Customer-centric and function-oriented development of mechatronic systems

T. Zerwas[1] · **G. Jacobs[1]** · **L. Brand[1]** · **S. Dehn[1]** · **K. Spütz[1]** · **G. Höpfner[1]** · **C. Matz[2]** · **C. Guist[2]** · **J. Berroth[1]** · **C. Konrad[1]** · **J. Kohl[2]**

Abstract

Successful products at least precisely meet the customers' expectations and, in the best case, exceed them. To develop successful products, customer expectations must be translated into requirements. With the increasing functionalities of products in recent years, the customers' expectations regarding product interaction and its behavior in different environmental conditions have also become more extensive. Current approaches of model-based systems engineering (MBSE) enable developing complex mechatronic products seamlessly from requirements to functions and solutions on a parameter level. However, there is a lack of approaches that systematically translate complex customer expectations into functional and design requirements as a starting point for further development.

In this contribution we present a method and a corresponding meta-model that allows to systematically formalize the dependencies of different stakeholders and their expectations as well as different environmental conditions and constraints. From these dependencies, operating states are elicited that represent a set of simultaneously valid stakeholder expectations with their corresponding constraints. From these operating states, functional and design requirements are systematically derived as a basis for the model-based design of the system under development. Our meta-model is compatible to the established modeling language SysML, thus, existing approaches for the function-oriented model-based system development can benefit directly from these formally modeled requirements.

Our publication signposts the potential for systematic and formal translation of customer expectations into operating states as well as requirements and thus enables a targeted, customer-centric and function-oriented development of mechatronic systems. We applied our method in an interdisciplinary, industrial project using the example of a thermal management system of a battery electric vehicle.

✉ T. Zerwas
  post@imse.rwth-aachen.de

1 Institute for Machine Elements and Systems
   Engineering, RWTH Aachen University,
   Eilfschornsteinstraße 18, 52062 Aachen, Germany

2 BMW Group AG, Petuelring 130, 80788 Munich, Germany

Kundenzentrierte und funktionsorientierte Entwicklung mechatronischer Systeme

Zusammenfassung

Erfolgreiche Produkte erfüllen die Erwartungen der Kunden und übertreffen sie im besten Fall. Um erfolgreiche Produkte zu entwickeln, müssen die Kundenerwartungen in Anforderungen umgesetzt werden. Mit der zunehmenden Funktionalität von Produkten in den letzten Jahren sind auch die Erwartungen der Kunden an ihre Interaktion mit dem Produkt und dessen Verhalten in unterschiedlichen Umgebungsbedingungen umfangreicher geworden. Aktuelle Ansätze des Model-based Systems Engineering (MBSE) ermöglichen es, komplexe mechatronische Produkte nahtlos von den Anforderungen zu den Funktionen und Lösungen bis auf Parameterebene zu entwickeln. Es fehlt jedoch an Ansätzen, die komplexe Kundenerwartungen systematisch in funktionale und gestalterische Anforderungen als Ausgangspunkt für die weitere Entwicklung übersetzen.

In diesem Beitrag stellen wir eine Methode und ein entsprechendes Metamodell vor, das es erlaubt, die Abhängigkeiten zwischen verschiedenen Stakeholdern und deren Erwartungen sowie unterschiedlichen Umweltbedingungen und Randbedingungen systematisch zu formalisieren. Aus diesen Abhängigkeiten werden Betriebszustände abgeleitet, die eine Menge gleichzeitig gültiger Stakeholder-Erwartungen mit den entsprechenden Einschränkungen darstellen. Aus diesen Betriebszuständen werden systematisch funktionale und gestalterische Anforderungen abgeleitet, die die Grundlage für den modellbasierten Entwurf des zu entwickelnden Systems bilden. Das Metamodell ist kompatibel zu der etablierten Modellierungssprache SysML, so dass bestehende Ansätze zur funktionsorientierten modellbasierten Systementwicklung direkt von diesen formal modellierten Anforderungen profitieren können.

Diese Veröffentlichung zeigt das Potenzial für eine systematische und formale Übersetzung von Kundenerwartungen in Betriebszustände sowie Anforderungen auf und ermöglicht so eine zielgerichtete, kundenzentrierte und funktionsorientierte Entwicklung mechatronischer Systeme. Die Methode wurde in einem interdisziplinären, industriellen Projekt am Beispiel des Thermomanagementsystems eines batterieelektrischen Fahrzeugs angewendet.
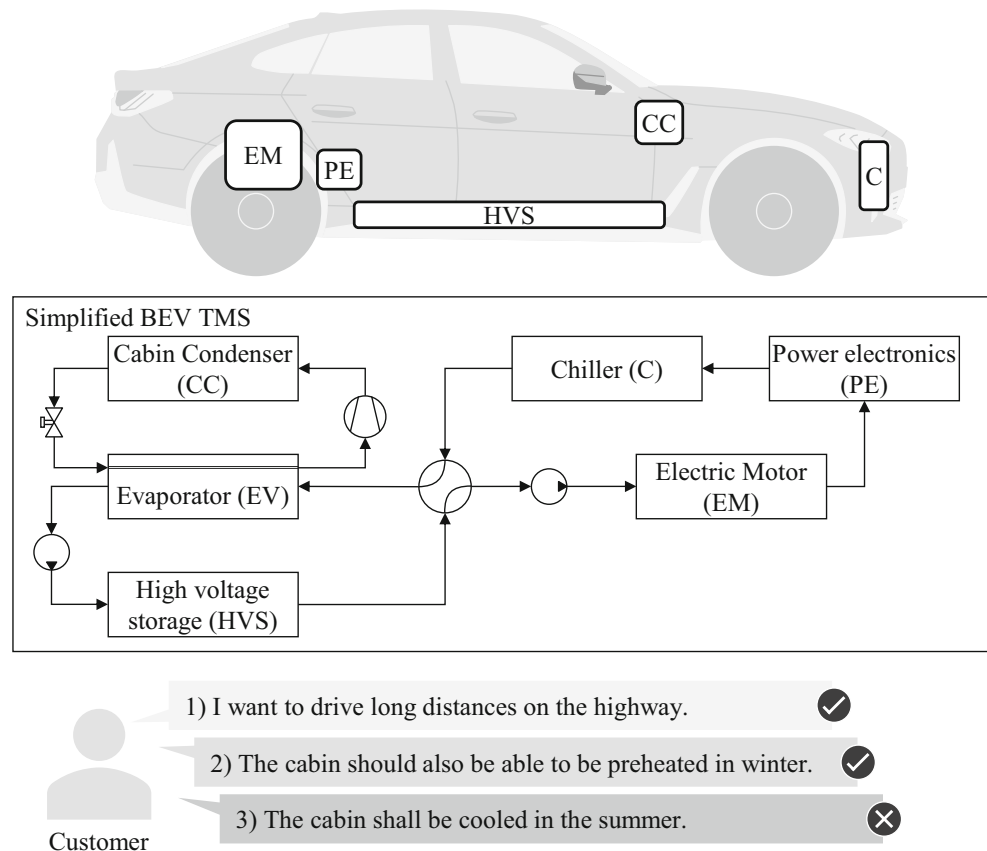
1 Introduction

The key objective of product development is to precisely transfer the customers' expectations into a product that reliably fulfills these expectations. To do this, the first step is to fully elicit and formalize the customers' expectations and to transform them without losses into requirements and specifications for the following development of functions, solutions and components [1]. Any (aspect of a) customer expectation that is not elicited or translated into requirements is at risk of not being considered in subsequent development resulting in a suboptimal product. On the one hand, the growing international competition and the need to develop more energy-efficient products increase the necessity to fulfill customer expectations as precisely and loss-free as possible and to avoid over-specification. On the other hand, the increased complexity and networking of mechatronic products with many operating states [2, 3] makes it increasingly difficult for developers to elicit and formalize requirements completely and adequately. Typically, stakeholders and customers have different expectations of the system to be developed. However, these expectations and the associated requirements typically do not always have to be met, but only those that are simultaneously present in an operating state. Therefore, in this paper an operating state is defined as a set of simultaneously relevant expectations of stakeholders and context systems as well as their associated requirements.

