vehicles

MDPI

Article

Digital Integrated Design and Assembly Planning Processes for Sports Vehicles Using the Example of a Skateboard

Timo Schuchter [1], Markus Till [1,*], Ralf Stetter [1] and Stephan Rudolph [2]

[1] Department of Mechanical Engineering, Ravensburg-Weingarten University (RWU), 88250 Weingarten, Germany; timo.schuchter@rwu.de (T.S.); ralf.stetter@rwu.de (R.S.)
[2] Institute of Aircraft Design, University of Stuttgart, 70569 Stuttgart, Germany; rudolph@ifb.uni-stuttgart.de
* Correspondence: markus.till@rwu.de

Abstract: The current product and assembly processes of system development in the vehicle industry are characterised by a multitude of different model formats, a relatively low level of data integration, and an unsatisfactory management of information. This article presents an integrated design and assembly planning process which applies several model-to-model (M2M) transformations in order to ensure a seamless transition from product requirements to an assembly system layout and design. The digital process employs a framework based on graph-based design languages (GBDLs) and achieves an integration in a model-based systems engineering (MBSE) industrial context. The underlying hypothesis that this seamless transition is possible is tested on the basis of the product and assembly system development of a sports vehicle. In this article, a skateboard is used for detailing and explaining the different modelling perspectives throughout the engineering and assembly process of this product. Due to a conscious application of GBDLs in an MBSE framework, it is possible to achieve a continuous sequence of M2M transformations which guarantees a maximum level of information integrity. These two aspects are cornerstones for a future integrated design automation of a product and its assembly system. It is important to note that the presented approach is universal and can be used in the production of components for the automotive industry, entire vehicles, and their assembly.

Keywords: graph-based design languages; assembly planning; layout planning; assembly system development

check for updates

Academic Editor: Mohammed Chadli

Received: 28 January 2025
Revised: 19 February 2025
Accepted: 21 February 2025
Published: 25 February 2025

Citation: Schuchter, T.; Till, M.; Stetter, R.; Rudolph, S. Digital Integrated Design and Assembly Planning Processes for Sports Vehicles Using the Example of a Skateboard. *Vehicles* **2025**, *7*, 22. https://doi.org/10.3390/vehicles7010022

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The design and the production of vehicles is currently characterised by enormous challenges. Due to intensified competition and increasing customer demands a worldwide, the product variety is escalating. Globally distributed manufacturing has further led to increasingly complex supply chains where the management of heterogeneous and scattered information has become a key issue [1]. This necessitates more flexible, integrated, and intelligent product and assembly system development processes. These processes are characterised by several product and assembly system modelling attempts which are currently not connected at all or are only partly connected. Consequently, several data or process entities such as product components, assembly processes, and resources are represented many times in multiple digital models. This fact and other challenges in assembly-oriented design such as a systematic digital process chain *"from data to information to knowledge"* highlights the urgent need for a better digital interaction [2].

The main aim of this paper is to propose an integrated design and assembly planning process that enables a better digital integration as well as a higher digital consistency

between several product and process models ranging from early product development to a concrete assembly system coupled with the necessary robot code. The proposed digital design and assembly planning process applies several M2M transformations in order to enable a seamless transition from initial product requirements to an assembly system design and robot code. A sports vehicle—a skateboard—serves as a concrete example for explanation. However, the proposed approach is universal and can be used in the production of components for the vehicle industry, entire vehicles, and their assembly.

In order to explain the proposed design and assembly planning process, the main information entities in the process are described here. Important results of the product and assembly system development are the EBOMs (engineering bill of materials—product parts, their quantity, and their assembly relation) from an engineering point of view) and MBOMs (manufacturing bill of materials—production process, including the manufacturing product structure) [1,3]. The bill of materials (BOMs) conveys key information that guides development at different stages and is an attempt to manage heterogeneous and scattered information [1,4]. A key step in the process of assembly planning is the generation of a sensible assembly sequence; the automation of this generation has been researched for decades but the solutions are computationally excessive, require many human operations, or require the automatic generation of constraint rules [5].

