

The DeSCAS Methodology and Lessons Learned on Applying Formal Reasoning to Safety Domain Knowledge

Jan Gačnik¹, Henning Jost², Frank Köster¹, and Martin Fränzle²

¹ German Aerospace Center, Lilienthalplatz 7,
38108 Braunschweig, Germany

{jan.gacnik, frank.koester}@dlr.de

² University of Oldenburg, Ammerländer Heerstr. 114-118,
26129 Oldenburg, Germany

{henning.jost, martin.fraenzle}@informatik.uni-oldenburg.de

Abstract. Functional safety has become an important aspect for engineering activities in the automotive domain due to the upcoming introduction of the safety standard ISO 26262. This paper proposes a methodology to guide the safety related requirements engineering process by means of OWL (Web Ontology Language) ontologies. These ontologies formalize necessary domain knowledge and serve as reference models to support semi-automated requirements discovery and to ease the certification process. Using OWL's logical base, knowledge inference is applied to reason about safety measures for ensuring compliance with the reference process (guidance). The proposed methodology has been implemented in a prototype toolchain and applied to a simple lane departure warning system as an example assistance and automation system. Lessons learned refer to conceptual (expressiveness) and technical (tooling efficiency) issues.

Keywords: Certification, ISO 26262, Domain Knowledge, Ontology, Process Framework, Assistance and Automation System, Semantic Reasoning

1 Introduction

Safety critical systems like assistance and automation systems (AAS) in the automotive domain demand a clearly defined proceeding during development, especially to support certification and qualification processes. In order to reduce the risk of a hazardous system failure, standards have been defined which propose a certain proceeding, requirements and associated methods and measures during development. One of these standards is the upcoming ISO 26262 for functional safety in the automotive domain [6]. Due to the informal representation of such standards in natural language text, there is an inherent

risk of misinterpreting the standard's content or following an incorrect or incomplete sequence of development activities. This threat is further increased by interdisciplinary issues in developing safety critical systems: in order to avoid misunderstandings between different disciplines involved in the development, a universally comprehensible representation of system and process requirements – on a common and generic basis – is essential. In the context of ISO 26262, process requirements in particular are important, as most of the requirements from the standard are process related. A formally sound specification of process requirements based on a well-defined terminology is crucial for supporting a precise presentation of requirements. Furthermore, through the formal base, process requirements and associated activities become computer-readable, and some analysis steps (e.g. checking consistency) can be automated. Especially in the context of certification, traceability between process and system artifacts is of high importance.

1.1 Design of Safety Critical Automotive Systems

The *Virtual Institute DeSCAS* (Design of Safety Critical Automotive Systems), which is funded by the Helmholtz Association, strives for defining a process framework and related methods to support interdisciplinary development of safety critical assistance and automation systems in the automotive domain. The process framework builds on generic as well as domain dependent concepts to *interweave* different interdisciplinary development activities. This includes the formalization of relevant standards (e.g. ISO 26262) in order to allow the automation of certain analysis methods (e.g. hazard analysis and risk assessment) and the derivation of requirements, which may vary due to variations in safety level classification.

1.2 Structure of the Paper

This paper will outline how the use of a standard can be improved by formalizing its structure. This comprises OWL (Web Ontology Language) based formalization of the risk analysis from ISO 26262, which is applied to an example application (lane departure warning system). Since safety requirements and associated methods as well as process phases have been modeled in OWL, the result of the risk analysis is furthermore used to infer concrete requirements for a system under development. These requirements and their dependencies are utilized to derive a safety related workflow. The different steps of analysis and formal reasoning have been integrated in a prototype toolchain, which especially operates on safety requirements from ISO 26262. This paper gives a summary of the DeSCAS process framework concerning implementation aspects and insights gained from the prototype implementation (proof of concept). This especially concerns the performance of formal reasoning on OWL ontologies.

