Towards a Skill- And Ability-Based Development Process for Self-Aware Automated Road Vehicles

Marcus Nolte, Gerrit Bagschik, Inga Jatzkowski, Torben Stolte, Andreas Reschka and Markus Maurer
Institute of Control Engineering
Technische Universität Braunschweig
Braunschweig, Germany
Email: {nolte, bagschik, jatzkowski, stolte, reschka, maurer}@ifr.ing.tu-bs.de

Abstract—**The development of fully automated vehicles imposes new challenges in the development process and during the operation of such vehicles. As traditional design methods are not sufficient to account for the huge variety of scenarios which will be encountered by (fully) automated vehicles, approaches for designing safe systems must be extended in order to allow for an ISO 26262 compliant development process. During operation of vehicles implementing SAE Levels 3+ safe behavior must always be guaranteed, as the human driver is not or not immediately available as a fall-back. Thus, the vehicle must be aware of its current performance and remaining abilities at all times. In this paper we combine insights from two research projects for showing how a skill- and ability-based approach can provide a basis for the development phase and operation of self-aware automated road vehicles.**

I. Introduction

The recent years have shown a growing trend towards the development of automated vehicles in the automotive industry. Research institutions as well as original equipment manufacturers (OEMs) and startup companies are working towards the market introduction of vehicles which are capable of taking over the whole driving task without or with little supervision of the human driver. Results from different research projects and already available partially automated driving systems (nomenclature according to SAE standard J3016 [1]) show potential customer benefit. However, all of the recently presented automated vehicles driving on public roads (e.g. [2], [3]) still require a human driver and/or external supervisor(s) to monitor the driving performance and to allow manual intervention if the vehicle behaves in an unsafe way. As a result, current production systems have to be considered SAE Level 2 systems. Until highly or even fully automated vehicles are ready for series deployment, manufacturers must ensure that the vehicles are safe in all regards of the complex driving task. Thus, for higher levels of automation, the system must be aware of its performance limits in other words be *self-aware* at the functional as well as on the technical level [4], [5]. Self-awareness means, that the vehicle must not only be aware of its external context (other vehicles, infrastructure, etc.), but also of its internal state in order to relate the vehicle's current performance to the system's performance limits. When it comes to designing such systems, a development process is required, which provides means to seamlessly integrate the development of monitoring mechanisms.

Regarding the development process, the ISO 26262 standard defines the state-of-the-art for functional safety of E/E-systems in automotive applications [6]. During the concept phase of the development process from the ISO 26262 standard, a hazard analysis and risk assessment (HARA) based on expert knowledge has to be conducted to identify and classify hazards originating from the considered system. For fully automated vehicles, this is particularly challenging, as the category of controllability for the driver during risk assessment is not applicable, as the driver is not present in the vehicle's control loop. Controllability for surrounding traffic participants is difficult to determine without a detailed scenario description. From the resulting hazardous scenarios (definition according to [7]) safety goals have to be derived which yield top-level functional safety requirements. Afterwards, requirements for functional components in the system must be identified and further detailed to derive technical safety measures. Monitoring is a key enabler for automated vehicles and must supervise the system's adherence to the derived safety and technical requirements during operation.

This contribution will summarize recent activities at the Institute of Control Engineering of TU Braunschweig towards a specification of an ISO-26262-conforming concept phase for safe self-aware automated vehicles as well as strategies for runtime monitoring. Examples from two current public research projects at the institute are utilized to describe the concept phase of a development process and how the results from the different process steps can be used to set up a framework for runtime monitoring and self-awareness.

The project *Controlling Concurrent Change* (CCC)[1] aims at the design of model-based methods and mechanisms for runtime based integration processes. One project goal is to develop a middleware which implements these methods. At the same time it provides monitoring at different system levels which is required to supervise vehicle operation on the road.

