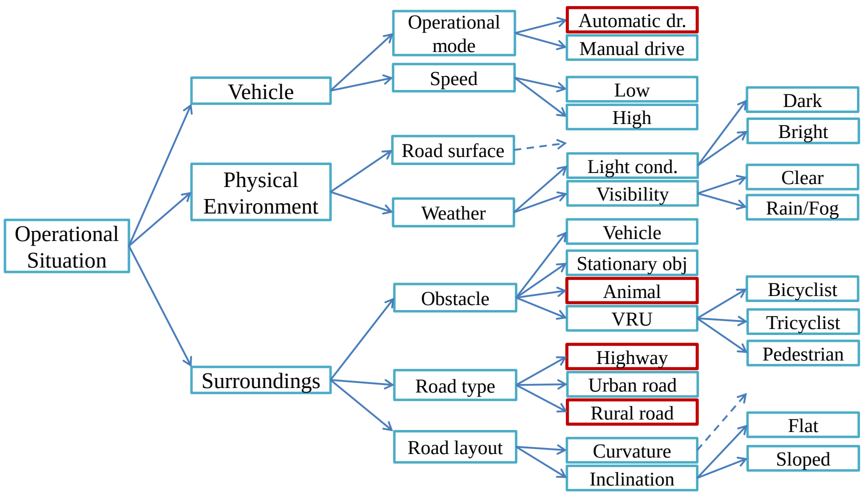

Fig. 2. Example of a generic situation tree.

The initial selection of operational situations is guided by the preliminary feature description. However, this selection does not have to be perfect. If relevant situations or hazards have been omitted, or selected despite being out of scope for the feature, this will be corrected as the iterative process progresses. For the AEB example, the specific obstacles mentioned in the user stories are obvious candidates: vehicle, animal and tricycle. The operational mode is also given: automatic drive. Beyond that, it is up to using one's best knowledge to add properties that may affect the way safety goals and function requirements will look like. **Table 2** includes some operational situations that may result from such a process. When selecting situations and hazards, adding a rationale to explain why certain aspects in the tree are not relevant for the function will strengthen the argument of completeness of the hazard analysis and could be used as evidence in a safety case.

Hazard Identification. Hazards are handled similarly to operational situations. **Fig. 3** shows an example of a generic hazard tree. Again, for space reasons, the number of hazards shown is limited. The basic structure is similar to the situation tree with the hazard at the root, and several levels of subcategories. As opposed to the situation tree, however, there is no combination of leaves. Each leaf by itself represents one possible hazard that can be included in the hazard analysis if relevant for the function being defined. For AD vehicles, where the E/E system moves beyond the operative level and is also responsible for tactical decisions, we have included a class of tactical level hazards to capture events where avoiding the tactical mistake may be the only way to prevent an incident. Potential driver/operator interference for AD functions is included as foreseeable misuse hazards. In **Table 2**, a hazard is combined with each operational situation to form hazardous events for our AEB.

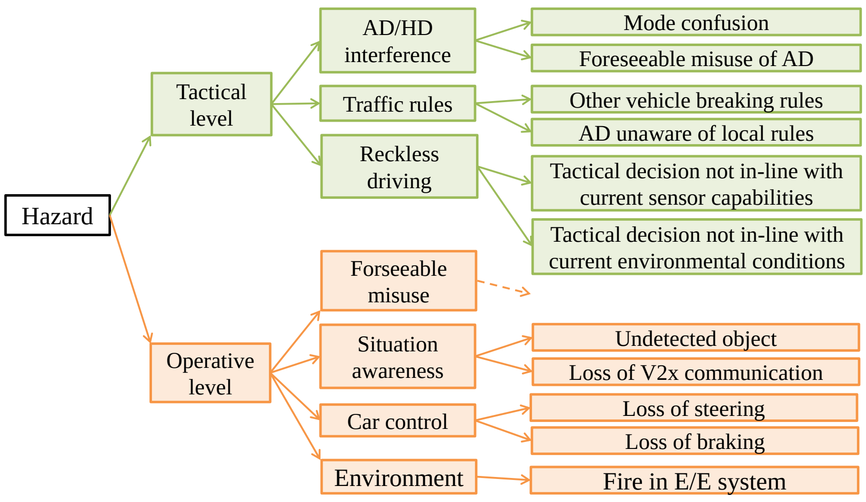

Fig. 3. Example of a generic hazard tree with operational and tactical hazards.