A striking example of complex mechatronic products with many operating states are thermal management systems (TMS) of battery electric vehicles (BEVs), which have become the focus of research due to political objectives for greenhouse gas reduction (Fig. 1; [4, 5]). TMS are characterized by the customers' expectation for a comfortable way of traveling in different driving situations. In terms of comfort, the customer expects, among other things, that the cabin is always at a pleasant temperature, i.e. independent of the ambient temperature and the respective driving situation (e.g. city or highway). Due to the shift from internal combustion engines to more efficient, electric drive trains, one challenge within BEV TMS is the insufficient usable heat within the BEV drive train for optimal temperature conditioning of the cabin in multiple usage states [6].

For example, in winter during slow city driving, not enough waste heat is generated in the drive train to heat the cabin comfortably [7]. In this case, the thermal power for the cabin must be generated using the stored electrical energy of the battery. If the heat is generated from electrical energy by resistive heating elements, it can reduce the BEV driving range by up to 40% [8–10] and thus negatively impact one of the most important purchase criteria from the customers' point of view [11]. For this reason, many BEV TMS now have heat pumps that can provide the necessary heat much more efficiently using ambient heat [9].

This example shows that *operating states* (e.g. city driving in winter) are characterized by three dimensions: the

Fig. 1 Simplified example of a battery electric vehicle thermal management system with customer expectations that are partially not satisfied

1) I want to drive long distances on the highway. ✓

2) The cabin should also be able to be preheated in winter. ✓

3) The cabin shall be cooled in the summer. ✗

Customer

stakeholder expectation of the customer (e.g. comfortable city driving), the environmental state (e.g. winter 0 °C) and the context system behavior (e.g. drive train does not generate enough waste heat). *Context systems* are systems (or subsystems thereof) that are not part of the system of interest but interact functionally with it and/or impose requirements on the SOI. The combinatorics of stakeholder expectations, environmental states and context system behaviors results typically in a high number of operating states, which need to be considered in design and test of mechatronic systems. Due to stakeholder expectations and context system behavior TMS operating states are characterized by heat demands and heat surpluses of the individual components. These demands and surpluses need to be served as energy-efficiently as possible in specific environmental states.

Analyzing the state of research, it is important to distinguish between elicitation and formalization of information. While *elicitation* refers to the act of obtaining an information in terms of its content regardless of its form, the term *formalization* describes the transformation process of an existing and unchanged content into a certain form. So far, the literature has mainly focused on approaches for the elicitation and formalization of stakeholder expectations, environmental conditions and context system requirements. However, there is no modeling method that allows the systematic elicitation and formalization of operating states and

their resulting requirements based on stakeholder expectations, environmental conditions and context system requirements.

Therefore, the contribution of this paper is a MBSE modeling method, that enables

- the formalization of stakeholder expectations, environmental states and context systems,
- the systematic elicitation of relevant operating states, and
- the derivation of requirements from operating states for the customer-centric and function-oriented design of mechatronic systems.

This contribution builds on the premise of a complete elicitation of stakeholder expectations, environmental conditions and context system requirements and starts with their formalization.

The paper is structured as follows: Chap. 2 includes a detailed description of the problem resulting in six challenges. Chapter 3 provides an overview of the state of research on identifying and modeling operating states. Chapter 4 defines the research question and hypotheses of this paper. Chapter 5 presents a modeling method for the customer-centric and function-oriented elicitation and formalization of operating states for the development of mechatronic products and illustrates the application of the method using a BEV

TMS as an example. The results are discussed and concluded in Chap. 6 and an outlook is given.

2 Problem statement

BEV TMS are characterized by a multitude of component specific heat demands and heat surpluses depending on environmental conditions, context systems as well as the customer behavior. Therefore, distributing heat from components with cooling demand to components with heat demand and generating lacking thermal power as efficiently as possible is one key challenge in the design of BEV TMS [7]. To enable this efficient heat distribution in all relevant operating states, modern TMS are highly integrated, mechatronic systems that connect heat exchangers of each context system to be tempered (Fig. 1). The context systems primarily include the vehicle cabin as well as the drive train consisting of the high-voltage storage (HVS), the power electronics (PE) and the electric motor (EM). Thermal energy is transported between the heat exchangers of these context systems by coolant or refrigerant of the system of interest (TMS) via various pipes, valves and pumps as well as other supporting components (e.g. reservoir, accumulator). Controllable elements such as valves and pumps allow the TMS to adapt its system architecture and behavior to specific operating states in order to provide the best possible heat distribution. If certain switching conditions, components, coolants, or pipes are missing or incorrectly designed, the TMS will not be able to create a requirement-fulfilling behavior in certain operating states. The simplified BEV TMS depicted in Fig. 1 can for instance fulfill customer expectation 2 by preheating the cabin with ambient heat from the chiller via the heat pump (evaporator, compressor, condenser, expansion valve). Customer expectation 2, on the other hand, is not fulfilled, since the TMS shown does not provide a technical solution for absorbing heat from the cabin and dissipating it to the environment.

The optimal system architecture depends strongly on the operating states [7, 12]. This is why TMS can rarely be reused in their entirety and almost always have to be developed anew for each vehicle project, which is also reflected in the high variance of TMS on the market [13]. The example of the TMS shows that it is inherently important for the design of integrated and efficient mechatronic systems to know all operating states [14]. To elicit all operating states, especially the stakeholder expectations from the customers representing the product usage must first be known (*challenge 1*). Since all mechatronic products are in use somewhere in the world after their manufacture and are subject to the laws of nature [14], it is important to capture all relevant environmental conditions (*challenge 2*). In addition, mechatronic systems are becoming increasingly physically

interconnected with other (sub) systems in their context [3]. Therefore, these context systems, including their influence on the system of interest (SOI), must also be fully considered (*challenge 3*). As shown in the example, it is not sufficient to design mechatronic systems by treating the three dimensions separately—the simultaneous consideration of stakeholder expectations, environmental state and context system behavior is decisive. Therefore, an operating state describes multiple stakeholder expectations and context system requirements as well as one environmental state. For the precise design of mechatronic systems all relevant operating states must be elicited [12] (*challenge 4*).

In addition to the premise that operating states must be completely known, it is also necessary that operating states are comprehensible for humans, processable for machines and compatible to development artefacts of subsequent development stages. Due to the size, complexity, and interacting domains of mechanical, electrical, and software, TMS are increasingly being developed using Model-based Systems Engineering (MBSE) approaches. MBSE describes the formalized modeling of technical systems to support system requirements, design, analysis, verification, and validation activities in all lifecycle phases [15].

In order to be able to leverage the operating states for the model-based design and test of mechatronic systems with common methods concrete functional and design requirements must be derived [16]. In particular, the simultaneous validity of requirements must be considered. For example, a BEV TMS must be designed for simultaneous cooling of all drive components during a summer highway trip, but never for simultaneous defrosting of the windshield and cooling of the cabin. Therefore, when deriving requirements from operating states, it is critical to determine which requirements really need to be met simultaneously and which do not, in order to avoid over-specification (*challenge 5*).

