



## **SYSTEM OF SYSTEMS APPROACH FOR THE DESCRIPTION AND CHARACTERIZATION OF VALIDATION ENVIRONMENTS**

A. Albers, C. Mandel, S. Yan and M. Behrendt

### **Abstract**

Validation reveals differences between created objects and their objectives. The increasing number of mechatronic systems drives the interdisciplinary approach of Systems Engineering for product development and thus also validation. Complexity and scale of the individual subsystems of a system may increase to such an extent, that the system can be considered a “System of System” (SoS). These SoS can also be found in validation systems. To establish a common understanding, this contribution describes an approach to transfer Systems- and SoS Engineering to the domain of system validation.

*Keywords: validation, systems engineering (SE), system of systems, design models, large-scale engineering systems*

### **1. Introduction**

Validation is an essential part of product engineering (Albers, 2010; Albers et al., 2016a). Triggered by the system of objectives (e.g. by product requirements, application aims) validation activities represent the link between the current product state in the development process and the underlying system of objectives. The activity of validation reveals differences between created objects (like the product) and their objectives hence, creating new objectives or completing the existing ones and serves as a source for new design solutions (Albers, 2010). Validation induces the development of product models (and further models) for various development phases like idea generation, conception, and design. In this understanding, validation pulls further models as an outcome from the product engineering phases for following validation activities. Therefore, this approach is called "pull approach of validation" (Albers et al., 2016a). As a result, both product system and validation system related results need to be generated simultaneously during product engineering. With today's more and more complex and interdisciplinary product development, Systems Engineering is a way to address and manage these challenges. The increasing number of mechatronic systems drives the interdisciplinary approach of Systems Engineering even further. Complexity and scale of the individual subsystems of a system may increase to such an extent, that those subsystems can be considered as individual systems on their own. These so called “System of Systems” (Maier, 1996) can also be found in validation systems. In this contribution it is shown, how approaches from System of Systems engineering can be transferred to the domain of validation in order to support the modeling and characterization of validation environments.

## 2. State of the art

### 2.1. Systems Engineering and System of Systems

Current product development processes are characterized by increasingly complex customer requirements and the worldwide competition for market shares (Friedenthal et al., 2014). Systems Engineering is, depending on the definition, a perspective, a process and a profession trying to address those challenges in an interdisciplinary and holistic way (Walden et al., 2015).

The interdisciplinary character of Systems Engineering is driven by the circumstance, that the majority of systems developed today can be regarded as mechatronic systems, whereas mechatronic describes “*the synergetic integration of mechanical engineering with electronic and intelligent computer control in the design and manufacturing of industrial products and processes*” (Harashima, 1996). Contrary to systems developed until the beginning of the last century, the complexity of modern mechatronic systems makes it impossible for a single engineer to understand a developed system completely in its whole (INCOSE, 2004). Systems Engineering aims at supporting the development of complex mechatronics systems from early requirements definition to operation and the late phases of the product lifecycle, integrating views and necessities of different disciplines, such as mechanical or electrical engineering.

Systems Engineering is based on the concepts of system science (compare Ropohl, 1975) Therein, the result of product development process can be seen as a sociotechnical system, characterized by the fundamental concepts of Ropohl (2009). The system itself can be decomposed into several interconnected subsystems (Ropohl, 2009). Those interrelationships of (sub-) systems as well as the relations between the System in Development (SiD) to its Connected Systems (CS) in terms of IPEK-X-in-the-Loop (IPEK-XiL) (Albers et al., 2016a) are a key point in Systems Engineering, as the function of a system is determined by the emergent properties, occurring when integrating and connecting discipline specific subsystems (Walden et al., 2015).

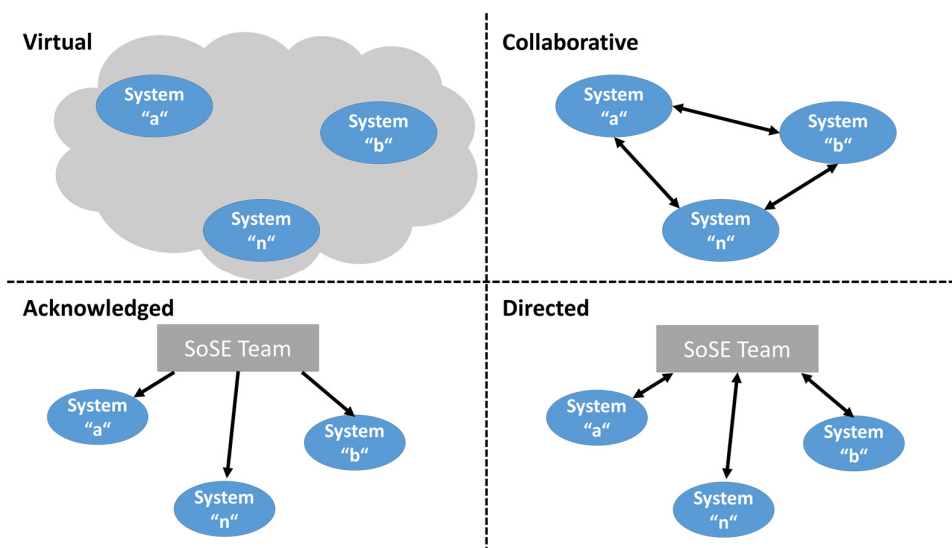
The complexity and scale of individual subsystems may rise to an extent where those subsystems can be regarded as individual systems on their own, as for example in the context of the internet or global satellite networks (Maier, 1996). In this context, the term “System of Systems” (SoS) is used to describe such configurations (Maier, 1996). Although there exists no commonly accepted definition for Systems of Systems, several aspects can be found in literature to characterize constituent systems in a Systems of Systems and differentiate them from subsystems in a large scale monolithic systems of subsystems (MS) (see Maier, 1996; Boardman and Sauser, 2006; Gorod et al., 2008; Haverkort, 2013; Albers et al., 2017c). Based on those aspects, 9 attributes and their manifestations for a MS or a SoS, respectively, have been consolidated and described in Table 1.