[1] http://ccc-project.org/

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

The project *Unmanned Protective Vehicle for Highway Hard Shoulder Road Works* (aFAS^2) serves as an example for the design phase. aFAS aims at developing an unmanned protective vehicle which is able to operate without human supervision on the hard shoulder (SAE Level 4). A more detailed outline of the project is presented by Stolte et al. [8].

II. Project aFAS

A major cause of fatalities of road workers in Germany are trucks crashing into the protective vehicle on the hard shoulder. This is a main motivation for the development of an unmanned, fully automated protective vehicle. In the project context, the consortium has produced results which are demanded by the concept phase of the ISO 26262 standard. The concept of the protective vehicle includes four operating modes (cf. Fig. 1).

At the beginning of operation (black circle), the protective vehicle will be driven manually to the construction site. From *Manual Mode*, the operator can change the system's state to *Safe Halt*. *Safe Halt* is considered the safe state of the protective vehicle. This has proven to be an enabling factor of the project, as for automated vehicles with a wider range of functions, it is difficult to define a safe state for every possible scenario [4]. Stopping is an applicable safety maneuver in any scenario for the planned use-case due to the low velocity and the limited area of operation. In this operating mode the protective vehicle is slowed down to and eventually kept in standstill. *Safe Halt* also acts as the transition point into automated operating modes *Coupled Mode* and *Follow Mode*. On- and off-ramps are passed in *Coupled Mode* because they differ from normal operation on a hard shoulder and are more challenging for the vehicle's perception system. In *Coupled Mode* the protective vehicle is virtually coupled to the motion of the leading vehicle and follows its trajectory without consideration of lane markings. This virtual drawbar is implemented by vehicle-to-vehicle communication between the leading and the protective vehicle. The main operating mode for safeguarding on the hard shoulder is *Follow Mode*. In *Follow Mode* the protective vehicle follows the left lane marking of the hard shoulder and keeps a distance to the leading vehicle of about 80 to 100 m. Due to regulations from UN-ECE R79, the velocity of the protective vehicle is limited to 10 km/h.

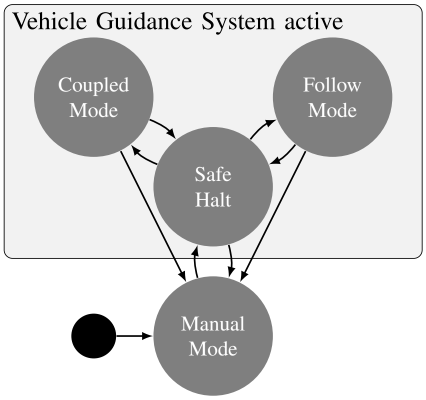


Fig. 1. Operating modes of the unmanned protective vehicle

After having defined the functional range and the operational environment in the item definition, a HARA was conducted, as demanded in the ISO 26262 standard. This yielded 59 hazardous scenarios and 17 resulting safety goals. As this contribution aims at proposing methods for different parts of a development process, we choose one safety goal for the *Follow Mode* as a use-case.

One of the obvious hazards is the protective vehicle crossing (multiple) lanes on the highway and entering moving traffic. This hazard can manifest in two different ways: Table I shows the corresponding safety goals which were chosen to mitigate (1) and eliminate (2) this hazard in the safety concept.

TABLE I
Hazardous events for protective vehicle entering traffic, related safety goals and Automotive Safety Integrity Levels.

Hazardous Event	**Safety Goal**	**ASIL**
(1) Protective vehicle enters moving traffic with full left lock (maximal steering angle).	Limit left lock to 3°.	D
(2) Protective vehicle enters moving traffic with limited left lock.	Always maintain a safe distance to the left lane marking.	B

The Automotive Safety Integrity Level (ASIL) is a combined classification of severity of potential harm, the exposure of the driving scenario and the controllability of the hazardous event [6, Part 3]. In case of the protective vehicle controllability levels have to be differentiated by the capability of other traffic participants to control the hazardous event, as there is no driver or permanent supervisor available for the vehicle. The first hazardous event, the protective vehicle entering moving traffic with full left lock, is classified as ASIL D because

- the hazardous event can cause a fatal crash with multiple cars and high relative velocities (severity S3),
- the driving scenario is the most common for this vehicle (exposure E4), and
- a fast crossing protective vehicle is hard to control for other traffic participants (controllability C3).

