Towards a System of Systems Engineering Architecture Framework

Mandel, Constantin
IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
constantin.mandel@kit.edu

Guenther, Matthias
Fraunhofer Research Institute for Mechatronic Systems Design IEM
Paderborn, Germany
matthias.guenther@iem.fraunhofer.de

Martin, Alex
IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
alex.martin@kit.edu

Windisch, Emily
IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
emily.windisch@gmx.de

Bursac, Nikola
TUHH - Technische Universität Hamburg
Hamburg, Germany
Nikola.Bursac@tuhh.de

Rapp, Simon
IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
simon.rapp@kit.edu

Anacker, Harald
Fraunhofer Research Institute for Mechatronic Systems Design IEM
Paderborn, Germany
harald.anacker@iem.fraunhofer.de

Albers, Albert
IPEK – Institute of Product Engineering at Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
albert.albers@kit.edu

***Abstract*— Model-based approaches such as Model-Based Systems Engineering (MBSE) are widely regarded as a necessity to enable and support the development of ever more complex Systems of Systems (SoS). A key to the application of model-based development approaches in companies is the introduction of systems engineering architecture frameworks (SEAF) to structure the models of systems to be developed. This paper displays the research to develop a SEAF for SoS. Through workshops and interviews with 18 partners from industry and research, we derived perspectives to be considered for describing and modeling SoS. By analyzing existing SEAFs from literature we concluded, that not all perspectives we identified as relevant are covered by a single framework. Based on our research, especially the explicit consideration of SoS characteristics such as autonomy and interdependence, the integrated consideration of product, production system and validation system development, as well as consideration of different product generations and (re-)use of models appears to be not profoundly addressed. Advancing the state of research, we therefore propose a six-dimensional SoS engineering architecture framework to cover all perspectives we identified as relevant. The proposed framework can further serve as a reference for developers to derive and build specific frameworks tailored to their needs while at the same time supporting consistency, reusability and compatibility of the specific frameworks. The presented framework is an initial proposition which continuously being used and evaluated by its application to structure modeling for a variety of real systems from different industries. Insights from the application are fed back for its further development.**

Keywords—ASE, MBSE, Ontology, PGE, Viewpoints

I. Introduction

Technology advances e.g. in digitalization are driving innovation and changing both, systems and the associated engineering. The continuous increase of hardware and especially software in systems increases the performance as well as the interconnectivity of systems, so that characteristic properties and behaviors may change over the lifecycle. Developing such systems requires powerful model-based development approaches that coordinate different perspectives from several engineering domains. In this contribution we outline an SoSE – System of Systems Engineering architecture framework, that includes different perspectives derived from the state of the art as well as needs from examples of industry partners to describe architectures of today's and future technical systems. This paper is based on research conducted in the BMBF funded research project MoSyS. In MoSyS, 18 partners from industry and research aim at developing new methods and tools to support the development of technical systems and the associated value networks as parts of complex Systems of Systems. Therein, the harmonization of different aspects of engineering under the premise of human orientation is a focus.

II. State of Research and Related Work

A. ASE - Advanced Systems Engineering

ASE is a guiding principle that will be essential for the successful design of innovative products, services and product-service systems and their development process in the future. The topics of Advanced Systems, Systems Engineering and Advanced Engineering are considered holistically and in an integrated manner. [1]

Advanced Systems are intelligent, cyber-physical systems with a high degree of networking, autonomy and socio-technical interaction and are mostly integrated into a larger system that provides capabilities that cannot be provided by a single system. Those Systems of Systems (SoS) come with new challenges that need to be addressed while developing tomorrows architectures [2]. Systems engineering is an approach that describes the collaboration of different disciplines and the handling of the associated complexity in project and organization [1]. The INCOSE – International Council on Systems Engineering, states, that the future of systems engineering is model-based [2]. In this understanding, the term Model-Based Systems Engineering (MBSE) is described as a formalized approach for creating consistent cross-domain system models to support in the areas of requirements management, analysis, verification and validation over the entire product lifecycle [3].

