# Ability and skill graphs for system modeling, online monitoring, and decision support for vehicle guidance systems

Andreas Reschka[1] and Gerrit Bagschik[1] and Simon Ulbrich[1] and Marcus Nolte[1] and Markus Maurer[1]

***Abstract*— In this paper, the *ability and skill graphs* are introduced for modeling vehicle guidance systems in the concept phase of the development process (abilities), for online monitoring of system operation (skills), and to support driving decisions (skill levels) of automated road vehicles and advanced driver assistance systems. Both graphs rely on a decomposition of the human driving task.**

**An *ability* is the entirety of conditions which are necessary to provide a certain part of the driving task. The ability graph can be developed in parallel to the item definition according to the ISO 26262 standard in the concept phase of the development process and can be used for supporting further development steps. A *skill* is defined as an abstract representation of a part of the driving task including information about the skills current performance. The skill graph is used to monitor the current system performance during operation and skill levels are input to driving decisions. Abilities and skills cover all aspects of the driving task including environment and self perception, data processing, decision making, and behavior execution.**

**During operation of the developed item, the skill graph is instantiated as a (distributed) software component to process online information for assessing current skill levels. Each skill uses one or more performance metrics, which represent its current performance capability in relation to the maximum (inherent) ability level. The resulting information could replace the monitoring of the system by a human driver and can be used as an input to driving decisions of the vehicle to support appropriate and safe decisions.**

## I. PROBLEM STATEMENT

In current research projects, prototypes of automated vehicles are monitored by a human safety driver at any given time (cf. [1]–[4]). According to Ohl [5], the still necessary human monitoring classifies these vehicles as Level 2 - *Partial Automation* according to [6] and *teilautomatisiert* according to [7] respectively.

In future applications, parts of the driving task or the whole driving task are to be executed by an electric/electronic programmable system, called a *vehicle guidance system* (german translation: *Fahrzeugführungssystem*), without human monitoring, e.g. the *Autobahn-Chauffeur* published in [7]. Thus, it is necessary that a vehicle guidance system executes all the tasks a human driver is otherwise responsible for. These tasks cover the control of the vehicle, the monitoring of passengers, and the monitoring of the vehicle condition and the vehicle guidance system itself. In [8]–[10] the tasks necessary to control a vehicle are described. They focus on perception of the environment and control of the vehicle's actuators. The monitoring tasks are only described implicitly in traffic rules, e.g. [11, § 23] in the traffic regulations in Germany. This paragraph states, that a driver has to check the vehicle condition, the passengers, and the goods (or load) transported before driving [11, § 23]. While the monitoring of small sets of system states, e.g. the state of charge and/or health of the vehicle's battery, the tire pressure, or system temperatures, is common practice in vehicle systems engineering, holistic approaches, which provide complete self-assessment of the current skills of a vehicle are rarely seen in literature. For example the monitoring of suspicious sounds and vibrations of mechanical parts is still not covered by technical systems. The driving experience and the sensitivity of humans is not yet reproducible in control algorithms [12]. Figure 1 shows the basic tasks a driver is responsible for while driving a vehicle in public traffic.

[1]All authors are with the Institute of Control Engineering, Technische Universitaet Braunschweig, 38106 Braunschweig, Germany reschka, bagschik, ulbrich, nolte, maurer@ifr.ing.tu-bs.de

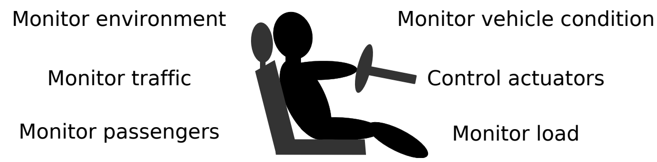


Fig. 1. Tasks of the human driver, while driving a vehicle in public traffic.

Human drivers perform quite well in today's traffic, also because the current traffic structure has been developed by humans to fit human driver's needs. For automated systems a comparable or even better performance seems appropriate, but is still a matter of research. [13] provides an overview of the current research in societal aspects of vehicle automation.

Besides detecting the described initiating and occurring defects, error handling is another important task, the human driver is responsible for. In most situations, most drivers are able to stop the vehicle at a safe location. Thus, a vehicle guidance system has to detect defects and eventually stop the vehicle at a safe location, as well. More generally speaking it is necessary that the vehicle guidance system detects defects and reacts appropriately to avoid unsafe driving. In [14], Bergmiller describes an approach which uses probabilistic methods to detect and handle faults. The focus of [14] is on a stability control system, but can be extended to other systems, as well.

The demand for self-monitoring does not only occur in vehicle guidance systems which enable automation levels like SAE Level 5 - *Full Automation* (not yet defined in [7]) and Level 4 - *High Automation* (*Vollautomatisiert* in [7]), but also in Level 3 - *Conditional Automation* (*Hochautomatisiert* in [7]) applications for the duration of its activation unless the human driver has taken over control [6].