Lane (2013) consolidates four categories used in literature to describe SoS, primarily based upon the level of authority and responsibility in the management of the SoS and its evolution (see Figure 1):

- **Virtual SoS:** The SoS lacks a central management authority and a centrally decided common purpose. Not all constituent systems are necessarily known, so that reliance and maintenance becomes a challenging task.
- **Collaborative SoS:** The engineering teams of the individual constituent systems work together on an incentive-based basis (more or less voluntarily) to fulfill a shared central purpose. However, there is no central authority to manage SoS-related activities of the constituent systems.
- **Acknowledged SoS:** The SoS has a designated manager (the SoS Engineering- or SoSE Team), recognized objectives and shared resources on the SoS level. However, constituent systems remain independent in ownership, objectives, funding and development approaches. The central manager can provide guidance to the individual constituent systems but the constituent systems are not required to comply with requests of the central management nor to formally report to it.
- **Directed SoS:** The SoS is built to fulfill a specific purpose and centrally managed by a SoSE Team. Constituent systems remain able to operate independently but a central management authority can require them to develop and support capabilities of the SoS.

**Table 1. Attributes and their characteristic manifestations for MS and SoS (compare Maier, 1996; Boardman and Sauser, 2006; Gorod et al., 2008; Haverkort, 2013; Albers et al., 2017c)**

Attribute	Monolithic System of Subsystems (MS)	System of Systems (SoS)
<b>Operational independence</b>	Full functionality only in the context of the super-system	Able to operate independently, even if the SoS is disassembled
<b>Managerial independence</b>	Developed especially to be part of the super-system	Constituent systems are able to operate independently and are assembled and integrated from independent sources
<b>Connectivity</b>	Static interoperability, determined during the design of the MS	Dynamic interoperability, allowing for integration of new subsystems. Connections and interfaces may appear and vanish as required
<b>Diversity</b>	Hidden in discrete modules that are homogenous	Heterogeneous to fulfill a diversity of functions
<b>Emergence</b>	Can for the most part be foreseen and tested	Functionality of SoS exclusively arises in the SoS context. Unpredictable errors may occur
<b>Scale, size and black-box-character</b>	White-box information is potentially accessible by other systems or developers	Constituent systems appear as black-boxes. No developer would be able to oversee all details of every involved constituent systems
<b>Spatial distribution</b>	Usually operate close to each other. Legal constraints and norms of only a specific country apply	Spatially distributed and communication over long distances. May underlie different, potentially conflicting legal constraints/norms
<b>Different Product Lifecycles</b>	Basically follow the lifecycle of the super-system	Lifecycles may greatly vary from each other. Lifecycle of SoS is rather long and enfolds different generations of constituent systems
<b>Evolutionary character</b>	Usually undergo no significant changes once the design of the MS is set and finished	Permanent changes that affect the SoS. Overall functionality of SoS needs to be ensured at any time of its lifecycle



**Figure 1. Categories of Systems of Systems (adapted from Lane, 2013)**

## 2.2. Continuous Validation Concept (CVC) in product engineering

Validation is the central activity of the product development process and guarantees a successful product on the market (Albers, 2010; Albers et al., 2016a). According to the 'Rule of Ten' (Ehrlenspiel and Meerkamm, 2013) the cost for changes on the system concept or design increases by a factor of 10 for every design phase. Therefore, a continuous validation concept is necessary, starting within early phases of product engineering. Validation triggers the development of validation systems containing specific product models for validation and suitable validation environments and methods for various phases like idea generation, conception, and construction. In this understanding, validation activities pull further models and resources as an outcome from the product engineering phases. Therefore, this approach is called "pull principle of validation" (Albers et al., 2016a). With this, both product and validation related results are generated simultaneously during product engineering.

Numerous approaches and process models for the integration of validation activities in the product development process are described in literature (compare Eigner et al., 2014). Some authors emphasize the importance of frontloading and implementation of validation activities in early phases (compare e.g. Paulweber and Lebert, 2014). However, validation activities and appropriate validation environments are often strongly related to particular phases (compare Eigner et al., 2014). As a consequence, the IPEK-X-in-the-Loop (IPEK-XiL)-Approach has been developed as an integrated approach to support a continuous validation throughout the product development process (Albers and Düser, 2010; Albers et al., 2016a). Established validation approaches share the idea of testing a specific subsystem of the product "in-the-loop". The loop is regarded as a model-based integration into the subsystem's surrounding, like in Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) or Hardware-in-the-Loop (HiL) setups (Bringmann and Krämer, 2008). The IPEK-XiL-Approach extends the scopes of those approaches regarding requirements from mechanic/ mechatronic product development and individuals from different disciplines (Albers et al., 2016a). Hereby, the 'X' as System in Development (SiD) is connected functionally or energetically to the super-system in a virtual, physical or mixed virtual-physical way. For example, the SiD could be a specific subsystem integrated in the system "Vehicle" and connected to the Connected Systems (CS) "Driver" and "Environment". All models have no limitation either to particular levels of abstraction or to physical components, control units, software or virtual models (see Figure 2).