In addition, new technical and organizational trends in engineering are developing with Advanced Engineering, which enrich methods, processes, tools and work organizations with the aspects of creativity, agility and digitalization [1].

The SoSE Architecture Framework presented in this publication is intended to support the structuring of SoS models across domains and thus improve the handling of technical and organizational complexity in the context of systems engineering.

B. System Architectures in Product Development

According to ISO 42010, Architecture is defined as the *"fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution"* [4, p. 2].

To describe an architecture, rules and principles need to be established that align to the needs of the applied domain and community of stakeholders [4]. When describing SoS in particular, it is important to use models as a basis for all engineering disciplines. Holt and Perry explain that an architecture has to follow an architectural design process and has to conform to an architecture framework for a defined purpose [5]. Therefore, the use cases and concerns of the stakeholders of an architecture must be evaluated as they determine the purpose of the architecture framework. Especially when using models, a framework should be established that uses an ontology and different viewpoints as described in [5].

Therein, the ontology defines terms, concepts and their relations to form the basis for the language used for modeling. The developed ontology is used to customize and extend a general-purpose modeling language like SysML. The viewpoints use elements as subsets of the ontology to describe a "filter" for a specific modeling purpose. All of the used viewpoints are arranged in the model framework. [5]

According to Gurbuz and Tekinerdogan [6] there are three types of architecture frameworks in the context of systems engineering.

- software architecture frameworks (SOAF)
- enterprise architecture frameworks (EAF)
- systems engineering architecture frameworks (SEAF)

While SOAFs focus on software architectures and EAFs, such as TOGAF or UAF, focus on enterprise architectures, SEAFs aim at supporting systems modeling covering all disciplines required for the engineering from a more focused systems engineering point of view [6, 7]. In this contribution we focus on SEAFs for multidisciplinary technical systems without describing organizational structures as in EAFs.

C. Systems Engineering Architecture Framework

In literature, several existing frameworks, that can be considered SEAFs, can already be found. A prominent representative of such a framework is the MagicGrid framework [7]. The MagicGrid framework can be represented as a two-dimensional matrix containing a set of views to structure a model and the corresponding modeling process. The two dimensions of the framework represent the domain (problem, solution, and implementation) as well as the so called "pillars" (requirements, structure, behavior, parameters, safety & reliability) for modeling a product. Similar frameworks can e.g. be found in the SPES XT methodology [8] or Capella (arcadia methodology [9]).

While those SEAFs mainly focus on the modeling of products, the framework created in the project mecPro[2] targets the integration of product and production system modeling [10]. Further approaches for an integrated modeling of product and production system have been researched in the project I4TP [11]. In addition, a more high-level architecture framework concerning production is the RAMI – Reference Architecture Model Industry 4.0 [12].

Using MBSE to support the validation of technical systems appears to be not profoundly investigated [13, 14]. A framework with characteristics of a SEAF, is the IPEK-XiL-framework [15]. The IPEK-XiL-framework aims at supporting an effective validation by providing a framework for modeling a SiD – System-in-Development in interaction with its residual system, a user interacting with the system, the system environment as well as test cases to be performed. Therein, modeling can be performed over several system layers, from a system in its whole to individual working surface pairs.

III. Research Approach and Research questions

Analysis of the state of research shows, that the development and use of SEAFs is being pursued for various use cases. Those frameworks should support a consistent modeling and description of systems of the domain they are stemming from. However, a comprehensive SEAF covering aspects of product, production system and validation system description in an integrated manner appears to be missing, even though those three systems are closely interlinked in product development. Furthermore, specifics of SoS such as connectivity and autonomy, seem to be not comprehensible addressed. In addition, the consideration of different product generations and (re-)use of existing models appears to be not profoundly addressed in existing SEAFs.

