EasyChair Preprint

№ 1529

Model Checking of Integratively Designed Product and Production Systems

Sergej Japs, Rik Rasor, Lydia Kaiser and Roman Dumitrescu

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

September 16, 2019

Model Checking of Integratively Designed Product and Production Systems

Sergej Japs[1], Rik Rasor[1], Lydia Kaiser[1], Roman Dumitrescu[2]

[1]Fraunhofer Research Institute for Mechatronic Systems Design IEM,
Product Engineering, Zukunftsmeile 1, 33102 Paderborn, sergej.japs@iem.fraunhofer.de,
[2]University of Paderborn, Advanced Systems Engineering,
Zukunftsmeile 1, 33102 Paderborn, roman.dumitrescu@hni.upb.de

Abstract: Short product life cycles and the associated shortening of production periods require an integrative design of the product and production system. This paper presents a methodological approach for linking integrative design, model-based systems engineering and model checking. This approach supports the integrative development of product and production systems by formal description in the early design phases. Based on the example of a pedelec and the associated production system, model checking is used to verify the behavior of the production system fully automated with respect to specified requirements regarding the production process. The formalized description with SysML supports this automated analysis of the integrative models by a common semantic and syntax. The methodological approach has been evaluated comprehensively based on different product variants of the application example and on different configurations of the production system.

1 Introduction

The current market situation results in increasing complexity throughout the entire value chain and shorter product life cycles across all industries. Therefore, the future competitiveness of manufacturing companies strongly depends on the complexity management of products and processes and a short time-to-market [FG13] [Sc14]. In this context, the product and the associated production system must be closely aligned with each other in the conceptual phase of system design [Ga10].

Decisions in product and production system design are usually determined by mutual dependencies. An efficient development process therefore requires the integrative design of the different systems. Integrative approaches enable an evaluation of the system properties and requirement fulfilment, the comparison of alternative variants and the analysis of the impact of changes [GD+18]. These approaches are currently supported by formalized modelling and simulation. However, the behavior of integrative product and production system models cannot be checked by fully automated analyses. The complexity of relations between product and production models and the multitude of interaction possibilities in the detailed behavior simulation make a fully automated analysis even for very powerful computers impossible [CD+05]. For this reason, the existing approaches cannot adequately support the model analysis in the context of the integrative conception of product and production systems.

Using the example of a pedelec, this paper shows how the integrative specification can be used for evaluating design aspects of the product and for production system planning. For this purpose, the model-based implementation in SysML is described with a consideration of time information. This implementation provides the methodological foundation for a model transformation and analysis, in order to identify errors in the design of the product and production system at an early stage.

In the following, the fundamentals of integrative product system design are summarized in Section 2. Section 3 describes our approach of integrative modeling using the example of a pedelec. Section 4 describes the method for fully-automated checking the behavior of the integrative model against requirements regarding the production process. Section 5 presents the evaluation of our approach. The evaluation is based on a manual model transformation. Section 6 summarizes the contribution and gives an outlook on future work.

2 Fundamentals: Integrative Product System Design

In traditional design theory, product development is almost completed when production system development begins. The aim of simultaneous engineering is to parallelize the consecutive workflows of product design and production system design. This shortens the development time of a new product and avoids subsequent changes. In this context different methods and instruments have been established for simultaneous engineering and for an early analysis of product and production systems (e.g. [BW97] [EM13] [Ga10]). The parallelization of integrative design requires a close cooperation and a high need for coordination of product and production system development, as well as a continuous exchange of information between the specific departments. Traditional approaches have a low degree of formalization and require manual analysis of the models (see Figure 1 – 1). Especially the design of complex mechatronic systems requires that the impact of manufacturing technologies on the product concept and the multitude of interactions between product and production system are analyzed in the early design phases [Ga10].

The use of a common modelling language is a good way of facilitating coordination between the disciplines. In particular, the Model-Based Systems Engineering (MBSE) approach can combine the development activities of all participating disciplines through a single system model [SF+16]. For the consistent representation, graphical modelling languages such as SysML (Systems Modeling Language) [Ob15] are used. Suitable approaches for the modelling of the structure and the abstract behavior modelling of product and production systems with SysML are presented in [KV13] [SR09]. With the integrative modeling of product and production systems, their dependencies can be represented due to identify inconsistent designs and to avoid redundant or contrary activities [EL+05] [MP+18]. In this way, the numerous degrees of freedom in product and production system design can be addressed by a fast and targeted reduction of the solution space. The formal and model-based specification offers the possibility to evaluate relevant design aspects from production system planning at an early stage (see Figure 1 – 2). As far as possible, this analysis should be based on calculation and simulation and not only on the intuition of experienced experts [Ga10].

