Automated Safety Case Compilation for Product-based Argumentation

Eric Armengaud

Automated Safety Case Compilation for Product-based Argumentation

Eric Armengaud
AVL List GmbH, Graz, Austria
Email: eric.armengaud@avl.com

***Abstract*—The main goal of a safety case is to provide a seamless argumentation why the product developed is acceptably safe for the purpose it is intended to. It usually consists of different argumentations such as product-based and process-based in order to describe the means for avoiding systematic failures during development and controlling random failures during operation. The main challenge during the compilation of a safety case is to regroup and harmonize all the information available from the different development activities in order to provide evidence for the safety attribute of the product. The contribution of this paper is to provide a solution for compilation of a safety case based on automated extraction of information coming from existing work-products.**

***Index Terms*—ISO26262, functional safety, safety case, tool integration, interoperability**

I. Introduction

The introduction of electronics has led to several paradigm changes, transforming passenger cars from purely mechanical systems to complex mechatronics systems and further to computers on roads. Hence, the mechanical systems (e.g., engine, transmission) are now fully under control of dedicated electronic control units (xCUs). Innovative functionalities have been further developed thanks to the better sensing of the environment and to the local integration of the xCUs in the car (e.g., smooth gear change thanks to close cooperation between engine and transmission). The innovation trend further follows the information society by integrating information surrounding the vehicle (e.g., GPS, car-2-X communication) for improving the overall vehicle efficiency.

Consequently, the automotive electronic architecture become very complex. Current cars are running with several hundred millions of lines of software code that have been developed conjointly by large teams from different institutions. At the same time, the higher degree of integration and the safety-criticality of the control application raise new challenges. Hence, the correctness of the applications both in the time domain and in the value domain has to be guaranteed. Due to the safety-criticality of the E/E control system, the xCUs are developed according to safety standards like the automotive functional safety standard ISO 26262 [1].

A major challenge in this context is the development of safety cases. The purpose of a safety case has been defined in [2] as *communicating a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context*. Different challenge arise: (1) *System complexity* of the control systems, requiring the support of different engineering disciplines and making the understanding of the entire system by a single technical chief engineer challenging; (2) *Safety-critical attribute* and the resulting strong safety requirements, leading to complex development and validation processes and methods (e.g., semi-formal or formal requirements, MC/DC coverage analysis); (3) *Distributed development* by different teams and different institutions, thus reducing the availability of knowledge and information across the teams and making the compilation and review of safety argumentations more difficult.

At the same time, a development shift is currently occurring in the automotive domain with the introduction of the ISO 26262. Hence, the risk based development (identification of hazards, definition of safety requirements, execution of safety analyses and safety validation) is being integrated in the company internal development environments and processes. This is however an evolutionary process and paradigm changes as proposed in the CESAR project [3] with fully integrated tool-chains covering the entire development process and providing full traceability of the product attributes are not industrially available yet. However, implicit traces (e.g., provided by the safety goals as outcome from hazard analysis and risk assessment and as basis for safety concept development, safety analysis and safety validation) are already available in current industry projects.

The purpose of this work is to present a method for automated compilation of product-related safety argumentation based on automated extraction of information coming from existing work products of industrial development processes. This approach relies on (a) state-of-practice trace management tools (e.g., requirement management) which are explicitly linking information together within a specific development activity, (b) interoperability and integration concepts (as developed in CESAR[1] with the interoperability specification [4] and being further enhanced in e.g., MBAT[2], CRYSTAL[3], VeTeSS[4]) in order to automatically link the partial traces together, and (c) the implicit traces coming from the risk oriented development

[1] www.cesarproject.eu
[2] www.mbat-artemis.eu
[3] www.artemis-ia.eu/project/index/view?project=46
[4] www.vetess.eu

process defined by the ISO 26262.

The paper is organized as follow: Section II describes the state of the art with respect to (modular) safety cases and safety argumentation. Section III presents the proposed approach. Section IV provides an evaluation of the method applied on a pilot project. Finally Section V concludes this work.

II. State of the art: Safety case and tool integration

According to ISO 26262 [1] a safety case is defined as an "*argument that the safety requirements for an item are complete and satisfied by evidence compiled from work products of the safety activities during development*". ISO 26262 Part 2, Annex C states the three following review criteria for the completeness of the safety case: confirmation that the referenced work-products are (a) traceable from each other, (b) consistent (no contradictions within or between eachother), and (c) complete (no open issues that can lead to the violation of a safety goal). This definition is in line with [2] where the purpose of a safety case is to "*communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context*".