This hazardous event can be mitigated by limiting steering left lock in Follow Mode to 3° as stated in the first safety goal. A limited maximal steering angle leads to a reduced lateral velocity. When the protective vehicle is entering moving traffic with reduced lateral velocity, the hazardous event can be classified as ASIL B due to better controllability for other traffic participants (controllability C1). Turns on highways typically provide large radii, such that the protective vehicle can follow all turns with the demanded limited left lock. For this safety goal an ASIL D hard- and/or software system can be implemented with manageable effort. As the final safety concept shall not only mitigate but eliminate system level hazards, the mitigated hazard must be reclassified with the assumption that the limitation is implemented. To eliminate this hazard, we introduce the safety goal to *always maintain a safe distance to the lane marking of the hard shoulder*.

[2] This abbreviation is derived from the German project name.

For the following parts of this contribution we choose this second hazard and the corresponding safety goal to exemplify how to derive further safety requirements and how to use these requirements in online monitoring to maintain a safe operation.

III. Project Controlling Concurrent Change

As described above, current automotive systems are developed according to the ISO 26262 standard, which proposes a V-model-like development process. Following the V-model, the aforementioned functional specification from the aFAS project is located in the upper left part of the ”V”. While aFAS is following a rather classic implementation of the V-model (specification, implementation, integration, test), the project CCC aims at facilitating the integration process in the right branch of the V-Model, particularly with regard to the addition of updates at runtime.

Due to the complexity of modern automotive systems consisting of an increasing number of system components, the verification of a complete system is becoming more and more complex and time-consuming. Integration tests in particular often show the outcome of hidden dependencies between the requirements for single system components, which makes the integration of updated components even more challenging. CCC aims at an automated model-based integration process for safety-critical embedded systems which includes automated analysis of the aforementioned dependencies. For this purpose, the system is modeled from a variety of viewpoints (e.g. function, timing, platform) using the corresponding requirements as a basis. However, as not all side-effects, which occur during operation, can be anticipated in the design process, monitoring mechanisms are instantiated at runtime.

In this paper, we describe skills and abilities as one of the models applied at the functional viewpoint which is used in the project context. In the context of the aFAS use-case we present a knowledge-based approach for the derivation of performance metrics for abilities.

IV. Concept of skills and abilities

The basic idea behind the skill and ability based representation is to model the system along the human driving task at a functional level. Dependencies (edges) between skills and abilities (nodes) are expressed in the form of a directed acyclic graph. The main concept and its origins have been described by Reschka *et al.* [9]. Further literature on the need for an internal representation of the system's performance can also be found in [5].

This can yield a representation of what the system must do how well for fulfilling its functional specification. In this context we would like to correct a translation error (reversed terms ability and skill) which occurred in [9], originating from a misinterpretation of a German source [10]. For correct translation, we refer the definition of skills and abilities for technical systems given by Reschka [11]:

Definition 1: A *skill* describes an activity of a technical system which has to be executed to fulfill the defined goals of the system (*translated from German*).

Definition 2: An *ability* describes the quality level of an activity dependent on internal properties and the current operational situation of the system (*translated from German*).

The differentiation between skills and abilities allows to apply the concept to the development phase (skill graph, Section V) and the use phase (ability graph, Section VI) of the system (nomenclature of phases according to [12, Chap. 3]. In this context, the skill graph can be used to derive safety and technical requirements. The requirements can then be attributed to skills in the skill graph as annotations of the outgoing edges of a skill. For instantiating the ability graph at runtime, expressive metrics for performance monitoring are required. In combination with the requirements derived for the skill graph, these metrics yield the basis for determining a performance level for each ability node in the graph at runtime. The performance level of an ability is represented by the edge weight of the incoming edge and consists of the performance levels annotated to the outgoing dependency edges.