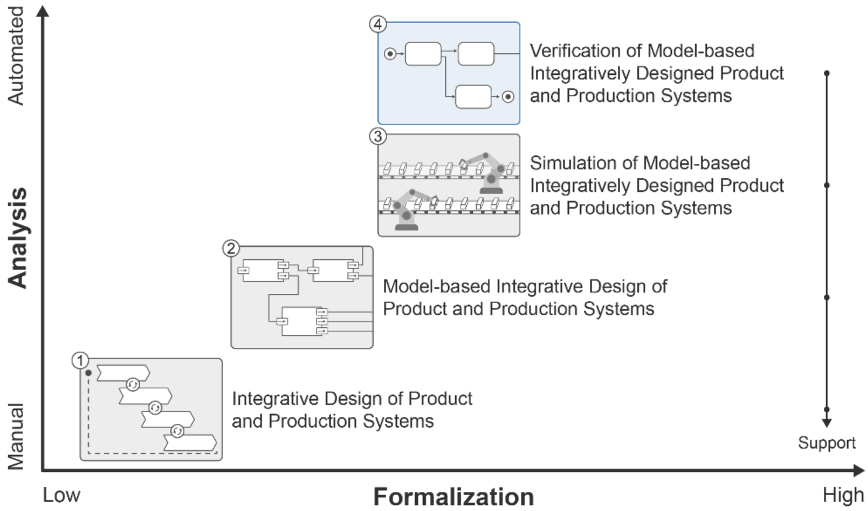


Figure 1. Evolution of supportive approaches in the context of integrative design

While modeling describes the process of building a model, simulation is the process of using a model to analyze its behavior. SysML models can be used to describe production systems independently of simulation tools, but are less suitable for detailed behavioral simulation or for describing time behavior [SR09] [AM+17]. For this reason the simulations of the integrated product and production system development are carried out by dedicated simulation tools. As shown in Figure 1 - 3, the semi-automated interpretation of the models requires a formalized description. This description can use common semantics and syntax as well as the definition of constraints [SF+16].

Summarizing, model-based approaches and modelling languages can be used in the early phases of integrative product and production system design - simulation-based approaches can be provided by external tools. A high degree of formal description can support automation of the analysis of integrative models. However, the simulation of systematically designed test cases is time-consuming and can only partly guarantee the correctness of the behavioral simulation. Methods for fully automatic verification of the integratively designed production systems against requirements regarding the production process do not exist (see Figure 1 – 4).

3 Integrative Design of Product and Production Systems

In this section we introduce the integrative design of the product and production system with SysML. This is illustrated by the application example pedelec and pedelec production system. SysML is a graphical and standardized modeling language based on UML [Ob17]. It is used as a de facto standard in the field of model-based systems engineering. SysML offers a wide range of descriptive tools for the holistic and interdisciplinary specification of intelligent technical systems.

Figure 2 shows an overview of the approach used in this paper. First, a product model and a production system model are integratively designed (see Sections 3.1 and 3.2). The production system considers already modelled properties of the product model. The internal relationships, especially the resource flows of the production system are then modelled taking into account the product model (see Section 3.3). The next step is to model the behavior model of the production system based on the structural model and on the resource model of the production system (see Section 3.4). In Section 3.5 we describe requirements for the production process of the production system. During the model analysis, the specified behavior of the production system is fully automated checked against the specified requirements (see Section 4).

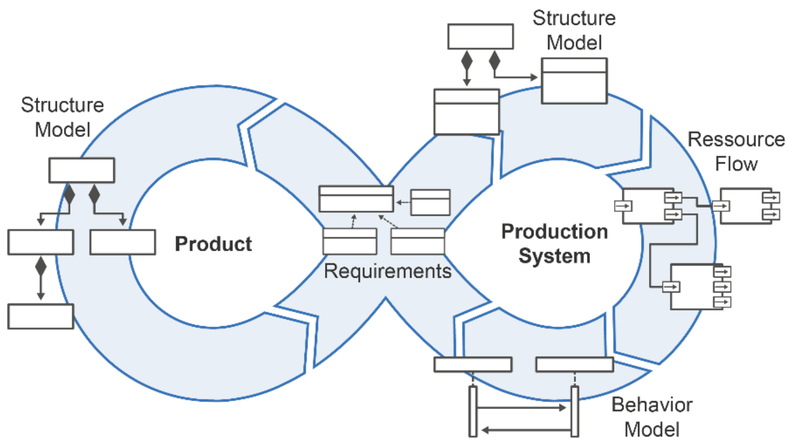


Figure 2. Approach for the automated model analysis of integrative design of product and production systems based on SysML

3.1 Product Model

The modeling of a system from the point of view of a black box is made possible by the Block Definition Diagram (BDD). Figure 3 shows an excerpt of a product model for the pedelec in the form of a BDD. In the context of this work, the hierarchical relationships of the product model are in focus. These relationships are expressed with a composite aggregation. The pedelec consists e.g. of a frame, a fork and wheels. The wheel is further divided into inner tube, folding tire and spokes. Relationships between subsystems can be modelled with Internal Block Definition Diagrams (IBD) (see Section 3.3).

