Conceptual Introduction of required development capabilities for Model-Based Systems Engineering

Marvin M. Schmidt [1], Simon Schmidt [2], Thomas C. Zimmermann [1] and Rainer Stark [3,1]

[1] Fraunhofer Institute for Production Systems and Design Technology IPK, Division Virtual Product Creation, 10587 Berlin, Germany
[2] Volkswagen AG, 38436 Wolfsburg, Germany
[3] Technische Universität Berlin, Institute of Machine-Tools and Factory Management, Department of Industrial Information Technology, 10623 Berlin, Germany
† This paper has been published and presented at 19th Drive Train Technology Conference ATK 2021.

Abstract: In the last years, multiple tools and methodologies have emerged that try to support the engineers to conduct Model Based Systems Engineering (MBSE). Nevertheless, the tools are usually bound to a certain methodology, which often requires sequential steps instead of supporting the iterative character of modern product development. In this paper, we conceptually introduce the development capabilities of the 5D model for MBSE as a convenient way to cover all relevant aspects of MBSE in an iterative way while being open to methodology and tool choice.

1. Introduction

The potential of Model Based Systems Engineering (MBSE) is evaluated at multiple companies and domains. Prostep ivip and the Association for Standardization of Automation and Measuring Systems (ASAM) e.V. announced a deeper cooperation regarding the application of MBSE for the development of SAE International level 4 and 5 automated driving systems. But MBSE is not yet harmonized [1] and thus multiple forms of working in MBSE exist and are used for the development of complex systems. In this paper, we would like to introduce the development capabilities of the 5D model for MBSE as a methodology- and tool-independent approach to conduct and master MBSE on the example of an automated driving system with the purpose of merging from standstill into traffic.

2. State of the Art

Automated driving systems support the driver by taking over longitudinal and lateral control of the vehicle to a defined scope. The degree of automation is divided into five levels according to SAE J3016 [2], whereby level 0 means no automation (driver only) and level 5 is autonomous operation (no human driver required). From level 3 onwards, the system takes over the vehicle guidance completely in certain situations and for a limited time. This results in higher safety requirements because malfunctioning behavior in the system can lead to critical situations and accidents. Automated driving systems are subjected to extensive and rigorous testing due to safety requirements, which increase with progressing levels of automation [3]. Safety is an important criterion in the development of automated driving systems [4] and has to be proven for a socially accepted market introduction. How safety is measured has not yet been determined, but approaches comparing the statistical performance of human drivers to automated driving systems exist [5]. The behavior of automated driving systems is influenced by perception of environment. This perception is realized by environmental sensors (e.g. radar sensors, lidar sensors, cameras, ultrasonic sensors) and is subjected to many influences from the vehicle environment like measurement-specific effects or weather conditions. The use of environment simulations enables tests to be carried out in a virtual vehicle environment during the system development.

In the view of automotive product development, the V-model is often applied. It helps to structure the development, but two basic difficulties remain. On the one hand, using V-model like shown in literature, requirements must be known as completely as possible at the beginning of the development to enable linear procedure. In reality, the development of automated driving systems is characterized by an incremental and iterative approach. On the other hand, the development is interdisciplinary and distributed, with many domains involved.

So a single V-model would not be enough to cover different strands of development. Different reference models try to solve these problems (V-model XT, Spiral Model, Systematic Design Model etc.). On the process level, agile frameworks are sometimes used as addition to these approaches to address the iterative nature of the development, e.g. as hybrid process of scrum and V-model [6]. As an example of an macroscopic reference model with iterative character, the Systematic Design Model for Automated Driving Functions [7] from Graubohm et al. is introduced briefly.