With an increased automation level, the driver becomes a surveillant of the technical system. Humans can not perform this task very well according to Bainbridge [15] and the Yerkes-Dodson-Law [16]. Thus, it is imaginable that increased automation leads to dangerous situations, because due to a mental underload humans could be unconcentrated and can be distracted from their remaining tasks, as well.

According to the ISO 26262 standard, the controllability of a situation by the driver or by other traffic participants has to be considered when assigning an automotive safety integrity level (ASIL) to a system [17]. As there is no permanent monitoring of the system in these automation levels, the controllability of the vehicle by the driver is not given. The controllability of the situation by other traffic participants relies heavily on the behavior of the automated vehicle. Thus, it is not only necessary to stop the vehicle in case of failure, but also to avoid endangering others, e.g. by stopping at a safe location. The appropriate reactions of the vehicle guidance system to failures is still a matter of research.

In this paper, the ability and skill graphs are introduced as tools for modeling, monitoring, and decision support. In the item definition according to the ISO 26262 standard [17] the ability graph (cf. IV-A) for the item can be developed and detailed. In the hardware and software development phases it can be extended by technical details of the system and transferred into a skill graph (cf. IV-B). This skill graph can then be used for online monitoring of the vehicle guidance system (V-A). The resulting skill levels can be used for decision support (cf. V-B).

## II. RELATED WORK

The basic concept of abilities in vehicle guidance systems has been published in [18]. Maurer outlines the demand for a permanent online monitoring of the vehicle capabilities. Siedersberger [19] and Pellkofer [20] further developed this idea with a strong focus on taking driving decisions based on an *ability network*. The ability network is basically a graph, which divides the driving task into sub-tasks (abilities) that are provided by so called experts and executed in parallel. Experts are software and hardware components, which run sub-tasks. The current abilities and their performance are used to influence driving decisions. The modeling of abilities in a graph with dependencies is similar to *Behaviour-Based Systems* in [21].

In [22], Bergmiller describes the skill network as part of a self-concept, which enables monitoring and especially fault-detection in full-by-wire vehicles. The self-concept allows a self-representation of the vehicle's current performance for each skill. It combines a vehicle model with a performance model to represent the physical state of the vehicle and its possibilities. Bergmiller also gives an outlook to the applicability of the self-concept for vehicle automation. As [22] focuses on by-wire-vehicles and does not consider environment perception and decision taking explicitly, the concept itself may be applicable for vehicle guidance systems, but it still lacks one important issue. In the requirements phase for vehicle guidance systems the exact specification is not available, because there are too many unknown conditions in public traffic. Thus, it is difficult to find performance metrics for skills, which rely on incomplete requirements and have to deal with high uncertainty. This is one of the topics of our current research, e.g. in the Stadtpilot project [23], the aFAS project [24], and the future development of the research vehicle MOBILE, amongst others published in [25].

Knoll [26], [27] uses the term *self-representation* which describes the inner condition of a robot and considers the external conditions of the environment. The inner condition is acquired by *self-perception* (cf. [28]). The information provided by a combination of these two allows a self-representation which can be used for safe decision taking. This understanding fits our approach, which combines self-perception and external conditions to a self-representation in the skill graph. Knoll [26], [27] also describes an approach for execution monitoring, which can also be found in [19]. Execution monitoring slightly differs from online monitoring of components. In normal operation of a mobile system a maneuver, e.g. a lane change, is planned and executed. The execution monitoring component compares the planned maneuver, e.g. a desired pose of the vehicle or robot, with the real resulting pose. The delta can be used as input to the self-representation. A drawback of this approach is the fact, that a maneuver is only evaluated after its execution. Thus, at least one unsafe maneuver could be planned and executed. For a safe operation it is necessary, that unsafe maneuvers have to be avoided in the planning algorithms.

Thuy et al. [29] describe a safety system for the experimental vehicle *MUCCI* at the University of the Armed Forces in Munich, which uses the skill network proposed by Pellkofer and Siedersberger for the performance evaluation of the vehicle's system. The concept is based on execution monitoring and is using the gathered information for online performance evaluation and its results for decision support.

In [23], an online monitoring approach from the Stadtpilot project has been published. It includes less capabilities and is not as generic as the ability and skill graphs are for advanced driver assistance systems and automated vehicles. But nevertheless, the approach uses sensor data and performance criteria (performance metrics) for online monitoring and for safety decisions.

Although, the ability and skill graphs for modeling human-like behavior are relatively new to technical systems with safety critical functions, it is already well-established in pen and paper and computer gaming. Especially in role-play games where the player takes the role of a fictive character, the skill tree is a common tool to model abilities and skills of the fictive human or human-like character. The skills have dependencies are interdepended and they are used in combination with basic attributes of a character. A famous pen and paper game is *Advanced Dungeons and Dragons*, a famous role-play computer game is *World of Warcraft*. In Advanced Dungeons and Dragons abilities represent attributes like strength, dexterity and wisdom of a character, whereas skills are attributes with a specific skill level that rely on an attribute, e.g. *Acrobatics* is a skill which relies on the attribute dexterity and some other factors [30]. In World of Warcraft attributes are similar to Advanced Dungeons and Dragons, but the so called talents are organized in *talent trees*. These talent trees are similar to the skill graph, because each higher level talent relies on the lower level talents. Talents can be learned and change during the game. Thus, all talents in the talent tree change their performance during the game, depending on the game situation [31].