In order to harmonize terms to describe SEAFs, we will use the terms viewpoints/views, perspectives and dimension. The term viewpoint/view will be used according to the descriptions in paragraph II.B. Following the description of Weilkiens et al. [16], we use the term perspective to cluster several viewpoints. Following this convention, e.g. the "pillar" requirements of the MagicGrid (see paragraph II.C) would be considered a perspective. In addition, we will use the term dimension to cluster perspectives to an axis of the (visual) representation of the proposed framework.

The goal of this contribution is to identify necessary perspectives to support the modeling of SoS and consolidate them in dimensions of an SoSE architecture framework. The developed framework aims at supporting the modeling of the product and is to be understood as an SEAF, not an EAF. We therefore formulate three research questions (RQ):

RQ 1) Which perspectives are required to describe and model the architecture of SoS?
RQ 2) How can those perspectives be clustered in a set of dimensions?
RQ 3) How can an SoSE architecture framework, that considers all the dimensions, look like?

For the research described in this contribution, we base on the approach by Holt and Perry mentioned in section II.B. Therefore, the development of an SoSE architecture framework encompasses an ontology, a set of viewpoints and associated views as well as an architecture framework to arrange the viewpoints in a reusable structure. As described in the ISO 42010, a viewpoint is governed by the concerns of one or more stakeholders with an interest in the (model of) the system architecture [4]. Furthermore, the concerns expressed to construct the viewpoints serve as a foundation to identify and abstract relevant elements that need to be described in the ontology.

The research for this paper largely takes places in the MoSyS project and is supported by input and several workshops with some of the 18 partners. The partners stem from different backgrounds in engineering (e.g. automotive, mechanical and plant engineering or production system engineering), research and a union. Our iterative research approach is schematically shown in Figure 1.

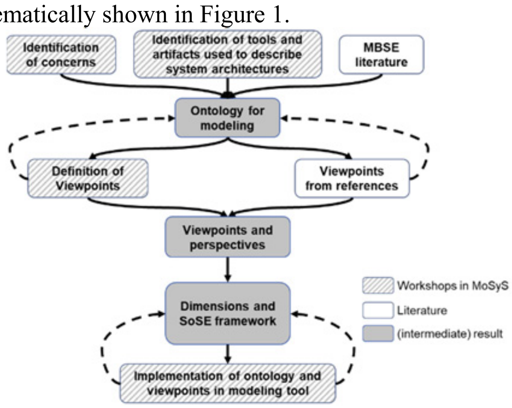

Figure 1: Schematic visualization of the research approach

To answer RQ 1, we initially identify concerns of relevant stakeholders. This was done with two parallel activities. On the one hand, four companies participating in MoSyS filled out templates to directly formulate their concerns on a SoSE architecture framework. On the other hand, eight partners described tools and artifacts they are using at the moment to describe their system architectures. Based on the input from the partners as well as existing approaches described in literature (see section II.C), a first ontology was developed. Using the developed ontology and references from SEAFs from the state of research, multiple workshops have been performed to define viewpoints. Results from these workshops were used to extend and concretize the ontology. In addition, the viewpoints were initially clustered in a set of perspectives.

For RQ 2, we propose a set of perspectives and dimensions for the SoSE architecture framework.

Using the developed dimension, an initial abstract version of the SoSE architecture framework is created to answer RQ 3. Oriented in the framework, the ontology and theoretically described viewpoints/views are implemented in a software-tool environment (iQuavis [17] and Cameo Systems Modeler [18]). The software tools with the implemented ontology and viewpoints/views are continuously used to exemplarily model systems. This exemplary modeling serves as an input to further challenge the ontology, viewpoints and the SoSE architecture framework by using real use cases and systems from the MoSyS partners. The modeling includes a variety of diverse systems and use cases. In this paper, the theoretical foundations of the framework are presented. The case studies and further evaluation in the MoSyS project are still ongoing and will be subject of future research.