The following sections will present possible applications of skills and abilities in the development and use phase in more detail. If not mentioned otherwise, we use terms defined by Avizienis *et al.* [12] and focus on the *safety* aspect in the given taxonomy of *dependability*. For the derivation of safety requirements, we draw the system boundary around the vehicle. In consequence we propose an extension of the terms given in [12]: Avizienis *et al.* define *behavior* as a sequence of *states*. The external state of a system is defined as ”the part of the provider's total state that is perceivable at the service interface” [12, p. 3]. Based on this, we differentiate *internal* and *external* behavior, as a sequence of *internal* and *external* states, respectively. As a result, the *internal behavior* describes what the system does internally to fulfill its function, while the *external behavior* is the behavior which directly influences other traffic participants and can thus cause hazards.

V. Skills in the development process

The ISO 26262 standard demands the formulation of functional descriptions and concepts for a functional safety concept without giving details of technical solutions. With this demand, the resulting concepts can be reused in further stages of development and assessed by external reviewers without struggling with a flood of information. Fig. 2 shows a shortened overview over steps in the concept phase.

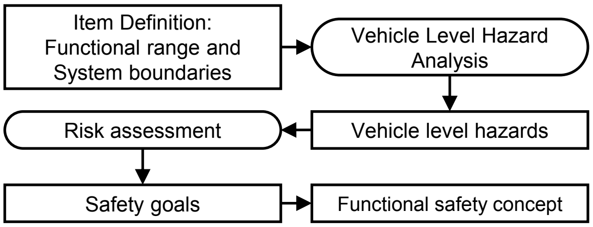

Item Definition: Functional range and System boundaries
Vehicle Level Hazard Analysis
Risk assessment
Vehicle level hazards
Safety goals
Functional safety concept

Fig. 2. Top-down development process

Reschka [11] proposes to use skill graphs in the *item definition* which is the first step in the development phase. After creating a use case description, the level of automation is determined. Followed by this, operational scenarios, the

