Investigation of Problems with High Initial and Update Efforts in the Modeling of Production Systems

A Review on System Modeling Approaches

Marius Heinrichsmeyer[1], Amirbabak Ansari[2], Nadine Schlueter[3], Christian Boehmer[4]
Product Safety and Quality Engineering
University of Wuppertal
Wuppertal, Germany
E-Mail: heinrichsmeyer@uni-wuppertal.de[1], aansari@uni-wuppertal.de[2], schluete@uni-wuppertal.de[3], christian.boehmer-hk@uni-wuppertal.de[4]

***Abstract* — The need to use Model-Based Systems Engineering (MBSE) has seen an upswing, especially in recent years, for example, due to the ever-increasing complexity of products and production systems. Nevertheless, evaluations of the current state of research and experience from our own completed and ongoing DFG projects (KAUSAL, ReMaiN, and FusLa show that the use of MBSE in the industry is underestimated, mostly because of the enormous initial and update efforts in the modeling. Approaches that support system modeling, such as Modelica, SysML, and eDeCoDe or approaches for their partial automation only help to a limited extent to reduce the modeling effort when mapping production systems. For this reason, the research group of Product Safety and Quality (PSQ) intends to research possibilities and opportunities for partial automation in the modeling of production systems. To achieve this, the problem of excessive initial and update efforts when using MBSE explicitly in the modeling of production systems should first be highlighted and developed as research potential.**

Keywords-Model Based Systems Engineering; Partial Automation; Failure Cause Localization; Production.

I. INTRODUCTION

Following our paper in ICONS 2020 about the validation of a Failure-Cause Searching and Solution-Finding Algorithm (FusLa) in production, it was stated that a detailed production system model forms the basis of localization of failure causes [1]. In this paper, the initial effort of system modeling and its updating is investigated.

System models can be used for many purposes, including the visualization of production systems. They are particularly important in order to master the increasing complexity of product and production systems as part of MBSE [2][3][4]. As a simplified representation of a complex system, system models form the basis for the design and improvement of processes according to failures and previous analyzes. However, the initial and update effort for creating a system model and the effort for the introduction and application of systems engineering is enormous, since companies have to use many tools or toolchains to be able to correctly map the complex information [5]. This effort shows itself particularly in high personnel costs and a considerable amount of time expenditure. In the coming years, a further increase in the resources, which are required for modeling the production system, is to be expected. It is because of the increasing number of components and their connectivity with each other and also the increasing variety of requirements, while the development and testing times for products or production systems are reducing [6]. Existing approaches to partial automation of the creation of system models are very specific and only consider just some aspects of the overall system, such as the requirements [7]. So, they cannot be used for a holistic system description. To reduce the initial effort for the creation and then the maintenance of a system model for companies and to reduce the resource expenditure, it is necessary to develop a practicable and scientific approach, with which systems can be modeled partially automated based on existing documents and information. However, in order to be able to implement such a development, three key questions need to be asked:

1) How does the modeling of a production system work?

The second section of this paper looks at how a production system can be represented as a model, and which elements are necessary for this. This is necessary since there are various considerations regarding the representation of models. Some approaches consider production systems as the interaction of the subsystems, while others consider inputs and outputs as well. Section II is primarily intended to describe the different forms of modeling of production systems and to specify their use cases.

2) Which approaches contribute to the modeling of a production system and how much effort is required?

Based on the modeling forms, the next step is to question, which approaches to modeling are already available and how they contribute to the mapping of a production system. This will not only indicate the limits of existing approaches regarding the modeling of production systems but also show the initial and update effort associated with their modeling. Overall, this makes it possible to determine a statement, to what extent the mentioned problems and efforts are already compensated or intensified by existing approaches.

3) Which approaches already contribute to the reduction of the initial and update effort. Are these sufficient?

In the last step, based on the initial and update efforts of each approach, it is then examined, which existing approaches already contribute and can contribute to the reduction of mentioned efforts. This step will provide a statement about whether current approaches are sufficient to eliminate the mentioned problem, or whether there must be further research projects and new approaches to be developed, which can contribute to an elimination of the problem.

To investigate these questions, Section II gives an overview of the types of system modeling. Section III discusses the state of the art in modeling approaches that deal with standardized modeling of systems and Section IV discusses those that contribute to partially automated modeling. Finally, Section V gives an overview of the research topics to be pursued.

II. MODELING OF PRODUCTION SYSTEMS

In the literature, there are numerous definitions of the term model, which originate from different industries and fields of application. What they have in common is that a model is an abstract representation of reality [8]. The systematic creation and the integrated use of digital system models in the context of the MBSE serve the purpose of making the increasing complexity of products and processes manageable [3][9]. However, how is the modeling of a production system accomplished?

Remarkably similar to the concept of the model, the related process of modeling is also defined in many ways. For the modeling of production systems, however, the modeling focuses on three main forms of representation, including functional, hierarchical, and structural modeling. Which form of presentation is most suitable depends largely on the object under consideration and the application [10].

1) Functional modeling

The functional form of modeling considers a model at the top level. As shown in Figure 1, this form of modeling models a production system as an operational conversion and transformation process, by which a set of outputs (e.g., products or services) is created from a set of inputs (e.g., material, energy) through the work of human and/or the use of work equipment [11]. This form of modeling is particularly suitable if a holistic view of the production system concerning other systems, such as product development or top-level use, should be achieved over the product life cycle [10].

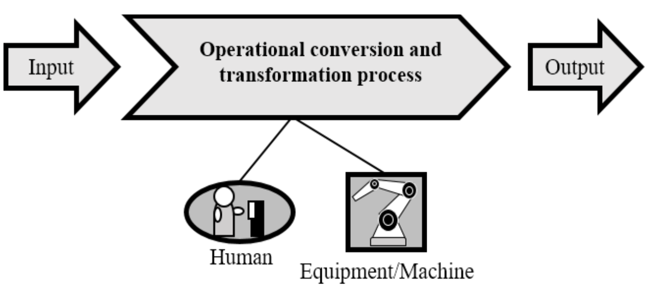

Figure 1: Functional modeling form of a production system.

2) Hierarchical modeling

The second form of representation of the modeling is called hierarchical modeling and covers production systems via subordinate and superordinate subsystems. In contrast to functional modeling, in which the highest level of detail is considered, hierarchical modeling already shows the first relationships between subsystems in more detail. This form of modeling is particularly suitable when the interaction of higher-level processes in the production system, e.g., purchasing or manufacturing, is to be analyzed. Above all, the recording of material and information flow is possible with this form of modeling [10].