In this paper, work from previous research by Maurer, Siedersberger, Pellkofer, Bergmiller, Knoll, and Reschka is combined to create tools for modeling and monitoring of vehicle guidance systems.

## III. TERMINOLOGY

The terms *abilities* and *skills* have not always been used consistently as Bergmiller points out. Especially in [18]–[20] the German terms "Fähigkeit" und "Fertigkeit" are used and cannot easily be translated [22]. In [10] these terms are defined as well and are focused on the human information processing and motoric actions. Thus, we expect great use of defining these terms in the context of vehicular system modeling as both terms are common in describing capabilities and performance of humans.

According to [32], an ability is defined as the entirety of conditions which are necessary to deliver a performance.

A skill is the description of an activity related to a certain task including a performance level (skill level). Further (human) skills can be separated to (senso-)motoric, cognitive, cognitive motoric, social, linguistic and perceptive skills [32].

In relation to vehicular systems the categories motoric, cognitive, perceptive, and, according to cooperative systems, social seem to be applicable, as well. Bergmiller mainly refers to motor abilities, as the work focuses on representation of observable actions carried out by the vehicle. He takes [33] into account to define the counterpart to motor abilities as cognitive abilities. At this point we propose to refine the cognitive abilities and point out that automated vehicles have, for now, no learning ability as humans do. The subcategories of cognitive abilities for automated vehicles can be defined as perceptive (for environment and self-perception), planning (for decision making and trajectory planning) and social (for upcoming cooperative systems).

Abendroth [10] defines (human-)abilities according to physiological abilities as the individual, time-dependent changes of human basic functions. This definition is not fully suitable for the technical systems in this article. Cognitive abilities are then named *intelligence* and define the hierarchical structured entirety of mental abilities. These mental processes determine the level and quality of a human personality. The concept of intelligence is applicable to vehicle guidance systems as we model abilities in a graph-based hierarchical structure to describe the system and the level of performance. Skills in [10] mainly depend on the level of experience and style of driving of the human driver. According to this definition, human drivers and technical vehicle guidance systems differ as the technical system does not gain experience in a human sense.

## IV. ABILITY AND SKILL GRAPHS

An ability graph can be used as a design tool in the concept phase of the development process, e.g. after the *Item Definition* according to the ISO 26262 standard [17]. It turns into a skill graph as soon as the abstract concept is instantiated into an implemented technical solution, which can provide numeric performance metrics.

### A. Ability Graph

An ability graph is used to model a -to be developed-system or *item* by its abilities. An ability is an abstract representation of a part of the driving task entailing necessary conditions to provide the ability to the system.

According to ISO 26262 standard [17], the development of a system begins with the *item definition*. In this definition, the item's functionality is described according to the desired extrinsic behavior. The extrinsic behavior describes the behavior which can be observed from an external point of view (e.g. a sensor or an external observer). On the contrary, the intrinsic behavior can only be observed through interfaces to the system (e.g. a debugging monitor). It will be modeled after the concept phase in the development process.

In an ISO 26262 standard inspired design process, an ability graph will be designed in the *concept phase* after the item definition and will initially be purely conceptional and thus fully independent from a technical implementation.

The design of the ability graph results in an abstract representation of the system's driving task which is inspired by the human driving task. During the development process the ability graph is refined and extended while the knowledge about the design of the system and its components increases. Finally the ability graph is transferred into a skill graph (section IV-B) for online monitoring of the resulting system.

An ability graph is directly derived from the system's requirements. Therefore, it is different from the technically motivated functional system architecture, which shall be developed in the concept phase, as well, but is mainly used as baseline for a functional system overview in the system development phase. A functional system architecture and an ability graph can be used in parallel to provide a holistic description and can be considered as different views on the same system. A functional system architecture provides an overview over the interaction of functional entities. An ability graph visualizes how abilities depend on other abilities, data sources and sinks. It visualizes functional or physical redundancies and provides an insight into how abilities and subsystems interrelate with system requirements.

Public traffic and its rules are designed for humans. Therefore, it stands to mind to design a system to act human-like. Thus the system's ability graph may strongly be correlated with humans' abilities.