2 Proposed Methodology

Taking into account the current design and properties of state-of-the-art development in the automotive domain (such as ISO 26262 and RESPONSE 3 *Code of Practice* [8]), DeSCAS has developed a methodology for the development of assistance and automation systems. In addition to interweaving interdisciplinary development activities of the automotive domain by reasoning about domain knowledge, a formal base is essential for the DeSCAS process framework. Most of the state-of-the-art standards lack a formal representation as they primarily consist of glossary-based, natural language text descriptions, informal checklists or questionnaires, complemented by some graphics and tables. As a consequence, inconsistencies concerning both the use of technical terms and the dependencies between process elements become visible when analyzing these standards in terms of a formal definition. The nomenclature is partially unclear or even ambiguous, as are some connections between process elements and requirements [3]. For that reason, the process model as part of the entire DeSCAS process framework builds upon a generic and formally defined process meta-model.

2.1 Interweaving Development Streams

The essential parts of the DeSCAS meta-model are constituted by the *development streams* which combine related design activities of an interdisciplinary system development to be synchronized via iterations within a V-Model. Concerning the development of AAS in the automotive domain, the DeSCAS process model defines the following three main development streams:

1. *Human factors*: Continuously involving human behavior and interaction with the system during system development (see RESPONSE 3 *Code of Practice*).
2. *Functional development and architecture*: The classical hardware and software development of AAS is highly affected by the other two streams.
3. *Safety measures*: Compliance with standards such as ISO 26262 is inevitable for ensuring the functional safety of safety critical systems.

Interweaving these three development streams represents the interdisciplinary collaboration during the design of AAS. A detailed description of the DeSCAS process model and its process meta-model can be found in [4]. The knowledge deduction within the streams *functional development and architecture* and *safety measures* will be focused in the following by means of an example application.

2.2 Formalization of Domain Knowledge

Each development stream of the DeSCAS process model comprises stream-specific ontologies which precisely capture knowledge and expertise within a

stream. In DeSCAS, OWL ontologies are used for the formal representation of such domain knowledge. OWL (Web Ontology Language) is a widely-used and open standard for describing ontologies and has been specified by the W3C³. The benefit and advantage of using OWL is its formal semantics which supports the application of a so-called *reasoner*. A reasoner is not only able to verify the consistency of the concepts modeled within an OWL ontology but also to perform logical deduction in order to automate analysis methods.

Domain standards such as ISO 26262 and the RESPONSE 3 *Code of Practice* supply domain terminology as in a glossary, reference process models and meta-models, as well as process- and product-related requirements. In order to show compliance of the development process to standards relevant for the system to be developed, these standards shall be formalized as done in DeSCAS with the committee draft of ISO 26262. The formalization of each standard involves modeling on the meta-model level and the model level:

- *Derive an ontology from the standard (meta-model level)*: This ontology model is often related to the document structure of the standard. Concerning the committee draft of ISO 26262, about 60 meta-model concepts had to be modeled.
- *Model actual requirements and relations (model level)*: This does not necessarily mean that the requirements need to be presented in a formal way (e.g. using formal languages) but rather documenting the relations and connections between several requirements and the relevant methods and measures, respectively, in a formalized way. The ontology of the committee draft of ISO 26262 exhibits up to 1000 instances of model concepts.

2.3 Semantic Reasoning

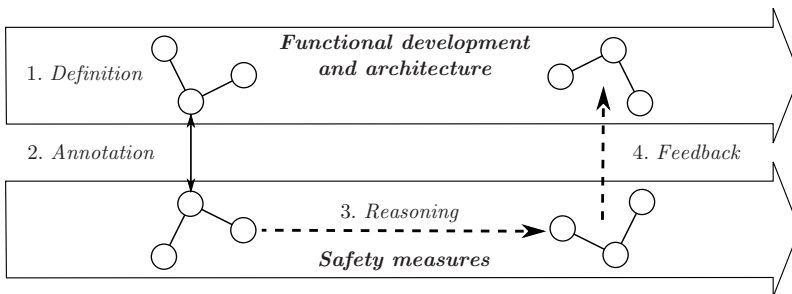


Fig. 1. Interweaving and reasoning within the DeSCAS methodology

Interweaving of models on the basis of formalized domain knowledge can be structured as follows, with Fig. 1 locating the steps within the respective development streams [2]:

³ OWL – <http://www.w3.org/TR/owl-ref/>

1. *Definition* of a concrete domain specific model, which could be a functional requirements model for a certain system under development.
2. *Annotation* (direct or indirect) of the concrete model, based on formal domain ontologies: Thus, it is assured that requirements have a meaning in the sense of the respective domain ontologies.
3. *Reasoning* about implications within and from other domains: This could be an impact on safety and vice versa, due to a given definition of a system under development and associated safety measures.
4. *Feedback* of consequences from another development stream for the very own development activities: These could be concrete safety measures and related activities which either impact the requirements model or concurrent implementation activities.

3 Example Application and Prototype Toolchain

To further illustrate the proposed methodology, the development of an exemplified assistance and automation system from the automotive domain will be sketched as depicted in Fig. 2 and Fig. 3. Both the figure and the following textual description refine the four steps described in Sect. 2.3 – the enumeration of the subsequent text references the numbers in Fig. 3. This example application refers to a lane departure warning system (LDWS) which alerts the driver as soon as the vehicle begins to move out of its lane [5].

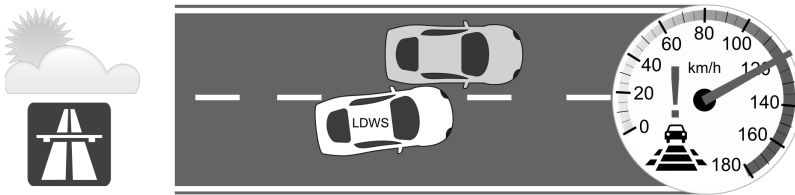


Fig. 2. Vehicle equipped with a lane departure warning system (LDWS) almost colliding with lateral traffic while driving at 120 km/h on a highway during daytime

1. *Requirements model*: The functional *requirements* of the system *component* (the LDWS) are formulated, e.g. the driver has to be alerted when the vehicle is leaving the traffic lane.
2. *Annotation*: The system component is annotated with domain attributes concerning the *operations* to be performed, the *environment* in which the system will be used, and the types of *accidents* which may occur due to malfunctions of the component. Regarding the LDWS, an example operation would be the continuous observation of the vehicle's lateral position within the traffic lane. The environmental conditions refer to driving situations on a normally frequented highway with a dedicated

road marking while driving with a velocity of 120 km/h during daytime in dry weather. They are annotated with the value for the probability of exposure (E) in this situation. Combining the component's operations with the environmental conditions automatically results in the respective *failure scenario*, e.g. a faulty detection of the driving lane on normally frequented highways while driving at 120 km/h during daytime in dry weather conditions. As system independent failure scenarios are referenced, the driver's controllability (C) in these scenarios can be specified independent of the system to be developed and prior to a risk analysis. The various types of possible *accidents* in the automotive domain are quantified via the potential severity (S) of the possible accident. As the LDWS observes the vehicle's position in the respective traffic lane, accidents resulting from unintentionally leaving the lane and colliding with surrounding traffic are possible in case of a potential component failure. Considering the failure scenarios and the possible accident types, the relevant *hazards* can be derived with the help of a generic hazard list that has been compiled in [1] and integrated into the DeSCAS ontology models. One of these hazards would be the possibility of undesired deviation from the traffic lane due to an error between set and actual value concerning the detection of the traffic lane.