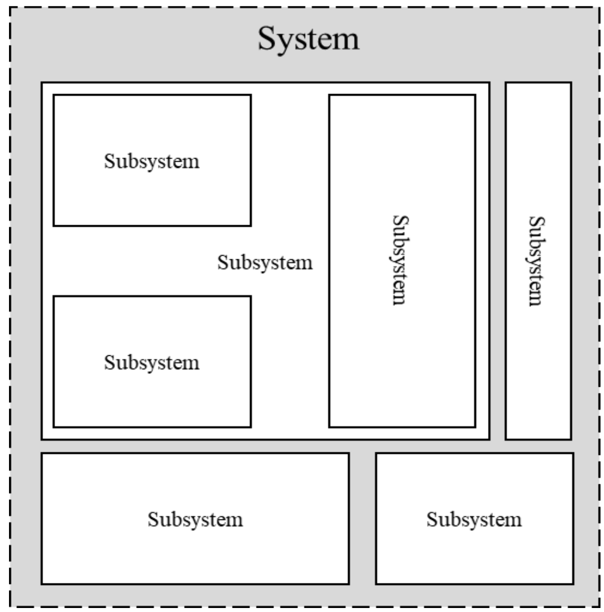

Figure 2: Hierarchical modeling form of a production system.

3) Structural modeling

Structural modeling represents the last form of modeling of production systems. Here, the production system is divided into different components, including system elements, their relations, inputs, outputs, the system environment, and the system boundary. This is the most detailed form of modeling. This is particularly suitable for understanding the interrelationships between different system elements and making the complexity of a holistic production system more manageable. In addition, this amount of detail makes it possible to ensure the traceability of system elements by evaluating their relationships [10].

As already mentioned, the selection of a suitable form of representation of the modeling largely depends on the object under consideration and the application. This suggests that the elements that are required to map a standardized production system model also vary on a case-by-case basis. However, experience from previous fundamental research projects, such as KAUSAL and in part, ReMaiN, showed that the structural modeling form, in particular, can be classified as suitable when it comes to analyzing and understanding the interrelationships within production.

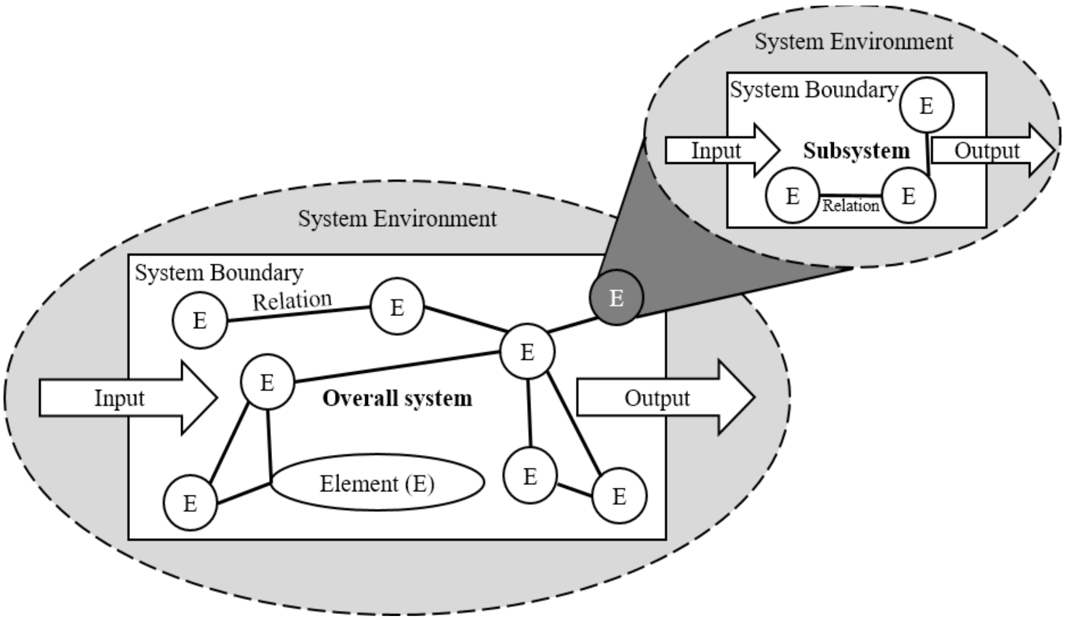

Figure 3: Structural modeling of a production system [12].

Nevertheless, this is also to be assessed disadvantageously, since the high level of detail of the structural modeling also entails an enormous challenge for the companies. The challenge is particularly noticeable in the initial and update effort already mentioned. Figure 3 demonstrates an exemplary structural system model and its elements

Specifically, structural modeling means that every system element, be it a machine, a person, or the input and output, must be recorded and related. Especially with extremely complex production systems, such as those found in the automotive industry, such modeling could hardly be carried out by individual people. Instead, individual partial models from different areas are developed. However, these are designed for a specific problem and do not help to understand the holistic production system model in detail. In order to counteract this problem and to simplify the modeling itself, different modeling approaches have been established in recent years. These specify which system elements are to be classified as necessary for the modeling and how their interrelationships are to be understood. The main aim of these approaches is to make the complexity of the production systems more manageable through suitable and, above all, less complex modeling.

To evaluate these approaches regarding their suitability concerning the modeling of production systems and their effort, some established approaches are presented below and critically examined. The subject of consideration is structural modeling, since, as already mentioned, this involves the greatest initial and updating effort.

III. Approaches to Mapping the Standardized Production System Model

Approaches that are considered in the context of the contribution are Modelica, CONSENS (Conceptual design Specification technique for the Engineering of Complex Systems), SysML (Systems Modeling Language), MES (Manufacturing Execution System), and Demand Compliant Design (DeCoDe). The initial and update effort was assessed after practical application of the respective approaches and is summarized using the assessment scheme ● = high effort, ◑ = moderate effort, and ○ = little to no effort.

A. Modelica

The first approach, “Modelica”, enables object-oriented modeling of complex heterogeneous systems. For this purpose, a description defined by a language code is translated using hierarchical object diagrams specified by a library. The interrelationships between the elements must always be physical [13][14].

Modeling with Modelica has both advantages and disadvantages. On the one hand, it is a simulation tool that enables the quantitative analysis of system behavior within the usage phase. A combination with other methods such as Fault Tree Analysis or Markow models can be implemented and the visualization also is not limited to a single medium. On the other hand, only the component view is considered in the visualization. Therefore, a statement regarding the involved functions, processes, and requirements cannot be made. This in turn means that the traceability of failures cannot be guaranteed. Regarding the effort involved in structural modeling, it was found that this, of course in direct comparison with other approaches, should be assessed with a moderate effort (◑). The background of this assessment lies in the focus on the component view. While other approaches consider other system elements, such as requirements or processes, and also take their interrelationships into account, the model with Modelica captures only one type of system element.