IV. Identifying Requirements and Perspectives to Consider for an SoSE Architecture Framework

A. Analyzing stakeholder concerns for a SoSE architecture framework in the project MoSyS

In the process of our research following the descriptions of the last chapter, several aspects were identified that are relevant for modeling across systems and use cases. For example, the representation of the functional, logical and physical architecture as well as the representation of requirements are required across use cases. Likewise, a representation of the time scale as well as the representation of product and development generations is needed. In addition to the hierarchical representation of the system on different levels, a clear separation of the problem space is essential. Furthermore, possibilities to characterize SoS-specific interfaces are required.

While there is a lot of research available for MBSE approaches to model requirements and system architecture, the initial findings from analyzing the state of research (section II) can be concretized. We identified four use cases that have not or only partially been supported by existing SEAFs and will be discussed in the following sections:

- Modeling while considering that the System in Development is part of a SoS or represents a SoS itself (IV.B)
- Modeling while considering that the development of a product and its individual increments is performed in generations and always bases on references (IV.C)
- Integratively modeling of product and production system, enabling product-production co-design (IV.D)
- Integratively modeling of product and validation systems (IV.E)

B. SoS – System of Systems

A SoS is described by characteristics, such as managerially and/or operationally independent constituent systems (CS), evolutionary development and emergent behavior [19, 20]. Since there is no central authority, that coordinates the interaction and development of SoS, it is important to put effort in the problem analysis when developing CS that integrate into an SoS (managerial independence). It is necessary to analyze and anticipate the SoS as good as possible including the structure, goals and behavior of the CS and the stakeholders involved (operational independent). This is described for example by Böhm et al., who propose the goal element as a central aspect, that needs to be considered as part of the SoS analysis [21]. In a SoS, CS interact and transfer information, energy or material in order to achieve outcomes, that a single system cannot achieve by itself (emergent behavior). It is therefore important to describe and analyze the interaction via use cases and structure diagrams that also include interfaces. Based on the problem analysis, requirements can be derived that describe the desired role of the CS in the SoS. These requirements can further be refined as part of the system design process through the solution space. Knowing that SoS will evolve over time, goals and even CS can change (evolutionary development) [22]. It is therefore important to use an iterative approach along with a central model, in which the stakeholder and goals are traced to the CS of the SoS.

As explained in chapter IV.A, the framework concept to consider SoS characteristics was evaluated by examples of the MoSyS project. In particular, problem analysis in terms of an extended analysis of CS and their goals and stakeholders, which may change over time, has proven to be important. For example, cooperation between companies can intensify over time in order to develop interface standards for SoS collaboration and improve interaction in the SoS.

C. SGE – System Generation Engineering

Systems are developed in generations by creating a new system through the deliberate variation of reference system elements [23, 24]. A new system generation (SG or product generation, PG, when referring to a product) is based on at least one reference system element. In addition to system generations, engineering generations (EG) can also be used to represent intermediate stages with evolving maturity in the development of a system generation [25].

To enable efficient and systematic system generation engineering, it is necessary to represent different system generations as well as the associated engineering generations over time and for different system types, such as: product, production system or validation system.

D. Integrated Modeling of Product and Production System

The approach product-production co-design describes a parallelized as well as highly networked creation of products and the production system necessary for it. This involves planning and development over several product generations and the associated production system evolutions. [26]

In order to address product-production co-design, elements and relations are needed that support cross-domain, consistent and traceable modeling of the product-production system architecture. In addition to the cross-system and cross-use case modeling needs (see Section IV.A), it must be ensured that interfaces between product features and production operations as well as the associated resources are represented by suitable relations. Furthermore, product- and production-side requirements and restrictions should also be considered in the modeling across domains.

E. Integrated Modeling of Product and Validation System

Following the definition of VDI norm 2221, validation means *"check as to whether the test results really show what is to be determined by the test"* [27, p. 10]. As described by Albers, validation is the central activity in product engineering and enabler for a product to be successful on the market [28]. However, validation and verification are rarely explicitly addressed in existing SEAFs and their applications [13]. Using SEAFs and, more general, MBSE can however support continuous and early validation, which is a necessity in product development [14, 15]. Therefore, the notion of validation system is introduced, which includes all systems, methods and process which are used of validation during the product development process [20]. MBSE can support in integrating the modeling of product and validation system, creating a traceability to support decision making in the planning and selection of suitable test cases and validation environments as well as in drawing conclusions from test results [14, 29]. Therefore, the integrated and traceable modeling of product and validation system needs to be included in the developed SoSE architecture framework.