The Systematic Design Model, which is depicted in Figure 1, has an iterative character expressed by an inner and outer cycle. The inner one is performed mostly theoretically to derive a mature concept or an early virtual prototype and the outer cycle for the long-term technical development leading to a market-ready system. Starting point and first step of the Concept phase is an identified customer need for driver assistance. From use cases an item definition is derived, which specifies a functional description of the system. Next, a hazard analysis and risk assessment (HARA) leads to safety goals and an Automotive Safety Integrity Level (ASIL) rating to specify a functional safety concept. Based on the item definition and safety requirements human factors are taken into account, to make the system controllable and user transparent. The last step of the Concept phase is considering marketing aspects. As result of the Concept phase, a functional concept of the system is specified. An Item refinement mechanism is installed to go back to the initial step and repeat the inner cycle and parts of the outer cycle until the specification of the functional concept is complete. During the System design phase a technical solution of the functional concept is developed. In the first step, the functional system architecture is specified. A following system analysis provides a technical safety concept based on the previous safety goals. As result, a system design specification is prepared to be implemented. The requirements for hardware and software are realized in the Domain-specific design phase. Subsequently the results of the Domain-specific design are integrated to the overall system. In the Testing and validation phase, the integration process is checked and the overall system is validated in reference to the specified use cases.

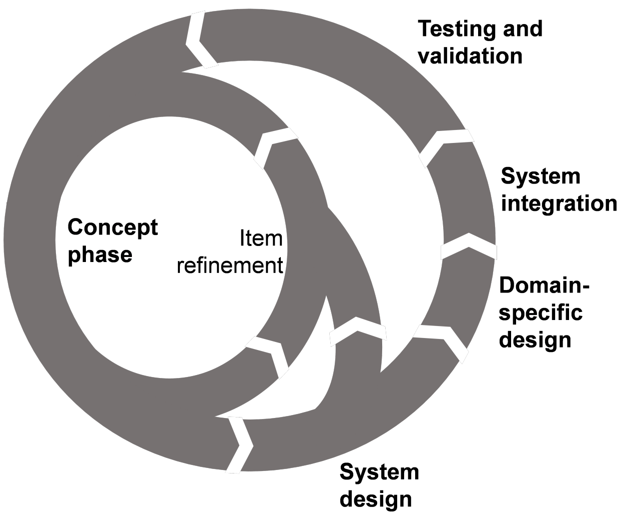

Figure 1: Systematic Design Model for Automated Driving Functions of Graubohm et al. [7]

While developing with such a development model as the one from Graubohm et al. the inherent characteristic of the developed system has to be considered. As depicted in Figure 2, Tomiyama et. al. classified modern systems as either mechatronic products, intelligent mechatronic products, cyber-physical systems (CPS) or smart products [8]. While mechatronic and intelligent mechatronic products rely on control of the mechanical components, CPS additionally are capable of communicating with other CPS or agents on the internet. Smart products gain additional functionalities by integrated internet based services.

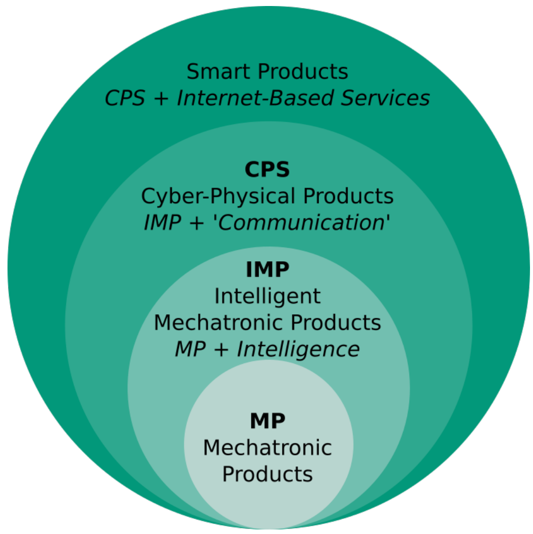

Figure 2: Product classification of Tomiyama al. [8]

The aforementioned automated driving systems can therefore be classified as either CPS or smart product, depending on how its functionality is achieved. We define the automated driving as a CPS in this context, as it is not mandatory to use internet-based services for its functionality, and while being the foundation for smart products, their specifications can be applied to smart products as well. With smart products being an enhanced form of CPS, we will only address CPS in this paper, even though the mentioned points can be mapped to smart products as well.