- 3a. *ASIL classification*: Once all relevant system hazards have been identified, the risk class and thus the automotive safety integrity level (*ASIL*) of each hazard can be determined by means of the hazard analysis and risk assessment of the ISO 26262. For this purpose, the three parameters S (severity), C (controllability), and E (probability of exposure), which can be derived from the accident types, failure scenarios and environmental conditions linked to the respective hazards, are evaluated. The ASIL of all hazards is calculated using SWRL (Semantic Web Rule Language⁴). Overall, the safety integrity level of the component is determined within the DeSCAS ontologies by the highest ASIL of all identified hazards (e.g. *ASIL B*). This is accomplished by the OWL reasoner *Pellet*. There are four ASIL classes A, B, C, and D, where D represents the highest safety integrity level. A fifth class – QM (quality management) – does not impose any additional safety requirements on the system under development, but rather demands a regular quality management during the development process.
- 3b. *Safety requirements*: Applying the formalized ontology model of the ISO 26262 (see [7]), the calculated ASIL is further processed to infer traceability links to the ASIL-related *safety requirements* which themselves are associated with related *safety methods* and process phases and steps (i.e. *safety clauses*). In this case, a sample requirement of the system design phase would be the system design verification for compliance and completeness, which involves deductive analysis, highly recommended for ASIL B obligation.

⁴ SWRL – <http://www.w3.org/Submission/SWRL/>

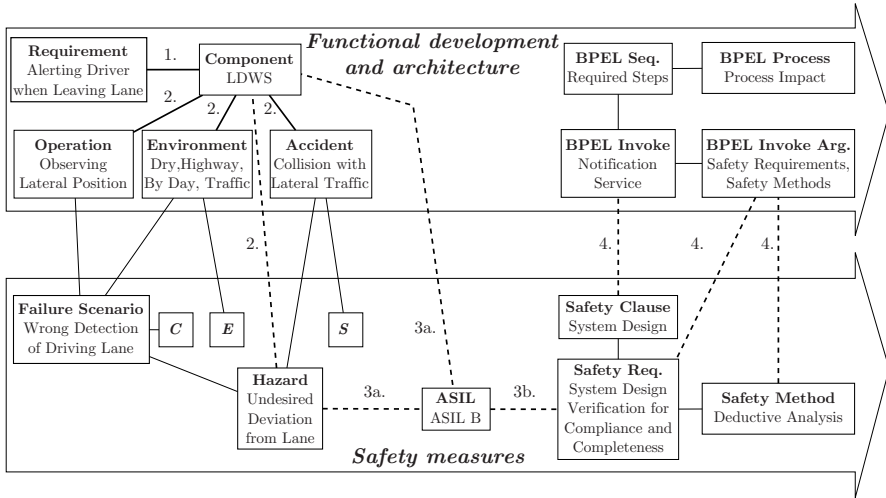


Fig. 3. A lane departure warning system is modeled with the design stream *functional development and architecture* (illustrated by the bold solid lines). Using inference techniques, product and process requirements can be reasoned automatically for the *safety measures* stream (see the bold dashed lines)

4. *Workflow model:* The inference on the formalized process model is used to derive a custom tailored workflow model for the individual developer in the development stream *functional development and architecture*. The tailored workflow model is further transformed into HTML documentation (more thoroughly described in [4]) and a BPEL (Business Process Execution Language⁵) workflow which can be instantiated. In the BPEL workflow, a *sequence* of notification services is *invoked* on the basis of derived ISO requirements and related methods. BPEL workflow monitoring can be used to track the progress of a workflow instance.

Regarding the implementation of these steps in a prototype toolchain, the transformation and reasoning steps have been implemented in Java, using the semantic reasoner *Pellet*⁶ and the *OWL API*⁷. Toolchain integration is built on top of the *Apache Ant* build tool⁸.