V. Developing Dimensions for a SoSE-Architecture Framework

To develop an SoSE architecture framework, that integrates the all of the identified perspectives, we propose different modeling dimensions based on the descriptions from section IV as well as references from the state of research.

The first dimension consolidates perspectives, that are derived from standard literature as well as carved out approaches from industry to describe technical systems using MBSE. Namely this includes problem space, requirements, functional, logical, and physical. We call this the **PS-RFLP-dimension**.

There are multiple perspectives that have a particular focus and need to be analyzed and defined across the architecture, such as safety and security or traceability. These topics are consolidated in the **crosscutting perspectives dimension**.

In addition, we derived different types of systems from the industry examples in MoSyS. These are part of the **system type dimension**, which consists of product, production system and validation system.

A central perspective of systems thinking describes, that systems consist of system elements, which can also be systems themselves. Especially in complex systems with many elements, this leads to several system levels that need to be described. We summaries this perspective in the **hierarchy dimension**.

During the development of systems, architecture descriptions evolve and can be changed over time. Especially in industrial engineering processes, different concept stages are defined in the product development with regard to a defined time. Therefore, we propose a generic **time dimension** to consider the evolution of architectures over time.

The planning of maturity stages in systems engineering is done in engineering- and system generations. The **system and engineering generation-dimension** represents the different system generations.

VI. SoSE Architecture Framework

A. Ontology

An excerpt of the developed ontology as the basis for the SoSE architecture framework is shown in Figure 2. The ontology focusses on relevant terms and their relations to build a common basis for a shared modeling language (as described in section II.B). The displayed excerpt focusses on functional and logical elements to model a product. In this form, the ontology is independent from an implementation in a modeling/tooling environment. However, in MoSyS implementations are performed using SysML and Cameo Systems Modeler as well as iQuavis.

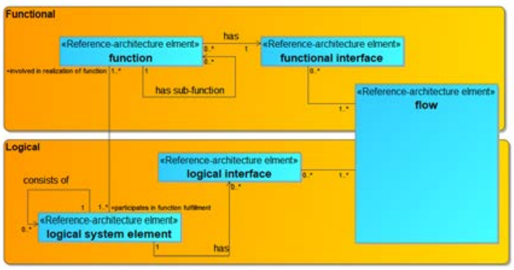

Figure 2: Excerpt of the developed ontology

B. *Model of the SoSE Architecture Framework*

In the following, a first state of the SoSE Architecture Framework is presented, considering all dimensions consolidated in Section V. The representation is oriented on the visualization of the RAMI 4.0 model [12]. For clear presentation, the SoSE architecture framework is divided into three parts and finally presented in a consolidated manner.

For the first part, Figure 3 shows three dimensions: “System Hierarchy”, “System Type”, and “Time”. The first axis maps the dimension of the system hierarchy. The number and detail of the system levels is usually company-specific. The second axis shows the dimension of the system types. This can be used to describe the parallel development of e.g. product, production system, validation system and other system types. The third axis represents the time dimension, which represents the life cycles of the various systems.

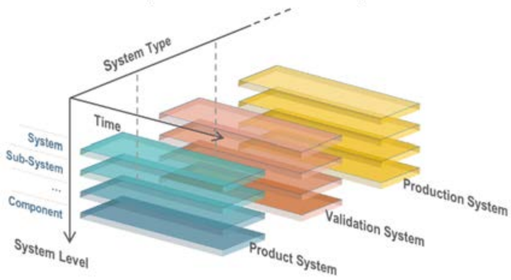

Figure 3: Representation of the dimensions: System Type, Time and System Level

Figure 4 illustrates relevant dimensions within a system type, e.g. Product System. Here, in addition to the already presented dimensions "Time" and "System Level", the dimensions "System Generation" and "Engineering Generation" are introduced. "Engineering Generations", representing the maturity level of a system, are arranged within a "System Generation" over the time dimension. Thus, a parallel development can be represented.