As could be seen from the described requirements for the perception, there is a great degree of connectivity as well as a large amount of information to be taken into consideration for automated driving systems. Tomiyama et. al. [8] mentioned the tools and methods of MBSE as one feasible approach to handle this complexity. Vettermann [10] supports the view that the development of autonomous driving systems requires virtual validation and verification of the whole system and mentions MBSE as a possible approach to reach that goal. This can be transferred to automated driving functionalities as well.

As most widely used definition, INCOSE defines MBSE as "[...] the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" [11]. Huldt and Stenius [1] mentioned, that this definition is open to interpretations, as MBSE has not yet been internationally standardized. The core of it is the usage of a model as "[. . .] the focal point for systems engineering activities and data management" [1].

Brusa et. al. [12] listed methodology, language, tools and data management as the four main aspects considered in MBSE. Vaneman and Carlson [13] rather focus on model-based processes, modeling language, presentation framework as well as the structure and mention that current tools are only taking parts of these aspects into consideration. Regarding the survey of Huldt and Stenius [1] the current methodologies (covering both processes and methods), notations and tools are already mature to support systems engineering activities, while on the other hand stating that the MBSE application is yet immature. In conclusion, these aspects cannot be the only relevant parts. MBSE is currently too young and unexplored [1] to be easily implemented in industry.

Independent from the previous aspects, Stark and Auricht [9] presented the MBSE development capabilities that have to be considered and mastered to efficiently conduct a MBSE approach in the 5D model of MBSE, which is shown in Figure 3. Zimmermann et. al. [14] investigated it further regarding the development of digital twins. The development capabilities include:

- **Systems Environment Analytics (SEA)** covering the definition of system boundaries and the interaction of the system of interest with the elements outside of it,
- **Systems Definition and Derivation (SDD)** comprising the creation of relevant models and describing it from different viewpoints,
- **Systems Interaction Modeling (SIM)** focusing on modeling and simulating the behavior of the system,
- **Systems Lifecycle Engineering (SLE)** for capturing and managing the system states throughout the whole system lifecycle and the
- **MBSE Capability and Maturation Matrix (CMM)** addressing the acquisition and mastering of the MBSE competencies.

These development capabilities are independent from the previous mentioned main elements language, tool, methodology, structure and presentation framework and thus the user is free to choose the tools or methodologies that best fit his preferences while still being able to master MBSE. The application of these development capabilities shall be shown on a more visual use case presented in the following section.