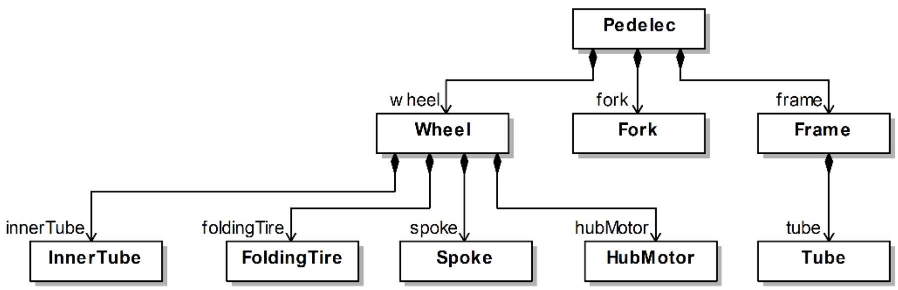
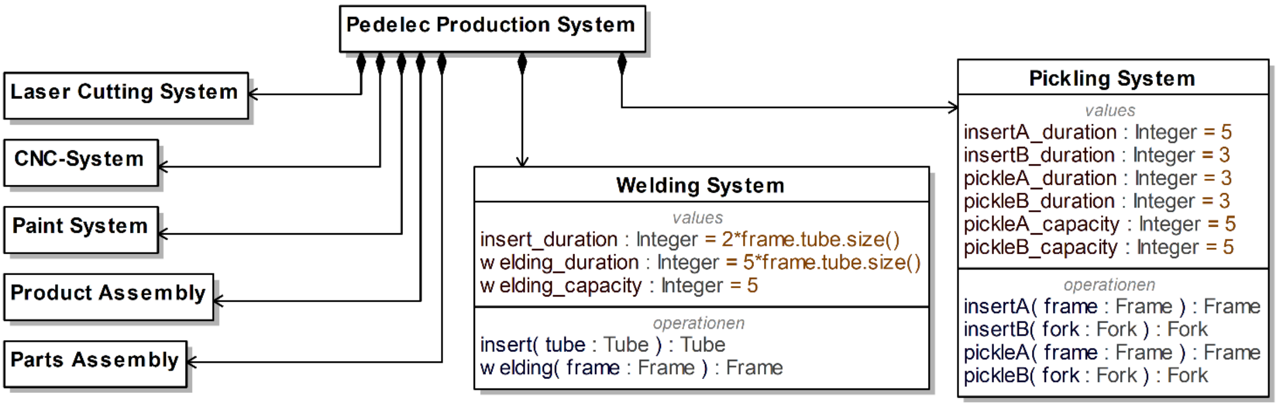


Figure 3. Pedelec product model

3.2 Production System Model

The manufacturing of a product is a process. This process is realized by a production system. The production system can be described analogous to the product by a BDD. Figure 4 shows an excerpt of a model for the pedelec production system in the form of a BDD, which represents hierarchical relationships. The model of the production system consists of several subsystems such as the product assembly, the welding system and the pickling system. Flow relationships of materials and information between the subsystems are addressed in Section 3.3. Processes of subsystems like the welding system can be represented in the SysML by operations. An example is the operation welding, which is responsible for the welding process. In the context of the pedelec product model, the welding process refers to the welding of a frame.

Using attributes, subsystems of the production system and their associated processes can be specified more precisely. Attributes allow, for example, the specification of the lead time of a process. The lead time is in the following also referred as duration. For example, a single welding process requires 5 time units. The total time for the welding process of an entire frame can be integratively related to the product model. A simple example is the multiplication of the number of tubes of a frame by the duration of a single welding operation. Attributes can also be used to specify the storage capacity of individual subsystems of the production system. For example, the storage capacity of the welding system is 5 welded frames.



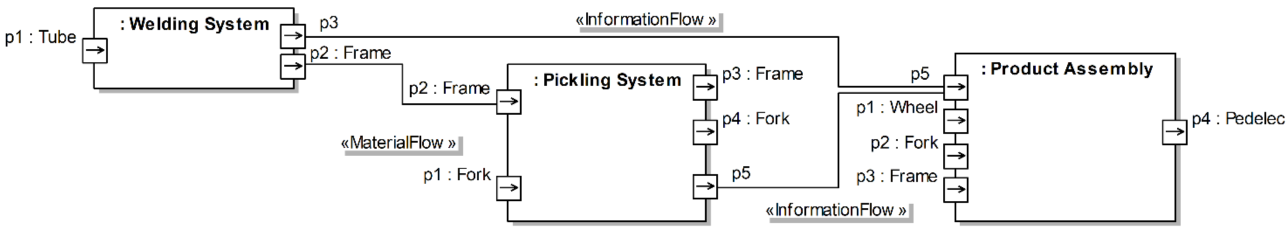
Picture 4. Integrative specified model of the pedelec production system

3.3 Production System Information and Material Flow Model

The production process determines the relationships between the subsystems of the production system. These relationships can be represented in SysML using an Internal Block Diagram (IBD). In particular, the production process is determined by flow relationships such as the material or information flow.

Figure 5 shows an excerpt of the internal relations within the pedelec production system in the form of an IBD. The relationships between individual system parts are named in the SysML connectors. Connectors can be linked using ports. Flow ports allow the specification of flow directions. Using stereotypes such as «InformationFlow» or «MaterialFlow», connectors can be differentiated according to different flow types. For a systematic classification of flow types we refer to [Ka13]. In total, the classification of flow types enables discipline-specific views on the system model.