Finally, the potential of MBSE methods for the integrated simulatability of a system architecture can only be utilized if all development artifacts and models are seamlessly connected via uniform parametric interfaces. Therefore, it is essential that stakeholder expectations, environmental states and context systems as well as the elicited operating states and requirements are available in a formalization that is perfectly aligned with the corresponding MBSE modeling method (*challenge 6*). Up to now, the operating states as well as the requirements are partly stored in documents [6], so that they are not directly usable for MBSE.

3 State of research

In the problem statement (cf. Chapter 2), six challenges were derived that arise in the development of mechatronic systems with multiple operational states. In the following,

Table 1 Review of the state of the art

Challenge		[17, 18]	[19]	[20, 21]	[1, 22, 23]	[24, 25]
1	Representation of stakeholder expectation (e.g. customer)	●	○	●	●	●
2	Representation of environmental conditions	○	●	●	○	●
3	Representation of context system requirements	◐	●	●	●	●
4	Complete derivation and formalization of operating states	○	○	○	○	◐
5	Derivation and formalization of requirements for the system design	○	○	○	○	◐
6	Formalization of operating states must be compatible with model-based system design	●	●	●	○	●

relevant publications from the state of the art are presented that at least partially address these challenges and provide methods for requirements elicitation for model-based system design.

Czriharz et al. and Wölkl et al. propose methods for requirements elicitation involving relevant stakeholders and to some extent context systems [17, 18]. In the contribution of Gausemeier, the influences of the environment and context systems on the developed system are considered extensively, but stakeholder expectations and use cases are not considered. Furthermore, the environment is not represented by a standardized variable set that can be instantiated as specific environmental states [19]. Pohl et al. and Weilkiens et al. both present a detailed approach for modeling stakeholders, context systems and use cases using SysML [20, 21]. However, both concepts are rather designed for software products and only partially transferable to mechatronic products. Glinz et al., Bender et al. and Robertson et al. contribute methods and procedures for identifying potential stakeholders and context systems as well as provide a guide for requirements elicitation [1, 22, 23]. Though, environmental influences are not included in the process. In addition, the processes presented are only partially suitable for model-based design of systems. The approaches of Lamm and Zingel [24, 25] completely include the representation of stakeholder expectations, environmental conditions and context systems in a model-based framework. But Lamm sets the main focus after determining the requirements on linking them to the functional level [25]. Zingel represents operating states as test cases without detailed explanation of their derivation and composition [24].

Overall, no approach can be identified, that systematically composes operating states and derives verifiable requirements for model-based system design, thus fulfilling all challenges derived in the introduction (cf. Chapter 1; Table 1).

Challenge 6 states that the approach to formalize operating states must be compatible with an MBSE development method, since one of the goals is to use the operating states for the simulation of mechatronic systems. According to the analysis in [26], the motego method is characterized by a deep parametric integration of simulation models into system models [16, 27]. Therefore, the motego method is selected as a foundation for the meta-model elaborated in this contribution.

Nowadays, information on operating states is mainly found in company-specific guidelines or technical standards [6]. Standards typically describe specific test procedures that must be followed for safety reasons, such as defrosting and demisting of windscreens [28]. Customer expectations regarding their mobility behavior are at most implicitly included, so it is difficult to create a genuine added value for the customer. As a result, customer expectations concerning the mobility behavior, environmental states and context system behavior are in some cases only partially taken into account [6, 29]. It is to be expected that the increasing use of sensors in vehicles and data collection by manufacturers will provide a better picture of operating states in the future. However, it should be noted that the usage data collected does not necessarily provide a clear indication of the customers' original expectations (e.g., if the customer prefers to use his second car with an internal combustion engine in winter and leaves the BEV at home).

4 Research question and hypothesis

In the problem statement, six current challenges in the model-based and function-oriented development of mechatronic products on the basis of customer-centric operating states were derived (cf. Chapter 2). While the state of research provides multiple solutions for the formalization of stakeholder expectations, environmental conditions and

context system requirements (challenge 1–3) the complete derivation of operating states and corresponding requirements from this information (challenge 4–5) remains unsolved (cf. Chapter 2). Therefore, we first elaborate a suitable formalization for challenge 1–3, which is the basis for overcoming challenge 4–6 as a focus of this publication. Hence, the research question of this publication is:

How can operating states and requirements be elicited in a systematic and customer-centric manner as well as formalized for function-oriented, model-based system design?

To answer this question, three research hypotheses are formulated:

1. Operating states of mechatronic systems can be systematically derived from the combination of stakeholder expectations, environmental states and context systems.
2. Operating states of mechatronic systems can be composed into sequences of operating states that represent mobility behavior.
3. Relevant requirements for the system design can be derived from the dependencies of the stakeholder expectations, environmental states and context systems occurring simultaneously within an operating state.

5 Method and meta-model for the determination of operating states for mechatronic products

In this chapter a method is presented, that enables the systematic derivation of customer-centric operating states for the model-based development of mechatronic systems from stakeholder and context system requirements. In parallel, a meta-model is introduced that supports the application of the method through a suitable formalization. The method consists of four main steps (Fig. 2) which are explained in the following subsections. The method begins with the formalization of use cases as well as their derived requirements from stakeholders and context systems. In the next step, logical relations between the use cases are specified in order to elicit all possible operating states. In the third method step, environmental states relevant for the SOI are elicited and formalized with their respective variables. The final step involves sequencing operating states, defining a specific en-

vironmental state and deriving emerging requirements. For each method step we describe the methodological procedure and the formalization of the development artifacts in the meta-model as well as illustrate the application using the example of a simplified BEV TMS.

5.1 Definition of the system of interest as well as its stakeholders and context systems

Method The method begins with the definition of the *system of interest* (SOI), for whose development relevant operating states and requirements are to be determined. The SOI can be a completely newly developed system or a newly developed subsystem within a surrounding *super system*. In the latter case, the super system and the context systems interacting with the SOI are also defined at the beginning. Super system, SOI and context systems are all represented as functions according to the function-oriented development [16]. In the next step, all potential stakeholde*r*s that can influence the SOI or that are influenced by the SOI are identified. Here, *stakeholders* are understood to include all (groups of) natural and legal persons as well as relevant organizations in contrast to context systems representing all technical systems that interact physically or logically with the SOI. The approaches of [1, 22, 23] can provide assistance in the identification of stakeholders and in the derivation of use cases. Subsequently, use cases regarding the SOI are defined for all stakeholders and context systems and broken down into further sub use cases when necessary, as in [17, 25]. From these use cases, *functional requirements* (FR) are derived, which describe the functional behavior that needs to be implemented to realize these use cases [16, 26]. In order to further specify the parametric boundary conditions for the implementation of the functionality *design requirements* (DR) can refine the respective FRs. This results in a set of FRs and DRs for each stakeholder and context system. Further information regarding the formulation of requirements and different attributes describing requirements can be found in [18, 21, 23].