Obviously, during an integrated product and assembly system development process, an enormous amount of data in the form of different models is created and the management of the abundance of models requires a holistic approach. Over the last several decades, MBSE was found to be a versatile approach to address this challenge [6–8]. The processes described in the scope of MBSE are based on the well-known V-model (see, e.g., VDI guideline 2206 [9,10]). Figure 1 shows a V-model-based representation of the main elements of an integrated product and assembly system development process.

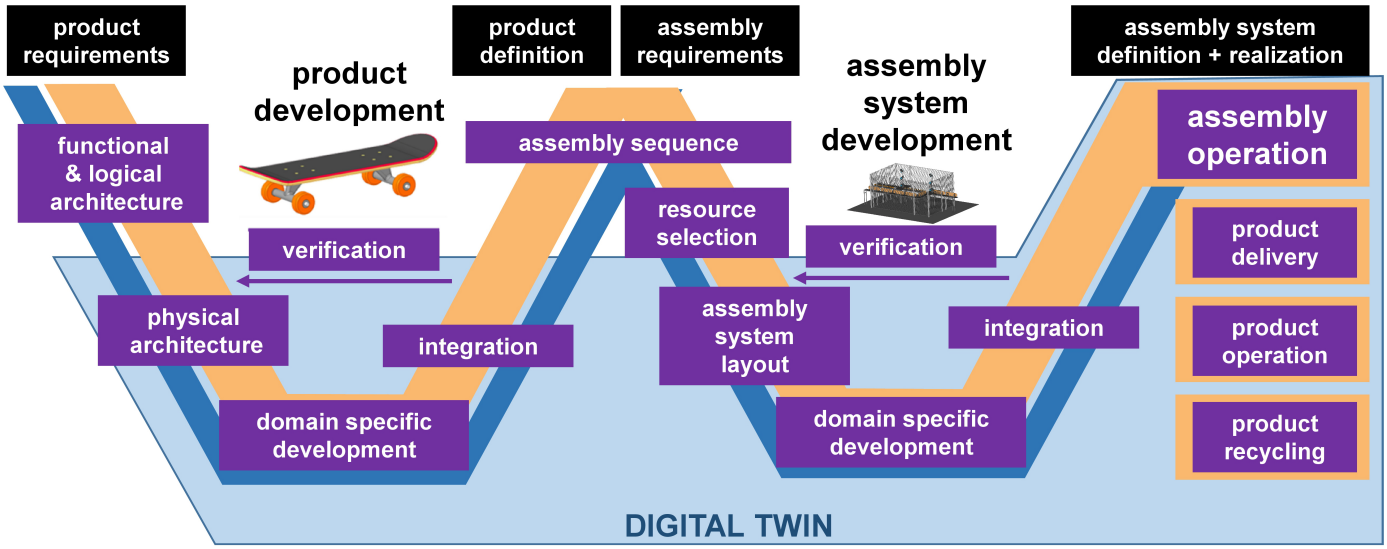


Figure 1. Life-cycle with important elements in product and assembly system development.

It is important to note that both the development cycles of the product and the assembly system are presented based on a V-model (Figure 1). The initial product model consists of the requirements concerning the functionality and decisive properties of the product to be assembled. The first cycle–product development–starts with the development of the functional and logical architecture. This architecture is decisive for a sensible realisation of the functionality and ultimately for the success of the product [11]. The next logical phase is the determination of the function carriers concerning the acting physical relationships—the physical architecture [12]. Further development is carried out in the different domains:

mechanics (including hydraulics and pneumatics), electrics/electronics, and software. It is an important quality of the V-model [9] that the integration of the domain-specific solutions is explicitly included; this decisive step enables the realisation of coordinated multi-domain systems, but requires intensive cross-domain communication. Depicted in the centre of the first cycle is the verification at different levels, which finally leads to a validation with the original product requirements. The first cycle leads to the complete set of models of the product—the product definition. Connecting the two cycles is the assembly sequence, which is the focus of this paper. In later sections of this paper, it will be shown that a model-to-model transformation of the product model to the assembly sequence is possible and that an intelligent product model can contain an implicit determination of the assembly sequence. Together with the assembly requirements, the assembly sequence enables the development cycle for the assembly system, which is essentially analogous to the first cycle. It is important to note that the verification stage may include approaches such as virtual commissioning [13] and pre-validation with virtual reality [14], with systems such as Unreal Engine, Epic Games, Cary, NC, USA [15] or Unity, Unity Technologies, San Francisco, CA, USA [16]. The second cycle culminates in the assembly system definition and realisation. On this basis, the next stages of the life-cycle, i.e., assembly operation, product delivery, product operation, and product recycling, are carried out.