Risk Assessment. In ISO 26262, the result of risk assessment is assignment of an automotive safety integrity level (ASIL) to each hazardous event. ASILs range from A to D, where D is the most critical and hence requires the most elaborate measures in order to avoid failures. ASIL is assigned by first evaluating the exposure (E0 to E4), controllability (C0 to C3) and severity (S0 to S3) for occurrence of the hazard in the given situation. Based on the value of those factors, the standard prescribes a certain ASIL. All these values are shown in **Table 2** for the example HEs. For an autonomous function, we generally assume controllability, i.e. the driver's ability to mitigate the effect of a failure in the E/E system, to be C3 (difficult to control or uncontrollable) since the driver is out of the driving loop. The risk assessment has to be updated in every iteration as new or modified situations and hazards are added.

Table 2. AEB example: Hazardous events and risk assessment.

HE	Operational Situation	Hazard	E	C	S	ASIL
1	AD on highway/rural road with animal obst.	Undetected object	2	3	2	A
2	AD with tricycle obstacle	Undetected object	2	3	3	B
3	AD with other vehicle obstacle	Loss of braking	4	3	3	D
4	AD in bright light with other vehicle obst.	Loss of braking	4	3	3	D
5	AD in high speed with stationary object obst.	Mode confusion	2	3	2	A

Iteration and Extending the Trees. New hazardous events will be created in the same manner in each iteration. It should be stressed that the generic situation and hazard trees are only a starting point since generic trees cannot cover the needs for all items and all contexts. As the function description matures and more known detail about the context is taken into account, the generic trees should be extended with context-aware specializations and additions. Such context information can be, for instance, functionality or performance characteristics of the target vehicle or environment. The properties should be selected by finding limits where one of the risk assessment factors (E,C,S) in the resulting hazardous events is affected. Such changes may go into a context-aware tree created for a specific project. Furthermore, new general aspects can be added to the generic tree, thus contributing to the situation knowledge base to be reused in future projects. Such additions can be made at any level of the generic tree below the root.

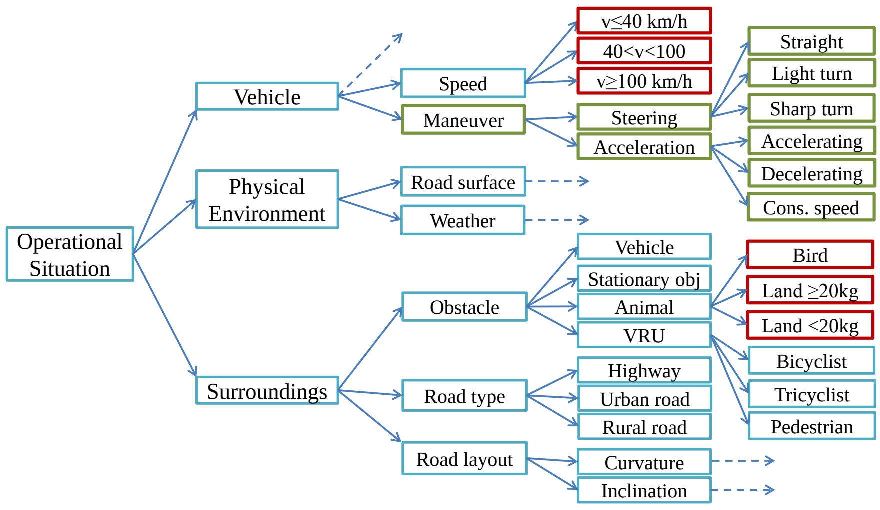

Fig. 4. Situation tree extended with new general and context-specific properties.

The *"speed"* property in **Fig. 4** shows how more generic properties (*"high"* and *"low"* in **Fig. 2**) are replaced with context-aware and more specific properties of the same kind. For instance, 40 km/h might be a speed limit where the severity factor of a collision is reduced. The *"animal"* property under obstacles is also extended with a new level containing more detailed context-specific information. These are changes that may go into a context-aware tree created for a specific project. Another addition shown in the figure is *"maneuver"* with its own sub-properties. This is a more general aspect, where extending the generic tree would be suitable.

3.3 Categorizing and Finding Dimensioning Hazardous Events

The previous step may easily give rise to a very large number of hazardous events (HEs). While the situation and hazard trees are helpful in classifying HEs, they will not directly show which ones are dimensioning, that is, which HEs are sufficient to cover all the safety goals. In order to understand the relationship between HEs, we use the method described in [3], which enables an arbitrarily long list of HEs to be reduced to only contain the dimensioning ones by applying a set of rules. There are eight rules of which four are to identify dominance, and four for non-dominance. In [3] it is shown that this set of rules is consistent and complete. This means that by comparing each pair of HEs in the list and applying these eight rules, the HEs which are sufficient to identify all safety goals of concern can be identified.