B. CONSENS

CONSENS is a specification technique used to describe the principle solution of mechatronic systems and the associated production system [15]. With this approach, ten partial models are defined, seven of which, as shown in Figure 4, describe the problem solution (environment, application scenarios, requirements, functions, active structure, shape, and behavior) and the remaining three (processes, resource, and shape) describe the production system. The language uses a visual syntax and since the semantics are already defined, it can be used effectively without any adjustments. This can also be extended via profiles [16].

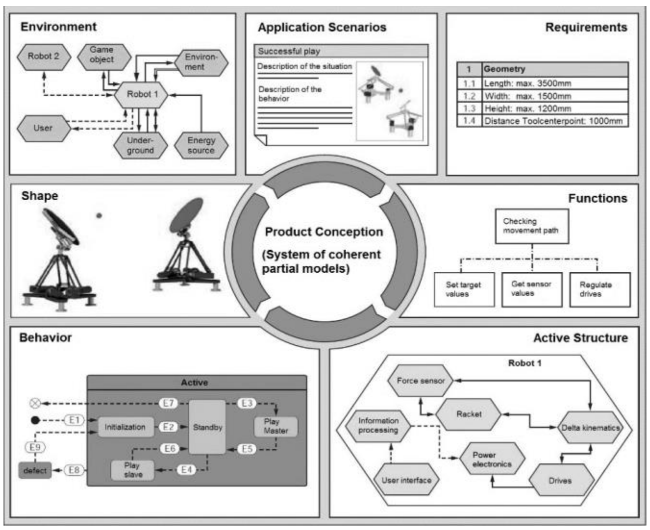

Figure 4: CONSENS approach according to [17].

In this model, requirements are listed, classified, and connected with the functions and system elements. The structure and the mode of action are represented by the structure of action, the core of the model [18].

One advantage of this model is that it forms a basis for discussion and documentation, especially in the planning phase. On the other hand, there is a connection between the views of the requirements, functions, and components. In comparison to Modelica, CONSENS records the behavior of the system model with the help of application scenarios. A disadvantage is that although there is a network being formed, there is no consideration of its interrelations. Besides, due to the numerous and, above all, extensive diagrams, the overall model quickly becomes confusing and even more complex. It should also be added that traceability is only partially guaranteed with this model. Regarding the initial and update effort with CONSENS, one can see that the modeling is of high effort (●). The acquisition of all system elements via the corresponding partial models as well as continuous updating by changes to the system are extremely resource-intensive. Above all, taking system behavior into account via corresponding application scenarios can be classified as a great effort, since the scenarios have to be individually adapted to the respective production system models.

C. SysML

SysML is a modeling language based on the Unified Modeling Language (UML) [19]. In contrast to CONSENS, SysML visualizes additional elements (e.g., requirements and functions) and offers a modeling of use cases as well as further possibilities [19]. It has its own notation so that system elements and relationships can be assigned. SysML is widely used because it is highly extensible and adaptable to the respective development task, e.g., through ready-made profiles [20]. However, adaptability is also necessary, since the semantics contained in SysML are only rudimentary compared to less frequently used alternatives [16].

As shown in Figure 5, the system model is characterized by various diagrams (e.g., diagrams of structure, behavior, requirements, parameters, and use cases) [21].

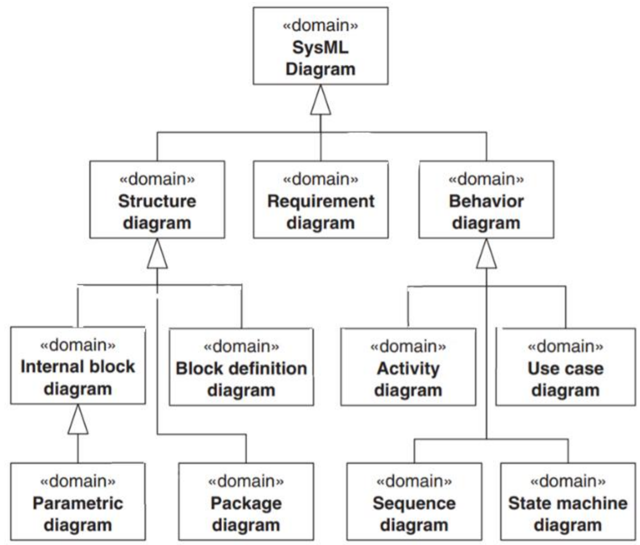

Figure 5: SysML diagrams according to [21].

The relatively large number of diagrams makes it possible to visualize the system model from different perspectives. At the same time, however, this is also a disadvantage, since the enormous number of diagrams and their defined structures do not allow intuitive use [19]. In addition, SysML was originally used in software development and later adapted for product development and is therefore not suitable for modeling production systems.

The application of SysML also involves a high effort (●). Although SysML can be simplified by supporting software systems such as Cameo Systems Modeler, numerous diagrams must be worked out and related to each other. The advantage of SysML, but not the decisive factor, is that the system elements are available across the diagrams. This means that when an explicit system element is changed, all system elements with the same identifier will also change. Above all, this reduces the update effort, since not every system element has to be changed individually.

D. User-oriented System Modeling

Florian Munker presents in [22] his approach to user-oriented system modeling. It aims at developing a concept that allows an easy entry into interdisciplinary system modeling while maintaining agility and flexibility. By determining boundary conditions and based on different approaches investigated, a user-oriented and integrated initial approach was developed, which should consist of language, method, and tool. This approach was then presented at the Systems Engineering Day 2015 [23].

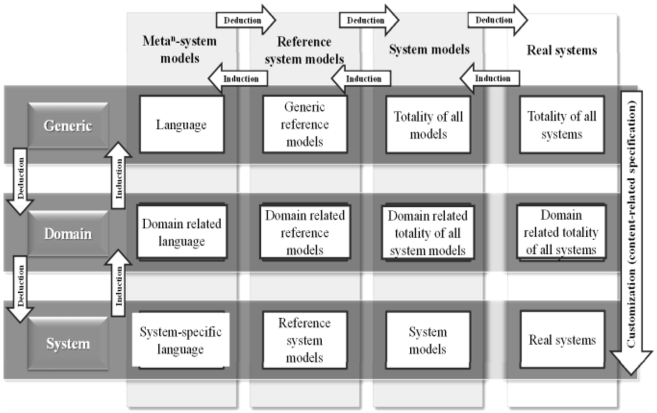

Figure 6: Abstraction of the system model [22].