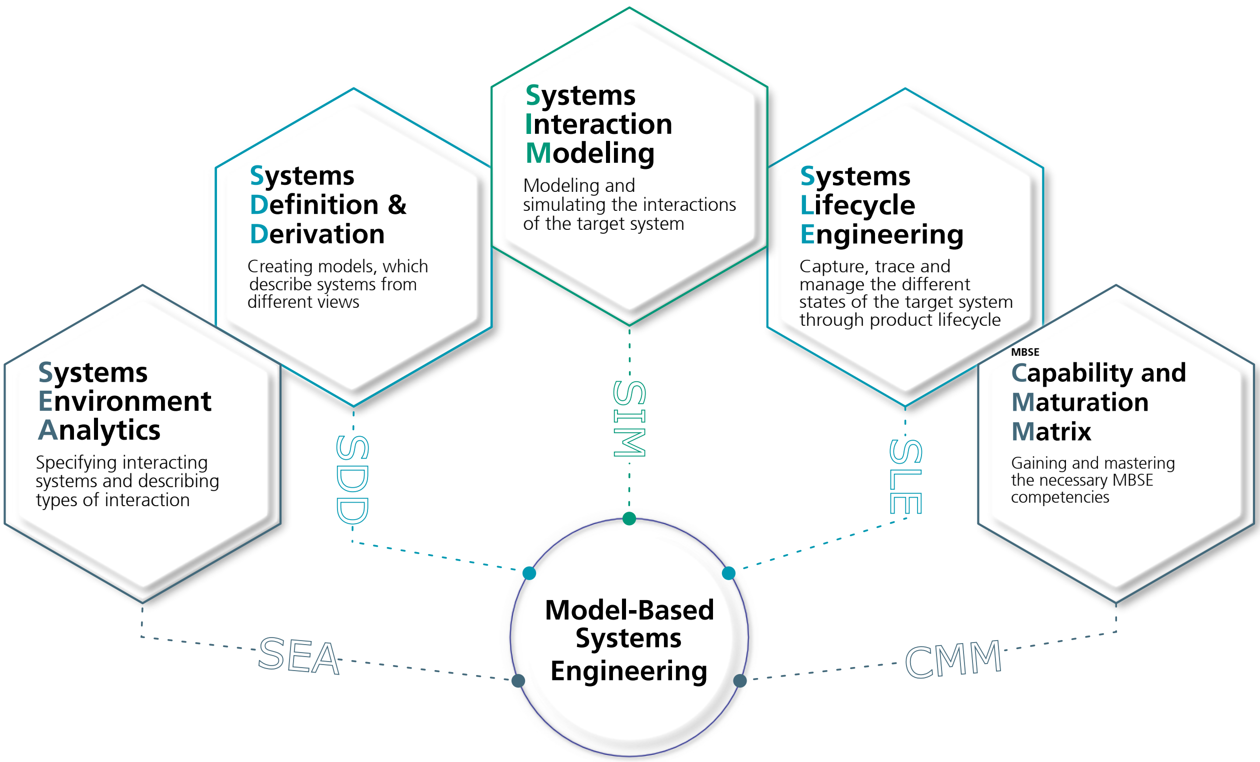

Figure 3: MBSE development capabilities based on the 5D model presented by Stark and Auricht [9]

3. Exemplary development of an automated driving sytem

In this section an exemplary automated driving system is developed according to the Systematic Design Model presented from Graubohm et al. [7]. The system is to take over the merging into the flowing traffic from a standstill. The use case in which the system has to support the driver is shown in Figure 4.

For system development, the Concept phase is executed first. Many accidents happen when merging into flowing traffic [15]. In order to support the driver, the system should take over the merging process. Thus, a customer need is identified, serving the aim to improve safety and decrease the number of accidents in situations the system takes over control. From this a functional description is developed. As initial condition, the system can only be activated when the car is at a standstill. The driver has to decide first in which direction to merge. Then the system checks whether merging is possible and no obstacles are in the route.When all conditions are met, the system starts driving onto the street and merges into the right lane. It accelerates to the target speed or adjusts the distance to the vehicle in front, if necessary. Based on this functional description a safety analysis is performed. Here critical situations are identified and an appropriate system behavior is determined for them. Obvious examples are not recognized obstacles (vehicles, cyclists, pedestrians) or an unauthorized use of the system in disallowed situations.

Laying out the system behavior, during the System design phase, a technical solution of the functional concept has to be specified. The environment perception of the system could be realized using radar sensors to detect moving obstacles like cars, cyclists or pedestrians. Radar sensors are suitable for distance measurements and can be used to monitor the flowing traffic. Cameras can be used to recognize lane makings and observe relevant traffic signs, since image processing is very powerful when classifying objects. The driver can activate the system via a touch display in the vehicle. In addition, the current system status and the environmental perception can be visualized there in order to make the system behavior transparent and understandable. Finally, the system needs interfaces to the steering and the powertrain in order to move the vehicle. After the components and technologies involved have been determined, the Domain-specific design begins. Here the requirements for software and hardware are broken down, implemented and tested in detail. Then the components are at first integrated into the system architecture and then step by step to an overall system, the vehicle. Extensive Testing and validation of the overall system follows to complete the development.