4 Lessons Learned from Applying Semantic Reasoning

On a conceptual level, OWL ontologies offer a convenient way for formally capturing and analyzing domain knowledge. The ability of defining descrip-

⁵ BPEL – <http://www.oasis-open.org/committees/wsbpel/>

⁶ Pellet – <http://clarkparsia.com/pellet/>

⁷ OWL API – <http://owlapi.sourceforge.net/>

⁸ Apache Ant – <http://ant.apache.org/>

tion logics axioms within OWL ontologies supports the automation and prototyping of analysis methods. Building upon an open-world assumption, OWL is very flexible and open so that one cannot rely on implicit assumptions. On a more technical level, formal reasoning on OWL ontologies may take a lot of computation time on a state-of-the-art personal computer due to the description logics axioms. To lower this time, the rule language *SWRL* and the query language *SPARQL* (SPARQL Protocol and RDF Query Language⁹) have been used in performance critical situations. Dividing major analysis steps into several individual steps, the computation time of reasoning can also be optimized. In terms of the DeSCAS ontologies, large ontologies with many axioms have been split up into smaller ontologies. However, this results in several consecutive reasoning steps.

5 Conclusion

To sum up, this paper has presented a methodology for interweaving differently geared development activities represented by the three development streams *functional development and architecture*, *safety measures*, and *human factors*, which are relevant within the automotive domain. The methodology forms a development proceeding for the design of safety critical automotive systems heavily relying on a formal base in contrast to most standard proceedings available. For this purpose, domain knowledge has been formalized using OWL ontologies illustrating how design decisions in one development stream may impact other domains and how this information can be used to reason about consequences of design decisions related to the current product development. However, formalization entails additional modeling effort when it comes to formalizing domain knowledge and standards, since a vast number of concepts have to be included in the OWL ontologies. On the other hand, once formalized domain knowledge and standards can be reused in other projects. The prototype toolchain of DeSCAS which has been used for the example lane departure warning system clarifies the advantages and disadvantages of applying OWL ontologies to logical reasoners for formal reasoning, and how to overcome problems arising from long computation times during reasoning.

Nevertheless, future research of DeSCAS will focus on extending and refining the proposed methodology by detailing the modeled domain knowledge (e.g. analyzing accident statistics to determine the severity of system hazards) and integrating more analysis methods (such as ASIL decomposition).

References

1. D. Beisel, C. Reuß, E. Schnieder, and U. Becker. Automotive Generic Hazard List. In *Automatisierungs-, Assistenzsysteme und eingebettete Systeme für Transportmittel (AAET)*, 2010.

⁹ SPARQL – <http://www.w3.org/TR/rdf-sparql-query/>

2. J. Gačnik. Providing Guidance In An Interdisciplinary Model-Based Design Process. In *Proceedings of the 13th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2010)*, Carmona, Spain, May 2010. IEEE Computer Society.
3. J. Gačnik, H. Jost, D. Beisel, J. Rataj, and F. Köster. DeSCAS Design Process Model for Automotive Systems – Development Streams and Ontologies. In *Safety-Critical Systems 2009*, number SP-2222 in Special Publications. SAE International, 2009.
4. J. Gačnik, H. Jost, F. Köster, J. Rataj, K. Lemmer, W. Damm, M. Fränzle, and E. Schnieder. DeSCAS – Formale Ontologien zur Verwebung von interdisziplinären Entwicklungsprozessen. In *AUTOMATION 2009*, number 2067 in VDI-Berichte. VDI Wissensforum GmbH, 2009.
5. ISO – International Organization for Standardization. ISO 17361: Intelligent transport systems – Lane departure warning systems – Performance requirements and test procedures, 2007.
6. ISO – International Organization for Standardization. ISO/DIS 26262: Road Vehicles – Functional Safety, December 2009. Draft International Standard.
7. H. Jost. Automating the Risk and Hazard Analysis via Generic Domain Concepts in Formal Ontologies. In *ESREL 2010, European Safety and Reliability Conference*, 2010.
8. J. Schwarz et al. RESPONSE 3 – Code of Practice for the Design and Evaluation of ADAS. In *PreVENT project deliverable D11.2*. Europe’s Information Society, October 2006.

FORMS/FORMAT 2010

Formal Methods for Automation and Safety in Railway
and Automotive Systems

Schnieder, E.; Tarnai, G. (Eds.)

2011, XI, 257 p., Hardcover

ISBN: 978-3-642-14260-4