As shown in Figure 6, a framework should be used, which should enable access to essential information on past product generations by accessing older system models. Thus, the modeling effort shall be reduced by transferring this information. The prototype worked was then translated into program codes by an assistant within one year. The different modules include the reading of the project and the Metadata, the graphical representation, the processing of the information according to the user stories, which served as requirements, and the saving of the data. However, the prototype was developed with some limitations, so that only the boundary conditions identified as mandatory were considered. These include user-oriented object modeling, graphical modeling, and view generation. The restrictions are thus "essentially the limitation of the realization to the partial model 'system structure' so that a fundamental system modeling can be tested on it [22 p. 70]". Furthermore, the modeling of the remaining partial models has been simplified. The created prototype was then used and evaluated by a test group. Part of the application study was also the Graphical User Interface, which is divided into a graphical modeling interface, buttons for modeling partial models and features, an administration area, and an area for the structure trees of the partial models. The bottom line is that this type of modeling also involves a high effort (●) and the suitability in practical application is rated as low, while the necessity of such an application and the potential of this prototype are confirmed.

E. MES

Another approach that is already established, especially in industry, is MES. These systems form an interface between the planning systems used, including ERP for example, and the equipment or personal interfaces present in the production systems. MES systems are primarily used to capture all processes in production systems, e.g., which equipment produces which product, to process them in real-time and to control them accordingly. This makes it possible to determine the process capability of running processes throughout the entire production system and to initiate measures to restore process capability in case of any deviations. In addition, material bottlenecks, e.g., in value creation with suppliers, can also be detected and compensated for at an early stage. The corresponding modeling of MES can vary depending on the company. While some companies embed CAD models of the facilities into the production system model, other companies only consider data evaluation [24]. Overall, however, it can be said that MES is quite capable of capturing corresponding processes, facilities or requirements to be implemented. However, an extensive acquisition of the persons including their competencies is missing. This has the background that MES systems are currently not yet developed for the optimization of people in the production system model, but focus primarily on the optimization of the process view [25].

The effort of MES modeling, especially concerning the initial implementation, is a challenge for companies. To be able to work with MES, companies must have a corresponding infrastructure within their production system. This means that the data of the facilities and machines must also be accessible and personal interfaces must be available. If this is not the case, massive intervention in the actual production system is required first. For this reason, the effort involved in dealing with MES is also classified as very high (●).

F. eDeCoDe

eDeCoDe is an approach for the standardized description of a sociotechnical system model under the principles of systematical thinking and acting [26][27]. The eDeCoDe model is used to mentally decompose sociotechnical systems into five different views. These include requirements (R), functions (F), processes (P), components (C), and persons (Pe) of the system under consideration. These views are arranged in the form of matrixes, which are linked to each other. There are also some tools and questions that are provided to help capture these links. eDeCoDe is a procedure for creating a transdisciplinary system model [26].

The eDeCoDe tools, including the Design Structure Matrix (DSM), Domain Mapping Matrix (DMM), and Multi-Domain Matrix/Multi-Domain Graph (MDM/ MDG), statically map the technical system under analysis. By adding the fifth view, the eDeCoDe tools also make the modeling and investigation of sociotechnical systems possible. The DSM allows the qualitative capture of different elements of the same view (e.g., functions).

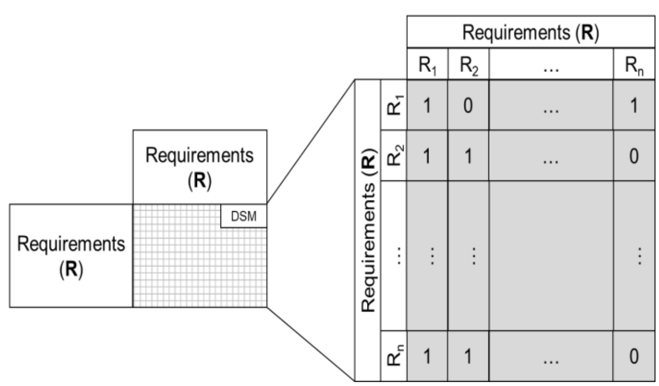

Figure 7: The DSM Matrix (Requirements View) [27].

As shown in Figure 7, by listing all elements equally on the axes of the square matrix, interrelationships between the elements can be identified using the notation 1 = relationship, 0 = no relationship [26] [28] [29].

The DMM is an extension of the DSM. While the DSM only considers the elements of a single same view, the DMM comprises the elements of two different views (e.g., functions and requirements). This makes it possible to capture the interrelationships between the elements of the views and thus also to link the views with a visualized notation [30].

As shown in Figure 8, the combination of DSM and DMM is called MDM. Similar to DSM, MDM is also a square matrix with equal axes, but this time it captures all views (requirements, functions, processes, components, and people), elements, and relations. By representing each element of the system through the views, it enables the derivation of indirect dependencies of the system elements under consideration.

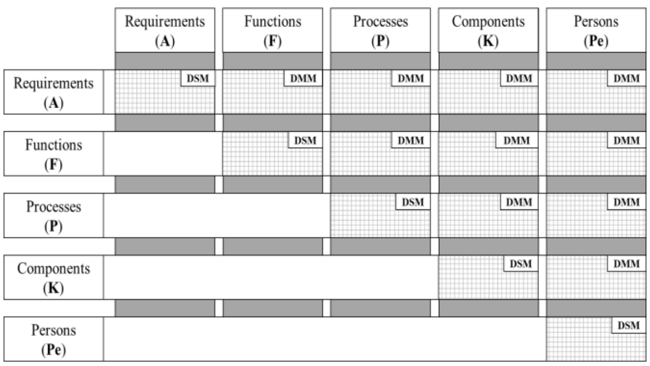

Figure 8: Combination of eDeCoDe Matrixes according to [27].

An advantage of the DeCoDe tools is that the system does not have to be completely mapped before it can be analyzed and designed [26]. This results in a system model that is reduced in complexity, although according to [31], this is associated with increased environmental complexity for this system. Furthermore, it is possible to illustrate the results resulting from the matrices in the form of graphs, so that the understanding of complex issues can be simplified by this kind of modeling [30].

The application of eDeCoDe has also proven to be extremely complex (●). The background of this is that each system element must first be worked out separately and then, in a further step, they will be related to each other. With extremely complex and continuously changing production systems, this task seems to be almost impossible to be accomplished by individual employees. Similar to SysML, eDeCoDe can also be supported by appropriate software in the actual process, such as LOOMEO. However, the initial and update effort remains almost identical.

After evaluating the corresponding efforts by applying the respective approaches to structural modeling, the result seems to show clearly that Modelica, in terms of effort, seems to be the most suitable. Nevertheless, at this point, it is necessary to critically question whether Modelica is sufficient to describe a production system model holistically since it only represents the component view. So capturing of interrelated processes or requirements is completely absent. However, this is necessary if an analysis of the facts within the production system is to be carried out. For this reason, the evaluation allows the statement that Modelica is not sufficient to model a production system, despite the lower initial and update effort required for modeling. However, which of the approaches then seems to be the most suitable concerning the respective effort involved?