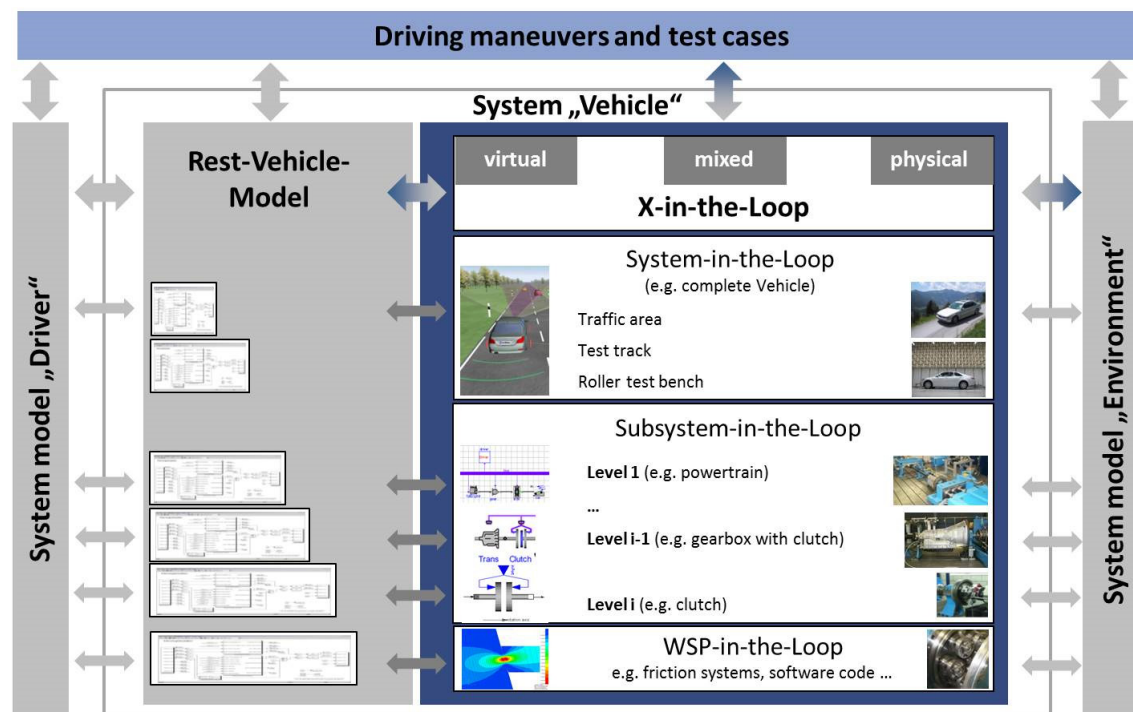


Figure 2. IPEK-X-in-the-Loop-Framework (Albers et al., 2016a)

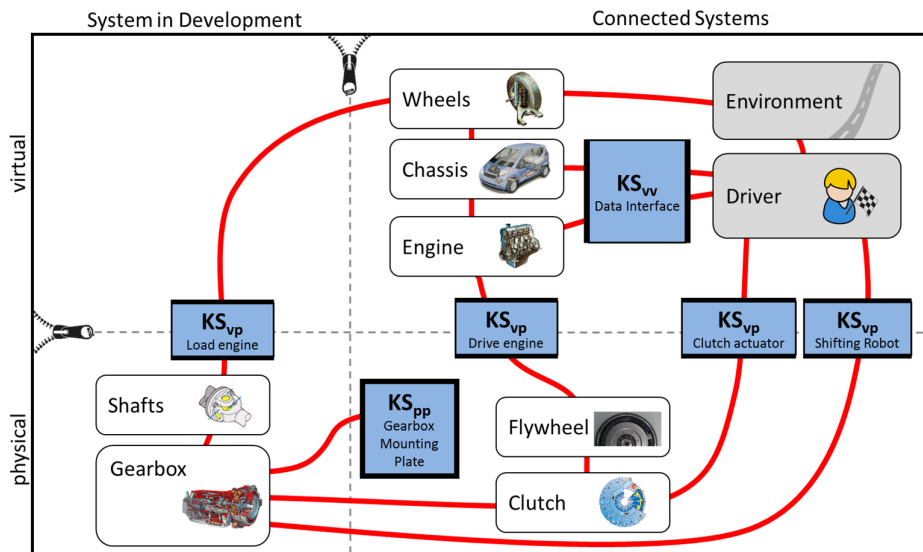
### 2.3. Validation environment

According to the integrated Product engineering Model (iPeM, compare Albers et al., 2016c) the **validation system** (VS) includes all developed elements (systems, methods and processes), which enable the validation of the product. The development of a validation system can be seen as a distinct development process, as e.g. a test bench has to be planned, designed and validated as well.

A suitable **validation environment** (VE) according to the IPEK-XiL-Approach provides the conditions to conduct validation activities, where a (sub-) system is validated in the context of the super system and interacting systems. Hence, a validation environment is a concrete manifestation of an operation system for validation in terms of methods and the system of resources for at least one combination of a product and a validation objective at a specific point in the product lifecycle (Albers et al., 2016a). Thereby, the system of resources for the specific validation environment is a sub-set of the available validation system. This includes for example simulation models on real time platforms being the manifestation of the virtual models, maneuver catalogs or software for post processing. A validation environment is always defined in relation to the engineer's validation objective.

Furthermore, a validation environment consists at least of one **validation configuration** (VC), which is a specific combination of methods, test cases and resources and parametrization derived from the validation environment.

For the practical implementation of a validation environment and its configuration, a top-down modelling approach is of advantage (Geier et al., 2012; Albers et al., 2016a). This approach divides the super systems "Driver", "Environment" and "Vehicle" (context of vehicle development) into the relevant sub-systems, which are then divided into a SiD and further CS. Hereby, the fidelity of a single product system model can vary depending on the validation objective (compare Stachowiak, 1973). Figure 3 shows a model of the IPEK-XiL-Architecture for a validation environment of the IPEK Powertrain-in-the-Loop test bench (Pinner et al., 2013). In this context, the term IPEK-XiL-Architecture is used to describe functional and structural elements of the validation environment. To identify the powertrain behavior, a gearbox with side shafts is regarded as the SiD. Further parts of the drivetrain (e.g. flywheel and clutch) are physically present on the test bench. Other components of the vehicle (e.g. combustion engine or tires) and the super systems environment and driver are virtually represented by real-time simulation models. Every product (sub-) model, regardless of whether it is realized physically or virtually, is symbolized as a small text box with an icon.