As mentioned before, the two cycles are connected with the assembly sequence, which is one of the main sources of information for assembly system development, but cannot be generated without knowledge concerning the assembly system concept. It is also important to note that in this representation, the development process progresses to the recycling stage in order to be able to represent the complete scope of a digital twin (DT) (see Section 5).

Furthermore, the necessity of the continuous consideration of a circular economy is emphasized. When focusing on the models that drive the MBSE process, it becomes apparent that a shift from product information to assembly system information is just logical and inevitable. A more detailed representation of the most important models in the integrated product and assembly system development process in shown in Figure 2.

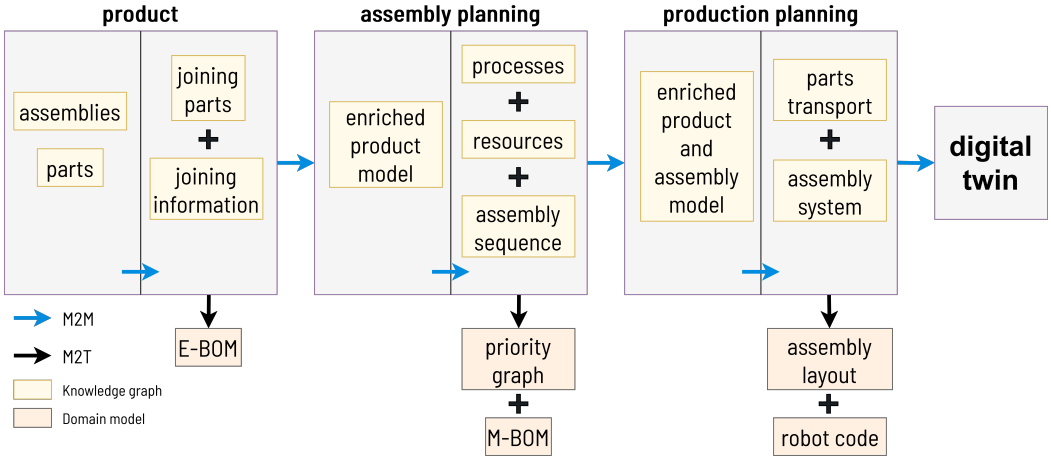


Figure 2. Model transformations in product and assembly system development.

On the very left, the items are shown, which are conventionally created and modified with computer-aided design (CAD) systems and are stored using product data management (PDM) or product life-cycle management (PLM) systems: the product parts and the assemblies are resulting from a combination of these product parts. As it will be shown in the later parts of this paper, an M2M transformation to the information entities' "joining parts" and their respective "joining information" is possible. The result of a model-to-text (M2T) transformation is the EBOM. The resulting enriched product model can be the basis of another M2M transformation determining the joining (or assembly) "processes", the nec-

essary "resources" for the joining operations (e.g., robots), and the "assembly sequence" (see Section 3). An M2T transformation leads to a priority graph and an MBOM. The result is an even more enriched model of the product and its assembly. As it will be shown in the later parts of this paper, an M2M transformation to the information entities' "parts transport" (i.e., the manner how parts and sub-assemblies are transported within the assembly system from station to station) and a geometric and kinematic representation of the "assembly system" is possible. An M2T transformation can result in the two-dimensional assembly system layout (i.e., the assembly system floor-plan) and the robot code (i.e., the different movement commands for all manipulators within the assembly system). The stepwise enriched product and assembly model, together with the geometric and kinematic information, can be the basis for a digital twin (see Section 5).