G. *Which approach is best suited for modeling a production system model?*

As already mentioned, structural modeling varies according to the consideration and application of the model. Therefore, to answer the above question, it must first be clarified, which object of consideration and application is involved. These can always be different. Thus, the production system model can be used to evaluate the information flows regarding data protection or to identify the causes of failures in the model based on detected failures in the use phase. Despite the variation of the objects of consideration and use cases, the literature shows that a model of a production system can be considered from five standardized views [32]. The five views, visualized in Figure 9, are a superordinate grouping of the individual system elements of the model. These include requirements (R), processes (P), people (Pe), functions (F), and components (C). According to [26], these views are necessary to represent a sociotechnical system, including a production system, in its entirety [33]. Besides, these views enable the traceability of individual system elements via the interrelationships within the production system model.

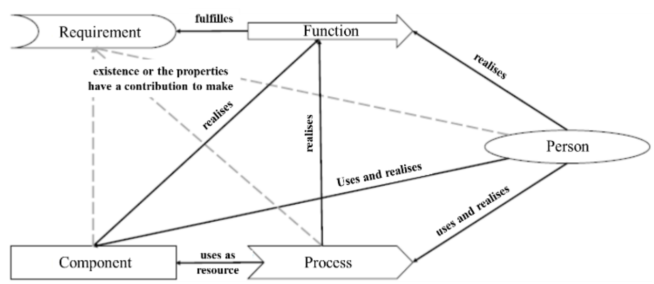

Figure 9: Interrelationships of system elements in eDeCoDe [27]

Based on this prerequisite, the eDeCoDe approach offers the greatest potential for structural modeling of a production system model. Despite the possibilities of eDeCoDe, it has already been shown that this approach involves an enormous initial and update effort. Therefore, the eDeCoDe approach is certainly suitable to make the complexity of a production system more manageable. Nevertheless, its modeling poses a great challenge to companies in terms of the effort involved. To compensate for this challenge, approaches were researched and evaluated, which can contribute to the partial automation of eDeCoDe modeling. The aim was to investigate whether partial automation of such a modeling is already possible, or whether the problem of excessive initial and update efforts in the modeling of production systems still exists.

In order to evaluate these approaches about their suitability for the partial automation of the modeling of production systems and concerning their limits and effort, some established approaches are presented and critically questioned in the following. The object of consideration is the structural modeling with eDeCoDe, since this, as already mentioned, involves the greatest initial and updating effort.

IV. Approaches to Partially Automated Modeling

In the literature, some individual approaches can be used to support the modeling of production systems. Especially the aspect of partial automation is considered to have great potential. Approaches that are included are, e.g., AAES and ARIS, which are described in detail below.

A. *AAES – Requirements View*

AAES is a method, with which the step from document-based to model-based requirements engineering (RE) as a starting point for MBSE is facilitated. With this method, specifications can be automatically broken down into individual requirements, which are subject to comprehensible versioning and are efficiently transferred to RE tools [34]. Finally, this also serves to quickly evaluate new requirements and initiate the implementation of these. Thus, the efficiency can be increased and at the same time, an increased acceptance of the changes by the users can be achieved. The starting point for the development was that many requirements are currently still stored in text-based documents that cannot be read by MBSE tools. If these continuous texts are now to be transferred to RE tools or modeling tools, this would mean that all requirements would have to be transferred manually. According to [34], this would go hand in hand with reduced quality and speed of the transmission, reduced profitability, and reduced user acceptance and motivation. However, since AAES can automatically read PDF-based documents, such as the specifications document, and forward them in ReqIF format to RE tools, which in turn can be linked to modeling tools, these effects can be counteracted preventively.

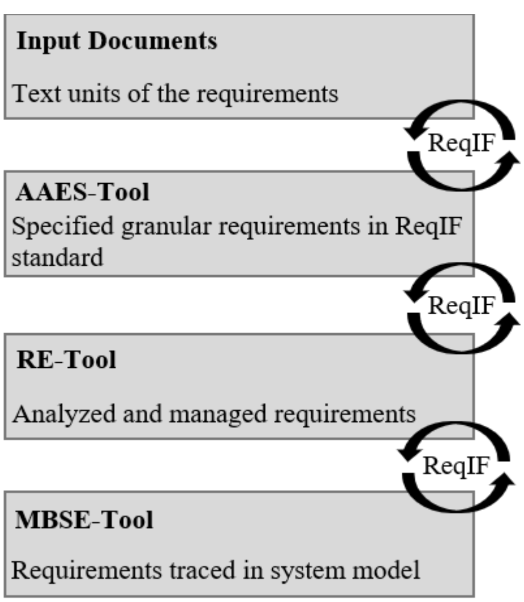

Figure 10: Process of requirements work from text to system model according to [34].

Besides, due to the growing complexity and its dimensions of variety, connectivity, dynamics, and globalization, a company must be able to act agilely and flexibly and at the same time guarantee traceability [35]. This means the networking of requirements with the product structure, tests, and the "atomic requirements gathering" are more relevant than ever [26][34]. The prerequisite for AAES is that requirements documents must be a structured set of data created and stored as a unit. If this requirement is met, the process of transfer based on the INCOSE manual or the phases of the V-Modell can be initiated. First, stakeholder requirements must be defined for this purpose, followed by a requirements analysis. This is followed by the architecture design, the design definition, and finally the system analysis.

B. *Analysis-simulation Models*

This approach is intended to contribute that reduces the manual effort required for simulation-based analyses. System simulations combined with fault injections can be used, for example, to support an FMEA, i.e., to assess the reliability of systems. Such a procedure is also recommended in ISO 26262:2015 "Functional safety of motor vehicles" to estimate the achieved Automotive Safety Integrity Level (ASIL). Model-Driven Development techniques are used for the specification of the failure effect simulation so that the effort of the failure effect simulation can be reduced by automated code generation and efficient reuse of simulation models using a component library. The effort of documenting the analyses according to ISO 26262 is also reduced. The UML profiles are also used because some extensions such as SysML and MARTE are already established in the automotive industry [36]. The connection to existing modeling languages is done with Model-to-Model transformation techniques (M2M). Furthermore, code generation techniques are used to automatically generate the structural part of the program code from the class descriptions: The code is highly reusable so that only the functional part of the code has to be added manually. A kind of top module instantiates, configures, and links the models of the simulation. The linking of the analysis results with the specifications of the system models can be done in two ways, semi or fully automatic.