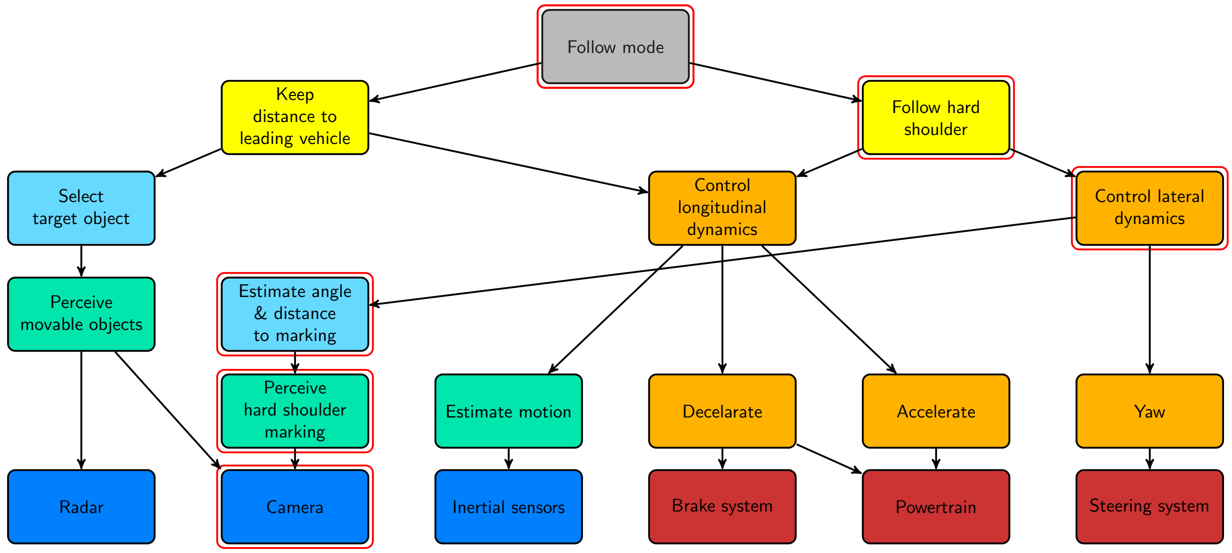


Fig. 3. Skills for follow mode with categories main (grey), observable external behavior (yellow), perception (green), planning (light blue), action (orange), sensors (blue), actuators (red). Examplatory path for structured derivation of performance metrics from safety goals is depicted by red frames.

intended behavior, and safe states for each scenario must be described. To cluster and describe operational scenarios of the vehicle, maneuvers can be used to express the behavior of vehicles from an external point of view.

Reschka also proposes a list of driving maneuvers to cover possible traffic scenarios based on literature review and further refinements towards automated vehicles. Each maneuver can be expressed by physical measures (vehicle A going faster than vehicle B) and semantic relations (lane change from right to left). Therefore maneuvers describe the *external behavior* of the vehicle. Which maneuvers have to be fulfilled by the vehicle can be derived from the intended use case and the operational scenarios. The chosen maneuvers can then be modeled using skill graphs. This model shall be independent from specific situations which the vehicle could encounter at run-time. Skill graphs can provide information how a system is able to fulfill the driving task and which dependencies exist between the system's skills in a functional manner. In this context, external behavior is characterized by skills which are in direct relation to the described top-level function (cf. Fig. 3). The majority of skills in the skill graph describe the *internal behavior* of the system.

Bagschik et al. [13] used skill graphs combined with computer-aided generation of operational scenarios to derive potentially hazardous scenarios according to the designed use case. This analysis was conducted without any technical details of implementation. One resulting skill graph was designed for the maneuver of following the hard shoulder in the state *Follow Mode*. Each skill from the graphs in the item definition was assigned to the category *main*, *sense*, *plan* or *act* (cf. Fig. 3).

As the ISO 26262 standard formulates that the HARA "is based on the item's functional behavior; therefore, the detailed design of the item does not necessarily need to be known." [6, Chap. 3.72] We further define that the mentioned functional behavior is the *external behavior* of the item.

The main item for assessment is called AFA-logic as described by Stolte *et al.* [7] and describes the service provided by the system on the top-level. Thus, we have to identify which external system states are hazardous in each possible operating scenario of the item. As mentioned above, the identification of the resulting hazardous scenarios which result from the *external* behavior of the system, yields the basis for safety goals for the *external behavior*. This directly relates the list of functional safety requirements (cf. Fig 4) to the externally visible behavioral skills (cf. Fig 3). For the underlying skills, more specific technical requirements can be derived in order to fulfill those functional safety requirements.

The given safety goal *always maintain a safe distance to the left lane marking* would thus entail an exemplary functional safety requirement of the form *the vehicle must always keep a distance of at least* 0.28 m *from the left lane marking when following the hard shoulder* for the skill *follow hard shoulder*. For the underlying skill *control lateral dynamics*, this would translate to a control-performance metric of the form: *the lateral controller must guarantee a maximal overshoot of* 0.28 m. These requirements must be met during operation of the vehicle for guaranteeing safe operation. A concept for monitoring (safety-related) system performance will be presented in the next section.

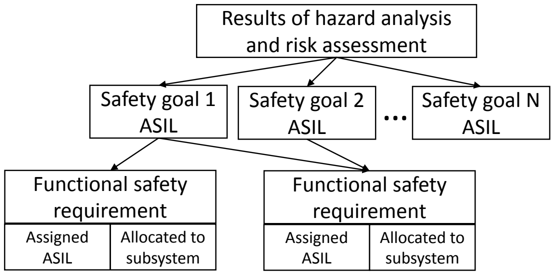


Fig. 4. Relation of HARA, safety goals and functional safety requirements according to [6].