Figure 2 shows the conceptual structure of a basic ability graph. The main node represents the desired ability of an item. Below this, ability nodes are hierarchically organized to provide sub-abilities to superordinate nodes. Thus, abilities and their dependencies can be modeled as a hierarchical,

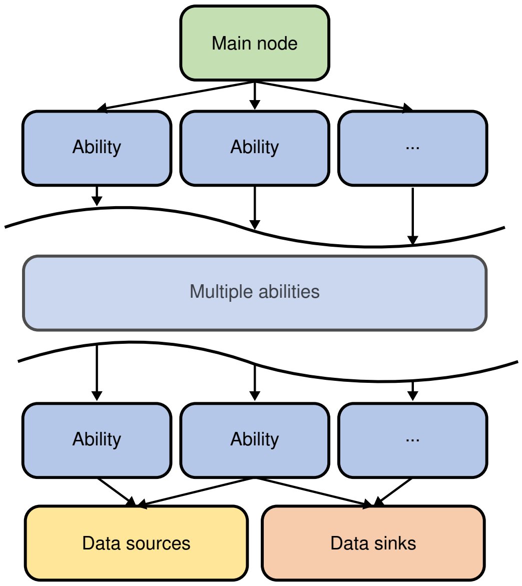


Fig. 2. Basic ability graph. A main node represents the item's main ability.

acyclic graph. In such a graph, nodes and edges are defined as following:

*1) Ability Nodes:* An ability node is characterized by its input requirement dependencies and the capabilities resulting from it. Each ability node (A) is decomposed into at least two other sub-abilities (B,C), which are necessary to fulfill the capabilities of ability A. This decomposition is repeated until the abilities can no longer be decomposed in underlying abilities. Hence, *basic abilities* are fringe ability notes, only depending on data sources and sinks, e.g. sensors and actuators to perceive or manipulate aspects of the environment or the ego-vehicle. If only parts of the overall system shall be modeled, it is possible to *hide* further decompositions of an ability node to limit the complexity of the graph.

*2) Quality Requirement Edges:* Edges between ability nodes represent quality requirements to subordinate nodes. An edge is directed from a superordinate node towards a subordinate node. It expresses a dependency between two nodes and represents a vector of quality requirements between abilities. These quality requirements can be derived from the functional system requirements, e.g. from the other work products in the concept phase of the ISO 26262 development process.

*3) Designing an Ability Graph:* Figure 3 shows a simplified ability graph for an Adaptive Cruise Control (ACC) system (Main ability). The represented function is a system that can control distances to in front driving vehicles (A1), control the speed (A2) to a value set by the human driver, and keeps the vehicle controllable for the human driver (A3). To follow another vehicle, a selection of the ACC target object is necessary (A4). To select a target, it is necessary to detect all dynamic objects on the road in front of the equipped vehicle (A5) and to detect the drivers intention (A6). The drivers intention is necessary to determine the correct lane for selecting the target object and to get the desired maximum speed while controlling speed (A2) without a valid

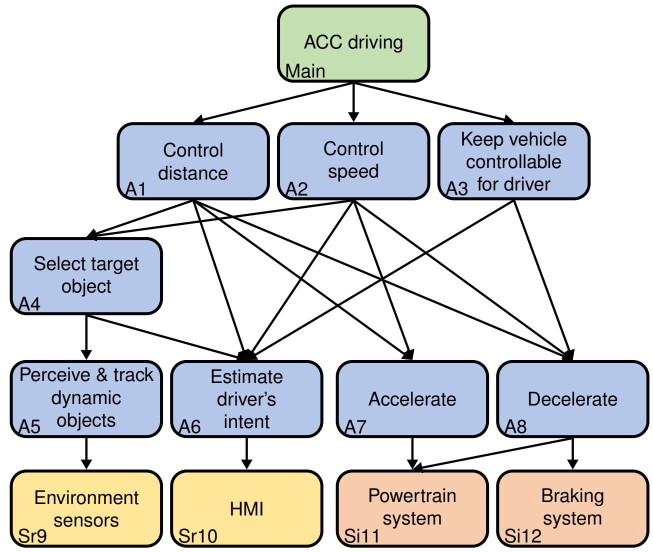


Fig. 3. Simplified ACC ability graph. For better visibility the graph is reduced to a minimum of abilities.

target object. Abilities *Accelerate* (A7) and *Decelerate* (A8) represent the vehicle's ability to accelerate and decelerate. On the lowest level, sensors are data sources and actuators are data sinks. The *Human-Machine-Interface (HMI)* (Sr10) includes pedals, the steering wheel, switches and buttons. The *Environment sensors* (Sr9) are used for dynamic object detection. The *Powertrain system* (Si11) can be used to accelerate the vehicle and with limitations to decelerate it. The *Braking system* (Si12) decelerates the vehicle.

In Figure 3 the ACC system is modeled with a reduced level of detail. The node *Perceive and track dynamic objects* entails several skills like object segmentation, dynamic classification, object tracking, etc.

In the concept and system design phase, this decomposition will become increasingly detailed. It may depict necessary redundancies for safety goals, subsystem requirements as well as error propagation and performance degradation within the system. Degradation is inspired by graceful degradation in biology (s. [34]) and allows a continuing operation with reduced functional capabilities. If the system is fully designed and implemented, the ability graph can be translated into a skill graph.