The «InformationFlow» connector between the welding system and the product assembly enables the early identification of a communication channel between these system parts. This communication channel can be used to inform the product assembly employees about the completion of a frame so that further steps can be initiated (see Section 3.4). Another example is the «MaterialFlow» connector between the welding system and the picking system. This connector specifies the material flow of the frame from the welding system to the picking system.



Picture 5. Material and information flow of the pedelec production system

3.4 Production System Behavior Model

In this section we consider the modeling of the behavior of the production system. The SysML offers various representation types for modeling of the system behavior. These include sequence diagrams, among others. Sequence diagrams, in contrast to the other representation types, enable the explicit modeling of interactions between individual subsystems of the production system. In the context of this work, interactions here correspond to processes of the production system. The IBD of the production system (see Figure 5) is used to determine which subsystems in the sequence diagram can interact with each other. The modeled processes of the individual subsystems (see Figure 4) determine which processes are responsible for the interaction between the subsystems in a sequence diagram. On this basis, the sequence diagrams model the corresponding process sequences (see Figures 5 and 6).

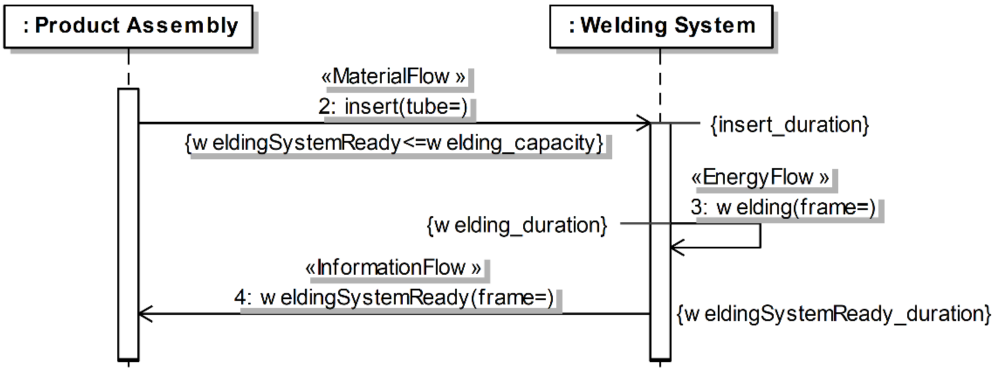


Figure 6. Process sequence between the system parts product assembly and welding system

Figure 6 shows a process sequence between the product assembly and the welding system for welding a frame. The specified time information and storage capacities of the production system (see Figure 4) are taken into account as a function of the specified product model (see Figure 3). If there is still space in the welding system storage, tubes can be inserted for the welding process. The time required to insert the tubes depends on the number of tubes required for a product variant (see Figure 3). The pipes are then welded together by the welding system to form a frame. The welding time of a frame depends on the product variant and on the welding time of the welding system used (see Figure 4). If the frame is welded together, the product assembly is informed by the welding system.

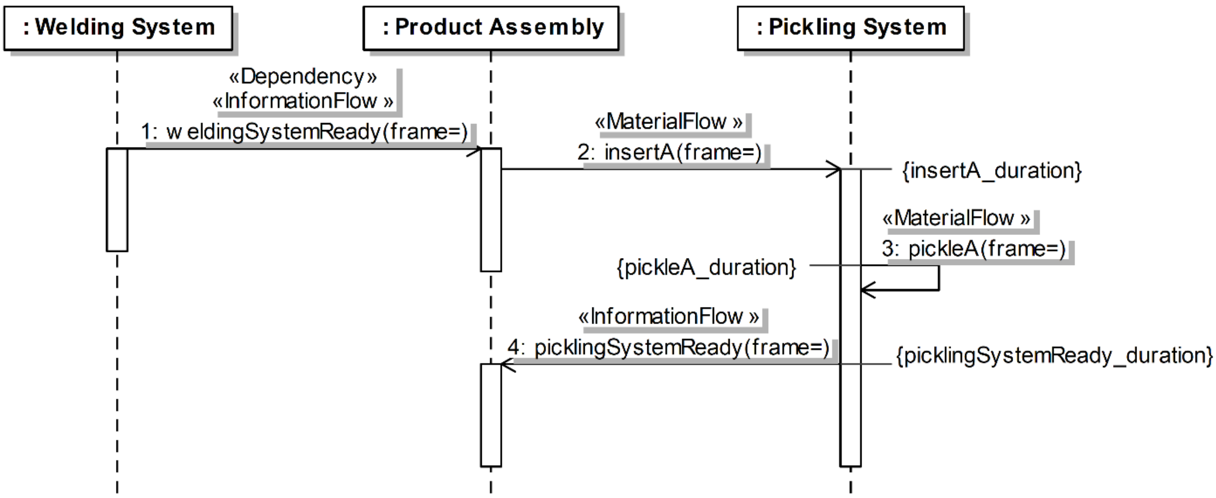


Figure 7. Process sequence between the system parts welding system, product assembly and pickling system

Figure 7 shows the process sequence between the welding system, the product assembly and the pickling system, for the process of pickling a welded frame. The specified durations of the production system (see Figure 4) are taken into account. The pickling of a frame in this case depends on whether the frame has already been completely welded. To model dependencies between sequence diagrams, we use the stereotype «Dependency». A possible alternative would be to set and query Boolean variables. If a welded frame is available, the frame has to be inserted into the pickling system by product assembly employees. The frame is then pickled in the pickling system for the specified duration. Finally, the pickling system informs the product assembly so that further steps can be initiated.