This approach is also pursued in other methods. For example, there are overlaps with the method described in [37]. This approach presents a method of automatic generation of simulation models for production planning. It allows the automatic generation of simulation models of production systems based on data from the production planning and control system (PPC system). Thereby methods of data mapping, data transformation, data storage, and an intermediate data model are used. Thus, the effort for simulation projects, which accounts for about 30-40% of the total duration of data collection and up to 35% of model preparation exists, can be reduced [37]. The aim is to prevent serious failures in the design phase of the model by determining restrictions, definitions, and structures.

C. MDSOA

The approach of "model-driven service-oriented architecture" describes the use of different methods and notations to refine models through automated model transformations and the generation of artifacts [38]. MDSOA can be applied to any software development process. It uses model transformations to automate recurring tasks. Among other things, the quality assurance process can be automated.

The approach is based on the OMG's MDA standard for model-based software development and is similar to the "modeling and simulation as a service" (MSaaS) approach presented in [39]. It also introduces automated model transformations that should enable users to model in their languages. The model-to-text transformation is the core of the model-driven process: the generator model serves as input; the output is the memory library in JavaScript and an HTML file that ensures the actual implementation.

D. Machine modeling – component view

According to [40], the effort for creating the machine model in simulation projects is often higher than the benefits derived from it. To prevent this effect, a method was developed, with which a machine model can be created automatically from the engineering documents. No detailed knowledge of the machine is necessary, and consequently, no expert has to be involved in creating the machine model. The modular approach used in the Aquimo project automatically configures interdisciplinary engineering documents and the machine model. Among other things, company and project-specific parameter values and the installation diagram are used for this purpose [41]. The behavior models of the components are also created automatically. Another approach is the approach by Reinhart et al. presented also in [40], in which a meta-model is created and the interfaces of the required modules are manually coupled. The subsequent parameterization is also done manually, while the machine model is generated in a partially-automated manner. As shown in Figure 11, the approach in [40] itself uses the documents that were created during the engineering process anyway to create the machine model automatically and in a resource-saving manner.

Other approaches use manually created, company-specific building blocks and rules to create machine models based on module and parameter lists or use a transformation of source code or models of a certain type to create the target model. These M2M transformations are partially supported by additional algorithms, for example, by taking degrees of freedom from 3D CAD models to create behavior models for individual components. The problem often arises that the information from the engineering documents is incomplete and the relationships in the initial models cannot be clearly assigned, so that manual rework is required. The effort of post-processing is about half as high as the total effort would have been without the method [40]. The degree of automation can be increased further, but additional work would be required in the engineering process to create additional documents.

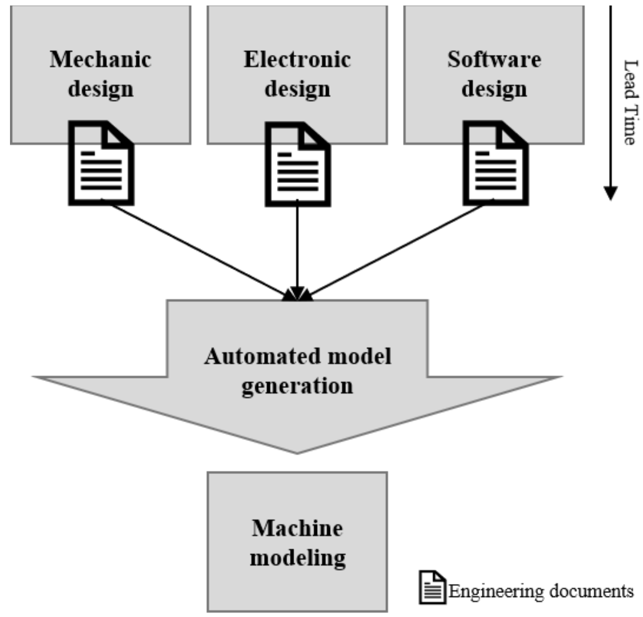

Figure 11: Automated generation of the machine model according to [40].

E. ARIS – people and process view

ARIS, developed by Scheer amongst others in cooperation with the German software company SAP-SE, is an acronym for the architecture of integrated information systems [42]. The underlying model of this approach, which is particularly well-known in Germany, consists of five description views, each with three description levels. The previous form, the so-called ARIS House, is used to reduce complexity and simplify process modeling. The (a), functional view, describes processes and their hierarchical relationships. The (b), organizational view, contains the organizational chart. The (c), data view, contains all company-relevant information objects. The (d), performance view, shows all service, material, and financial services and finally, the (e), process view or control view, integrates all other views (a) to (d) in a time-logical flow chart, such as event-driven process chain (EPC). The description levels are the technical concept, the IT concept, and the implementation level. They serve to represent the business processes for specialists, the implementation of the technical concept in IT-related description models, and the IT-technical realization of the process parts. The software tool ARIS has evolved steadily since its introduction and now consists of several software modules. These enable, among other things, the import of data from data sources such as CRM systems, ERP reports, data warehouses, or Excel tables. In addition, models from UML, MS Visio, BPMN WSDL, XSD, or BPEL can be integrated into the software. Thanks to the uncomplicated import of various file formats and their linking, new information can be implemented quickly. Besides, compatibility with supplier system models can be made easier. Once the system model has been implemented, the ARIS Toolset can be used to automatically create the Quality Management manual, the process and work instructions, job descriptions, the creation of key figures, and process cost accounting.

Furthermore, EIS (Executive Information Systems) takes over the filtering and preparation of decision-relevant information for the management, i.e., data from different sources is merged, and information is offered in a user-friendly way according to different views and levels of aggregation. In addition, data mining techniques are used, which enable the business process owner to navigate in a targeted manner to processes relevant to the investigation. If information objects or attributes are removed from the data model, added to it, or changed, this information automatically leads to an adjustment of the user mask in the system. Automation is also aimed at through the use of object-oriented code generators [43], whereby additional code must be generated manually in some cases and re-delegation takes place in the case of failures caused by the design itself.

F. Which approaches already contribute to reducing the initial and update effort and are they sufficient?

The approaches to partially automated modeling presented here all serve the purpose of reducing the effort involved in creating and updating system models. It will only make sense to use such approaches if this goal can be achieved. The reduction of effort is to be achieved by modeling the five views of eDeCoDe presented above, i.e., only those aspects are considered, which are useful for this purpose. The extent, to which the above-described approaches complement, contradict, or exclude each other as well as eDeCoDe must also be considered. The eDeCoDe views of requirements, components, processes, and, in some cases, the view of the people can be found to some degree in the examined approaches. At least one of the approaches relates to these views, but there is no possibility of partial automation regarding the view of the functions. At the same time, it is noticeable that although each of the approaches is based on a model including its definition, these approaches have little or no overlap.