## *B. Skill Graph*

The skill graph is used for online monitoring and decision support of vehicle guidance systems. Its structure can be derived from a system's ability graph. A skill is derived from an abstract ability by instantiating this ability into a usable activity with a performance level of execution (skill level). Thus, an ability graph transforms into a skill graph as soon as the abstract concept is instantiated into an implemented technical solution, which can provide numeric performance measures.

Skill limitations may directly be caused by ability restrictions of the system *per se* or by a degraded system performance at the current time.

A skill graph can be used for a *self-representation* of the system. It can be used for online decision support and/or online system operation monitoring. This may entail to monitor the availability of redundant paths, remaining safety margins, or options for behavioral degradation.

Figure 4 shows a derived skill graph for the ability graph from Figure 3. The structure is identical. However, the graph is further detailed and each skill node has skill-specific current performance levels. These levels are given by its performance metrics and each performance impact edge provides weights of subordinate metrics for the above skill node. Additionally, redundancy mechanisms are highlighted (cf. IV-B.3).

The structure of a skill graph is identical to the ability graph it is derived from. In such a graph, nodes and edges are defined as following:

*1) Skill Nodes:* Similar to an ability node, a skill node is characterized by its input requirement dependencies and the capability resulting from it.

Each skill includes one, or more performance metrics, which are used to quantify a current skill performance levels. Such metrics represent a specific property of a skill, e.g. the skill *Accelerate* (S14) in Figure 4 has the properties *maximum possible acceleration* and *reaction time*. The first property represents the current skill to accelerate the vehicle as an upper boundary. The second property represents the time gradient between commanding an acceleration and obtaining a response from the engine. The skills *Control distance* (S1) and *Control speed* (S2) utilize the skill performance levels to calculate derived performance metrics for their particular skills.

*2) Performance Impact Edges:* Edges in a skill graph represent performance impact weights for performance metrics of a subordinate node to the performance level of the superordinate node. An edge is directed from a superordinate node towards a subordinate node. It is characterized by a vector of performance impact factors to weight the impact of different metrics for the superordinate skill.

*3) Redundancy in the Skill Graph:* The skill graph can be used to determine the current availability of different types of redundancy and degradation opportunities in the system. The different redundancy mechanisms can be separated in the groups hardware redundancy, software redundancy, and functional redundancy. Hardware redundancy means that identical hardware devices are installed multiple times and that these can be switched between in case of failure, e.g. Sr16 and Sr17 in Figure 4. This includes of course the software running on the devices. Software redundancy can be diverse and it can be implemented with or without a degradation of functionality. This means that software components are implemented in different ways (diverse), which allows a majority vote for calculation results due to the different approaches. Functional redundancy is always diverse, but can be implemented with (S9 and S10) or without (S5 and S6) degradation of functionality.

## V. Utilizing a Skill Graph

Revisiting the example of the skills *Control distance*, *Control speed* and *Accelerate* from Figure 4: The skill *Accelerate* might have a *Maximum possible acceleration* of currently $0.1\,m/s^2$ as the vehicle might drive close to its maximum speed. The *Dead time* of the system might be $1\,s$. The performance impact edges provide a vector of impact weights, how much the performance metrics *Maximum possible acceleration* and *Dead time* impede the skills *Control distance* and *Control speed*.

The *Maximum possible acceleration* might have little impact on the skill *Control Distance* as it only slightly impedes the control offset of the distance to an ACC target. However, assuming the user has set a target velocity higher than the current speed, it will impact the skill *Control speed* as the vehicle can no longer keep the control offset between the current speed and the target speed within the intended bounds.

### A. Online System Monitoring

The skill graph can serve as a tool for monitoring the execution of the designed function. As described above, each edge in the underlying graph can be weighted with one or multiple metrics for the actual *quality of service* of the respective parent nodes. Starting at the outermost nodes of the modeled graph, the quality of service can be propagated from subordinate nodes to superordinate nodes. Possible degradation of underlying skills always result in a degradation of its corresponding superordinate skills so that changes in the quality of service are always propagated through the skill graph.

Assuming, that the quality of service of each skill can be measured and that a sufficiently detailed mathematical model of the aforementioned propagation of the execution quality through the entire graph is available, the skill graph can be used as a tool for online monitoring of the complete system. Thus, the system can determine at any time how changes in the quality of service of subordinate skills affect the quality of superordinate skills.

In order to simplify the inherent complexity of the propagation of the performance metrics through the graph, the approach taken by Bergmiller [22] is based on fuzzy logic. There the current performance metrics of single skills are fuzzified. With a growing level of abstraction, the fuzzified skill metrics are further combined and further fuzzified. This results in raw estimates of the current skill level of each skill.