Production planning is an anticipation of the manufacturing and assembly operations and systems for the purpose of initial development and optimisation; the production planning process in industrial companies involves various management levels and concerns different planning horizons [17]. This important industrial field is influenced by several current trends. For several years, digital twins (DTs) that provide a digital representation of a real system; enable a bidirectional exchange of parameters, states, and other data; can represent the whole system life-cycle and allow for elaborated monitoring, control, diagnosis, and optimisation processes [18–20] have been available. In this context, integrated control frameworks, both in the product itself—the vehicle—and its assembly system, are gaining importance [21,22] and their development is frequently supported by hardware in the loop (HiL) experiments [23,24], especially in later development stages. The early stages of production planning are highly influenced and altered by the widespread availability of rapid prototyping technology [17]. Based on the same technology, additive manufacturing is meanwhile applied in the production process itself—mainly for low-volume or rather small product and assembly system components [25].

In this context, important research work concerns the design for manufacturing and assembly (DFMA). A concise overview is given by Formentini et al. [26]. Bouissiere et al. proposed methods for the assessment manufacturing and assembly aspects of product architectures [27]. Conceptual design for assembly based on a mathematical framework is also a topic of current research [28]. Some current research works focus on assembly planning in the vehicle industry [29–31]. The context of sports vehicles is usually not the focus of most research work. One example is a study by Arifin et al. who address the design for manufacturing and assembly of sports equipment with the example of an aerobic walker [32]. The modular design of sports vehicles, which also considers assembly processes, is the focus of the work of Sali [33].

Obviously, an essential part of production planning is assembly planning. A complete review of the state-of-the-art automated assembly sequence generation is beyond the scope of this paper; some prominent results will be discussed below. Demoly et al. [2,34] described a product-relationship management approach that enables concurrent product design and assembly sequence planning. He et al. [1] proposed a unified bill of materials (BOMs) model based on a single source of product data. A near-optimum assembly sequence search algorithm employing an objective function (scoring assembling ability as well as the efficiency of the assembly process) is presented by Enomoto et al. [5]. A related study focuses on near-optimal assembly task sequencing for multi-arm robot systems [35]. The optimisation of task allocation for human–robotic assembly processes was researched by Tram et al. [36]. The planning of optimal assembly sequences is addressed by Culbertson et al. [37]. Liu et al. [38] applied deep reinforcement learning in order to enable a physics-aware combinatorial assembly planning. The compatibility of different product models was researched by Paehler and Matthiesen [39].

Upon summarising the literature analysis concerning concurrent and integrated product and assembly system development, it can be pointed out that promising approaches for partial problems exist, but that a continuous M2M transformation-oriented approach is yet to be developed. The following sections explain elements of this kind of approach. Section 2 focuses on the product stage of the integrated process. The stages that follow concern the assembly and production planning and are addressed in Section 3 and Section 4, respectively. The dynamic simulation of the assembly system can be employed as a digital twin (DT); this is analysed and described in Section 5. This paper is concluded by a conclusion and outlook (Section 6).

2. Product Modelling in Graph-Based Design Languages

In the proposed framework, the culmination point for product modelling and the compilation of GBDLs is the so-called Design Cockpit 43® software environment [40], which is based on the fundamental research activities of Rudolph [41] and many of his co-workers (see [42,43] for two of the most recent contributions). The Design Cockpit 43® software was developed, maintained, and further extended by IILS mbH (Trochtelfingen, Germany) [40]. This software environment enables the storage of design information in the form of ontologies, the coding of product and assembly system information as vocabulary, as well the transformation of the coded knowledge into a unified central model—the design graph. The digital design process with GBDLs is shown in Figure 3 (adapted from [41,44]).