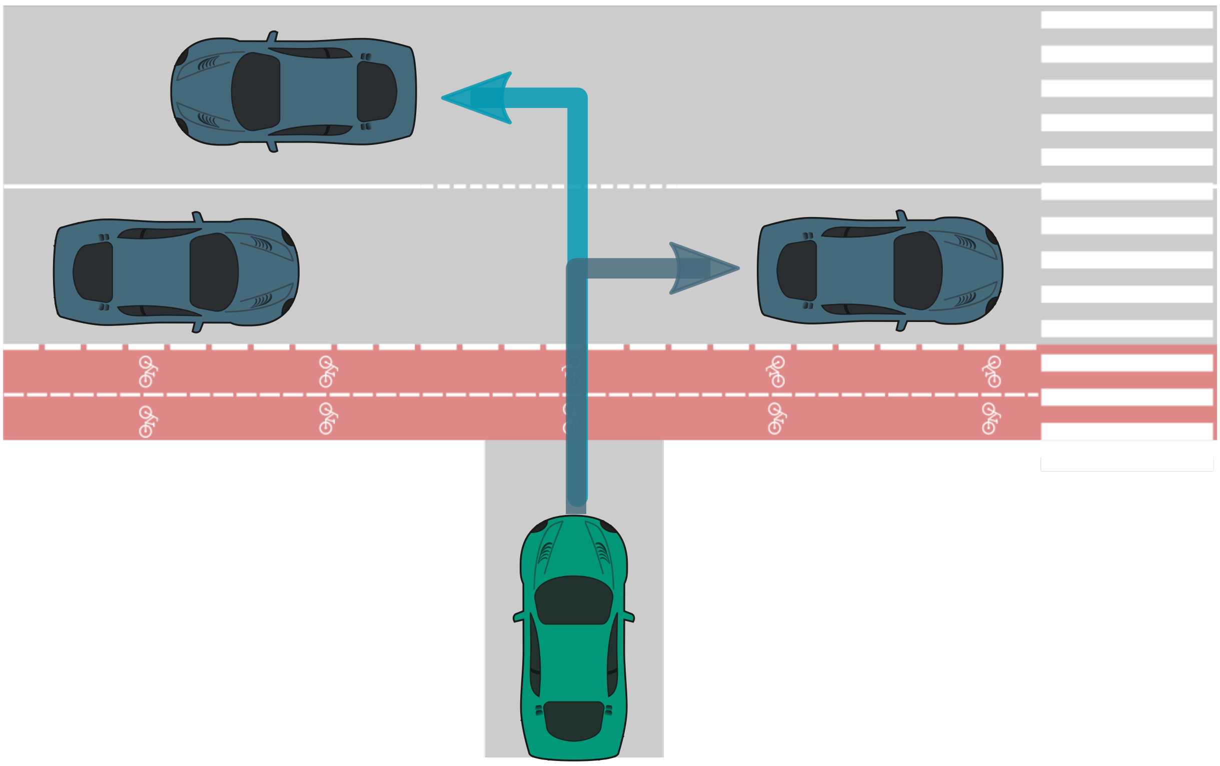

Figure 4: Use Case for an automated driving system: Merging into flowing traffic.

4. Conceptual application of the five development capabilities of MBSE

In the previous section, an exemplary automated driving system in development has been presented. The steps of the presentation were oriented on the development approach of Graubohm et. al. [7]. In this section, the five development capabilities of MBSE shall be applied on this example, to better understand their content and importance.

4.1. Systems Environment Analytics (SEA)

In the SEA development capability the context of the system of interest has to be specified in sense of customer requirements, system boundaries and thus interacting systems and the types of interaction. In most systems engineering methodologies, these are among the first activities that have to be conducted to develop a complex system.

The development capabilities can be characterized rather as a discipline and not as an activity, which shows it is nothing one does once. In development models like Graubohm et. al. [7] this is tried to respect by integrating iterative processes, but this would evoke an endless loop, as the environment might always change and thus a step back to the environmental analysis might be necessary. As a discipline, you should always keep the system environment analytics in mind when developing a system. The usage of agile frameworks as iterative approaches leverages the need for these continuous consideration even more. Even though, the focus of this paper shall be placed on the macroscopic development scope and thus be oriented on the example from Graubohm et. al. [7].