With this knowledge, countermeasures against the degradation of skills can be taken by the system. On the one hand, the vehicle could for example trigger fallback strategies to stand-by components in order to avoid a degradation of the overall quality of execution. This would be typical redundancy mechanisms as they are already applied to ensure functional safety in vehicle or avionic systems.

On the other hand, it is imaginable that the same metrics used for monitoring can also serve as an input to optimization algorithms. These algorithms could also use additional information such as the underlying network topology or available

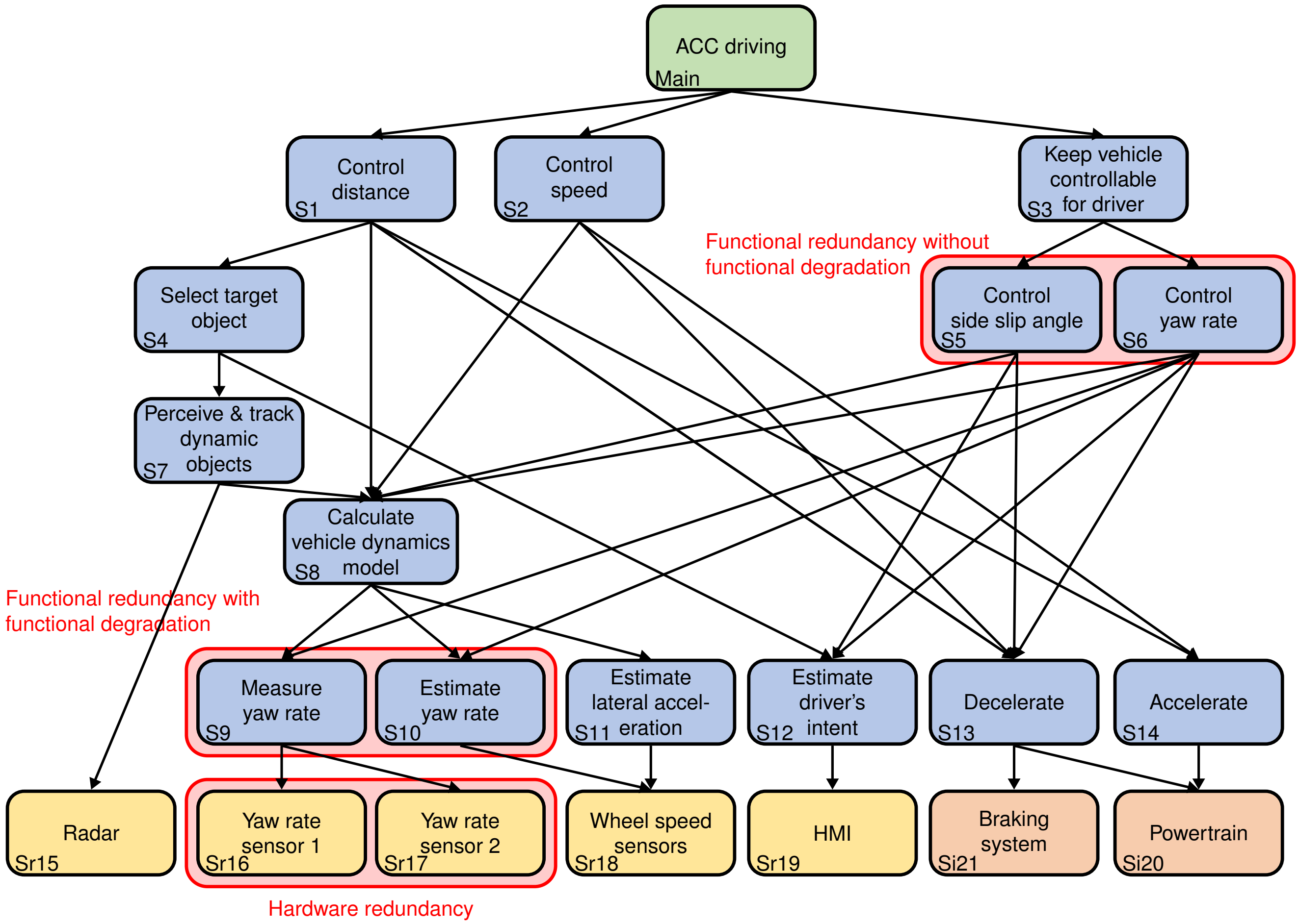


Fig. 4. Further detailed ACC skill graph with redundancies and exemplary metrics. Still with a reduced level of details.

resources as optimization constraints. In case of a detected loss of quality, the system could trigger an optimized reconfiguration of the affected software components in order to keep the quality of service at the highest possible level.

Methods of optimization and reconfiguration are currently under research in the project Controlling Concurrent Change (CCC) [25]. If the above mentioned methods fail or are not available, the vehicle can still adapt its driving strategy in order to being able to operate safely with a reduced quality of service. In this case the skill graph can be considered as a tool for *decision support* for the system.