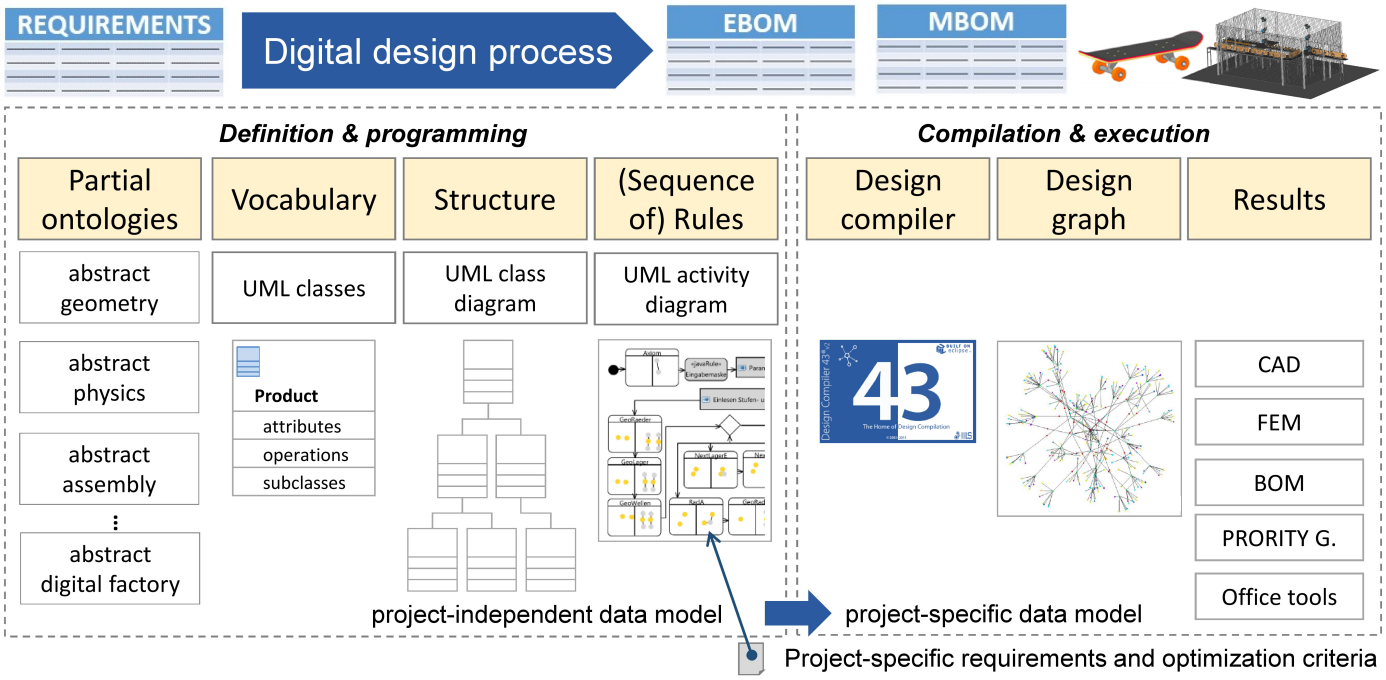


Figure 3. Digital design process with graph-based design languages (adapted from [41,44]).

In the integrated digital design process, two main phases can be distinguished. In the definition and programming phases, the main focus is on project-independent data. Based on partial ontologies, vocabulary, and structure, a sequence of rules is generated. The compilation and execution phase focuses on project-specific, instantiated data. The data from the project-independent phase are compiled leading to the design graph as a central model. From the design graph, different kinds of domain-specific data models can be compiled such as CAD files. The current research concentrates on Open Cascade, 7.8.0, Open Cascade SAS, Guyancourt, France [45], and CATIA, Dassault Systèmes, Vélizy-Villacoublay, France [46], but other CAD systems can also be integrated. For finite element analyses,

the software systems Ansys, Ansys, Inc., Canonsburg, PA, USA [47], or Abaqus FEA, Dassault Systèmes, Vélizy-Villacoublay, France [48], are widely used in industry. BOMs can be represented in Microsoft Excel, Microsoft Corporation, Redmond, WA, USA [49], but professional CAD systems such as PTC Creo, 10.0, PTC Inc., Boston, MA, USA [50], can also store BOMs.

The digital design process shown in Figure 3 has already been described in detail in earlier publications [44,51]. It is important to note that the general procedure is applicable to all M2M and M2T transformations in integrated product and assembly system development. In the following, the architecture design and the underlying methodology, which is illustrated in Figure 2, will be discussed using the product model of a skateboard (see Figure 4).