V. Conclusion and Further Work

The use of system models is accompanied by many advantages, which are necessary for the success of a company. Especially in the context of the increasing complexity of product and production systems, the system model plays an important role. Therefore, new approaches to system model creation are constantly being published.

In this article, the problem of excessive initial and update efforts in the modeling of production systems was highlighted. It was shown that there are different approaches to depicting production system models and that these contribute to reducing their complexity. However, these approaches have the commonality of manual implementation. Because production systems are made up of numerous system elements and relationships, it is hardly possible for them to be created by individual people. For this reason, the article also critically questioned how far the development of partially automated approaches has progressed. Therefore, approaches of partial automation were also examined, which should reduce the effort of system model creation and updating.

Existing approaches of partial automation of model creation are very branch specific or consider only partial aspects of the overall system, such as the requirements, so that they cannot be used for holistic system description and modeling in a multi-dimensional way, such as eDeCoDe. Other, unspecific approaches to partial automation, on the other hand, do not offer any significant reduction in effort.

The result of this investigation clearly shows that there are approaches that could map individual views of the modeling with, e.g., eDeCoDe in a partially-automated manner. Because these approaches are view-specific, however, the question arises as to whether it is possible to link the view-specific approaches to a holistic approach of partially automated modeling. If this is not the case, it is necessary to develop a new approach to partial automation.

Acknowledgment

The authors thank the German Research Foundation (DFG) for their support of the projects KAUSAL [WI 1234/21-1], ReMaiN [WI 1234/28-1], and FusLa [SCHL 2225/1-1].

References

[1] M. Heinrichsmeyer, N. Schlüter, F. Kösling, and A. Ansari, “Validation of a Failure-Cause Searching and Solution-Finding Algorithm in Production based on Complaint Information from the Use Phase,” in The Fifteenth International Conference on Systems, International Journal on Advances in Systems and Measurements IARIA Lisbon, Portugal, pp. 7–12, 2020.

[2] B. S. Onggo, N. Mustafee, A. Smart, A. A. Juan, and O. Molloy, “Symbiotic simulation system: hybrid system model meets big data analytics," 2018 Winter Simulation Conference (WSC), pp. 1358–1369, 2018.

[3] H. Hick, M. Bajzek, and C. Faustmann, “Definition of a system model for model-based development," SN Applied Sciences, vol. 1074, pp. 1074, 10.1007/s42452-019-1069-0, 2019.

[4] A. Canedo, E. Schwarzenbach, and M. A. Al Faruque, “Context-sensitive synthesis of executable functional models of cyber-physical systems,” in Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems - ICCPS '13, Lu, C.; Kumar, P. R.; Stoleru, R. ACM Press; IEEE New York, New York, USA, pp. 99, 2013.

[5] C. Torens, L. Ebrecht, and K. Lemmer, “Starting Model-Based Testing Based on Existing Test Cases Used for Model Creation,” in 2011 IEEE 11th International Conference on Computer and Information Technology (CIT 2011), IEEE Computer Society IEEE Piscataway, NJ, pp. 320–327, 2011.

[6] R. Haberfellner, O. L. de Weck, E. Fricke, and S. Vössner, “Systems Engineering,” Orell Füssli Verlag; Orell Füssli, vol. 14, Zürich, 2018.

[7] L. Kof, “From Requirements Documents to System Models: A Tool for Interactive Semi-Automatic Translation,” in 2010 18th IEEE International Requirements Engineering Conference (RE 2010), University of Technology, Sydney; IEEE Computer Society; Institute of Electrical and Electronics Engineers; IEEE International Requirements Engineering Conference; RE IEEE Piscataway, NJ, pp. 391–392, 2010.

[8] A.-W. Scheer, “ARIS - Vom Geschäftsprozess zum Anwendungssystem (ARIS - From business process to application system),” Springer, vol. Vierte, durchgesehene Auflage, Berlin, 2002.

[9] B. Stützel, L. Borchardt, T. Illa, and C. Gerling, "*Systems Engineering in Deutschland*," [Online]. Available from: https://www.gfse.de/Dokumente_Mitglieder/se/pubs/downloads/2018-11-06_Prozesswerk_Broschuere_digital.pdf, 01.12.2020.

[10] F. Ehrenmann, “Kosten- und zeiteffizienter Wandel von Produktionssystemen (Cost and time efficient change of production systems),” Springer Gabler, vol. 1, Wiesbaden, 2015.

[11] T. Tomiyama, M. Mäntylä, and S. Finger, “Knowledge Intensive CAD, vol. 1,” Springer US, Boston, MA, s.l., 1996.

[12] M. Heinrichsmeyer, “Entwicklung eines zielgerichteten Fehlerursachensuch- und Lösungsalgorithmus [FusLa] (Development of a targeted error search and solution algorithm [FusLa]),” Bergische Universität Wuppertal, vol. 1, Wuppertal, 2020.

[13] P. Fritzson and P. Bunus, “Modelica - a general object-oriented language for continuous and discrete-event system modeling and simulation,” in Proceedings / 35th Annual Simulation Symposium, SS 2002, IEEE IEEE Computer Society Press Los Alamitos, Calif., pp. 365–380, 2002.

[14] P. Fritzson and V. Engelson, “Modelica — A unified object-oriented language for system modeling and simulation,” Springer, vol. 1, Berlin, 1998.

[15] J. Gausemeier and B. Behmann, “Produkte und Produktionssysteme integrativ konzipieren (Integrative design of products and production systems),” Carl Hanser Fachbuchverlag, vol. 1. Aufl., s.l., 2012.

[16] J. Heihoff-Schwede, C. Bremer, M. Rabe, and C. Tschirner, “Werkzeuge für den Mittelstand–MBSE leicht (Tools for medium scale- MBSE light),” in Tag des Systems Engineering, Schulze, S.-O.; Tschirner, C.; Kaffenberger, R.; Ackva, S. Carl Hanser Fachbuchverlag s.l., pp. 35, 2016.

[17] V. Salehi, G. Florian, and J. Taha, “Implementation of Systems Modeling Language (SysML) in consideration of the CONSENS approach,” in Proceedings of the DESIGN 2018 15th International Design Conference // Design 2018, Marjanović, D.; Štorga, M.; Škec, S.; Bojčetić, N.; Pavković, N. Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia; The Design Society, Glasgow, UK; Fac. of Mechanical Engineering and Naval Architecture Univ Zagreb, pp. 2987–2998, 2018.

[18] M. Maurer and S.-O. Schulze, “Tag des Systems Engineering (Day of Systems Engineering),” Hanser, vol. 1, München, 2015.

[19] S. Friedenthal, “A practical guide to SysML,” Morgan Kaufman, vol. Third edition, Waltham, MA, 2015.