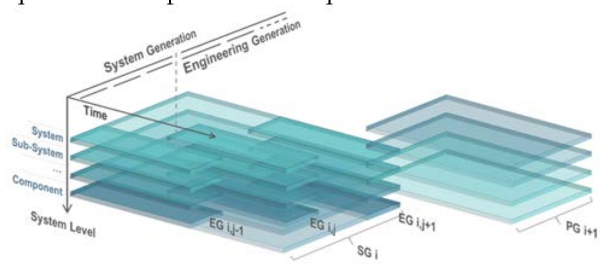

Figure 4: Representation of the dimensions: System- (SG) and Engineering- (EG) Generations, Time and System Level within a System Type

Within an engineering generation, each system level of a system is described by two dimensions (see Fig 5). Here, the "Time” dimension is not represented. The PS-RFLP dimension introduced in Chapter V, which is based on frameworks from the literature such as MagicGrid, is supplemented by another dimension, the Crosscutting Aspects, to which any number of additional aspects can be added. This enables the location of application-specific, system-specific and system-typical crosscutting views, e.g. regarding traceability. In the abstraction area "Problem Space" SoS aspects are considered

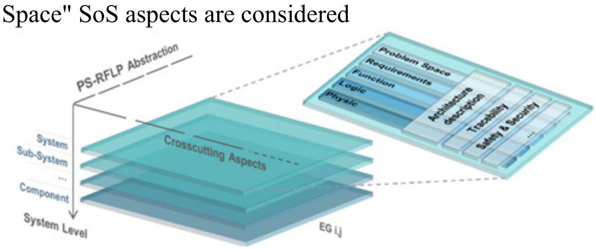

Figure 5: Representation of the dimensions: PS-RFLP Abstraction, Crosscutting Perspectives and System Level within a Engineering Generation.

Figure 6 shows the consolidated SoSE Architecture Framework with all dimensions presented before.

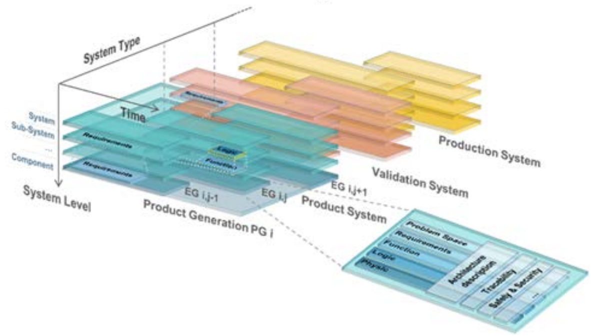

Figure 6: Consolidated SoSE Architecture Framework

VII. STATEMENT OF CONTRIBUTION

The developed SoSE architecture framework and its visualization aim at gathering different dimensions for modeling of SoS in an integrated description. The dimensions have been consolidated from existing SEAFs and in various workshops and applications with academia and industry in the project MoSyS. Our analysis showed, that no existing SEAF covers all dimensions necessary for modeling SoS.

The overarching and abstract framework supports the integration of different system types, such as product, production system and validation system, and can be extended to include additional system types. The framework considers the integration of system- and engineering-generations across the time axis and supports the modeling of different system levels. In addition, possible SoS characteristics are systematically considered in the problem space of the individual system types. The SoSE architecture framework can therefore serve as a reference to enable researchers and companies to derive and build specific frameworks tailored to their needs. The reference supports the consistency, reusability and compatibility of the specific frameworks as it establishes dimensions and perspectives necessary for the description of their specific systems.