The rules depend upon the ability to categorize situations and hazards using set theory, i.e. determine if two situations are mutually exclusive, subsets, identical or overlapping. An advantage of using the trees is that the situations and hazards are already clearly defined making this trivial. For instance, one of the dominance rules state that from the three HE table columns situation, hazard, and integrity (ASIL): *Dominance exists if two columns show relation 'identical' and the third one has the relation '$\subset$' or '<'*. With this rule, one can show that HE4 in **Table 2** can be removed since it is dominated by HE3; the situation in HE4 is a subset of that in HE3 and the other properties are identical.

This method enables a fine-grained approach when categorizing hazards and situations, which mean the safety margins can be designed more precisely, avoiding unnecessarily conservative ASIL implications on the entire E/E architecture.

3.4 Function Refinement

In this step requirements elicitation for the intended nominal functionality is performed. The requirements should define a function that fulfills the user stories, and as is always the case when designing a new function, additional concerns such as cost and technical feasibility must be taken into account. The provisional list of HEs from the previous step will complement the user stories by providing information of what it takes to maintain safe operation. This knowledge will be used when defining the scope of the function in more detail. The hazardous events are considered as part of the requirements elicitation, and one of three actions is performed for each HE:

1. Decide that the HE is within the scope of the function and add requirements to reflect this. For instance, in our AEB example, HE5 considers what should happen when encountering stationary objects on the road, which was not covered in the user stories. In this case it may make sense to include stationary objects in the scope

of the function, since a braking function that reacts to i.e. moving vehicles but not stationary would probably not be useful.

2. Break down the problem further since the HE is too general or abstract to classify as within or outside scope. For instance, look at the HE on emergency braking for animals. It may be the case that it will be technically difficult to design a sensor system that detects birds and small fast animals like a rabbit. On the other hand, these are not likely to pose a threat to the passengers of the car, i.e. the severity factor is low. Therefore, the situation tree is expanded with a new subclass for different types of animals (see **Fig. 4**), and the HE updated to reflect what is within the scope of the function. **Table 3** shows that for the next iteration the original HE1 is marked as out of scope and a new HE1b that deals with *"land animal ≥ 20 kg"* is added to the list. It will also be necessary to update the classification as exposure should be lower but severity higher compared to a collision with any animal.
3. Restrict the scope of the function in order to find a feasible solution. Consider a case where the analysis shows that constructing a function that works in any surrounding poses too many technical or cost obstacles. Instead, it is decided to build an AD that can only be used on highways. In the AEB example, this would likely change the exposure for tricycles to E0, since highways are restricted areas where it is extremely unlikely a tricycle would ever be. **Table 3** shows how HE2 is updated to HE2b to reflect this; but in fact, HE2b shows that the integrity level drops to QM, so even if the user story about tricycles is kept, it will not result in any safety requirement that needs to comply with ISO 26262. The other HEs are also updated to reflect that the function is only available on highways.

Table 3. AEB example: Changes to hazardous events for second iteration.

HE	Operational Situation	Hazard	E	C	S	ASIL
~~1~~	~~AD on highway/rural road with animal obst.~~	~~Undetected object~~	~~2~~	~~3~~	~~2~~	~~A~~
1b	AD on highway with **land animal ≥20 kg** obst.	Undetected object	**1**	3	**3**	A
~~2~~	~~AD with tricycle obstacle~~	~~Undetected object~~	~~2~~	~~3~~	~~3~~	~~B~~
~~2b~~	~~AD **on highway** with tricycle obstacle~~	~~Undetected object~~	~~**0**~~	~~3~~	~~3~~	~~**QM**~~
3	AD **on highway** with other vehicle obstacle	Loss of braking	4	3	3	D
5	AD **on highway** in high speed with stationary object obstacle	Mode confusion	2	3	2	A

After the first iteration, the function is delimited by a number of HEs, but the exact scope may still be uncertain. For each subsequent iteration of refinement, it becomes clearer what the capabilities of the function should be. Note that if more than one function/item is involved in AD operation, it should be considered that the relevant HEs may fall within the scope of any of those functions, so the aim of the analysis is to make sure all HE are taken care of by these functions together.

The preliminary feature description is of less importance once we have an initial list of HEs and requirements to continue refining in subsequent iterations. However, the user stories could still serve a purpose, especially in an agile organization, if they are kept updated. While HEs and function requirements are more detailed, they may not be as suitable for e.g. prioritization and communication with a product owner. In the AEB case, one would add stories reflecting that we have restricted AEB to highways and added stationary obstacles after the first iteration.

HEs considered outside the scope are removed from the dimensioning list, but retained separately together with a rationale why it was not included. Where relevant, these scope restrictions will become part of the user manual for the vehicle, e.g. *"The AEB function of this vehicle can only be used on highways"*.