**Figure 3. Model of the IPEK-XiL-Architecture of the Powertrain in the Loop validation environment**

In order to link the different product system models which are not able to establish a linkage directly due to incompatibility (e.g. between virtual and physical domain), so-called Koppelfunktionen are

needed. They are necessary for the realization of the overall system behavior but shall not change the relevant system behavior. These Koppelfunctions are implemented by Koppelsystems, whereof there are three types (Albers et al., 2016b):

- physical/physical Koppelsystems (KS\_PP) between two physical models (e.g. the connection between the gearbox and the test bench environment)
- virtual/virtual Koppelsystems (KS\_VV) between two virtual models (e.g. the connection between distant virtual models via network)
- virtual/physical Koppelsystems (KS\_VP) between one virtual and one physical model (e.g. the robot shifter that transforms a virtual gear change demand into physical action)

Altogether, this validation environment is able to provide a vehicle's system behavior to parts of a vehicle powertrain, which is relevant for the investigation and validation aim. The SiD is set up to behave like being used within a real vehicle.

Driven by distributed development environments of today's automotive industry the elements required to conduct validation activities can also be distributed. Consequently, the concept of a distributed validation is derived from distributed product development. The latter has its origins in the division of labor, which is a major prerequisite for handling the high complexity of product development (Ehrlenspiel and Meerkamm, 2013). Important characteristics of a distributed product development are the collaborative solution of an overall problem, the exchange of information and the integration of partial solutions (Krause et al., 1998). Distributed product development can occur between departments of the same company or different companies. In this understanding, the spatial separation is one of the relevant factors for a distributed validation environment (DVE). Not all parts of the operation system (e.g. simulation models, test benches or validation engineers) have to be at the same location. The reasons for the distribution of the validation environment are various and range for example from different development focuses and resources of the company resp. company department or even different legal entities to confidentiality issues. On the other hand, there is also the possibility to share models for efficiency. However, in a DVE enabling model-based validation activities, the model behavior has to overcome spatial separation.

### 3. Motivation and research aim

Profound research is conducted on the topic of validation in product development. However, shared languages, methods and tools for the description and the design of validation environments seem to be missing. However, those are crucial to create a common understanding of the validation environment across different engineering teams and engineering disciplines. The importance of a common understanding and communication continuously increases as the design and management of validation environments becomes more and more complex, as in the case of DVEs.

To support the foundation of such a common understanding, this contribution describes an approach to transfer Systems- and SoS Engineering to the domain of system validation. Therefore, it is described how a validation environment or validation configurations respectively, following the IPEK-XiL-Approach, can be modeled as a system or a SoS. By this means, languages, tools and methods investigated in systems- and SoS engineering research, can be used for the design and management of validation systems.

First of all, the proposed description model enables the identification and quantification of differences between specific validation environments based on the aspects described in Chapter 2.1. This supports the comparison of different validation environments, so that similar validation environments, for example used for the validation of previous product generations, can be identified and used as references.

As described in Chapter 2.2, validation is a crucial activity in product development and should therefore be carried out in parallel to the product development process, starting in early development phases. Developing and managing validation environments as a SoS supports in linking the development and modeling of the SiD and the respective validation environments and validation configurations, respectively. Therefore, already established Systems Engineering languages, tools and methods can be used.

Finally, the proposed approach should support in planning the potential use of a validation environment, especially in the context of a DVE, in advance. Consequently, supportive measures for the use of the validation environment as a constituent system in the SoS context of an DVE can already be planned

and realized during the design of the validation environment. This should help at leveraging the DVE on a more controlled category of SoS (see Chapter 2.1), for example from collaborative SoS to acknowledged SoS.

## 4. Validation environment as a System of Systems

### 4.1. New description model for validation environments

In order to find a shared definition of the term DVE and to support the characterization of different validation environments, the proposed approach connects the System of Systems perspective with the IPEK-XiL-Approach.

As described in Chapter 2.3, a validation environment (consisting of at least one validation configuration) for a specific validation objective considers a set of methods, test cases and resources, including the physical manifestation of the product system models and Koppelsystems. For the implementation of a specific validation environment, the top-down modelling approach from the super systems “Driver”, “Environment” and “Vehicle” (context of vehicle development) into the relevant product subsystems is applied. The resulting product (sub-) system models can be realized either in physical, virtual, or a mixed physical-virtual form. Further Koppelsystems link the different models, which are not able to establish a linkage directly due to incompatibility. Based on the explications above, the new description model for validation environments can be described:

*The new description model for validation environments views each specific combination of at least one product system model and required resources for its physical manifestation, one test case and one method as an individual system.*

Those individual systems could e.g. be an already operable test bed with a physical product system model or a real-time simulation platform with several virtual product system models. Consequently, the validation environment with its validation configurations can be regarded as a system or SoS, respectively, consisting of several interconnected individual systems (constituent systems).