VIII. DISCUSSION AND OUTLOOK

The developed SoSE architecture framework and the underlying ontology are still being refined based on feedback from their application in MoSyS. Especially, the tailoring of frameworks to a specific application area that base on the SoSE architecture framework is still under investigation. Therefore, future research will further investigate case studies for application and tailoring of the framework with examples from industry. To supplement the analysis from application, a more detailed comparison of similarities, interfaces but also differences of the developed framework and existing SEAFs and EAFs like TOGAF or UAF will be subject of future research. In addition, implementation of the framework in software tools needs to further demonstrate the usability for companies and serves as a valuable source for refinement. In this way, the necessity to refine or add dimensions or aspects therein is being investigated. Another area of research focusses on the interaction of the developed framework and models for process descriptions, as e.g. from the VDI 2221 [30] or the Systems Engineering Handbook/ISO 15288 [3].

IX. ACKNOWLEDGEMENT

This article is part of the research project MoSyS - Human-Oriented Design of Complex Systems of Systems. This project is funded by the German Federal Ministry of Education and Research (BMBF) within the "The Future of Value Creation – Research on Production, Services and Work" program (funding number 02J19B106 - 02J19B090) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the content of this publication.

X. REFERENCES

[1] R. Dumitrescu, A. Albers, O. Riedel, R. Stark, and J. Gausemeier, Eds., "Advanced Systems Engineering - Value Creation in Transition: Engineering in Germany – Status quo in Business and Science," Paderborn, 2021. Accessed: Jan. 4 2022. [Online]. Available: https://www.advanced-systems-engineering.de/

[2] S. Friedenthal *et al.,* "Systems Engineering Vision 2035: Engineering Solutions for a Better World," INCOSE, 2021. Accessed: Feb. 10 2022.

[3] D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and T. M. Shortell, *SYSTEMS ENGINEERING HANDBOOK - A GUIDE FOR SYSTEM LIFE CYCLE PROCESSES AND ACTIVITIES*. Hoboken, NJ, USA: John Wiley & Sons, Inc, 2015.

[4] *42010-2011 - ISO/IEC/IEEE Systems and software engineering -- Architecture description*, Piscataway, NJ, USA.

[5] J. Holt and S. Perry, *SysML for Systems Engineering A Model-Based Approach A model-based approach*. Stevenage: Institution of Engineering and Technology, 2018. Accessed: Dec. 2 2021.

[6] H. G. Gurbuz and B. Tekinerdogan, "Analyzing Systems Engineering Concerns in Architecture Frameworks – A Survey Study," in *2018 IEEE International Systems Engineering Symposium (ISSE)*, Rome, 2018, pp. 1–8.

[7] A. Morkevicius, A. Aleksandraviciene, D. Mazeika, L. Bisikirskiene, and Z. Strolia, "MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems," *INCOSE International Symposium*, vol. 27, no. 1, pp. 136–150, 2017, doi: 10 1002/j.2334-5837.2017.00350.x.

[8] K. Pohl, M. Broy, H. Daembkes, and H. Hönninger, *Advanced Model-Based Engineering of Embedded Systems*. Cham: Springer International Publishing, 2016. Accessed: Dec. 9 2019.

[9] P. Roques, "MBSE with the ARCADIA Method and the Capella Too," in *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, 2016.

[10] M. Eigner, T. Dickopf, M. Schneider, and T. Schulte, "MECPRO² - A HOLISTIC CONCEPT FOR THE MODEL-BASED DEVELOPMENT OF CYBERTRONIC SYSTEMS," in *21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN (ICED17) Vol. 3 Product, Services and Systems Design*, Vancouver, Canada, 2017. Accessed: Dec. 15 2020.

[11] C. Mandel, T. Stürmlinger, C. Yue, M. Behrendt, and A. Albers, "Model-Based Systems Engineering Approaches for the integrated development of product and production systems in the context of Industry 4.0," in *SysCon 2020 14th Annual IEEE International Systems Conference*, IEEE, Ed., 2020.

[12] Plattform Industrie 4.0, "RAMI4.0 - a reference framework for digitalisation," Aug. 2018. Accessed: Feb. 11 2019. [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.html

[13] R. Cloutier, "2018 Model Based Systems Engineering Survey: Conducted December 2018," in *Proceedings of the 2019 INCOSE MBSE Workshop*, Torrance, CA, USA, 2019. Accessed: Jan. 14 2022.