When taking a look at the customer and technical system requirements (supporting the merging process to enhance safety and thus decrease the number of accidents), the environment of the system has to be derived. With complex systems most of the time consisting of multiple subsystems and thus often being systems of systems, the analysis has to be done on various levels. On the vehicle level (product), the analysis would for example identify pedestrians, additional vehicles, the street, trees, animals, traffic signs as well as the driver of the vehicle. On the system level enabling automated driving the braking system, powertrain and human-machine-interfaces (HMIs) like a steering wheel might come up.

For all of the identified external systems, the interaction has to be defined. Obstacles (e.g. animals, trees and pedestrians) have to be avoided, traffic signs and additional vehicles have to be analyzed and this analysis has to be used to adapt the driving profile (as well as avoided).

It is critical to maintain a meaningful separation between the technical system which is under development focus (also referred as *target system*) and those other systems which are in the environment of the target system. Only then can the sum of all internal components of a system that are bound by defined requirements and unambiguously verifiable against an established set of requirements be established. Within the SEA it is therefore important to distinguish the target system outbound direction (which active influence to the outside system environment takes place) and the system environment inbound direction towards the target system (which influence does the system environment have on the target system). The language how to describe influence in more generic terms than the software development oriented use cases needs to be further researched on.

Any other details like which system elements, which type of behavior or logic etc. might not be of interest in SEA but will be intensified with the development capability System Definition and Derivation (SDD) and then further on with the development capability System Interaction Modeling (SIM).

4.2. Systems Definition and Derivation (SDD)

The SDD development capability enables complete modeling coverage of all system elements of the target system. In various methodologies, this involves deriving more and more detailed model representations of the systems of interest starting at a higher level moving down to subsystem and component level where the design processes often transition into domain-specific methods and tools. The sequential character of these methodologies should be applied to all activities of this development capability, as it supports a more structured approach to model the system. A possible approach might be the following:

1. **Definition of the target system:**
   The core purpose of the target system has to be defined (often referred to as *mission statement*) as well as the elements that influence and are influenced by the target system. The SEA development capability can be used as an impactful input in this step. Applied on the example of the automated diving function, the basic function is the merging into the flowing traffic and influencing as well as influenced element are the HMI and the powertrain.

2. **Description of system architecture:**
   Based on the previous step, the system architecture can be basically described. The actual realization of this step depends on the chosen methodology and framework. Feasible approaches can be a functional decomposition followed by the definition of a logical architecture (RFLP-approach) or the direct definition of solution elements based on the definition of the target system. A possible branch for the functional decomposition of the automated driving system might be 'merge into traffic' → 'check if merg-

ing is possible' → 'check for obstacles' → 'scan environment'. A direct definition of a solution element might be a camera, since cameras might already be available and be part of other vehicle systems.

3. **System Derivation:**
   If the solution elements could not be directly determined in the description of the system architecture, they have to be derived from the combination of target system and system architecture. For the elements that cannot be derived directly, technical or economic assumptions have to be made for a further system synthesis. Here, a close collaboration of various disciplines is required to make these assumptions. For the automated driving system, such assumptions might be the maximum cost for a sensor as well as the minimum field of view of all sensors combined.

4.3. Systems Interaction Modeling (SIM)

The models defined with the SDD development capability can be perfectly used for descriptive tasks but lack one feature to be suitable for an optimal decision making: they are not executable by default. They describe the structure of the system in sense of system architecture, functional architecture or context models, but not its interactions, e.g. the behavior of the system.

Here, the SIM development capability comes into consideration. With this development capability, the interaction of the target system with other systems as well as the interaction inside the target system is modeled and simulated where required. Interaction in this scope can be either in logical form, e.g. the comparison of opportunities one system element can offer based on the SDD development capability with demands to fulfill the target systems functionality in respect to the system environment, or in form of generic messages, states or full behavior descriptions.

With respect to the automated driving system, a functional SDD output has been 'merge into traffic'. SEA showed that vehicles, pedestrians and further obstacles have to be avoided when the merging process is started. The elements of the system architecture like optical and radar sensors then have to be checked for their interaction with these possible obstacles. When the high-level interaction modes are determined, the behavior of the system might be modeled and simulated as well. The information generated as output of the simulation can then be used for engineering decisions, uncertainty reduction and improvement of system quality [1].