In [2] the Goal Structuring Notation (GSN) is introduced. It aims at formalizing the argumentation based on the three following pillars (a) safety requirement, (b) safety argument and (c) safety evidence – the safety argument making the link between objective and evidence. Important care is set to a good balancing between argument and evidence: an argument without evidence is unfounded, and an evidence without argument is unexplained. An extension of the GSN language is proposed in [5]. During this work the author draws the parallelism between software architecture and safety case architecture, and provide an extension of the GSN approach that is modular and compositional, and thus better suits the modular structure of the software architecture.

A first important aspect is the importance of providing both **process-based arguments** (to minimize probability of systematic design faults) and **product-based arguments** (to control random faults and provide evidence that the system behaves acceptably safe), as discussed in [6]–[8]. In [6] industrial experiences for building safety cases in the context of ISO 26262 are presented. Focus is set on the necessity to have both process-based arguments (to minimize probability of systematic design fault) and product-based arguments (to minimize random fault and provide evidence that the system behaves acceptably safe). An important outcome is the challenge for the companies to align their development process for explicit full compliance to the standard. At the same time the existence of work-products in the company internal development process covering different major aspects required by the standard is acknowledged.

The work discussed in [7] describes the concept of "generic safety cases" in the context of railway domain according to the safety standard for railway applications EN 50129 [9]. During this approach, the argumentation is split into "generic product safety case" (independent of application), "generic application safety case" (for a class of applications), and "specific application safety case" (for a specific application). These three levels are mapped to different part of the development having a potential of reuse in other projects (e.g., communication architecture). All the three levels present the same structure (system definition, quality management, safety management, technical safety, related safety cases and summary) with product-related argumentation and process related argumentation.

A second important aspect is the **parallelism between the safety argumentation and the safety requirements** [10], [11]. The work presented in [10] describes an approach for generating safety concept trees out of component fault trees. The approach is based on a risk-oriented development and the development of fault trees for each component. These fault trees are then logically inverted to generate safety requirements trees (identifying the counter measures in a systematic way) for the system specification.

Similar to that, the work presented in [11] provide classes of requirements to describe and formalize the technical refinement required during the definition of the safety concept. This proposed formalization is based on vertical safety interface following the layered scheme of control systems – starting with application specific argumentations that are refined and integrated with platform specific argumentations (covering the aspects of basis services, operating systems, communication). This language enables pre-certification of components and the later certification of the entire systems, such improving re-use capability and partitioning the efforts for certification.

Based on these two observations, the proposed approach in this work will be to use the information already available within the development process in order to perform automated safety case compilation for the product-based part of the argumentation. This compilation (and thus access to the information within the development process) shall be strongly supported by the work on tool integration and interoperability performed during the CESAR project, see [3] for a detailed technical overview on the project and [4] for the aspects on interoperability.

The works described in [12], [13] describe the different aspects of tool integration on the example of an automotive tool chain. Important aspects from the integration point of view are the *integration platform* (providing basic services such as workflow engine, repository, transformation engine, standard APIs) and the *semantic understanding* (providing a common semantic between different methods and tools, which is required to exchange information and describe the relationships between the elements). From the end-user point of view, the *development process* (performing the links between the related standard(s), the development methods and the relying tools) is the main driver. During this work, main focus will be set on the semantic aspect – how to transform and relate the existing information (safety analysis, safety concept, safety validation)

together.

Regarding the process-based argumentation, a loosely integrated approach is proposed in [14]. Databases are provided to list the safety activities required by the ISO 26262 as well as the requirements allocated to these safety activities. By filling the databases with evidence of completion of these activities, the process-based argumentation can be set-up. Note that this has a direct impact on the product-related argumentation: Hence, the outcomes of these activities (e.g., safety analysis, safety validation) can be then integrated into the automated safety case compilation.

III. Automated compilation of product-related safety argumentation

A. Overview of the approach

The proposed approach for automated compilation of product related safety argumentation is to automatically extract summary information from the different work-products and map these information together. The expected benefits are (a) to provide a technical overview of the product being developed, and (b) to ease consistency review between the work-products. Figure 1 illustrates the approach. The data generated by the different activities are summarized, grouped and linked together in order to illustrate how the safety objectives, arguments and evidence are mapped together. The information is grouped around the safety goals. Hence, they represent the safety objectives and are thus the central point of interest.