Besides application-specific metrics, additional information like the skill's calculation time, its current state of redundancy, its current state of degradation, or even the uncertainty of the metrics' values may be propagated through the skill graph. This allows a detailed representation of the current state of the system. The information from each skill is aggregated in the layers of the skill graph, such that superordinate skills have detailed information about their subordinate skills.

### B. Decision Support

The monitoring data available in the skill graph can be used to influence driving decisions. Besides the situational influences to the driving decisions the skill graph integrates the vehicles current performance capabilities. This results in more reliable and thus safer driving decisions.

In decision making, multiple inputs are used and processed to determine possible action and to select the best solution. These decisions have to be taken under high uncertainties due to limitations in the environment perception, the traffic participant's intent estimation, and uncertainties resulting from the propagation of performance metrics through the skill graph. For instance, such decision making for executing lane changes with a basic consideration of current skill levels is described in [35].

## VI. Conclusions and Outlook

The concept of abilities and skills and the skill graph for technical systems was introduced by Maurer, Pellkofer and Siedersberger and further extended by Bergmiller. It allows modeling of the driving tasks of driver assistance systems and vehicle guidance systems for every automation level of the SAE definitions. In this paper, we presented how the approach can be used for several tasks concerning vehicle automation. Similar approaches to model human-like behavior in technical systems have been introduced in pen and paper and computer games. The skill graph modeled in the concept phase can be used for online monitoring of the system and the resulting data can be used to improve decision taking in vehicle guidance systems. The main challenges for future usage will be the identification of abilities and skills, their dependencies, and the necessary metrics. These challenges are currently part of our research in the Stadtpilot project [3], the development of an unmanned safeguarding vehicle for highway construction sites in the aFAS project [24] and the Controlling Concurrent Change project [25].

In future work, the approach will be further detailed and its applicability to other domains like robotics will be investigated. As the approach is generic and can be applied to every technical system which imitates human behavior, a generic implementation in most-used programming languages and for most-used platforms seems useful.

## ACKNOWLEDGMENT

We would like to thank our colleagues in the projects Stadtpilot, aFAS, CCC, and MOBILE for their support of our research. The aFAS project was made possible by the German Federal Ministry of Economic and Affairs and Energy (BMWi). The CCC project was made possible by the German Research Foundation (DFG).

## REFERENCES

[1] Google Inc. (2014) Google Self-Driving Car Project. [Online]. Available: https://plus.google.com/111118414189048552116/posts/9WBWP2E4GDu

[2] J. Ziegler, P. Bender, H. Lategahn, M. Schreiber, T. Strauß, and C. Stiller, "Kartengestütztes automatisiertes Fahren auf der Bertha-Benz-Route von Mannheim nach Pforzheim," in *Workshop Fahrerassistenzsysteme*, Walting, Germany, 2014, pp. 79–93.

[3] T. Nothdurft, P. Hecker, S. Ohl, F. Saust, M. Maurer, A. Reschka, and J. R. Böhmer, "Stadtpilot: First fully autonomous test drives in urban traffic," in *2011 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Washington DC, USA, 2011, pp. 919–924.

[4] J. M. Wille, F. Saust, and M. Maurer, "Stadtpilot: Driving autonomously on Braunschweig's inner ring road," in *2010 IEEE Intelligent Vehicles Symposium (IV)*, San Diego, CA, USA, 2010, pp. 506–511.

[5] S. Ohl, "Fusion von Umfeld wahrnehmenden Sensoren in städtischer Umgebung," Ph.D. dissertation, Technische Universität Braunschweig, 2014.

[6] SAE, "Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems," Society of Automotive Engineers (SAE), 2014.

[7] T. M. Gasser, C. Arzt, M. Ayoubi, A. Bartels, L. Bürkle, J. Eier, F. Flemisch, T. Hesse, W. Huber, C. Lotz, M. Maurer, S. Ruth-Schumacher, J. Schwarz, and W. Vogt, *Rechtsfolgen zunehmender Fahrzeugautomatisierung : gemeinsamer Schlussbericht der Projektgruppe*. Wirtschaftsverlag NW, Verlag für neue Wissenschaft, 2012.

[8] E. Donges, "Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen," *Automobilindustrie*, vol. 27, no. 2, pp. 183–190, 1982.

[9] ——, "Fahrerverhaltensmodelle," in *Handbuch Fahrerassistenzsysteme*, H. Winner, S. Hakuli, and G. Wolf, Eds. Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH, 2012, pp. 15–23.

[10] B. Abendroth and R. Bruder, "Die Leistungsfähigkeit des Menschen für die Fahrzeugführung," in *Handbuch Fahrerassistenzsysteme*, H. Winner, S. Hakuli, and G. Wolf, Eds. Vieweg+Teubner Verlag, 2012, pp. 4–14.

[11] "Straßenverkehrs-Ordnung," Bundesministerium für Verkehr und digitale Infrastruktur (BMVI), Berlin, Germany, 2015.

[12] J. C. Gerdes, "Personal communication," Braunschweig, Germany, 2014.