Sufficiency Checklist. After this step the iterative process continues with a new round of situation analysis and hazard identification given the new decisions made during function refinement. Iteration should continue until the list of HEs and function requirements are stable, complete, and useful. Below is a checklist to help determine when this is fulfilled, which also requires some engineering judgment:

- Sufficient situation and hazard coverage:
  - The generic trees should be fully covered (i.e. all properties are used in the analysis or have a rationale why they are not relevant for the function).
  - It must be clear whether the remaining HEs are within or outside scope.
  - Rules for dominance and non-dominance (see [3]) have been used to find potential gaps among the identified HEs.
- Clarity for continued design process:
  - Function requirements are useable for next steps of refinement.
  - HEs have a suitable level of abstraction to make safety goals which will not result in overly conservative ASIL assignment.

3.5 Item Definition, Safety Goals and Further Refinement

When the HEs and requirements are sufficiently elaborated, the final step of the process is to create safety goals from the dimensioning HEs, and an item definition which includes the functional requirements. This is input to the functional safety concept in ISO 26262, and from this point on the standard can be followed without modification. The further work is likely to be iterative as well. For instance, the item's refinement will need to be complemented with prototyping of the item's elements, in correlation with preparation of the system architecture work. This work will provide feedback to item definition, functional safety concept and technical safety concept. It is possible that both prototyping and system architecture work will effectively call for modification of the item definition. It can be a result of, for example, not fulfilling physical constraints, too much cost, or too much of development effort needed. Hence, the item will evolve during the whole development process. In addition, in a constantly changing environment for autonomous vehicles, such as new rules and regulations introduced and new methods discovered, the item will likely have to be kept open to modifications even after the product is released to the market. Whenever such modifications are called for, the hazard analysis needs to be revisited to capture any necessary changes to the safety goals.

4 Conclusions and Future Work

In this paper we argue that completeness of safety goals is a new challenge for autonomous vehicles. When no driver is available to fill in potential gaps, under-specified functions can pose a danger. We propose a method that can help reach completeness of safety goals for the entire AD functionality by using hazard analysis as an aid when defining the scope of the function. Since the scope of an advanced function will likely make it difficult to perform all the steps manually while still keeping everything consistent, a suitable future improvement would be tool support, including further work to make the process itself more suitable for automation.

Acknowledgements. The research has been supported by the Swedish government agency for innovation systems (VINNOVA) in the FUSE project (ref 2013-02650).

References

1. Jang, H.A., Hong, S.-H., Lee, M.K.: A Study on Situation Analysis for ASIL Determination. Journal of Industrial and Intelligent Information, Vol. 3, No. 2, pp. 152-157 (2015)
2. VDA: Situationskatalog E-parameter nach ISO 26262-3. VDA 702, Verband der Automobilindustrie e.V. (2015)
3. Johansson, R.: Efficient Identification of Safety Goals in the Automotive E/E Domain. In: Proceedings of 8th European Congress of Embedded Real-Time Software and Systems (ERTS^2) (2016)
4. Bergenhem, C., Johansson, R., Söderberg, A., Nilsson, J., Tryggvesson, J., Törngren, M., Ursing, S.: How to Reach Complete Safety Requirement Refinement for Autonomous Vehicles. In: Critical Automotive applications : Robustness & Safety workshop (CARS) (2015)
5. Kemmann, S.: SAHARA - A structured Approach for Hazard Analysis and Risk Assessments, Technische Universität Kaiserslautern (2015)
6. Stålhane, T., Myklebust, T., Hanssen, G.: The application of Safe Scrum to IEC 61508 certifiable software. In: Proceedings of ESREL 2012, Helsinki, Finland (2012)
7. ISO: International Standard 26262:2011 Road vehicles -- Functional safety (2011)
8. Jesty, P.H., Ward, D.D., Rivett, R.S.: Hazard Analysis for Programmable Automotive Systems. In: Proceedings of 2nd International Conference on system Safety, IET (2007)
9. Vuori, M.: Agile Development of Safety-Critical Software. Technical report 14, Tampere University of Technology, Department of Software Systems (2011)
10. Bainbridge, L.: Ironies of Automation. Automatica, vol. 19, no. 6, pp. 775-779, Pergamon Press (1983)
11. National Highway Traffic Safety Administration (NHTSA): Preliminary Statement of Policy Concerning Automated Vehicles, http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf
12. Maher, G., Sporer, H., Berlach, R., Armengaud, E., Kreiner, C.: SAHARA: A Security-Aware Hazard and Risk Analysis Method. In: Proceedings of 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE) (2015)