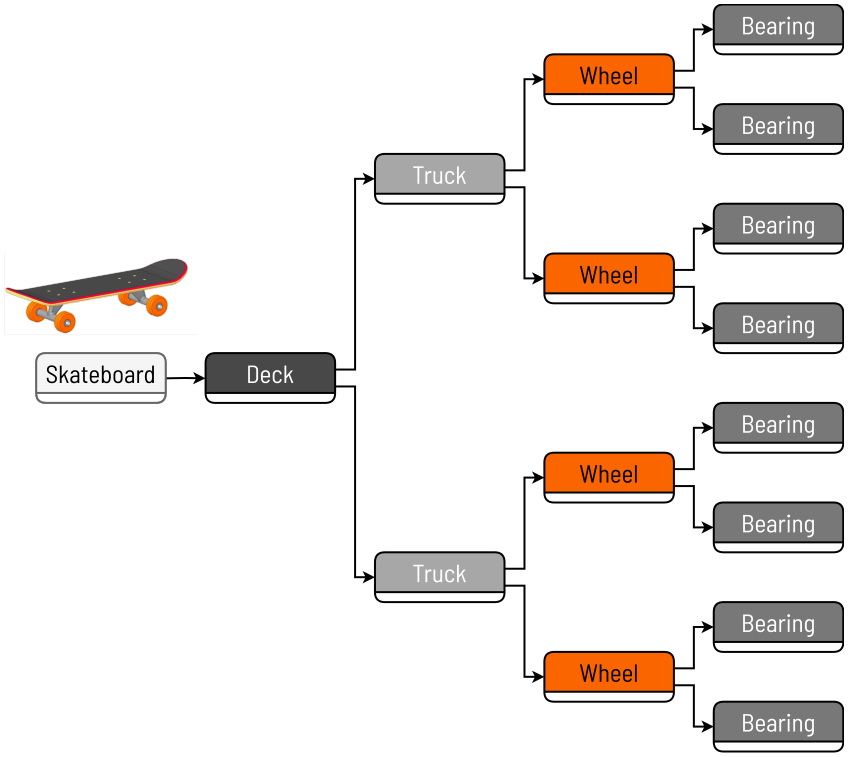


Figure 4. Product model of the skateboard as a knowledge graph.

Figure 4 represents the product model and is intended to familiarise the reader with the main structure of the product in order to ease the understanding of the assembly sequence. The model is based on the "Skateboard" node, which reflects the basic model. The "Deck" node refers to the top board of the skateboard and is therefore an essential component of the model. Two so-called "Trucks" are connected to it and represent the axles of the skateboard. Each axle is connected to two wheels, which in turn dispose of two bearings. In the following, the model described in Figure 4 is supplemented with joining information. This information describes the way in which two components are connected and contains all the information that is relevant for describing the joining process. The underlying product architecture is realised by means of three different assembly operations, which are modelled as rules. Figure 5 presents the assembly of a bearing in a wheel.

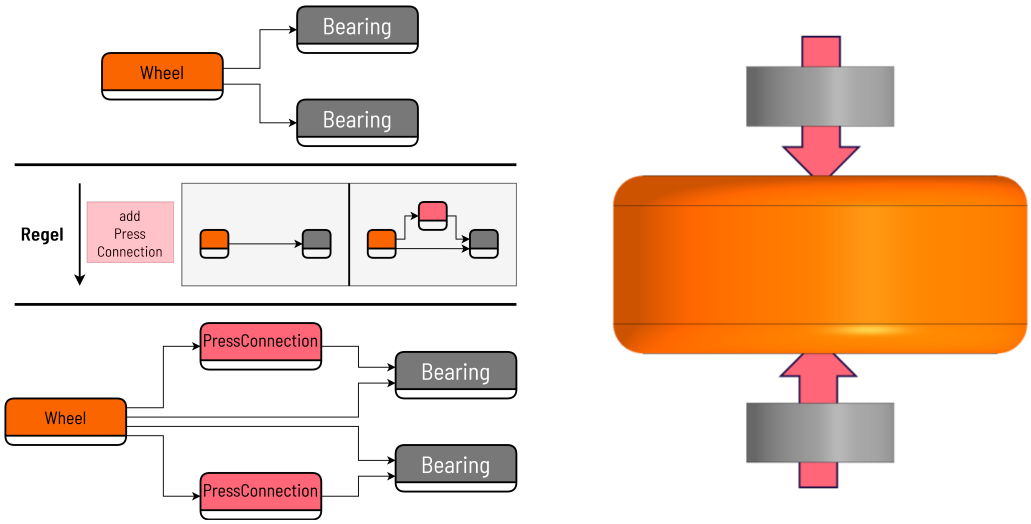


Figure 5. Assembly of bearings in wheels. (**Top left**) Knowledge graph before rule execution. (**Bottom left**) Knowledge graph after rule execution. (**Right**) A 3D CAD model.