### 4.2. Characterization of a validation environment and graphical representation

By regarding validation environments as MS or SoS, respectively, the individual constituent systems and their connections can be investigated with the help of the attributes described in Table 1. This helps in evaluating, for which attributes the validation environment shows characteristics of a MS and for which rather characteristics of a SoS. Therefore, manifestations of the attributes are differentiated for three different cases, which are represented by different numbers: 0 for a MS, 1 for a SoS (as already described in Table 1) and 0.5 for an intermediate form. Eventually, it is possible to display the manifestation of those attributes in a characteristic map for the validation environment (see Figure 4):

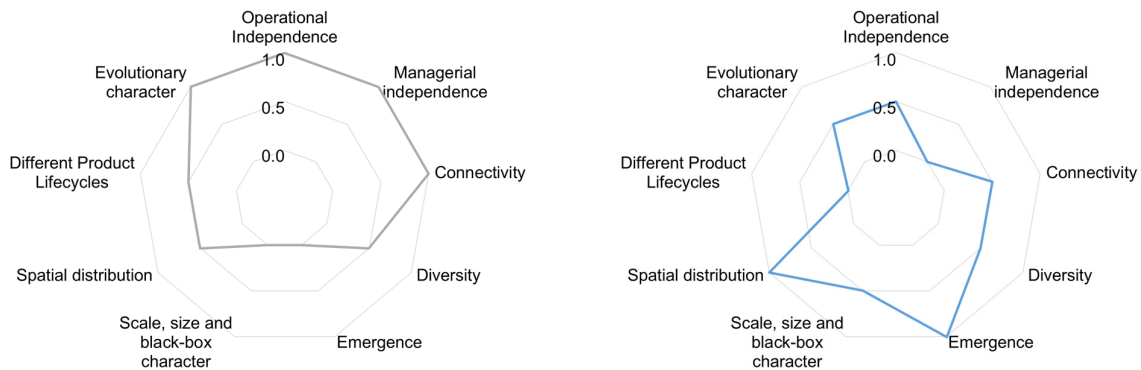


Figure 4. Exemplary validation maps

These maps can help in comparing different validation environments and draw e.g. conclusions for best practices or pain points from one validation environment to support the development of a comparable or supplementary one.

### 4.3. Distribution ratio of a validation environment

In a specific application of the characteristic maps for a validation environment, the so-called “distribution ratio” is introduced. The distribution ratio, derived from each single attribute of the map, aims at differentiating DVEs by quantifying the character of distribution for a validation environment. Therefore, it directly gives support for the management of validation activities by revealing the potential complexity of a specific validation environment.

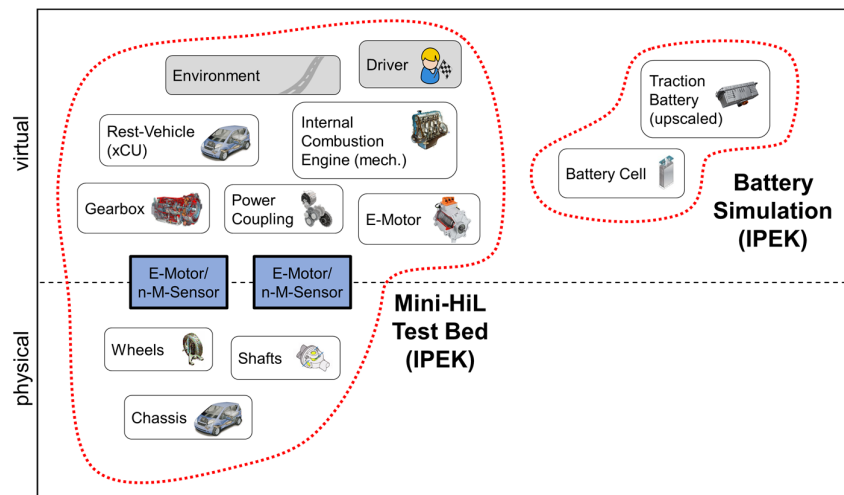
According to the generic formula for a cost-benefit analysis, the distribution ratio is defined as follows:

$$\delta_{VE} = \sum_{a=1}^m w_a \delta_a(f_a) \quad (1)$$

The overall rating of the distribution  $\delta_{VE}$  consists of the weighted ratings of all attributes of the validation environment map  $\delta_a$ . Every engineer decides which factors  $f_a$  are most important to draw conclusions from in his specific use case. Therefore, the weighting factors  $w_a$  are to be adjusted with respect to the validation objective and further constraints of the product development process. Given that  $\sum_{a=1}^m w_a = 1$  and  $\delta_a < 1$ , the maximum distribution ratio  $\delta_{VE}$  equals 1 (100%), describing the most pronounced SoS configuration, whereas 0 (0%) would describe the most pronounced MS configuration of the validation environment according to the attributes in Table 1.

### 4.4. Application example

The described application example shows, how the super systems “Driver”, “Environment” and “Vehicle” can be distributed to different locations (e.g. OEM and supplier) but still be interacting with the other product (sub-) systems of the validation environment. In this example, four partners of the project XiL-BW-e – Laboratory Network Baden-Württemberg for Electric Mobility (compare Albers et al., 2017b) contributed virtual and physical (sub-) system models to a joint validation environment according to the IPEK-XiL-Approach. Each partner contributed product (sub-) system models according to their available validation systems. The available validation system of the project partner IPEK in XiL-BW-e is shown in Figure 5. The dotted red lines mark the constituent systems in the context of SoS.