During the concept phase, the hazard analysis and risk assessment (based on the item definition) is generated. This document provides the safety objectives (list of safety goals) mapped to their context (list of system functions, malfunctions and hazards). The next phase is the development of the safety concept. During this phase, the safety argumentation is build based on the successive refinements from safety goals to functional and technical safety requirements down to finally software and hardware safety requirements. Note that this is not a monolithic activity – The safety concept is refined iteratively with the system specification and analyzed using different safety analyses. The safety analyses serves the verification of the safety concept, e.g., to ensure completeness of counter measures with respect to possible component failures. Finally, the test concept provides an evidence for the correctness of the safety requirements fulfillment (safety arguments).

Note that the purpose of this safety case is not to replace existing documents. On the contrary, it aims at providing a technical summary of the activities and mapping these activities together. Therefore, the full information is still available in the original documents.

The proposed view supports project reviews with respect to the following criterias:

1) The safety objectives (safety goals) have been completely identified based on the system functions and related hazards
2) The safety argumentation (hierarchy of safety requirements) is correct and consistent to the safety objectives
3) The safety argumentation (hierarchy of safety requirements) is complete with respect to possible component failures
4) The safety evidence (test concept) validates the safety argumentation by ensuring the correct operation of the proposed counter measures

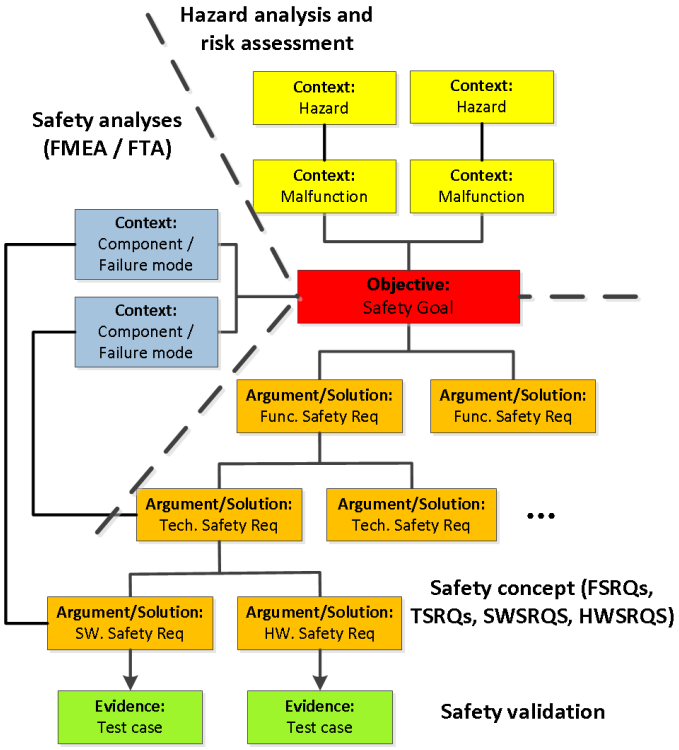

Fig. 1. Proposed data structure

Note additionally that this approach is very useful during development (and during milestone review) to illustrate attributes of the system. Hence, this approach can be used to monitor development status and perform some check such as *completeness of description* (are the elements complete – e.g., all requirement fields filled) or *completeness of traces* (are all the traces available – e.g., all requirements have a trace to at least a test case).

B. Hazard analysis and risk assessment – extraction of safety goals

The main purpose of the hazard analysis and risk assessment is to analyze the functionalities of the system in its intended context and to identify and classify the possible hazards. Main outcome here are the safety goals – which represent the safety objectives during the project. During the automated extraction,

each safety goal is mapped with its related hazards. The malfunction leading to the hazard is listed as well in order to provide a clearer understanding of the context. Note that the reasoning for the hazard classification is not integrated in the safety concept. Hence, the hazard analysis and risk assessment document remains available for more detailed reviews if required.

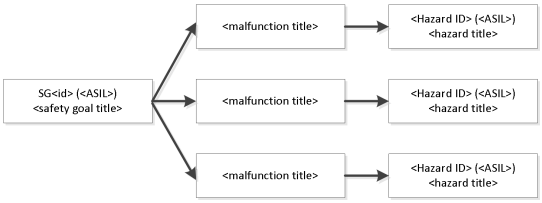

Fig. 2. Information extracted from the hazard analysis and risk assessment

Figure 2 illustrates the information automatically extracted from the hazard analysis and risk assessment document. It consists of the safety goals (unique identifier, ASIL, title), the related malfunctions and the hazards (unique identifier, ASIL, title). This view supports the documentation of the context leading to the safety goals. Note also that different tests can be performed automatically as well: Are the trees complete (all the malfunction have an hazard), are the elements complete (identifier, ASIL, title), are the ASIL level consistent (safety goal shall have the ASIL of the highest hazard).