[20] J. Holtman, J. Meyer, W. Schäfer, and U. Nickel, “Eine erweiterte Systemmodellierung zur Entwicklung von softwareintensiven Anwendungen in der Automobilindustrie (An extended system modeling for the development of software intensive applications in the automotive industry),” in Software Engineering 2010 - Workshopband, Engels, G.; Luckey, M. Ges. für Informatik Bonn, pp. 149–158, 2010.

[21] T. Weilkiens, “Systems Engineering with SysML/UML,” Elsevier professional, vol. 1. Aufl., s.l., 2008.

[22] F. Munker, “Ein Ansatz zur anwenderorientierten Systemmodellierung für die interdisziplinäre Produktentwicklung,” KIT; Karlsruhe, vol. 1, Karlsruhe, 2016.

[23] F. Munker and A. Albers, “SystemSketcher – Entstehung eines anwenderorientierten Ansatzes zur interdisziplinären Systemmodellierung (SystemSketcher - Development of a user-oriented approach to interdisciplinary system modeling),” in Tag des Systems Engineering, Schulze, S.-O.; Tschirmer, C. Carl Hanser Fachbuchverlag s.l., pp. 291–300, 2015.

[24] M. Gonia, “Using manufacturing execution systems (MES) to track complex manufacturing processes,” in IEEE/CPMT/SEMI 29th International Electronics Manufacturing Technology Symposium, International Electronics Manufacturing Technology Symposium IEEE Service Center Piscataway, NJ, pp. 171–173, 2004.

[25] J. Kletti, “Manufacturing Execution Systems - MES,” Springer-Verlag Berlin Heidelberg, vol. 1, Berlin, Heidelberg, 2007.

[26] P. Winzer, “Generic Systems Engineering,” Springer Vieweg Verlag, vol. 2, Berlin, Heidelberg, 2016.

[27] J.-P. G. Nicklas, “Ansatz für ein modellbasiertes Anforderungsmanagement für Unternehmensnetzwerke (Approach for a model-based requirements management for enterprise networks),” Shaker Verlag, vol. 1, Aachen, 2016.

[28] S. Schlund and P. Winzer, “DeCoDe-Modell zur anforderungsgerechten Produktentwicklung (DeCoDe model for requirements-based product development),” in "Das ist gar kein Modell!" (It is not a model), Bandow, G.; Holzmüller, H. H. Gabler Wiesbaden, pp. 277–293, 2010.

[29] A. Hahn, S. Häusler, and S. Große Austing, “Quantitatives Entwicklungsmanagement (Quantitative development management),” Springer Vieweg, vol. 1, Berlin, 2013.

[30] O. Bielefeld, H. Dransfeld, N. Schlüter, and P. Winzer, “Development of an Innovative Approach for Complex, Causally Determined Failure Chains," Management and Production Engineering Review, vol. 3, pp. 3–12, 10.1515/mper-2017-0023, 2017.

[31] D. Krause and N. Luhmann, “Luhmann-Lexikon,” Stuttgart: UTB; Enke, vol. 2., vollst. überarb., erw. und aktualisierte Aufl., Stuttgart, 2005 // 1999.

[32] O. Bielefeld, N. Schlüter, and P. Winzer, “Development of a methodological approach for requirements management in cross-company networks (ReMaiN),” in Proceedings M2D2017, Silva Gomes, J. F.; Meguid, S. A. Edições INEGI Porto, pp. 1109–1124, 2017.

[33] S. Marchlewitz, J.-P. Nicklas, and P. Winzer, “Using system engineering for improving autonomous robot performance,” in 2015 10th System of Systems Engineering Conference (SoSE 2015), Institute of Electrical and Electronics Engineers; IEEE Systems, Man, and Cybernetics Society; IEEE Reliability Society; International Council on Systems Engineering; System of Systems Engineering Conference; International IEEE Conference on Systems of Systems Engineering; SoSE IEEE Piscataway, NJ, pp. 65–70, 2015.

[34] A. Götz and C. Donges, “Automatisierter Übergang vom dokumenten- zum modell-zentrierten Requirements Engineering als Ausgangsbasis für MBSE (Automated transition from document to model-centric requirements engineering as a starting point for MBSE),” in Tag des Systems Engineering, Schulze, S.-O.; Tschirner, C.; Kaffenberger, R.; Ackva, S. Carl Hanser Verlag GmbH & Co. KG München, pp. 301–310, 2017.

[35] El-Haik. Basem and K. Yang, “The components of complexity in engineering design,” in IEEE Transaction, IEEE Springer Berlin, pp. 925–934, 10.1999.

[36] A. Burger, S. Reiter, A. Viehl, O. Bringmann, and W. Rosenstiel, "*Systemmodellierung zur Fehlereffektsimulation (System modeling for failure effect simulation),*" [Online]. Available from: https://www.researchgate.net/publica-tion/273004392_Systemmodellierung_zur_Fehlereffektsimulation, 01.12.2020.

[37] D. Krenczyk, “Automatic Generation Method of Simulation Model for Production Planning and Simulation Systems Integration," Advanced Materials Research, pp. 825–829, 10.4028/www.scientific.net/AMR.1036.825, 2014.

[38] G. Rempp, M. Akermann, M. Löffler, J. Lehmann, and M. Starzmann, “Model Driven SOA,” Springer, vol. 1, Berlin, 2011.

[39] P. Bocciarelli, A. D'Ambrogio, A. Mastromattei, and A. Giglio, “Automated development of web-based modeling services for MSaaS platforms,” in International Symposium on Model-driven Approaches for Simulation Engineering

(Mod4Sim'17), D'Ambrogio, A.; Durak, U.; Çetinkaya, D. Curran Associates Inc Red Hook, NY, pp. 1–12, 2017.

[40] A. Kufner, “Automatisierte Erstellung von Maschinenmodellen für die Hardware-in-the-Loop-Simulation von Montagemaschinen (Automated creation of machine models for hardware-in-the-loop simulation of assembly machines),” Jost-Jetter, vol. 1, Heimsheim, 2012.

[41] R. Angerbauer, R. Buck, U. Doll, M. Hackel, K.-H. Kayser, M. Klebl, H. Mack, R. Siegler, F. Wascher, and R. Würslin, “Aquimo,” VDMA-Verl., vol. 1, Frankfurt, M., 2010.

[42] A.-W. Scheer, “ARIS - Business Process Modeling,” Springer Berlin Heidelberg, vol. Third Edition, Berlin, Heidelberg, s.l., 2000.

[43] A.-W. Scheer, “ARIS - Modellierungsmethoden, Metamodelle, Anwendungen (ARIS - Modeling methods, metamodels, applications),” Springer Berlin Heidelberg, vol. Dritte, völlig neubearbeitete und erweiterte Auflage, Berlin, Heidelberg, 1998.