3.5 Requirements Regarding the Production Process

In this section we present the specification of requirements for the production process. These requirements must be satisfied by the production system. Thereby different product variants must be considered, and different configurations of the production system. Furthermore, potential boundary conditions must be satisfied. For the specification of these requirements we use SysML requirement diagrams.

Figure 8 shows a requirement in the form of a production order. The production of a specific number of pedelecs must be possible within a certain average production time. For this a special pedelec variant must be produced. For example a pedelec for women, which differs in the frame and thus in the number of tubes. The production is based on a configuration of the production system. The configuration is determined, for example, by the storage capacities and lead times of the subsystems of the production system (see Figure 4). In addition, the production must comply with certain boundary conditions. In this case, availability of a welded and a cleaned frame in the associated warehouses must be ensured after completion of the production order.

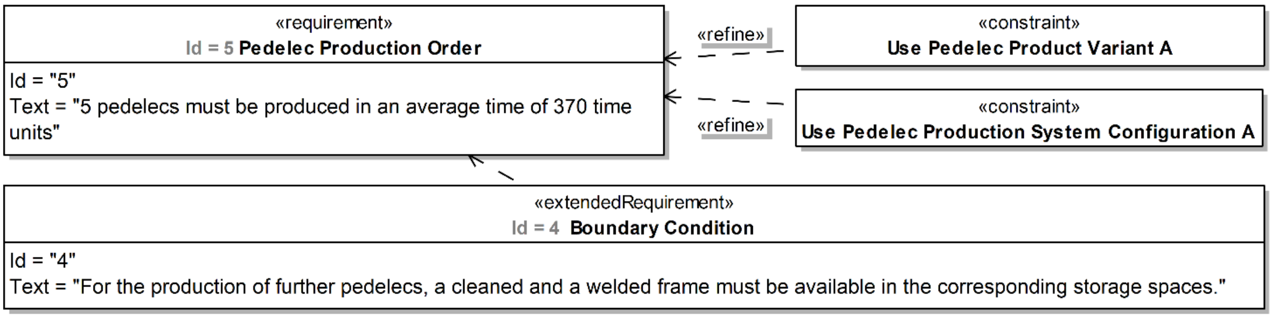


Figure 8. Requirements regarding the pedelec production process

4 Model Analysis

The modelled product and production system models allow an early analysis prior to the development of the product and the corresponding production system. The analysis of the models can generally be done manually by a review. The review can uncover ambiguities or erroneous assumptions.

In this section we introduce the automatic analysis of the behavior of the production system. The model analysis focuses on the verification of compliance with requirements regarding the production process (see Section 3.5). The analysis effort for behavior models increases exponentially with the size of the behavior models [CK+12]. This fact makes the analysis of large behavioral models impossible even for very powerful computers. For state diagrams, algorithms exist that enable an efficient model analysis [CD+05]. In this case, the analysis effort only increases polynomial with the size of the behavior models.

Section 4.1 introduces the model transformation from sequence diagrams to state diagrams. This enables an efficient model analysis. In Section 4.2 we introduce the fully automated verification of behavioral models of the production system.

4.1 Model Transformation

In this section we introduce the model transformation from sequence diagrams to state diagrams. Within the scope of the model transformation, modeled process sequences of sequence diagrams are mapped to transition sequences of state diagrams. To perform the model analysis, the state diagrams are enhanced with analysis information during the model transformation. This information ensures that requirements regarding the production process (see Section 3.5) can be covered in the model analysis.

The direct modelling of the behavior of the production system in the form of state diagrams is not in the focus of this paper. This is because analyzable state diagrams contain too much additional analysis information on the modeled behavior of the production system. This makes the modeling process more difficult.

The model transformation will be explained using Figures 9 and 10. To make the transformation easier to understand, we use the support functions init, inc, dec, checkMax and checkMin. The support functions are explained on the basis of the model transformation.