Meta-model To formalize and link the development artifacts of the first method step, the following meta-model is proposed (Fig. 3). In order for the modeling method devel-

Fig. 2 Overview of the method steps

1	Definition of the system of interest as well as its stakeholders and context systems
2	Systematic derivation of operating states
3	Definition of environmental states and variables
4	Creation of relevant sequences of operating states and derivation of new requirements

Fig. 3 Meta-model for the definition of the SOI with its stakeholders and context systems (cf. Chapter 5.1) and for the combination of operating states (cf. Chapter 5.2)
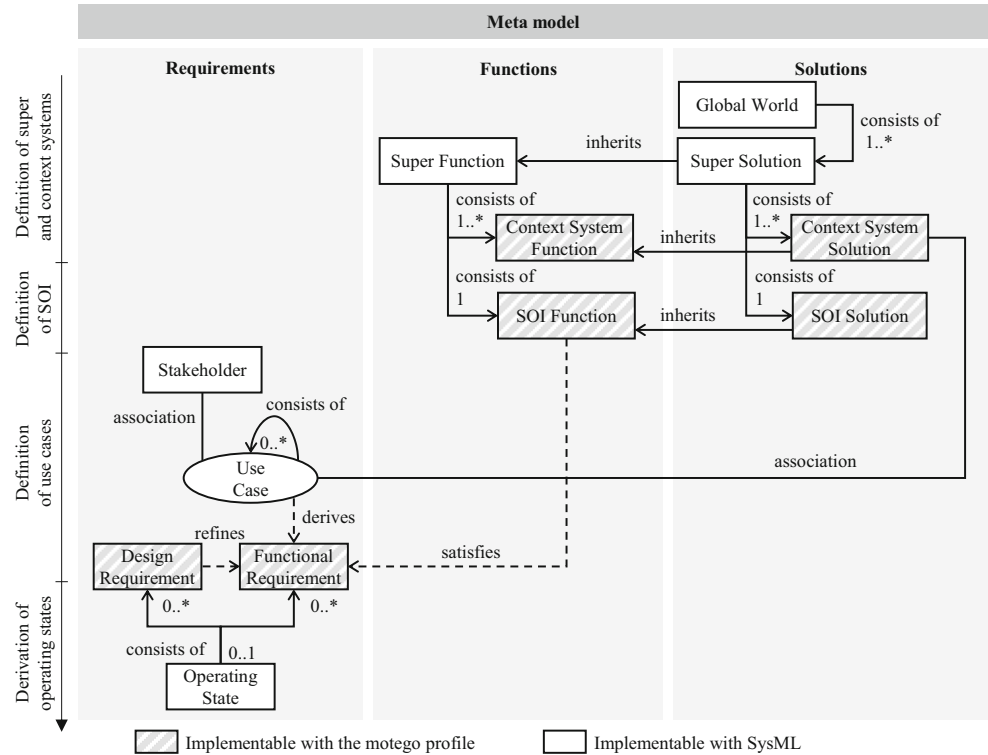
oped here to be a meaningful addition to the motego profile [27], the meta-model consists of already existing or compatible language elements and relationships. Artifacts that can be implemented with existing language elements of the motego profile are shown shaded. Artifacts that require an extension of motego, but are implementable with SysML, are drawn with white filling. The stakeholders and use cases are modeled at the requirements level. Use cases can be derived from both stakeholders and context systems and are treated in the same way. Functional requirements that are derived from use cases are satisfied by the SOI function or its decomposed sub functions [16, 26]. This further functional decomposition of the SOI as well as the associated technical solutions are deliberately not shown here, as they will be developed based on operating states and requirements resulting from the method described here.

Exemplary application Fig. 4 depicts the application of the method based on the meta-model to the exemplary BEV TMS. The customer as most important stakeholder is shown here with three use cases in relation to the SOI, representing "driving", "charging", and "parking" the vehicle. The driving use case is refined into two further use cases, between which only one can be selected at a time. Therefore, it is formalized as an abstract element, which is illustrated in italics. Several FRs for "country" and "city trip" can be derived from these further use cases as well as DRs that refine the FRs more specific, like for example the DR for the maximum speed. A possible implementation of FRs is

shown on the right side of Fig. 4 and contains attributes to describe the requirement textually, as well as its source and corresponding use case.

After this method step, all stakeholders and context systems are modeled systematically, and relevant use cases with FRs and DRs are derived.

5.2 Systematic derivation of operating states

Method One challenge in the development of mechatronic systems is the collection and consideration of simultaneously relevant requirements (cf. Chapter 1). Therefore, the FRs and DRs previously derived from the use cases of the stakeholders and context systems are combined into so-called *operating states* (OS). An operating state is defined as a set of use cases and FRs of the stakeholders and context systems that can be required simultaneously by the SOI. Since several use cases can exist simultaneously for each stakeholder (e.g. parking and charging), the operating state can contain several FRs for each stakeholder.

As a basis for the elicitation of operating states the relationships between the previously collected use cases are modeled first. Using the UML relationships *include, extend* and the *generalization* relationship, the logical dependencies between all use cases can be formalized in the use case diagram so that all possible operating states can be derived. If a use case is selected as active and has an outgoing include relationship, the use case at the other end of the relationship arrow must also be selected. If an extend
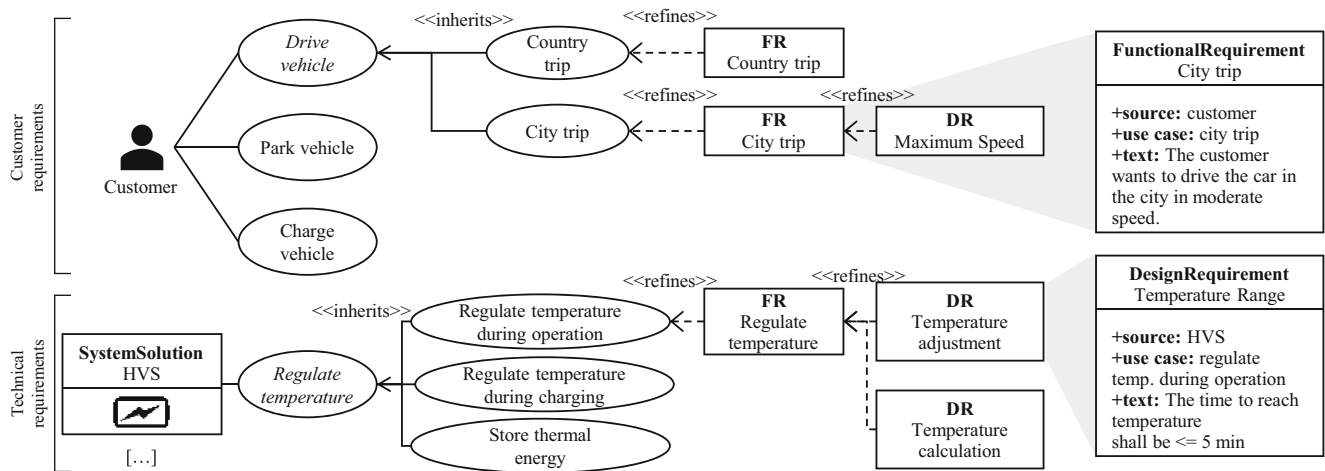
Fig. 4 Example use cases and requirements of stakeholders and context systems

arrow points to an active use case, it is optional whether the use case at the beginning of the extend arrow is also active. An extension point can be used to automatically activate an extend relationship as soon as the condition of the extension point is met [18, 20]. As explained in Chap. 5.1, all FRs derived from a certain use case are associated with the respective OS as soon as the particular use case is selected as active for the OS.

In addition to the FRs, their optional DRs are associated with the operating state. In this way, all simultaneously available use cases and requirements of all stakeholders and context systems can be logically combined into an operating state and made usable for further development. The SOI can therefore only be in a single OS at any given time.

It is crucial for the system design to elicit all relevant operating states (cf. Chapter 1). The difficulty of the method step lies in the definition of all include and extend relationships. This requires additional know-how and a manual work mode. Once these dependencies are all defined, the subsequent derivation of the operating states can easily be automated by external computational engines.