4.4. Systems Lifecycle Engineering (SLE)

Lifecycle Engineering (LCE) in general is a sustainability-focused approach that focuses on the impact of the system on its environment and not the system itself. SLE means to address that topic by investigating the system itself throughout its entire lifecycle. This comprises the capability to capture, trace and manage the different states the target system enters and is in in its lifecycle. This goes beyond current Product Lifecycle Management (PLM) approaches that lack some functionalities to fully support Systems Engineering [16].

When the system is already produced and in use, this development capability strongly overlaps with current approaches for digital twins. Zimmermann et. al. therefore investigated the MBSE development capabilities for the development of digital twins [14] and saw the focus mainly on managing the parameters. The capturing and tracing is not investigated in detail as the digital twin, as a digital representation of the active system, is already meant to capture and trace these parameters. In the development phase this capturing in tracing correlates with the SIM development capability, as the interaction of the system influences the states of the system.

The example of the activities conducted for the development of the automated driving system with the SIM development capability may have shown that the user shall interact with the system to activate and deactivate it. Thus, possible states for further productive use are 'active' and 'inactive'. These states are especially important for later over-the-air (OTA) update purposes, as the system should not be updated while actively in use as this may produce malfunction. During development of the system, the current development progress should be captured and managed as well. System elements like the radar sensors might be in testing, calibration phase or already validated. Interacting elements like the braking system might have similar states that are traced to each other. If for example the braking system is in a state that is refusing its use, this state should be traced to the depending systems as the automated driving system.

4.5. MBSE Capability and Maturtion Matrix (CMM)

While most methodologies and even the previous development capabilities focus on the technological aspects of MBSE, the human as an integral part of the development has to be considered. Albers and Lohmeyer already addressed this topic in 2012 with the Advanced Systems Engineering (ASE) [17]. Huldt and Stenius additionally mentioned the MBSE learning curve as one major obstacle for an efficient MBSE implementation [1]. The MBSE Capability and Maturation Matrix (CMM) tries to cope with these topics.

The core of the CMM development capability is the gaining and mastering of the necessary MBSE competencies. This includes training and educating the stakeholders included in the development of the system of interest especially in the previously mentioned development capabilities. For methodologies, Albers and Lohmeyer [17] mentioned Performance, Presentation and Process as the most vital aspects of Human-Centered Systems Engineering. This can be mapped on the development capabilities as well. The Performance in sense of 'what does the user get out of the knowledge in this development capability?', Presentation as an easy human-capability-interface in sense of trainings and continued information material and Process in sense of the organizations processes that should in best case be tailored to the consideration of these development capability are vital parts of CMM.

In the exemplary context of the development of the automated driving system, the contribution of various domains such as mechanical engineering, electrical engineering or software engineering supports the requirement for a common sense of the developed system and the development approaches of the various domains. Mastering competencies in MBSE enables more cooperatively working in development capabilities like SIM and thus getting more qualitative results. Additionally, the customer as a vital part of the development has to be included here as well and forms of conveying information to the customer have to be considered.

To deliver the best training and education possibilities, the current state of knowledge has to be tracked. Frameworks like The Open Group Architecture Framework (TOGAF) or Department of Defense Architecture Framework (DoDAF) already consider forms of defining and reviewing capabilities and may be mapped to MBSE competencies. The MBSE Capability and Maturity Model of Siemens PLM and Fraunhofer IPK [18] is another approach to rate and enhance the capabilities in MBSE relevant processes and model-based working.

5. Conclusion

In this paper the 5D-model for MBSE has been introduced on the example of an automated driving system. It has been shown, that under consideration of the five development capabilities MBSE can be conducted and mastered with the development methodologies and tools of choice, while still considering all important aspects.

There is still further research required on this topic. In upcoming work we will investigate the direct comparison of the application of MBSE with existing development approaches and the benefits and efforts from and for the application of these development capabilities. Additionally, the model types that are defined and used within the development capabilities and their overlap are of interest to us. These insights can be used to improve and refine the presented 5D-model further and thus leverage the application of MBSE in industry.