C. Safety concept – extraction of safety requirements

The main purpose while developing the safety concept is to systematically refine the safety argumentation by providing functional requirements, then technical requirements, and finally software (resp. hardware) requirements to mitigate the risks. Content of the automated extraction are the hierarchy of requirements with their identifier and their traces.

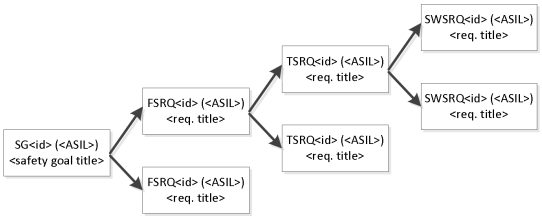

Fig. 3. Information extracted from the safety concept

Figure 3 illustrates the information extracted from the functional safety concept, technical safety concept and SW (resp. HW) safety requirements. The elements consists of unique identifiers, ASIL, requirement title. Note that further information such as e.g., full text, rationale, or acceptance criteria are not extracted here. The purpose of the safety case is to provide an overview of the product, detailed information needs to be retrieved in the respective documents. Such view enables different tests such as: completeness of the traces between the requirements, completeness of the requirements (are the fields filled), consistency of the ASIL within the safety argumentation.

D. Safety analysis – extraction of component malfunctions and required counter measures

The objective of safety analyses is to identify the conditions and causes that could lead to the violation of a safety goal, and subsequently to identify and confirm the safety requirements. The challenge is to integrate the safety analysis in context of the safety concept (both safety goals and safety requirements). Content of the automated extraction is the component failures mapped to the safety goals (effects on the system) and to the safety requirements to mitigate the effects.

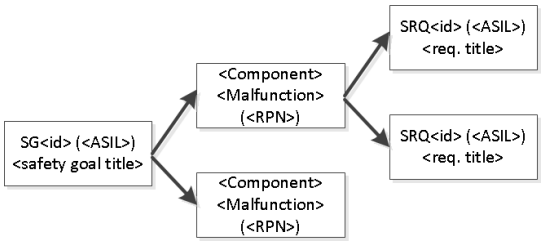

Fig. 4. Information extracted from the safety analysis

Figure 4 illustrates the information extracted from the safety analysis. The main target of this view is to list all the possible component malfunctions having an eventual impact for a given safety goal, and to further show the trace to the related safety requirement that mitigate the risk. Similar to the previous views, only a summary of the information is provided (in this case: safety goal identifier and text, component name and its malfunction, as well as safety requirement identifier and title). The full information is available in the main documents (e.g., FMEA report, safety concept), and this view illustrates the logical links between the different information and documents. This view supports the (verification) review of the safety concept against the possible component malfunctions. More especially, following topics can be reviewed:

- correctness: is the safety requirement appropriate for the malfunction
- completeness: are all the malfunctions linked to at least one safety requirement
- consistency: are the safety requirements consistent to each other for the possible component malfunctions identified

E. Test concept – extraction of evidence

The final step in the argumentation is the documentation of the evidences that the required counter measures have been implemented and are working properly. This step is usually performed by means of test execution. The main target in this view is therefore to illustrate the link between the safety requirements and the related test cases.

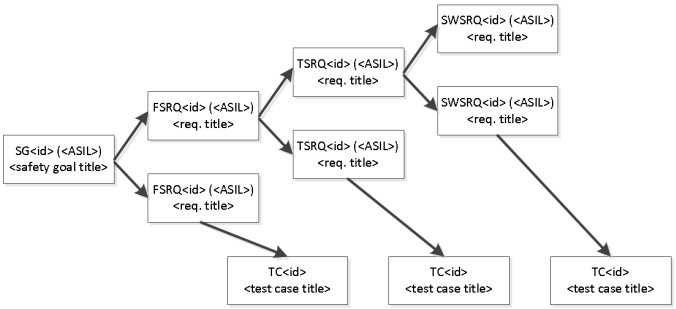

Fig. 5. Information extracted from the test concept

Figure 5 illustrates the information extracted from the test concept. It is mainly very similar to the safety concept and has in addition the test cases (identifier and title) mapped to the requirements. Note that this view is usually generated during different generations of tests. Hence, SW (respectively HW) safety requirements are usually validated in the context of component tests (e.g., MiL, SiL or PiL – Model, Software or Processor in the Loop). Technical safety requirements are usually validated in the context of entire control systems (e.g., HiL – HW in the Loop), while functional safety requirements and safety goals are usually validated in the context of the entire system or the vehicle (test beds, vehicle platform).