VI. Abilities for online performance monitoring

For runtime monitoring of the system's performance, the skill graphs derived in the development process can be transformed into ability graphs. The challenge for monitoring performance in specific situations at run-time is to find expressive metrics even if metrics from different domains are combined for calculating a level of performance.

Abilities can consist of multiple interacting software components which makes them a (sub-)system, according to [12]. As performance is measured at the interfaces of these subsystems, the performance attributed to abilities can be defined as *quality of service* (*service* cf. [12, p. 3]).

In the field of (autonomic) grid computing, substantial research effort has been spent in recent years to create structured approaches towards a meaningful computation of quality of service in heterogeneous computing systems. For this purpose, knowledge-based approaches have been developed which apply ontologies (formally ordered representations of terms, obects and their relations) as a basis for knowledge representation (cf. [14] for an overview). Possible applications range from a specification of quality of service during design- and runtime to reasoning about the quality of composed services [15]. These applications are in line with the development process for self-aware automated vehicles which has been proposed in the sections above.

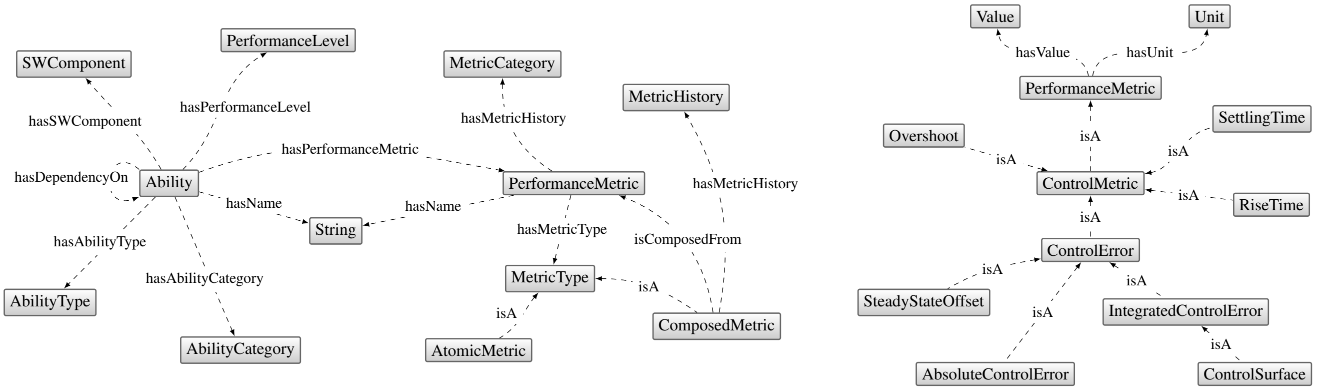
For our application, ontologies provide means to model performance measures for each skill and incorporate identified safety goals and resulting requirements for the abilities. Additionally, an ontology extending the semantics of the ability graphs allows to define mappings between the vehicle's abilities and software components implementing the respective ability. With this connection it is possible to explicitly define redundancies within the system to support degradation strategies. As ontologies provide a formal description of entities and their relations, they also enable reasoning about sensible combinations of entities at runtime. Considering reconfigurable systems, as in the CCC project, the relations can be used as a basis to synthesize redundant (software-)sub-systems and corresponding quality metrics at runtime.

However, simply propagating metrics from lower levels unchanged through the graph adds little to no value to the monitoring on higher levels. Instead, we aggregate lower level metrics to composed metrics on higher levels which allows us to add a weight to each metric and prioritize certain aspects over others during monitoring. During composition, it must be ensured that metrics remain expressive for effective monitoring. To ensure this, we equip each metric with a history of the metrics it was composed of and utilize this history for the composition of metrics on higher levels of the graph.