Meta-model The meta-model in Fig. 3 shows the formalization of the new operating state language element that is created in the requirement level. It consists of an arbitrary number of FRs and DRs that are simultaneously valid in this state. Using this assignment, the operating states to be considered can be formed from the finite set of all FRs and DRs relevant for the SOI. If a use case and thus a FR of a stakeholder changes, this change is automatically considered in all operating states that contain these FRs. This eliminates maintenance and adaptation efforts in the current document-based management of operating states [6]. There are several possibilities for implementing the element of operating states based on SysML in the motego profile [27], which will be the subject of future research. An important

aspect is that the operating states can be used as directly as possible and without avoidable modeling effort for the functional validation of solution elements [26, 30].

Exemplary application Fig. 5 illustrates how operating states can be generated from the customer's point of view based on the use cases of all stakeholders and context systems. The operating state shown as an example describes the city trip of a customer at a certain maximum speed. During this city trip the context system HVS requires temperature regulation, because the use case "city trip" includes the use case "regulate temperature during operation" and is therefore mandatory for the creation of the OS. Furthermore, the derived FR and DR specify that the desired temperature shall be reached within five minutes. In addition, an example of an extension point is shown in Fig. 5: If the parking time is longer than 30 min, the use case "store thermal energy" is activated, since it is connected to "park vehicle" via the extend relationship.

By analyzing the use cases and their logical relationships, all operating states can now be elicited. In our example, the customer is modeled in such a way that one customer use case is always active over the entire usage time. Therefore, all operating states can be identified here by systematically going through all include and extend relationships starting from the customer. In this simplified example, this results in six operating states (Fig. 5). For other technical systems, it may be necessary to form the operating states starting from all stakeholders and context systems, e.g., if there is no customer as a human user of the SOI.

After this step, all relevant operating states have been defined by reusing the previously derived use cases and requirements and can be used in Chap. 5.4 to realistically represent the usage behavior with sequences of operating states.
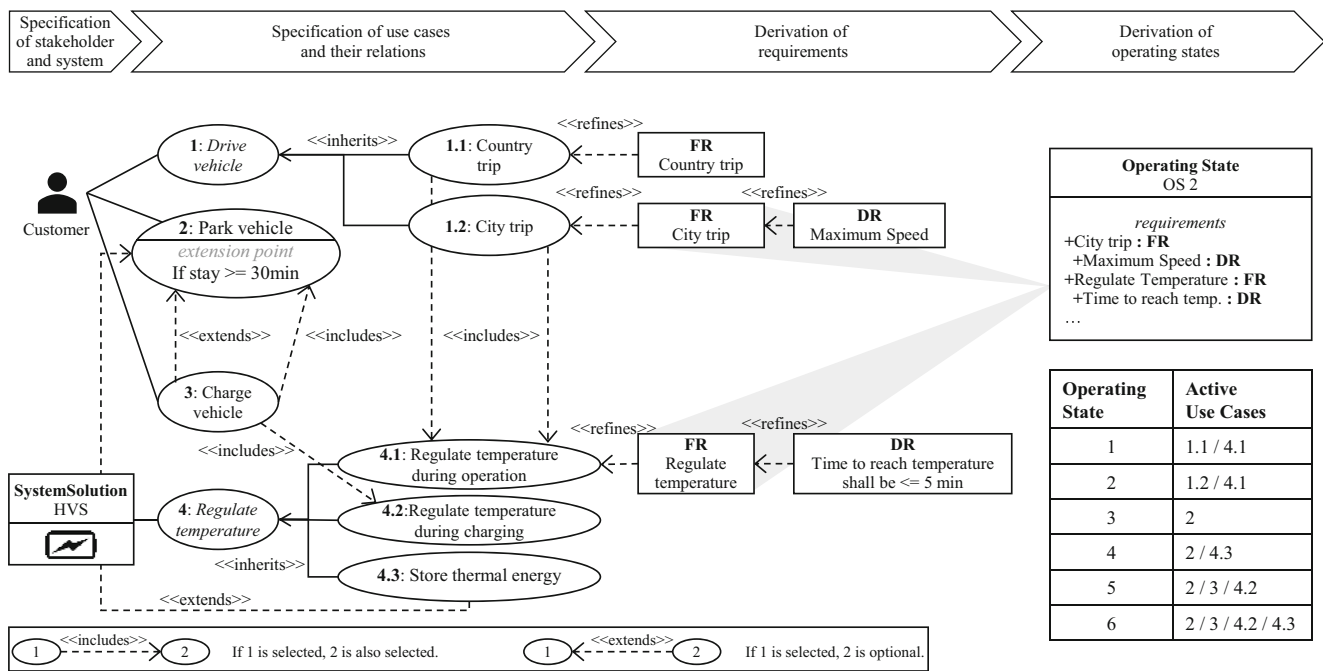
Fig. 5 Example of the procedure of creating operating states from the customer's perspective

5.3 Definition of environmental states and variables

Method The environment is a crucial factor in the design of mechatronic systems, since typically the SOI or context systems interacting with it are also physically dependent on the environment. For example, a BEV TMS dissipates heat to the environment during highway travel in summer, but at the same time needs ambient heat to warm the cabin during travel in winter (cf. Chapter 1). In both cases, the ambient temperature is a physical quantity that is relevant for the interaction and can take different values at one location during the course of the year as well as describe completely different annual curves globally [6, 9]. Therefore, in this step, environmental states relevant for the SOI are systematically formalized, which can later be reused for simulation. Each environmental state is described by a consistent set of physical parameters. The elicitation of relevant environmental conditions can be conducted based on standards and guidelines, expert surveys, specialized catalogues or video data. In case of the thermal management system, for instance, this could be data like driving cycles, weather data or street data [31–33].

For companies such as automotive manufacturers that develop several subsystems of a vehicle in parallel, it can be advantageous to define these environmental states not individually for each subsystem, but for the entire system or even company-wide. This would ensure, in the sense of a single source of truth, that the same environmental states are used for the design of all subsystems. Since in this case a very large number of physical parameters relevant for the overall system is to be expected, it is necessary to define which parameters are relevant for which subsystem. For example, the road surface is initially irrelevant for the development of the TMS, but it is certainly decisive for the development of the chassis.

Meta-model To support the reusable definition of consistent environmental states we propose the meta-model depicted in Fig. 6. We extend the previous meta-model (Fig. 3) by the artifact *environment* constituting the g*lobal world* with the s*uper solution*. The environment is characterized by one to an infinite number of *environmental state variables*. These physical quantities can be characterized by a description, physical quantity, unit, value, or range of validity, analogous to [34]. Creating instances of the environment allows to define specific environmental states with quantified variables that can be passed to simulation models integrated into the system architecture during functional validation according to [16, 35].