[14] C. Mandel, J. Böning, M. Behrendt, and A. Albers, "A Model-Based Systems Engineering Approach to Support Continuous Validation in PGE - Product Generation Engineering," in *7th IEEE ISSE International Symposium on Systems Engineering 2021*, 2021, n.p. Accessed: Sep. 14 2021.

[15] A. Albers, M. Behrendt, S. Klingler, N. Reiss, and N. Bursac, " Agile product engineering through continuous validation in PGE - Product Generation Engineering," *Design Science Journal*, vol. 3, 2017.

[16] T. Weilkiens, J. G. Lamm, S. Roth, and M. Walker, *Model-Based System Architecture*. Hoboken, NJ, USA: John Wiley & Sons, Inc, 2015. [Online]. Available: http://doi.wiley.com/10.1002/9781119051930

[17] TwoPillars, *iQuavis*. [Online]. Available: https://www.two-pillars.de/iquavis/ (accessed: Mar. 18 2022).

[18] Dassault Systèmes, *Cameo Systems Modeler*. [Online]. Available: https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/ (accessed: Mar. 18 2022).

[19] C. B. Nielsen, P. G. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, "Systems of Systems Engineering," *ACM Comput. Surv.*, vol. 48, no. 2, pp. 1–41, 2015, doi: 10.1145/2794381.

[20] A. Albers, C. Mandel, S. Yan, and M. Behrendt, "SYSTEM OF SYSTEMS APPROACH FOR THE DESCRIPTION AND CHARACTERIZATION OF VALIDATION ENVIRONMENTS," in *DS92 Proceedings of the DESIGN 2018 15th International Design Conference*, 2018, pp. 2799–2810.

[21] W. Böhm, M. Broy, C. Klein, K. Pohl, B. Rumpe, and S. Schröck, *Model-Based Engineering of Collaborative Embedded Systems*. Cham: Springer International Publishing, 2021.

[22] J. Dahmann and G. Roedler, "SYSTEMS OF SYSTEMS ENGINEERING STANDARDS," *INSIGHT*, vol. 19, no. 3, pp. 23–26, 2016, doi: 10.1002/inst.12102.

[23] A. Albers, N. Bursac, and S. Rapp, "PGE – Product Generation Engineering – Case Study of the Dual Mass Flywheel," in *Proceedings of DESIGN 2016*, Dubrovnik, 2016, pp. 791–800.

[24] A. Albers *et al.,* "The Reference System in the Model of PGE: Proposing a Generalized Description of Reference Products and their Interrelations," in *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, Delft, The Netherlands, 2019, pp. 1693–1702.

[25] A. Albers, J. Fahl, T. Hirschter, M. Endl, R. Erwert, and S. Rapp, "Model of PGE – Product Generation Engineering by the Example of Autonomous Driving," in *CIRP 2020 – CIRP Design Conference 2020*, 665-677.

[26] A. Albers *et al.,* "Product-Production-CoDesign: An Approach on Integrated Product and Production Engineering Across Generations and Life Cycles," in *32st CIRP Design Conference 2022 (CIRP Design 2022)*, 2022.

[27] *Entwicklung technischer Produkte und Systeme - Modell der Produktentwicklung Blatt 1*, VDI 2221, VDI, Nov. 2019.

[28] A. Albers, "Five Hypotheses about Engineering Processes and their Consequences," in *Proceedings of the TMCE 2010*, 2010.

[29] C. Mandel, M. Wäschle, S. Lutz, and M. Behrendt, "A Model-Based approach for automation and traceability of validation activities – clarified for Advanced Driver Assistance Systems," in *9th Int. Symposium on Development Methodology*, Wiesbaden, 2021, n.p.

[30] *Design of technical products and systems Configuration of individual product design processes Part 2*, VDI 2221, VDI, Nov. 2019.