Similar to the safety concept, the full information is available in the main work products (e.g., full requirement text, full test case). This view illustrates the logical links between the different elements and support the (verification) review of the test campaign. More especially, following topics can be reviewed:

- correctness: are the test cases appropriate for the safety requirements
- completeness: are all the safety requirements linked to at least one test case
- completeness (respectively): are all the test cases mapped to a safety requirement (or is there any new function to be tested that is not yet part of the system specification)
- consistency: are the different test campaigns consistent to each other

IV. Integration in company processes and application in industrial context

A. Overview of the approach

One main challenge for application of new methods in industrial context is the alignment with company internal processes, methods and tools. Therefore, an important requirement for this approach was the high flexibility to cope with the different environments and templates. The developed environment consists of three main components:

- Wrapper
- Database
- Report generator

Wrapper: The purpose of the wrapper is to provide interfaces to the different tools and environments, in order to get access to the data in an automated way. During this work, wrappers to MS Excel documents, MS Access databases and to requirement management tools such as PTC Integrity[5] were developed. Further interfaces, e.g., to Enterprise Architect[6] for SysML modeling, are currently being developed.

Database: Once the data has been imported, it has to be structured and taken into relation with the remaining data imported from the other sources. This activity strongly relates to *semantic understanding* and *model transformation* – the definition of a meta-model for the proper structuring of the information and the capability to link the different inputs with this defined meta-model (transform the format of the input data into the scheme defined by the meta-model). In this work, the meta-model presented in Section III has been used. Note that most of the tools present more complex data structure than the one described in this work - the additional information were not uploaded since not relevant in the context of the product-related safety case argumentation (keep in mind that the safety case presents an overview of the results and does not replace the work-products realized previously).

Report generator: Once the full information has been imported, harmonized and structured, then the post processing can take place. It consists of automated graph generation using the freeware DOT[7] and report generation using LaTeX. For each safety goal, four dedicated graphs are generated:

- Hazard and risk analysis: List of system malfunctions and related hazards identified for each safety goal
- Safety concept: Safety requirements refined from each safety goal
- Safety analysis: Related component failures having a potential impact on each safety goal
- Test concept: List of test cases mapped to the safety requirements related to each safety goal

Furthermore, automated checks can be executed on this database in order to check the correctness and completeness of the data – see [15] for more information. Finally, a PDF report is generated out of this information.

B. Hazard analysis and risk assessment – extraction of safety goals

The process of hazard analysis and risk assessment (HARA) in the context of ISO 26262 consists at identifying the different system functionalities, deducing the possible malfunctions and map these malfunctions to driving situations and vehicle environments in order to identify the possible hazards. In a second step, the hazards are then classified according to their severity, exposure and controllability, and finally grouped to safety goals. The safety goals represent therefore the main

[5]http://www.mks.com/platform/
[6]http://www.sparxsystems.com.au/
[7]http://www.graphviz.org/

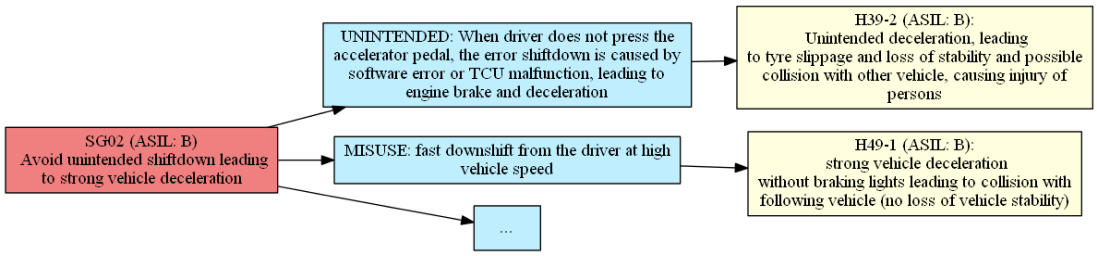

Fig. 6. Example: Mapping between hazards and safety goals

targets for the following safety development. It is therefore important to properly document each safety goal and its context - in this case the malfunctions and hazards related to the safety goal.

This step has been performed using an MS Excel wrapper. The HARA elements have been automatically extracted from the project document based on the company templates. Note that the example illustrated in Figure 6 has been modified and reduced due to confidentiality issues.