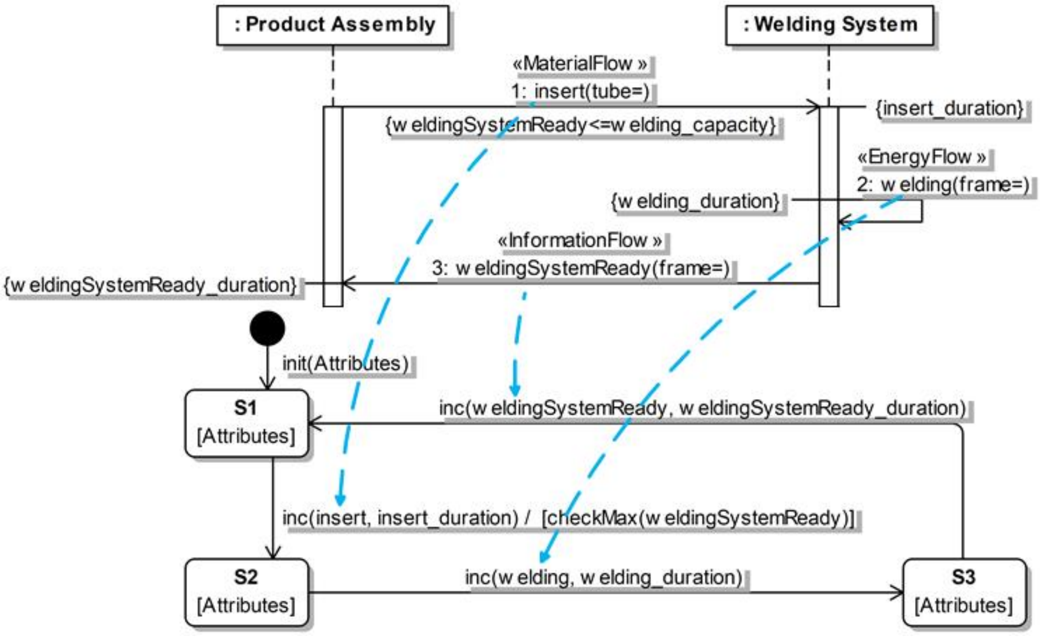


Figure 9. Model transformation from the sequence diagram from Figure 6 to a state diagram

Figure 9 shows a model transformation from a sequence diagram (welding process) to a state diagram. The process sequence of the sequence diagram is generally mapped to a sequence of transitions and states of the state diagram. Transitions can have events and can contain conditions. If an event is triggered, a state transition takes place. If a transition has a condition, the transition only takes place if the condition is fulfilled. In this paper, the support functions init, inc and dec correspond to events. The support functions checkMin and checkMax correspond to conditions. The events and conditions are executed or checked depending on parameters. The parameters correspond to the processes and lead times of the sequence diagrams.

For each sequence diagram an initial state S1 is generated, which declares the attributes. The declaration serves the identification of attributes to be used. The attributes correspond to lead times, usage times, stock levels and storage capacities. The lead times and storage capacities correspond to those of the production system model (see Figure 4). The usage times and stock levels are counted for analyses of the production system behavior. If a frame is welded, the welding time is increased by the lead time of the welding process, at the same time the stock of finished welded frames is increased by one.

For the process insert from the sequence diagram, a new state S2 is generated. This state is connected via a directional transition from S1. The process insert and the corresponding lead time insert_duration is mapped as a parameter to the event function inc. The inc function evaluates the parameters as follows: The counter for the process insert is increased by one. This captures how many tubes have already been inserted into the welding system. In addition, the time for inserting tubes is increased by the associated lead time. The event inc, is only activated if the condition checkMax is fulfilled. This is done depending on the value of weldingSystemReady from the sequence diagram. If there is still space in the storage for welded frames, then the condition is fulfilled and further tubes can be inserted into the welding system. Finally, the updated and non-updated attribute values are transferred to state S2.

The transformation of the welding and weldingSystemReady processes from the sequence diagram is performed analogously. For the last process, the associated transition in the state diagram is connected to the initial state of S1. This ensures that the processes can be performed as often as necessary. Simultaneously, modified and unmodified attribute values are transferred to the subsequent state.

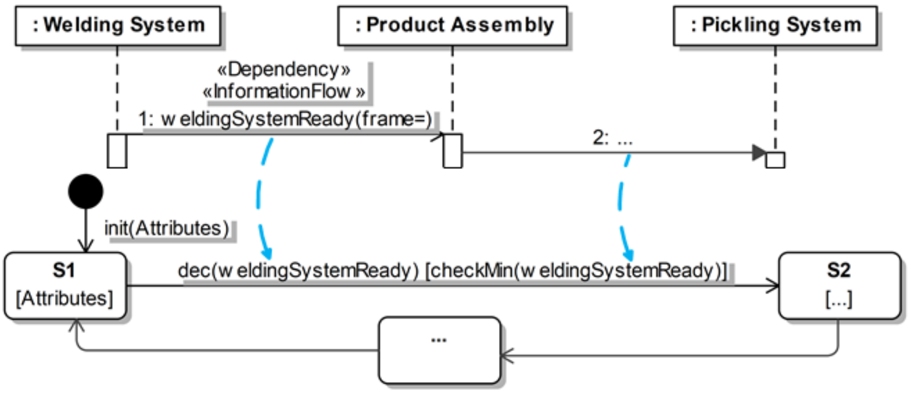


Figure 10. Model transformation from the sequence diagram from Figure 7 to a state diagram

Figure 10 shows a fragment of a model transformation from a sequence diagram (pickling process) that is depending on a different sequence diagram. The dependency is expressed by the stereotype «Dependency». In this case the pickling process depends on the welding process. If a dependency occurs, the process of the sequence diagram is mapped to a transition with an event function dec and a condition function checkMin being used as parameters. The dec support function reduces the counter for existing welded frames by one. This is performed by evaluating the checkMin function. It is only allowed to clean one frame if there is a minimum number of welded frames.