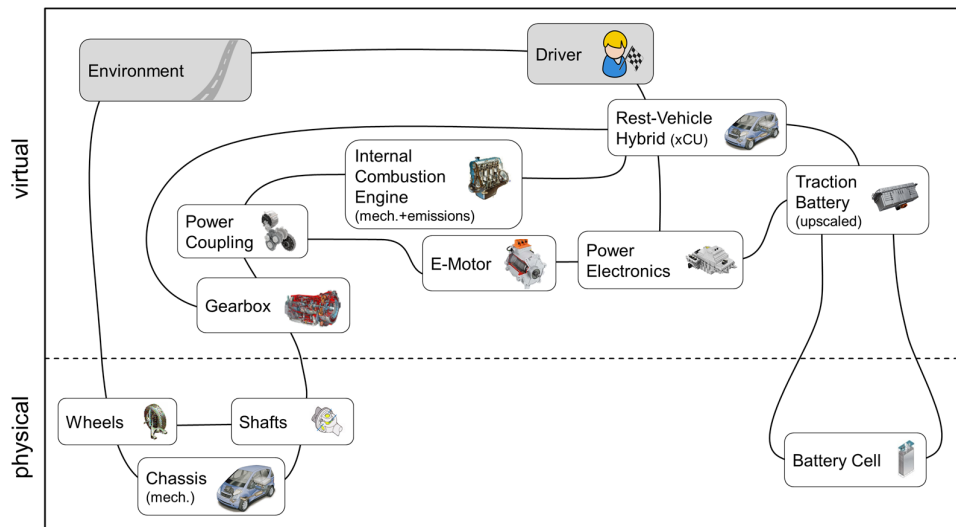


**Figure 5. Available validation system of the IPEK in XiL-BW-e**

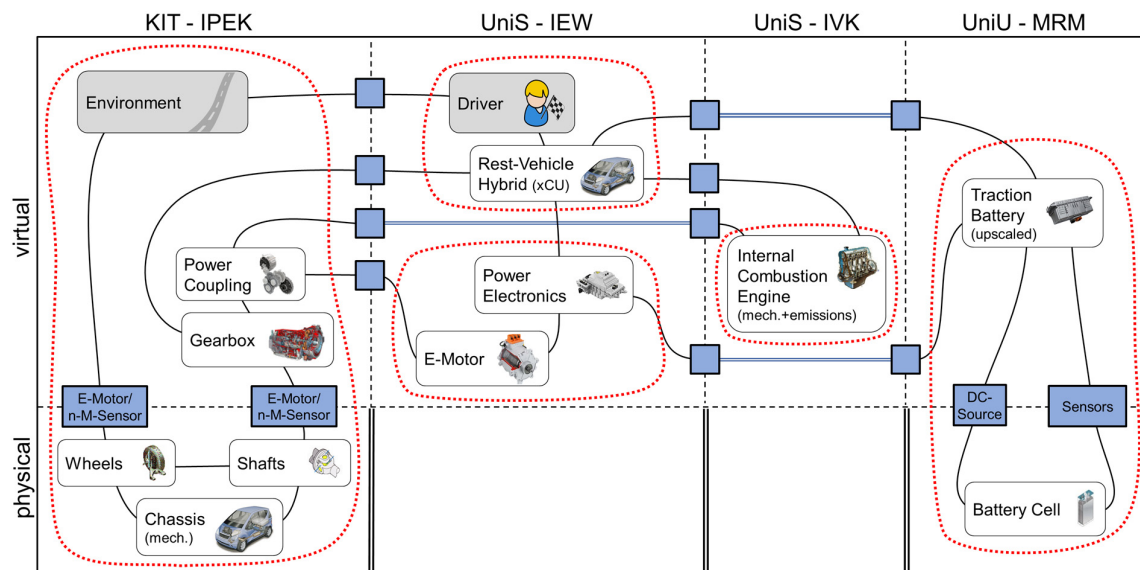
The validation objective of the project partners was to investigate the battery behavior and combustion engine emissions of a hybrid vehicle in the future Worldwide Harmonized Light Vehicles Test Procedure (WLTP). Therefore, supplementary models are required in addition to the models available in the validation system of IPEK, in order to fulfil the validation objective. The overall desired validation environment of all four partners is depicted in Figure 6 (as seen for the product models).



Interconnections between different locations are mostly possible in the virtual domain, thus all partners have standardized interfaces according to the networking concept of the laboratory network (Albers et al., 2017b). The overall system consists of subsystems, which belong to a domain and location (see Figure 7). With this, the Koppelsystems (blue boxes in Figure 7), especially the virtual/virtual Koppelsystems (KS\_VV) between different locations, play a major role. If a product model of a constituent system is substituted by a model of another constituent system, the corresponding KS\_VV has to disconnect the original connections and establish new ones to the related new product models of other constituent systems (e.g. E-Motor or Internal Combustion Engine of the IPEK Mini-HiL are replaced by models from UniS-IEW and UniU-IVK). They also ensure the real-time capability of the now distributed validation environment. On the other hand, they restrict the validation objective with regard to the speed of the phenomenon of interest and the transfer quality of the Koppelsystems. At the same time, the validation objective in general sets requirements for the validation environment.



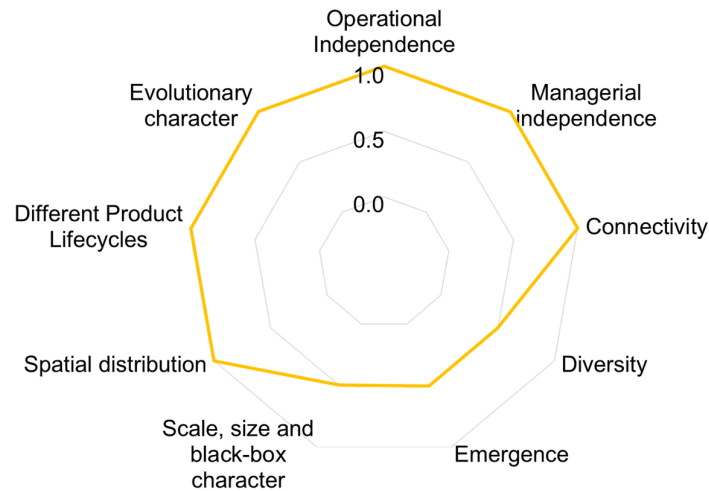
**Figure 6. Model of the required IPEK-XiL-Architecture of the validation environment in order to fulfill the validation objective**



**Figure 7. Model of the realized IPEK-XiL-Architecture of the distributed validation environment in XiL-BW-e**

The technical systems surrounded by a dotted red line in Figure 7 represent the systems in the context of the SoS. The characteristic validation map of the validation environment is shown in Figure 8.