References

[1] T. Huldt and I. Stenius. "State-of-practice survey of model-based systems engineering". In: *Systems Engineering* 22.2 (Sept. 2018), pp. 134–145. DOI: 10.1002/sys.21466.

[2] Society of Automotive Engineers (SAE). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.* 2018.

[3] Nidhi Kalra and Susan M. Paddock. "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" In: *Transportation Research Part A: Policy and Practice* 94 (Dec. 2016), pp. 182–193. DOI: 10.1016/j.tra.2016.09.010.

[4] ISO 26262. *Road vehicles — Functional safety.* International Organization for Standardization, 2011.

[5] Walther Wachenfeld and Hermann Winner. "Die Freigabe des autonomen Fahrens". In: *Autonomes Fahren.* Springer Berlin Heidelberg, 2015, pp. 439–464. DOI: 10.1007/978-3-662-45854-9_21.

[6] Otto Emmerich et al. "A Systems Engineering Framework and Application to an Open Automated Driving Platform". In: *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society.* IEEE, Oct. 2020. DOI: 10.1109/iecon43393.2020.9255208.

[7] Robert Graubohm et al. "Systematic Design of Automated Driving Functions Considering Functional Safety Aspects". In: *8. Tagung Fahrerassitenz.* Munich, Germany, 2017.

[8] Tetsuo Tomiyama et al. "Development capabilities for smart products". In: *CIRP Annals* (June 18, 2019). DOI: 10.1016/j.cirp.2019.05.010.

[9] Rainer Stark and Maik Auricht. "Beschleunigung der Validierung innerhalb der Entwicklung von mechatronischen Systemen mit einer integrierten Plattform". In: *ProSTEP Symposium 2017, Essen.* 2017.

[10] Steven Vettermann. "MBSE praktisch nutzen". In: *ProduktDaten Journal 2* (2018).

[11] INCOSE Technical Operations. *Systems Engineering Vision 2020.* INCOSE-TP-2004-004-02. Seattle, WA, 2007.

[12] Eugenio Brusa, Ambra Calà, and Davide Ferretto. *Systems Engineering and Its Application to Industrial Product Development.* Springer-Verlag GmbH, Apr. 13, 2018.

[13] Warren K. Vaneman and Ronald Carlson. “Model-Based Systems Engineering Implementation Considerations”. In: *2019 IEEE International Systems Conference (SysCon)*. IEEE, Apr. 2019. DOI: 10.1109/syscon.2019.8836888.

[14] Thomas C. Zimmermann, Christian Masuhr, and Rainer Stark. “MBSE-Entwicklungsfähigkeit für Digitale Zwillinge”. In: *ZWF* 115.special (Apr. 2020), pp. 51–54. DOI: 10.3139/104.112312.

[15] Statistisches Budensamt. *Verkehr - Verkehrsunfälle, Fachserie 8 Reihe 7*. 2018.

[16] Rainer Stark and Ernst-Eckart Schulze. “Need and potentials to improve Systems Engineering by future PLM Solutions”. prostep ivip Symposium 2010. Berlin, Ludwig Erhard Haus, Apr. 28, 2010.

[17] Albert Albers and Quentin Lohmeyer. “Advanced systems engineering - Towards a model-based and human-centered methodology”. In: *Tools and methods of competitive engineering: Proceedings of the ninth International Symposium on Tools and Methods of Competitive Engineering (TMCE 2012), Karlsruhe, Germany, May 7-11, 2012. Ed.: I. Horváth*. Ed. by I. Horvárt. Karlsruhe, Germany, May 2012, pp. 407–416.

[18] Robert Woll and Atakan Sünnetcioglu. “Reifegradsystem für Entwickler: Siemens PLM Software und Fraunhofer IPK ebnen Weg für ”Modellbasiertes Systems Engineering””. In: *interface. Aktuelles, Anwenderberichte und Fachartikel von Siemens PLM Software* 1 (2017), pp. 32–33.