4.2 Model Checking

Model checking is a procedure for the fully automated verification of a model against a specific property. For a given model M and a given property P, it shall be verified if M is a model for P. [CH+18]

Here the model of the behavior of the production system (see Section 3.4) is used as model M. The specified requirements for the production process (see Section 3.5) are used as the specified property P. Model checking can be used to verify whether the specified production process can satisfy the specified requirements, taking into account different product variants and different configurations of the production system. If the result of the model checking is negative, adjustments can either be made to the production process or to the product variants and configurations of the production system. If the result is positive, the production process is verified. For this production process it is guaranteed that it satisfies the specified requirements. The verified production process can be used as a basis for simulations with dedicated tools for the simulation of detailed production processes.

Figure 11 shows an excerpt of the state space for the behavior (see Section 3.4) of the production system. The state space represents possible production processes, based on the transformed state diagrams from Figures 9 and 10. For simplification, the transition sequence (insert, welding, wheelAssemblyReady) from Figure 9 was summarized to WELDING and the transition sequence (wheelAssemblyReady, pickle, picklingSystemReady) from Figure 10 was summarized to PICKLING. The states indicate how many frames have been welded and how many frames have already been pickled.

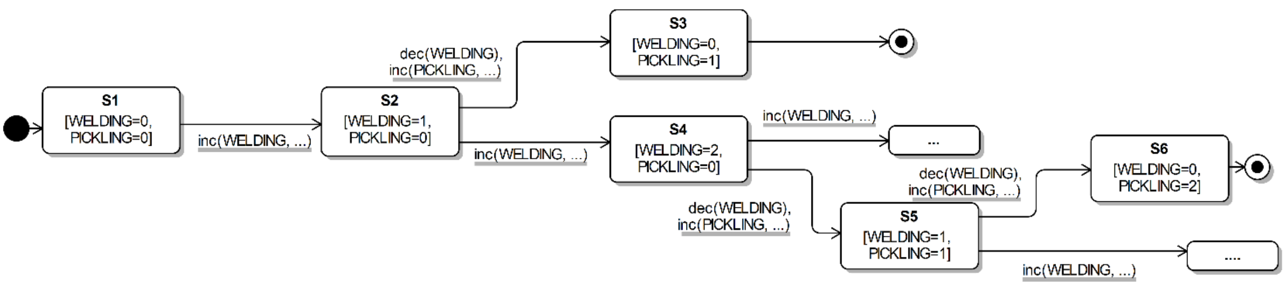


Figure 11. Excerpt of the state space of the production system behavior

Within the context of model checking, a reachability analysis [CD+05] on the state space (model M) is used to verify whether a state S can be reached from the initial state S1, which satisfies a specified property P. P corresponds here to the requirements of the production process (see Section 3.5). If the state S can be reached, a subgraph is identified, which consists of a state and transition path leading to the state S with the satisfying property P. The state and transition path represents a production process in the context of the production system. In this case it is guaranteed that the production process satisfies the requirements.

5 Evaluation

For evaluation we used the model checking tool UPPAAL [Up19]. This allows the verification of requirements (see Section 3.5) against state spaces (see Section 4.2) on the basis of state diagrams (see Section 4.1).

Requirements Regarding the Production Process

Production Order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Produced Pedelecs	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25	5	10	15	20	25
Average Production Time	370	355	345	270	250	370	335	260	335	330	360	340	325	320	250	250	325	325	315	310
Boundary Condition																				
Pickled Frames in Storage	1	0	1	0	0	6	0	1	1	1	2	0	0	0	1	1	2	0	0	0
Welded Frames in Storage	1	0	1	0	0	6	0	1	1	1	2	0	0	0	1	1	2	0	0	0
Product Variant	A					B					A					B				
Tubes per Frame	10					7					10					7				
Production System Configuration	A										B									
Pickled Frames Storage Capacity	5										1									
Welded Frames Storage Capacity	5										1									
Frame Pickling Lead Time	30										30									
Single Welding Lead Time	5										3									
Requirement Fulfilled	yes	yes	yes	no	no	no	yes	no	yes	yes	no	yes	yes	yes	no	no	no	yes	yes	yes

Figure 12. Evaluation of the fully automated analysis of requirements regarding the production process under the consideration of different product variants and configurations of the production system

To evaluate our approach (see Figure 12), 20 different requirements for the production process (see Section 3.5) were specified in the form of production orders to be satisfied. We use two different product variants and two different configurations of the production system. The product variants differ in the frame structure and in this case in the number of tubes for the frame. The configurations of the production system differentiate in the storage capacities for welded and pickled frames and in the welding and pickling time of a frame. All specified requirements for the production process were verified fully automated in the context of model checking.