In order to calculate the distribution ratio of the validation environment, the weights  $w_a$  of the individual parameters are set as displayed as shown in Table 2 (0 = minimum weight; 1= maximum weight). The weights are derived from the project context: As described by Maier (1996), the managerial and organizational independence are of major importance. This is also the case for the underlying project as many different entities are involved. On the other hand, the test benches and real time systems used in the validation environment share relatively similar characteristic, thus diversity does not play a major role. Eventually, this yields a distribution ratio of  $\delta_{VE} = 0.875$  (87.5%).



**Figure 8. Validation map for the XiL-BW-e validation environment**

**Table 2. Weight factor of the different attributes in the context of XiL-BW-e**

Attribute	$w_a$
Operational independence	0,2
Managerial independence	0,2
Connectivity	0,1
Diversity	0,05
Emergence	0,1
Scale, size and black-box-character	0,1
Spatial distribution	0,1
Different Product Lifecycles	0,1
Evolutionary character	0,05
$\Sigma$	<b>1</b>

## 5. Discussion

The new approach for describing validation environments as a SoS adds a new perspective on the topic of validation and directly relates to the analysis and design of validation environments. Especially, when already existing individual systems of validation environments are used to build up a new validation configuration, as in the context of DVEs, the new approach is expected to be beneficial. Being able to use established Systems Engineering languages, tools and methods can support the designer e.g. in identifying critical interfaces between individual systems of the validation configuration and thus prioritize development activities. In addition, it enables to integrate the development of a system and its dedicated validation environments by using for example approaches like FAS4X (compare Kleiner et al., 2017).

The introduced description model enables the characterization of different validation environments and the quantification of similarities and differences. This can prove beneficial as it allows to identify best

practices and potential pain points from comparable validation environments at early stages of the validation environment development. In the context of PGE - Product Generation Engineering (compare Albers et al., 2017a) the predecessor validation environment of different parties can now easier be analyzed and reused and thus serve as a knowledge- and development base for new development processes of validation environments. This is an important step to take full advantage of the synergy potentials between different generations of validation environments.

The introduced distribution ratio can be used as a single-value benchmark for different validation environments. Generally spoken, a high distribution value indicates that the validation environment can be regarded as a SoS, accompanied by all the challenges surrounding the management of SoS. It is still to be investigated, if and how the distribution ratio is related to other aspects of validation. Those aspects could e.g. be the gain of knowledge by the validation activity or up to which extend a distribution ratio can be accepted for a given point in the product development process.

## 6. Summary and outlook

In this contribution, a new approach to describe validation environments is proposed. Therefore, it is described, how a validation environment can be regarded as a System of Systems. This allows for the transfer of insights from Systems- and SoS Engineering research to the domain of validation. It is described how, based on SoS attributes, a validation environment can be characterized and thus made quantifiably comparable to other validation environments. In a specific application, a distribution ratio is introduced in order to gain a metric for the description and comparison of distributed validation environments.

Further research will apply the new approach to several validation environments and create their characteristic validation maps. With this, it can be evaluated, if the chosen attributes are appropriate and sufficient for the description of validation environments and how they depend on the validation context. Special focus will be laid on DVEs and the distribution ratio. Furthermore, it will be investigated how validation environments with similar validation maps can be grouped together with the goal of defining a set of categories for validation environments (as e.g. DVE).

In addition, the integration of system and validation system development by Systems Engineering means needs to be further evaluated and exemplarily realized.

## Acknowledgement

The distributed validation environment in XiL-BW-e (see Chapter 4.4) was partially supported by a grant from the Ministry of Science, Research and the Arts of Baden-Württemberg. The authors thank all partners of the research project “XiL-BW-e” for their assistance and support.

## References

- Albers, A. (2010), “Five Hypotheses about Engineering Processes and their Consequences”, *Proceedings of the 8<sup>th</sup> International Symposium on Tools and Methods of Competitive Engineering (TMCE 2010), Ancona, Italy, April 12-16, 2010*, TMCE, pp. 343-356.
- Albers, A. and Düser, T. (2010), “Implementation of a Vehicle-in-the-Loop Development and Validation Platform”, *Proceedings of the 33rd FISITA World Automotive Congress, Vol. 1, Budapest, Hungary, May 30-June 4, 2010*, pp. 173-182.
- Albers, A., Behrendt, M., Klingler, S. and Matros, K. (2016a), “Verifikation und Validierung im Produktentstehungsprozess”, In: Lindemann, U. (Ed.), *Handbuch Produktentwicklung*, Carl Hanser Verlag, München, pp. 541–569. <https://doi.org/10.3139/9783446445819.019>
- Albers, A., Behrendt, M., Klingler, S., Reiss, N. and Bursac, N. (2017a), “Agile product engineering through continuous validation in PGE - Product Generation Engineering”, *Design Science*, Vol. 3, pp. e5. <https://doi.org/10.1017/dsj.2017.5>
- Albers, A., Dietmayer, K., Bargende, M., Behrendt, M., Yan, S. et al. (2017b), “XiL-BW-e - Laboratory Network Baden-Württemberg for Electric Mobility”, *Proceedings of the 30th International Electric Vehicle Symposium and Exhibition (EVS30), Stuttgart, Germany, October 9-11, 2017*.
- Albers, A., Pinner, T., Yan, S., Hettel, R. and Behrendt, M. (2016b), “Koppelsystems: Obligatory Elements within Validation Setups”, *Proceedings of the DESIGN 2016 / 14th International Design Conference, Dubrovnik, Croatia, May 16-19, 2016*, The Design Society, Glasgow, pp. 109–118.