Exemplary application The right side of Fig. 6 shows an exemplary modeling of the environmental state "Munich winter" as an instance of the environment. The instance contains a standardized set of environmental state variables, which are specified with values representative for the town Munich in the winter season during the instantiation process, e.g. an ambient temperature of –5 °C or a high traffic level.
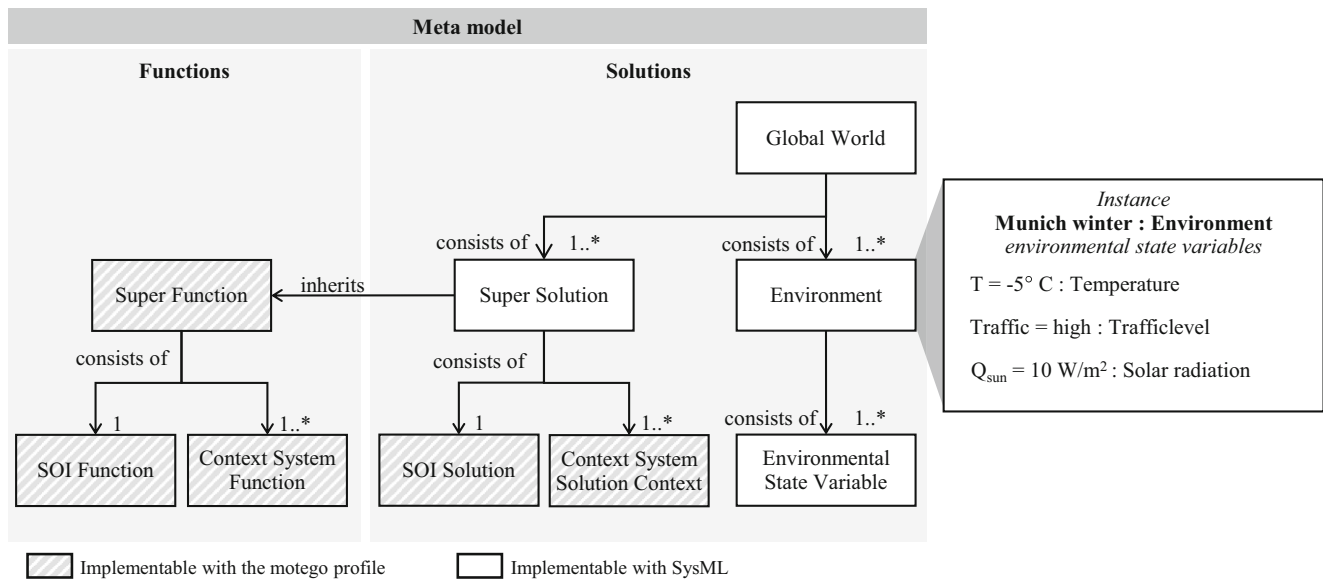
Fig. 6 Meta-model of environment and its state variables as well as an exemplary instance of the environment

Through this methodological step, all environmental states relevant for the SOI can be elicited based on different data sources, but incorporated in a formalized and uniform way. Thus, the environmental states can be efficiently reused in the following step to specify the environment at certain sequences of operating states.

5.4 Creation of relevant sequences of operating states and derivation of new requirements
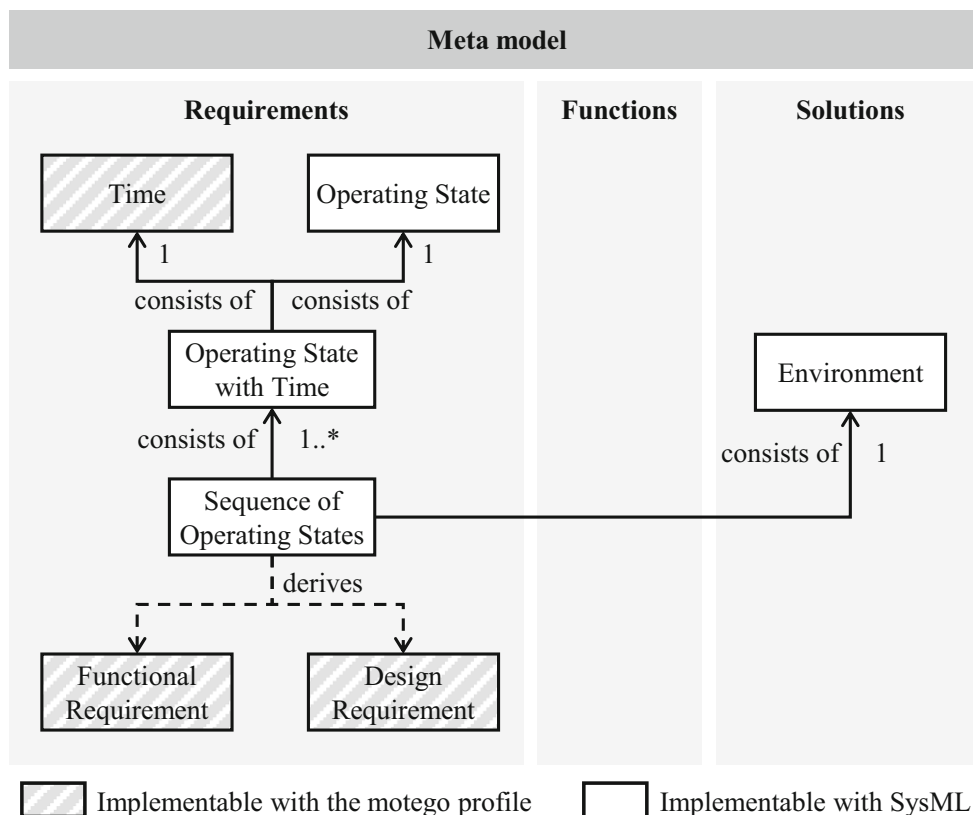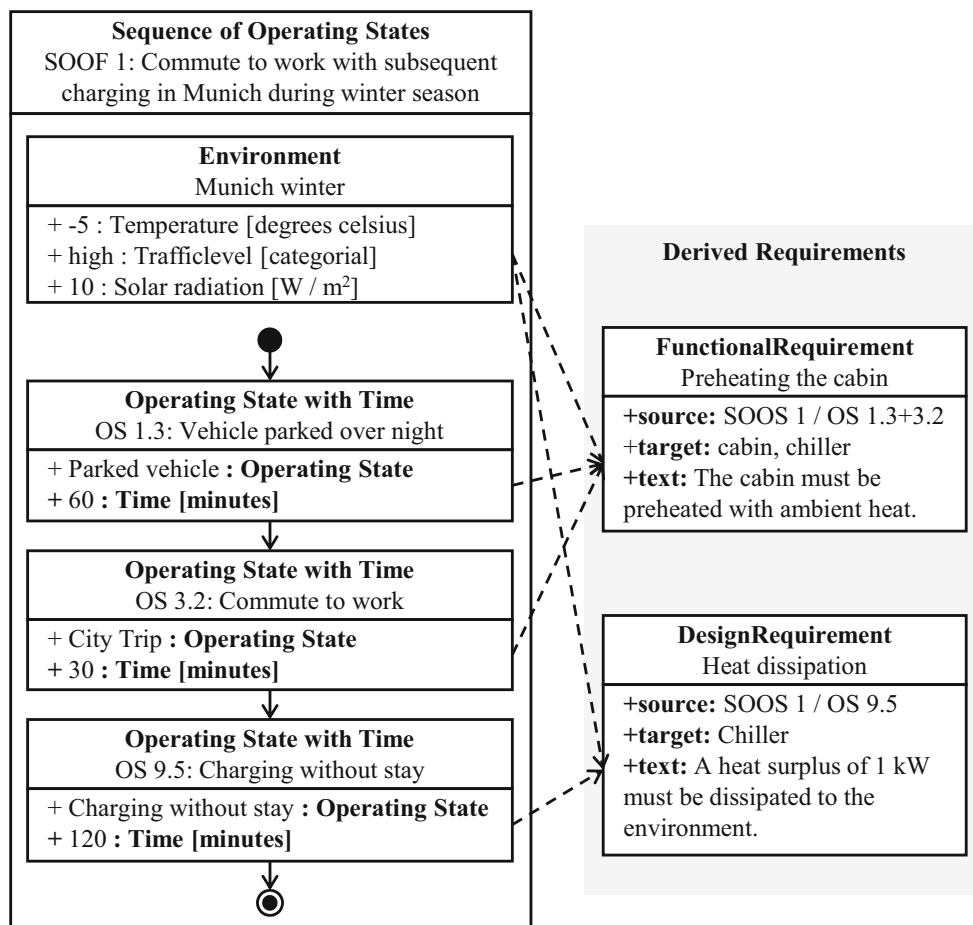
Method Mechatronic systems can increasingly no longer be designed and validated on the basis of stationary extreme cases [6]. For BEV TMS, for example, it is crucial to know how long a car will be parked before it is used again. Only this way, it can be estimated how much heat is lost to the environment via the cabin and whether it is worthwhile to store it. Therefore, it is not sufficient to consider only individual operating states with their simultaneously existing requirements of the stakeholders and the context systems, but their processes in particular. Thus, a *sequence of operating states* (SOOS) defines the consecutive and temporal sequence in which operating states occur with a specific duration and in a specific environmental state. An OS can be reused in several SOOSs and thus occur several times and in different order. To determine relevant sequences of operating states, it is important to use reliable data (Chap. 3). This can include either publicly available sources, such as WLTP cycles [36] or statistics, or usage data collected or purchased by the developing companies [37].