For this pilot project, the HARA document consists of five safety goals and 64 hazards analyzed. This relatively small project already illustrates the needs of such view: the information can be visualized (and therefore reviewed) from the perspective (viewpoint) of the safety goals, and not only from the perspective of the functions or hazards. This enables a review that can be performed more efficiently: the review can be performed more topic related and can be dispatched into different teams more easily.

C. Safety concept – extraction of safety requirements

The next step is the identification of the functional and technical safety concepts, and the further refinement to SW safety requirements (respectively HW safety requirements). This work focuses on the systematic refinement from the safety goals to single technical solutions to be implemented as combination of SW functions and electronic HW components (and eventually other technologies). This is a challenging task since a high level of accuracy is required, thus leading to a large number of requirements. At the same time, the development target (safety goal) shall be kept in mind, and a certain degree of redundancy between the safety goals exist (e.g., methods to ensure the integrity of the computing platform).

This activity strongly relies on industrial requirement management tools. Tools such as PTC Integrity, IBM Rational DOORS or Dassault Reqtify provide a solid framework for requirement elicitation (which information a requirement shall contain) and for requirement management (how the requirements are logically organized – traces between the requirements). During this work, we have implemented a wrapper to PTC Integrity in order to automatically extract specific requirement fields as well as the links between the requirements, see Figure 7. The identifiers at the begin of the fields are unique requirement identifiers within Integrity. With this ID, the full requirement text can be retrieved during review.

For this pilot project, 2915 requirements were parsed. From this database, 207 requirements were identified as being direct or indirect children from the safety goals identified previously. The identification of their related document field indicates their affiliation as functional, technical or SW safety requirement. This view played a central role during the review of the safety concept. Hence, it enables to split the large number of requirements per target (for each safety goal) and to review the soundness of the safety concept for each safety goal. Furthermore, this view combined with the HARA view enables a cross check of the safety concept in the context of the hazards identified previously. Note finally that this requirement export was also the basis for different checks indicating the quality of the requirements (e.g., availability and completeness of traces, checks if important fields are filled).

D. Safety analysis – extraction of component malfunctions and required counter measures

The execution of safety analyses is usually a parallel step to the safety concept discussed in the previous section. The aim of this step is to consolidate the safety concept by the systematic (inductive and deductive) analysis of the possible component faults and possible impacts on the safety goals. An important outcome of this work is the mapping between detailed faults at component level, safety goals at vehicle level, and safety requirements to properly identify and react to the faulty situations.

During this pilot project, three approaches have been compared: an MS Excel template, a dedicated database and industrial tools such as API IQ-FMEA and FaultTree++. From the safety analyses point of view, all approaches have their trade-off with respect to flexibility (tailoring of the safety analysis method to project needs), efforts to tailor and execute the method, and efforts to ensure consistency over the large amount of information. With regards to integrating safety analyses in this automated safety case compilation, all three methods present a structured way to access to the information (the semantics is known and similar between the approaches). The syntax, however, differs (XML formats, MS Excel inter-