- Albers, A., Reiß, N., Bursac, N. and Richter, T. (2016c), “iPeM - integrated Product engineering Model in context of Product Generation Engineering”, *Procedia CIRP*, Vol. 50, pp. 100-105. <https://doi.org/10.1016/j.procir.2016.04.168>
- Albers, A., Reussner, R., Kurrle, A., Burger, E., Moeser, G. et al. (2017c), “Continuity in the development of seamless mobility: An approach for a system-of-systems environment”, *International Research Journal of Engineering and Technology*, Vol. 4 No. 2, pp. 668–678.
- Boardman, J. and Sauser, B. (2006), “System of Systems - the meaning of of”, *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering, Los Angeles, CA, April 24-26, 2006*, IEEE, pp. 118-123. <https://doi.org/10.1109/SYSE.2006.1652284>
- Bringmann, E. and Krämer, A. (2008), “Model-Based Testing of Automotive Systems”, *Proceedings of the 1<sup>st</sup> International Conference on Software Testing, Verification, and Validation, Lillehammer, Norway, April 9-11, 2008*, IEEE, pp. 458-493. <https://doi.org/10.1109/ICST.2008.45>
- Ehrlenspiel, K. and Meerkamm, H. (2013), *Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit*, 5th ed., Carl Hanser Verlag, München, Germany. <https://doi.org/10.3139/9783446436275>
- Eigner, M., Roubanov, D. and Zafirov, R. (2014), *Modellbasierte virtuelle Produktentwicklung*, Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-662-43816-9>
- Friedenthal, S., Beihoff, B., Kemp, D., Nichols, D., Oster, C. et al. (2014), *A World in Motion. Systems Engineering Vision 2025*. [online] INCOSE. Available at: <http://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf>
- Geier, M., Jäger, S., Stier, C. and Albers, A. (2012), “Combined real and virtual domain product validation using top-down strategies”, *Proceedings of the ASME 2012 Verification and Validation Symposium (V&V 2012), Las Vegas, Nevada, May 2-4, 2012*, ASME.
- Gorod, A., Sauser, B. and Boardman, J. (2008), “System-of-Systems Engineering Management: A Review of Modern History and a Path Forward”, *IEEE Systems Journal*, Vol. 2 No. 4, pp. 484-499. <https://doi.org/10.1109/JSYST.2008.2007163>
- Harashima, F. (1996), “Recent advances of mechatronics”, *Proceedings of IEEE International Symposium on Industrial Electronics (ISIE '96), Warsaw, Poland, June 17, 1996*, IEEE. <https://doi.org/10.1109/ISIE.1996.548386>
- Haverkort, B.R. (2013), “The Dependable Systems-of-Systems Design Challenge”, *IEEE Security and Privacy*, Vol. 11 No. 5, pp. 62–65. <https://doi.org/10.1109/MSP.2013.124>
- INCOSE (2004), *INCOSE System Engineering Handbook*, Version 2a, INCOSE.
- Kleiner, S., Husung, S., Mandel, C., Albers, A. and Behrendt, M. (2017), “(Model-Based) Systems Engineering für die Digitalisierung der Produktentwicklung”, *Tag des Systems Engineering, Paderborn, Germany, November 8-10, 2017*, Carl Hanser Verlag, pp. 135-144. <https://doi.org/10.3139/9783446455467.015>
- Krause, F.L., Schultz, R. and Doblies, M. (1998), “Verteilte Produktentwicklung”, *Industrie Management*, Vol. 14 No. 1, pp. 14–18.
- Lane, J.A. (2013), *What is a system of systems and why should I care?* [online] USC Center for Systems and Software Engineering. Available at: <http://csse.usc.edu/TECHRPTS/2013/reports/usc-csse-2013-500.pdf>
- Maier, M.W. (1996), “Architecting Principles for Systems-of-Systems”, *System Engineering*, Vol. 1 No. 4, pp. 267-284. [https://doi.org/10.1002/\(SICI\)1520-6858\(1998\)1:4<267::AID-SYS3>3.0.CO;2-D](https://doi.org/10.1002/(SICI)1520-6858(1998)1:4<267::AID-SYS3>3.0.CO;2-D)
- Paulweber, M. and Lebert, K. (2014), *Mess- und Prüfstandstechnik: Antriebsstrangentwicklung · Hybridisierung · Elektrifizierung*, Springer Vieweg, Wiesbaden. <https://doi.org/10.1007/978-3-658-04453-4>
- Pinner, T., Berger, J., Stier, C., Geier, M., Albers, A. and Kohlhammer, K. (2013), “Gesamtfahrzeugvalidierung mit Schaltroboter”, *AutomobilKONSTRUKTION*, Vol. 2, pp. 40–41.
- Ropohl, G. (1975), “Einleitung in die Systemtechnik”, In: Ropohl, G. (Ed.), *Systemtechnik - Grundlagen und Anwendung*, Carl Hanser Verlag, München, Wien, pp. 1-77.
- Ropohl, G. (2009), *Allgemeine Technologie: Eine Systemtheorie der Technik*, KIT Scientific Publishing, Karlsruhe. [https://doi.org/10.26530/OAPEN\\_422388](https://doi.org/10.26530/OAPEN_422388)
- Stachowiak, H. (1973), *Allgemeine Modelltheorie*, Springer, Wien. <https://doi.org/10.1007/978-3-7091-8327-4>
- Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D. and Shortell, T.M. (2015), *INCOSE Systems engineering handbook: a guide for system life cycle processes and activities*, 4th ed., Wiley, Hoboken, NJ.

Constantin Mandel, Master of Science  
 Karlsruhe Institute of Technology, IPEK-Institute for Product Engineering, NVH and System Validation  
 Kaiserstraße 10, 76131 Karlsruhe, Germany  
 Email: [constantin.mandel@kit.edu](mailto:constantin.mandel@kit.edu)