The sequences of operating states determined in this way are the basis for simulating realistic uses of the SOI by the stakeholders, on the basis of which the SOI can be designed and validated. In addition, the sequences of operating states

are the first development artifact that combines the functional requirements of stakeholders and context systems with an environmental state. This combination gives rise to new requirements that can only be derived in this step and provide helpful restrictions for the further development of the SOI.

Meta-model Figure 7 shows the meta-model that we propose to formalize the sequence of operating states. In order to be able to reuse an operating state (e.g. slow city drive) in several sequences of operating states with different durations (e.g. 5 min drive to the bakery, 50 min drive through rush hour traffic), the language element *operating state with time* is inserted here as an intermediate stage. This element adds a time duration to an operating state and is used to define sequences of operating states. The sequence is also linked to an environmental state, which is valid for the entire sequence of operating states. This simplification avoids a high variance, which would result from the complete combination of all operating states with all relevant environmental states. Depending on the developed system it could be beneficial to assign multiple environment instances to an SOOS or to extend the formalization by a time-varying behavior which will be part of our future research. As described above, the definition of a sequence leads to new requirements that can be implemented via existing motego language elements and are linked to the sequence via a derived relationship.

Exemplary application Figure 8 shows an example sequence of operating states defined for our BEV TMS: The customer parks his vehicle at home overnight, drives through wintry city traffic in Munich morning, and then

Fig. 7 Meta model for the sequence of operating states

Meta model

| Requirements | Functions | Solutions |

Time — Operating State

consists of 1 — consists of 1

Operating State with Time

consists of 1..*

Sequence of Operating States

Environment — consists of 1

derives

Functional Requirement — Design Requirement

▨ Implementable with the motego profile   ☐ Implementable with SysML

Fig. 8 Example of a Sequence of Operating States with derived requirements

Sequence of Operating States
SOOF 1: Commute to work with subsequent charging in Munich during winter season

Environment
Munich winter
+ -5 : Temperature [degrees celsius]
+ high : Trafficlevel [categorial]
+ 10 : Solar radiation [W / m^2]

Operating State with Time
OS 1.3: Vehicle parked over night
+ Parked vehicle : **Operating State**
+ 60 : **Time [minutes]**

Operating State with Time
OS 3.2: Commute to work
+ City Trip : **Operating State**
+ 30 : **Time [minutes]**

Operating State with Time
OS 9.5: Charging without stay
+ Charging without stay : **Operating State**
+ 120 : **Time [minutes]**

Derived Requirements

FunctionalRequirement
Preheating the cabin
+**source:** SOOS 1 / OS 1.3+3.2
+**target:** cabin, chiller
+**text:** The cabin must be preheated with ambient heat.

DesignRequirement
Heat dissipation
+**source:** SOOS 1 / OS 9.5
+**target:** Chiller
+**text:** A heat surplus of 1 kW must be dissipated to the environment.

charges the BEV while he is in the office. This sequence was modeled by reusing the previously defined environmental states and operating states, each of which is assigned a specific duration. This allows the same operating state "park vehicle" to be reused in other sequences with a shorter duration or at another location with a different ambient temperature.

The example also shows that new requirements relevant for the development arise through the creation of sequences. The operating state "park vehicle" does not imply alone that the cabin must be pre-tempered, but only if it is followed by a stay in the vehicle. Through the environmental state "Munich winter" with an ambient temperature below the freezing point, the functional requirement can then be further specified that the cabin must be heated, not cooled. Likewise, only the combination of the last operating state "charge vehicle" with the environmental state determines how much thermal power must be dissipated from the TMS in total.

The sequences of operating states formed in this way can be used in their entirety for the model-based test and design of mechatronic systems across all relevant usage scenarios of the stakeholders and environmental states. The derived requirements are the basis for the further detailing of the functional architecture and the selection of solutions according to [16, 26].

6 Conclusion

In this paper, a method and a corresponding meta-model were presented that enable the customer-centric identification and function-oriented formalization of operating states for mechatronic systems. The method is based on the idea that the stakeholders and context systems of the system of interest are first modeled separately with respect to their use cases and requirements in relation to the SOI. Then, the logical dependencies between the use cases are formalized as a foundation for the systematic elicitation of operating states, which are linked to sequences according to relevant usage scenarios with specific durations. These sequences of operating states are extended by a concrete environmental state, which can be selected from a set of previously uniformly modeled states. This linking of stakeholder and context system requirements with environmental states results in new requirements that ensure the fulfilment of customer requirements in the subsequent development and at the same time represent a justified restriction of the solution space.

This contribution marks the first step towards the systematic collection and formalization of operational states. The set of sequences of operating states generated in this way contains relevant information for designing and function-

ally validating complex mechatronic systems with multiple operational states (such as thermal management systems for battery-electric vehicles).

An important boundary condition in the development of the meta-model presented here was compatibility with suitable MBSE development methods. Therefore, in the following research it makes sense to test and implement the extension of the SysML-based motego profile [27] by the meta-model developed here. The newly introduced elements have to be implemented in the motego profile and the method has to be reasonable adapted using the language elements, relations and diagrams available in the SysML. As part of the motego profile, the results of this paper can be efficiently modeled and used to extend the approach of Jacobs et al. [16] in practice.

Moreover, it is important to conduct further research on the usage of the sequences of operating states and derived requirements for the synthesis of system models [16, 26], for the rule-based design of system architectures [7] or by integrating simulation models [34, 35] for functional testing with workflows [30, 38].