Within the ontology, we differentiate between abilities and metrics as shown in Fig. 5 a), although slightly simplified for more clarity. An ability depicts a node in the ability graph and can depend on other abilities within the graph which is specified by a dependency relation. A dependency relation is required for each ability to reflect the edges of the graph as shown in Fig. 3. It can be described by its type, i.e. whether the ability node is the main node, a sub node, a data sink, or a data source. A category can be assigned in analogy to the skill graph, i.e. by grouping the nodes into visible external behavior, perception, planning, action, sensors, and actuators. Each ability has a performance level, and a dedicated mapping regarding the software component(s) which implement this ability and thus provide data for associated performance metrics. An ability can have one or several metrics from which the performance level is determined. Each metric can be grouped by whether it is a composed metric or an atomic metric. In addition to their composition history composed metrics are attributed with information about their composition rules in order to avoid fusion of correlated metrics. Still, finding appropriate and meaningful metrics for each ability and graph level, aggregation measures, as well as models for performance propagation poses a challenging task and is subject to ongoing research. From the safety goals identified in section II, we derive metrics for the main ability *Follow mode* and the visible behavior ability *Follow hard shoulder*. On the highest level *Follow mode* the identified safety goal of *Do not drive into traffic* must be adhered to at all times. Thus, it is essential to monitor whether the vehicle enters or is about to enter traffic of the rightmost lane of the motorway. Based on this, the safety goal for the ability *Follow hard shoulder* consists of maintaining a safe distance to the lane markings of the hard shoulder at all times. To derive monitoring metrics from this safety goal we need to define what constitutes a minimal safe distance. Thus, it must be evaluated whether the current distance to the lane markings lies within a specified range below the safe distance of 0.28 m.

In our case, the requirements have been defined in the item definition as stated in section V. The technical safety requirement for *control of the lateral verhicle dynamics* implies the need for monitoring the overshoot of the control algorithms and check whether it stays within the predefined range. As the control algorithms are dependent on the *estimated angle and distance to the perceived lane markings*, these estimations must provide some measure of uncertainty, e.g. the variance of the estimation. The obvious metric derived from this safety goal is to monitor whether the variance falls within a specified range to stay within the bounds of the hard shoulder. It should also be known whether the variance estimation is optimistic or pessimistic. Considering the *perception of the lane markings*, we require a representation of invalid results meaning a plausibility check must be performed to determine validity of the detected markings and to detect when existing markings were not detected.

Fig. 5 b) shows parts of a performance metric ontology which can be used for monitoring the control algorithms within the graph. This part of the ontology can be seen as a generalization and further specification of the class hierarchy presented in [16]. In our example, this set of available performance metrics would be applied to monitor control of the lateral dynamics of the vehicle. A meaningful performance level can e.g. be calculated by comparing the actual steady



a) Classes and object properties for an ontology describing abilities and corresponding performance metrics

b) Exemplary ontology for metrics addressing control quality

Fig. 5. Ontologies for structured derivation of performance metrics

state offset during operation with the maximal allowed steady state offset from the corresponding requirement (0.28 m).

Summarizing, the proposed ontology provides means for transforming skill graphs into ability graphs for runtime performance monitoring by providing additional semantics for metrics and performance. The approach of deriving metrics and thresholds from identified safety goals provides a useful method to aid in this process.

VII. Conclusion

We have outlined how skill and ability graphs can provide a valuable method for modeling the vehicle's capabilities in the development process while also enabling online monitoring. Employing skill graphs in the item definition allows the derivation of metrics for runtime monitoring directly from the safety goals identified in the HARA. Thus, since safety goals are defined for the externally visible vehicle behavior, we distinguish external from internal system behavior.

Although ontologies can provide means for derivation and synthesis of performance metrics, it must be mentioned that their power must not be overstated (cf. [15]). They do not provide ready-made solutions, but should be considered as a tool for coming to solutions in the task of introducing expressive monitoring mechanisms from a functional viewpoint.

Acknowledgment

This work is part of the DFG Research Unit Controlling Concurrent Change, funding number FOR 1800.