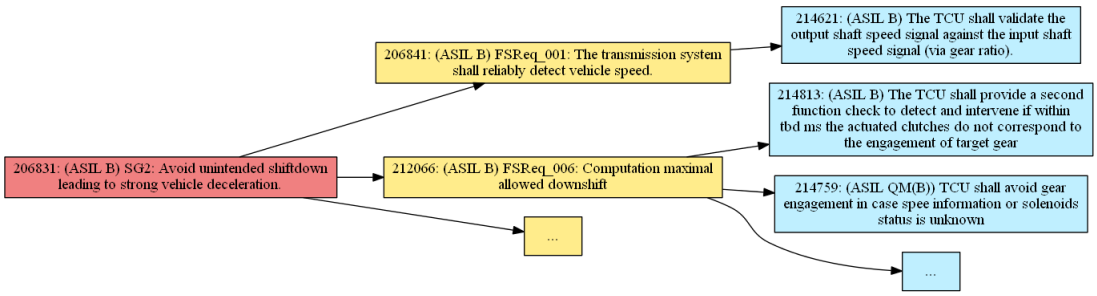

Fig. 7. Example: Mapping between safety goals and safety requirements

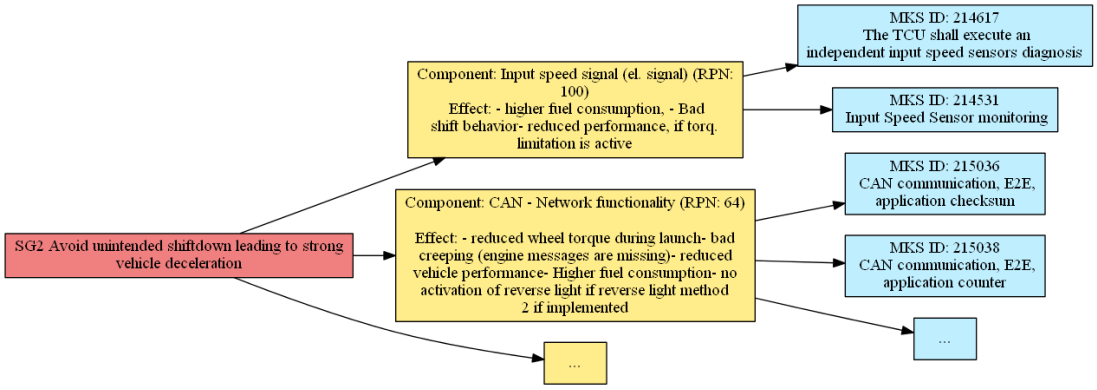

Fig. 8. Example: Mapping between safety goals and component malfunctions (FMEA)

face). Figure 8 provides a modified example for an interface FMEA.

Regarding the FMEA at control system level, a total of 56 interfaces were analyzed; 26 interfaces have been identified as having potential impact to one or more safety goal(s), and 18 safety requirements were mapped to these interfaces in order to identify and react to possible component failures. This view was important to review the appropriateness between safety requirements and system architecture (and possible component failures). The safety goal centric approach is useful to focus on one topic at one time and to enable to split the review in different teams.

E. Test concept – extraction of evidence

The compilation and execution of test cases is a central aspect to provide evidence of correct implementation of the safety measures. Typical challenges for test management are to check completeness of the test campaign with respect to the requirements (for requirement-based tests) and to illustrate the appropriateness of the test campaign for the safety context (hazards and component malfunctions previously identified). An additional challenge is the large number of test platforms (e.g., MIL, SIL, PIL, HIL, testbed, vehicle) and the large variability to manage the test documentation.

Similarly to the safety analyses, a large number of test environments needed to be supported in this industrial context. The structure of the information was similar for the different environments, the way to access the data differed. All test cases have a unique ID, title, description, mapping to requirement ID, status. This information is automatically extracted and mapped to the respective safety requirement, see Figure 9. Note that this figure has been modified for confidentiality issues. Furthermore, this figure shows only the test performed at SW level (and not at control system level or at vehicle level) – only the SW requirements are mapped. This illustrates that the validation at SW level has been accomplished, however the validation of the safety requirements at control system level and at vehicle level are still open.

In this pilot project, a total of 3772 test cases are available. A subset of 167 test cases are mapped to safety requirements. This view documents and supports the review of the completeness of the test campaign for the provided safety context. Note furthermore that the capability to import data from different sources and the needs to combine and harmonize the different inputs strongly support improvement of the quality. Hence, automated checks can be performed, e.g., to check the completeness of traces or check the correct fulfillment of different fields.

V. Conclusion

The compilation and maintenance of a safety case during the project duration is a tedious task. Hence, the safety case shall provide an argumentation that the product is acceptably safe for its indented purpose. This argumentation is based