6 Conclusions and Future Work

Decisions in product and production system design are usually determined by mutual dependencies. An efficient development process therefore requires the integrative design of the different systems. Integrative approaches enable an evaluation of the system properties and requirement fulfilment, the comparison of alternative variants and the analysis of the impact of changes. These approaches are currently supported by formalized modelling and simulation. However, the behavior of integrative product and production system models cannot be checked by fully automated analyses.

For the fully automated analysis, we present an integrative concept for the product and production system (PS) based on SysML. The approach was illustrated by the application example of a pedelec and an associated PS. Based on the product model in the form of a BDD, a PS in the form of a BDD was integratively modelled. The PS consists of subsystems such as the welding system, which executes processes such as the welding of a pedelec frame. For processes, lead times were given, while for subsystems storage capacities were given. Furthermore, in the form of an IBD it was presented how material flows or information flows of a PS can be modelled. Based on the BDD and the IBD, the behavior of PS was modelled in the form of sequence diagrams. Furthermore, integrative requirements for the PS were specified. These consider different product variants and configurations of the PS.

Using model checking, the behavior of PS against specified requirements regarding the production process were verified fully automatically. This required a model transformation from sequence diagrams to analyzable state diagrams. The approach was evaluated on the basis of 20 requirements regarding the production process. Different product variants and different configurations of the PS were used for this purpose.

Part of future work will be the prototypical implementation of the model transformation and a simplified integration of the model analysis into established MBSE tools. Furthermore, the integrative specification of PS under consideration of production lines is addressed.

Literature

[AM+17] Arndt, D.; Melzer, S.; God, R.; & Sieber, M.: Konzept zur Verhaltensmodellierung mit der Systems Modeling Language (SysML) zur Simulation varianten Systemverhaltens. Tag des Systems Engineering: Paderborn, 2017

[BW97] Bullinger, H. J.; Warschat, J.: Forschungs-und Entwicklungsmanagement: Simultaneous Engineering, Projektmanagement, Produktplanung, Rapid Product Development, Teubner, Stuttgard, 1997

[CD+05] Cassez, F.; David, A.; Fleury, E.; Larsen, K.G., Lime, D.: Efficient On-the-Fly Algorithms for the Analysis of Timed Games, Springer Berlin Heidelberg, 2005

[CH+18] Clarke, E.M.; Henzinger, T.A.; Veith, H.; Bloem, R.: Handbook of Model Checking, Springer Verlag, 2018, S. 86

[CK+12] Clarke, E.M.; Klieber, W.; Novacek, M., Zuliani, P.: Model Checking and the State Explosion Problem, in Tools for Practical Software Verification, Springer Berlin Heidelberg, 2012

[EL+05] Eversheim, W.; Luczak, H.; Pfeifer, T.; Schuh, G.; Kabel, T.; Kubosch, A.; Simon, M.; Witte, V.: Management integrierter Produktentstehungen, In: Eversheim, W.; Schuh, G. (Hrsg.). Integrierte Produkt- und Prozessgestaltung, Springer, Berlin, 2005

[EM13] Ehrlenspiel, K.; Meerkamm, H: Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz, Zusammenarbeit. Carl Hanser Verlag GmbH Co KG, 2013

[FG+13] Feldhusen, J.; Grote, K. H.: Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung, Springer-Verlag, 2013

[GD+18] Gausemeier, J.; Dumitrescu, R.; Echterfeld, J.; Pfänder, T.; Steffen, D.; Thielemann, F.: Innovationen für die Märkte von morgen. Strategische Planung von Produkten, Dienstleistungen und Geschäftsmodellen, München: Hanser, 2018

[KV13] Kernschmidt, K.; Vogel-Heuser, B.: An interdisciplinary SysML based modeling approach for analyzing change influences in production plants to support the engineering, In: 2013 IEEE International Conference on Automation Science and Engineering (CASE) (pp. 1113-1118), IEEE, 2013

[Sc14] Schuh, G: Produktkomplexität managen: Strategien-Methoden-Tools, Carl Hanser Verlag GmbH Co KG, 2014

[SR09] Schönherr, O.; Rose, O.: First steps towards a general SysML model for discrete processes in production systems, In Proceedings of the 2009 Winter Simulation Conference (WSC) (pp. 1711-1718), IEEE, 2009

[Ob15] Object Management Group: Systems Modeling Language Specification V. 1.4, 2015

[Up19] UPPAAL Model Checker V.4.1.20, 2019, http://www.uppaal.org

[Ob17] Object Management Group: Unified Modeling Language Specification V. 2.0, 2017

[Ga10] Gausemeier, J.; Dumitrescu, R.; Kahl, S.; Nordsiek, D.: Integrative Development of Product and Production System for Mechatronic Products, In: Proceedings of the 20th International Conference on Flexible Automation and Intelligent manufacturing (FAIM 2010), Oakland, USA, 2010

[Ka13] Kaiser, L.; Dumitrescu, R.; Holtmann, J.; Meyer, M.: Automatic Verification of Modeling Rules in Systems Engineering for Mechatronic Systems, In: Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, ASME, 2013