References

1. Bender B, Gericke K (2021) Pahl/Beitz Konstruktionslehre, 9th edn. Springer, Berlin
2. Isermann R (2007) Mechatronische Systeme: Grundlagen, 2nd edn. Springer, Berlin
3. Porter ME, Heppelmann JE (2014) How smart, connected products are transforming competition. Harv Bus Rev 92(11):64
4. Bundesregierung (2021) Klimaschutzgesetz 2021. https://www.bundesregierung.de/breg-de/themen/klimaschutz/klimaschutzgesetz-2021-1913672. Accessed 31 Oct 2022
5. Umweltbundesamt (2021) Emissionsquellen. https://www.umweltbundesamt.de/themen/klima-energie/treibhausgas-emissionen/emissionsquellen#energie-stationar. Accessed 31 Oct 2022

6. Menken JC (2016) Thermomanagement im batteriebetriebenen Pkw unter Nutzung eines Kaltdampfprozesses mit Sekundärkreislaufsystem. https://leopard.tu-braunschweig.de/receive/dbbs_mods_00064030. Accessed 31 Oct 2022

7. Weustenfeld T (2017) Heiz- und Kühlkonzept für ein batterieelektrisches Fahrzeug basierend auf Sekundärkreisläufen. Dissertation. Audi Dissertationsreihe, v. 122. Cuvillier, Göttingen

8. Tober W, Lenz H-P (2016) Praxisbericht Elektromobilität und Verbrennungsmotor: Analyse elektrifizierter Pkw-Antriebskonzepte. Springer, Wiesbaden

9. Homann G (2015) Energieeffizientes Heizen eines E-Fahrzeugs. https://leopard.tu-braunschweig.de/receive/dbbs_mods_00060032. Accessed 31 Oct 2022

10. Wawzyniak M, Art L, Jung M et al (2017) Thermomanagement als Grundvoraussetzung für die elektrische Mobilität. ATZ Automobiltech Z 119:52–57. https://doi.org/10.1007/s35148-017-0103-z

11. Verband kommunaler Unternehmen e. V. (2021) VKU-Umfrage zu Elektromobilität: Die Aufgabenverteilung ist klar: VKU. https://www.vku.de/presse/pressemitteilungen/archiv-2021-pressemitteilungen/vku-umfrage-zu-elektromobilitaet-die-aufgabenverteilung-ist-klar/. Accessed 31 Oct 2022

12. Cornet A, Hensley R, Hirschberg C et al (2019) Reboost: A comprehensive view on the changing powertrain component market and how suppliers can succeed. https://www.mckinsey.com/~/media/McKinsey/Industries/Automotive%20and%20Assembly/Our%20Insights/Reboost%20A%20comprehensive%20view%20on%20the%20changing%20powertrain%20component%20market%20and%20how%20suppliers%20can%20succeed/Reboost-a-comprehensive-view.pdf. Accessed 31 Oct 2022

13. Erriquez M, Morel T, Moulière P et al (2017) Trends in electric-vehicle design. https://www.mckinsey.de/industries/automotive-and-assembly/our-insights/trends-in-electric-vehicle-design. Accessed 31 Oct 2022

14. Koller R (2013) Konstruktionslehre für den Maschinenbau: Grundlagen zur Neu- und Weiterentwicklung technischer Produkte mit Beispielen, 4th edn. Springer, Berlin

15. INCOSE (2007) Systems engineering vision 2020. https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf. Accessed 4 Oct 2022

16. Jacobs G, Konrad C, Berroth J et al (2022) Function-oriented model-based product development. In: Krause D, Heyden E (eds) Design methodology for future products: data driven, agile and flexible, 1st edn. Springer, Cham

17. Wölkl S, Shea K (2009) A computational product model for conceptual design using SysML. In: 29th Computers and Information in Engineering Conference. ASMEDC, vol 2, pp 635–645

18. Czriharz T, Hruschka P, Queins S et al (2022) Handbuch Requirements Modeling: Aus- und Weiterbildung zum IREB Certified Professional for Requirements Engineering. International Requirements Engineering Board

19. Gausemeier J, Dumitrescu R, Tschirner C et al (2011) Modellbasierte Konzipierung eines hybriden Energiespeichersystems für ein autonomes Schienenfahrzeug. In: Maurer M, Schulze S-O, Ackva S (eds) Tag des Systems Engineering: Hamburg, 9.–11. November. Hanser, München, pp 45–54

20. Weilkiens T, Lamm JG, Roth S et al (2016) Model-based system architecture. John Wiley & Sons Inc, Hoboken

21. Pohl K, Hönninger H, Achatz R et al (2012) Model-based engineering of embedded systems: the SPES 2020 methodology. Springer, Berlin

22. Glinz M, van Loenhoud H, Staal S et al (2022) Handbuch für das CPRE Foundation Level nach dem IREB-Standard: Aus- und Weiterbildung zum Certified Professional for Requirements Engineering (CPRE)

23. Robertson S, Robertson J (2013) Mastering the requirements process: Getting requirements right, 3rd edn. Addison-Wesley

24. Zingel C, Albers A, Matthiesen S et al (2012) Experiences and advancements from one year of explorative application of an integrated model-based development technique using C&C²-A in SysML. IAENG Int J Comput Sci 39:165–181

25. Lamm JG, Weilkiens T (2013) Method for deriving functional architectures from use cases. Syst Eng 17:225–236. https://doi.org/10.1002/sys.21265

26. Zerwas T, Jacobs G, Spütz K et al (2021) Mechanical concept development using principle solution models. IOP Conf Ser Mater Sci Eng 1097:12001. https://doi.org/10.1088/1757-899X/1097/1/012001

27. Spütz K, Jacobs G, Zerwas T et al (2023) Modeling language for the function-oriented development of mechatronic systems with motego. Forsch Ingenieurwes. https://doi.org/10.1007/s10010-023-00623-4

28. ISO (2014) Passenger cars—Windscreen defrosting and demisting systems—Test method(ISO 3468)

29. Herlt A, Jana P, Kellner M et al (2022) Smartphones on wheels: New rules for automotiveproduct development. https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/smartphones-on-wheels-new-rules-for-automotive-product-development?stcr=FFFBFC808E894D67AC0C00BBD428F9A9&cid=other-eml-alt-mip-mck&hlkid=abb394b8239444b896b9515ffac61fb9&hctky=14231617&hdpid=ed4eea05-0946-4fad-8913-1ab75e41622d. Accessed 26 Oct 2022

30. Höpfner G, Jacobs G, Zerwas T et al (2021) Model-based design workflows for cyber-physical systems applied to an electric-mechanical coolant pump. IOP Conf Ser Mater Sci Eng 1097:12004. https://doi.org/10.1088/1757-899X/1097/1/012004

31. Strupp NC (2011) Betriebsverhalten von Verflüssigern in automobilen Kältekreisläufen. http://www.digibib.tu-bs.de/?docid=00042236. Accessed 7 Nov 2022

32. Strupp NC, Lemke N (2009) Klimatische Daten und Pkw-Nutzung: Klimadaten und Nutzungsverhalten zu Auslegung, Versuch und Simulation an Kraftfahrzeug-Kälte-/Heizanlagen in Europa, USA, China und Indien 224. https://www.tib.eu/de/suchen/id/TIBKAT%3A689343531. Accessed 7 Nov 2022

33. Flik M, Kampf H, Kies A et al (1996) Lastprofile zur Berechnung des Energieverbrauchs von Klimatisierungssystemen in Fahrzeugen. In: VDI-Tagung „Simulation und Berechnung im Fahrzeugbau", Würzbug

34. Zerwas T, Jacobs G, Kowalski J et al (2022) Model signatures for the integration of simulation models into system models. Systems 10:199. https://doi.org/10.3390/systems10060199

35. Spütz K, Berges J, Jacobs G et al (2022) Classification of simulation models for the model-based design of plastic-metal hybrid joints. Procedia CIRP 109:37–42. https://doi.org/10.1016/j.procir.2022.05.211

36. United Nations (2018) Global technical regulation on worldwide harmonized light vehicles test procedures (WLTP)

37. Schuldt F (2017) Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen. https://leopard.tu-braunschweig.de/receive/dbbs_mods_00064747. Accessed 31 Oct 2022

38. Habermehl C, Höpfner G, Berroth J et al (2022) Optimization workflows for linking model-based systems engineering (MBSE) and multidisciplinary analysis and optimization (MDAO). Appl Sci 12:5316. https://doi.org/10.3390/app12115316