[13] M. Maurer, B. Lenz, J. C. Gerdes, and H. Winner, Eds., *Projekt Villa Ladenburg "Autonomes Fahren"*. Springer-Verlag Berlin Heidelberg, 2015.

[14] P. Bergmiller, M. Maurer, and B. Lichte, "Probabilistic fault detection and handling algorithm for testing stability control systems with a drive-by-wire vehicle," in *2011 IEEE International Symposium on Intelligent Control (ISIC)*, Denver, CO, USA, 2011, pp. 601–606.

[15] L. Bainbridge, "Ironies of Automation," *Automatica*, vol. 19, pp. 775–779, 1983.

[16] R. M. Yerkes and J. D. Dodson, "The relation of strength of stimulus to rapidity of habit-formation," *J. Comp. Neurol. Psychol.*, vol. 18, pp. 459–482, 1908.

[17] *ISO 26262:2011 Road vehicles - Functional safety*, International Organization for Standardization (ISO) Std. ISO 26 262, 2011.

[18] M. Maurer, *Flexible Automatisierung von Straßenfahrzeugen mit Rechnersehen*. VDI-Verlag, 2000.

[19] K.-H. Siedersberger, "Komponenten zur automatischen Fahrzeugführung in sehenden (semi-)autonomen Fahrzeugen," Ph.D. dissertation, Universität der Bundeswehr München, 2003.

[20] M. Pellkofer, "Verhaltensentscheidung für autonome Fahrzeuge mit Blickrichtungssteuerung," Ph.D. dissertation, Universität der Bundeswehr München, 2003.

[21] M. Matarić and F. Michaud, "Behavior-based systems," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer Berlin Heidelberg, 2008, pp. 891–909.

[22] P. Bergmiller, "Towards Functional Safety in Drive-by-Wire Vehicles," Ph.D. dissertation, Technische Universität Braunschweig, 2014.

[23] A. Reschka, J. R. Böhmer, T. Nothdurft, P. Hecker, B. Lichte, and M. Maurer, "A Surveillance and Safety System based on Performance Criteria and Functional Degradation for an Autonomous Vehicle," in *2012 IEEE International Annual Conference on Intelligent Transportation Systems (ITSC)*, Anchorage, AK, USA, 2012, pp. 237–242.

[24] T. Stolte, G. Bagschik, A. Reschka, and M. Maurer, "Automatisch fahrerlos fahrendes Absicherungsfahrzeug für Arbeitsstellen auf Autobahnen (aFAS)," in *Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel 2015 (AAET)*, Braunschweig, Germany, 2015, to be published.

[25] A. Reschka, M. Nolte, T. Stolte, J. Schlatow, R. Ernst, and M. Maurer, "Specifying a middleware for singular or distributed embedded systems for ADAS," in *2014 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Hyderabad, India, 2014, pp. 117–122.

[26] A. Knoll and T. Christaller, "Selbstrepräsentation, Selbstwahrnehmung und Verhaltenssteuerung von Roboterrn," in *Selbstrepräsentation in Natur und Kultur.*, H. J. Sandkühler, Ed. Lang, 2000, pp. 109–132.

[27] A. Knoll, "Autonomie für situierte Robotersysteme - Stand und Entwicklungslinien," 2003, Forschungsbericht der Technischen Fakultät Abteilung Informationstechnik 99-06.

[28] R. Matthaei, "Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten für Fahrerassistenzsysteme und automatisches Fahren in urbaner Umgebung," Ph.D. dissertation, Technische Universität Braunschweig, 2015, to be published.

[29] M. Thuy, M. Goebl, F. Rattei, M. Althoff, F. Obermeier, S. Hawe, R. Nagel, S. Kraus, C. Wang, F. Hecker, M. Russ, M. Schweitzer, F. Leon, G. Färber, M. Buss, K. Diepold, J. Eberspächer, B. Heißing, and H.-J. Wünsche, "Kognitive Automobile - Neue Konzepte und Ideen des Sonderforschungsbereichs/TR28." in *3. Tagung Aktive Sicherheit durch Fahrerassistenz.*, Garching b. München, Germany, 2008.

[30] B. Slavicsek, R. Baker, and M. Mearls, *Dungeons and Dragons For Dummies 4th Edition*. Wiley, 2010.

[31] WoWWiki. (2014) World of Warcraft Wiki - Talent Tree. [Online]. Available: http://www.wowwiki.com/Talent_Tree

[32] 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*. Verlag Hans Huber, 2014.

[33] J. B. Carroll, *Human cognitive abilities: a survey of factor-analytic studies*. Cambridge; New York: Cambridge University Press, 1993.

[34] D. A. Norman and D. G. Bobrow, "On data-limited and resource-limited processes," *Cognitive Psychology*, vol. 7, no. 1, pp. 44 – 64, 1975.

[35] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, The Netherlands, Oct. 2013, pp. 2063–2067.