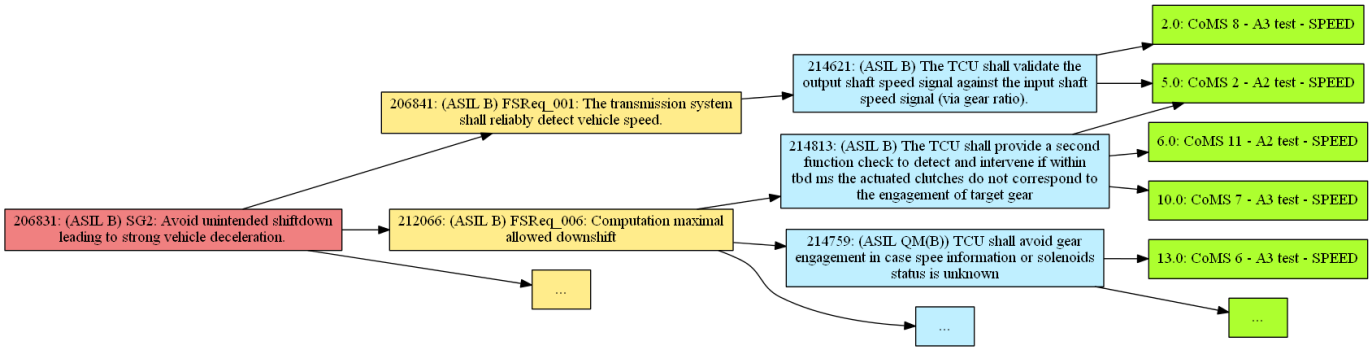

Fig. 9. Example: Mapping between safety requirements and test campaign

on and summarizes the different work-products generated during project execution – the consistency between the work-products has to be established and maintained, each change in a work-product has a potential impact on the safety case. A method for automated compilation of the product-based argumentation based on the existing work-product has been presented in this work. This approach strongly reduces the efforts for compilation of a safety case. Moreover, the generated document supports review of the safety argumentation and enables automated checks of different criterias (e.g., with respect to completeness and consistency of the argumentation). This is also useful during the project (e.g., at milestones) for technical management of the safety aspects. To summarize, quality improvement in project results from

- the capability to perform context aware reviews supported by dedicated views
- the possibility to run automated checks based on a criteria catalog (quality metrics) and to automatically assess the documentation

ACKNOWLEDGMENT

The author wants to thank R. Mader and G. Macher for fruitful reviews on early versions of this paper.

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement number 295311 and from specific national programs and/or funding authorities.

REFERENCES

[1] International Organization for Standardization, "ISO 26262 Road vehicles - Functional safety," 2011.

[2] T. Kelly and R. Weaver, "The goal structuring notation a safety argument notation," in *Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases*, 2004.

[3] T. Wahl and A. Rajan, *CESAR - Cost-efficient Methods and Processes for Safety-relevant Embedded Systems*. Springer, 2013.

[4] CESAR Consortium, P. Vasaiely et al., "Interoperability Specification V1.0, D_SP1_R1.6_M4," 2012. [Online]. Available: www.cesarproject.eu

[5] T. Kelly, "Using software architecture techniques to support the modular certification of safety-critical systems," in *Proceedings of the eleventh Australian workshop on Safety critical systems and software - Volume 69*, ser. SCS '06. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2006, pp. 53–65.

[6] R. Dardar, B. Gallina, A. Johnsen, K. Lundqvist, and M. Nyberg, "Industrial experiences of building a safety case in compliance with iso 26262." in *ISSRE Workshops*. IEEE, 2012, pp. 349–354.

[7] E. Althammer, E. Schoitsch, H. Eriksson, and J. Vinter, "The decos concept of generic safety cases - a step towards modular certification," in *Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications*, ser. SEAA '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 537–545.

[8] J. Birch, R. Rivett, I. Habli, B. Bradshaw, J. Botham, D. Higham, P. Jesty, H. Monkhouse, and R. Palin, "Safety cases and their role in iso 26262 functional safety assessment," in *SAFECOMP*, 2013, pp. 154–165.

[9] CENELEC EN 50129, "Railway Applications - Safety Related Electronic Systems for Signaling," 2003.

[10] D. Domis, M. Forster, S. Kemmann, and M. Trapp, "Safety concept trees," in *2009 Proc. Ann. Reliability & Maintainability Symp.* Piscataway, NJ: IEEE, 2009, pp. 212–217.

[11] B. Zimmer, S. Bürklen, M. Knoop, J. Höfflinger, and M. Trapp, "Vertical safety interfaces: improving the efficiency of modular certification," in *Proceedings of the 30th international conference on Computer safety, reliability, and security*, ser. SAFECOMP'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 29–42.

[12] E. Armengaud, M. Zoier, A. Baumgart, M. Biehl, D. Chen, G. Griessnig, C. Hein, T. Ritter, and R. T. Kolagari, "Model-based Toolchain for the Efficient Development of Safety-Relevant Automotive Embedded Systems," in *SAE 2011 World Congress & Exhibition*, Apr. 2011. [Online]. Available: http://papers.sae.org/2011-01-0056

[13] E. Armengaud, M. Biehl, Q. Bourrouilh, M. Breunig, S. Farfeleder, C. Hein, M. Oertel, A. Wallner, and M. Zoier, "Integrated tool chain for improving traceability during the development of automotive systems," in *ERTS2 2012 — Embedded Real Time Software and Systems*, Feb. 2012.

[14] E. Armengaud, Q. Bourrouilh, G. Griessnig, and P. Reichenpfader, "Using the CESAR Safety Framework for Functional Safety Management in the context of ISO 26262," in *ERTS2 2012 — Embedded Real Time Software and Systems*, Feb. 2012.

[15] R. Mader, E. Armengaud, G. Griessnig, C. Kreiner, C. Steger, and R. Weiss, "Oasis: An automotive analysis and safety engineering instrument." *Reliability Engineering & System Safety*, vol. 120, pp. 150–162, 2013.