The project aFAS is partially funded by the German Federal Ministry of Economics and Technology (BMWi). The consortium consists of MAN (consortium leader), ZF TRW, WABCO, Bosch Automotive Steering, TU Braunschweig, Hochschule Karlsruhe, Hessen Mobil - Road and Traffic Management, and BASt - Federal Highway Research Institute.

We would like to thank our partners and colleagues in both projects for valuable discussions, Mischa Möstl in particular for providing additional input from a different domain.

References

[1] SAE, *J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, 2016.

[2] Tesla, *Autopilot*, https://www.tesla.com/presskit/autopilot, accessed: 2016-12-12.

[3] A. Davies, *Uber's self-driving truck makes its first delivery: 50,000 beers*, https://www.wired.com/2016/10/ubers-self-driving-truck-makes-first-delivery-50000-beers/, accessed: 2016-12-08, 2016.

[4] A. Reschka and M. Maurer, "Conditions for a safe state of automated road vehicles," *it - Information Technology*, vol. 57, no. 4, pp. 215–222, 2015.

[5] J. Schlatow, M. Möstl, R. Ernst, M. Nolte, I. Jatzkowski, M. Maurer, C. Herber, and A. Herkersdorf, "Self-awareness in autonomous automotive systems," in *Design, Automation and Test in Europe (DATE)*, 2017.

[6] *ISO 26262:2011: Road vehicles – functional safety*, International Organization for Standardization, Geneva, Switzerland, 2011.

[7] T. Stolte, G. Bagschik, A. Reschka, and M. Maurer, "Hazard Analysis and Risk Assessment for an Automated Unmanned Protective Vehicle," in *2017 IEEE Intelligent Vehicles Symposium*, 2017, accepted to appear.

[8] T. Stolte, A. Reschka, G. Bagschik, and M. Maurer, "Towards Automated Driving: Unmanned Protective Vehicle for Highway Hard Shoulder Road Works," in *2015 IEEE International Conference on Intelligent Transportation Systems*, 2015, pp. 672–677.

[9] A. Reschka, G. Bagschik, S. Ulbrich, M. Nolte, and M. Maurer, "Ability and skill graphs for system modeling, online monitoring, and decision support for vehicle guidance systems," in *2015 IEEE Intelligent Vehicles Symposium*, 2015, pp. 933–939.

[10] J. Funke, P. Zwitserlood, J. Müsseler, K. Sonntag, J. Bengel, F. Petermann, S. Gauggel, G. Gründer, R. Volbert, H. W. Bierhoff, N. Döring, R. M. Puca, M. Hasselhorn, J. B. Asendorpf, S. Walper, H. E. Lück, M. Huff, G. Schwarzer, K. Moser, M. Wirtz, and J. Strohmer, *Dorsch - Lexikon der Psychologie*, German. Verlag Hans Huber, 2014.

[11] A. Reschka, "Fertigkeiten- und Fähigkeitengraphen als Grundlage für den sicheren Betrieb von automatisierten Fahrzeugen in städtischer Umgebung - English title: Skill and ability graphs as basis for safe operation of automated vehicles in urban environments," announced, PhD thesis, Technische Universität Braunschweig, 2017.

[12] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[13] G. Bagschik, A. Reschka, T. Stolte, and M. Maurer, "Identification of Potential Hazardous Events for an Unmanned Protective Vehicle," in *2016 IEEE Intelligent Vehicles Symposium*, 2016, pp. 691–697.

[14] V. Sorathia, L. Ferreira Pires, and M. Sinderen, "An analysis of service ontologies," *Pacific Asia Journal of the Association for Information Systems*, vol. 2, no. 1, pp. 17–46, 2010.

[15] G. Dobson, R. Lock, and I. Sommerville, "Developing an ontology for QoS," in *Proceedings of the 5th Annual DIRC Research Conference*, 2005, pp. 128–13.

[16] M. Maurer, "EMS-vision: Knowledge representation for flexible automation of land vehicles," in *2000 IEEE Intelligent Vehicles Symposium*, 2000, pp. 575–580.