This assembly is represented by "PressConnection" (Figure 6). The assembly operation modelled with this rule is implicitly visible on the right side of Figure 4; slightly further on the left in Figure 4 is the assembly of the wheels on an axle of a truck (Figure 6).

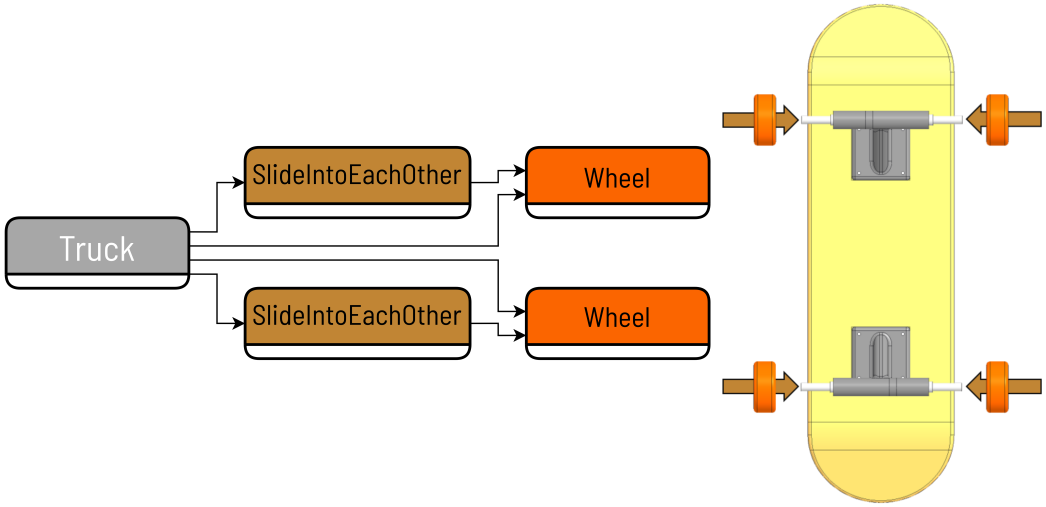


Figure 6. Assembly of wheels. (**Left**) Knowledge graph. (**Right**) A 3D CAD model.

This assembly is represented by "SlideIntoEachOther" (Figure 6). The next assembly step concerns the connection of each truck with the deck by means of screwing. This is presented in Figure 7.

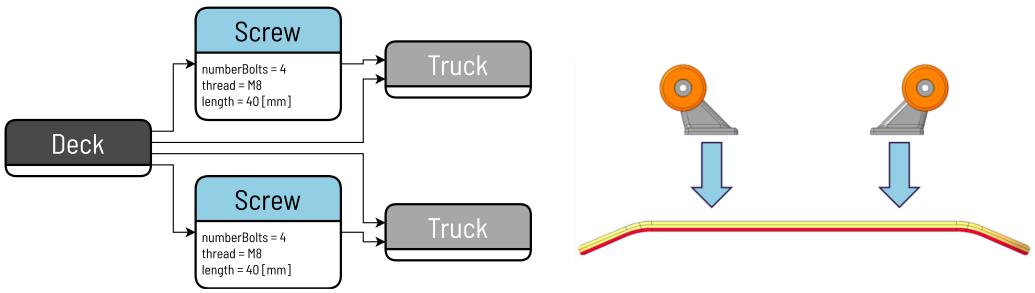


Figure 7. Assembly of trucks. (**Left**) Knowledge graph. (**Right**) A 3D CAD model.

This assembly is represented by "Screw" (Figure 7). Representing these assembly operations, the product model of the skateboard shown in Figure 4 is supplemented with joining information; the result is an enriched product model (see also Figure 2). This enriched product model is shown in Figure 8.

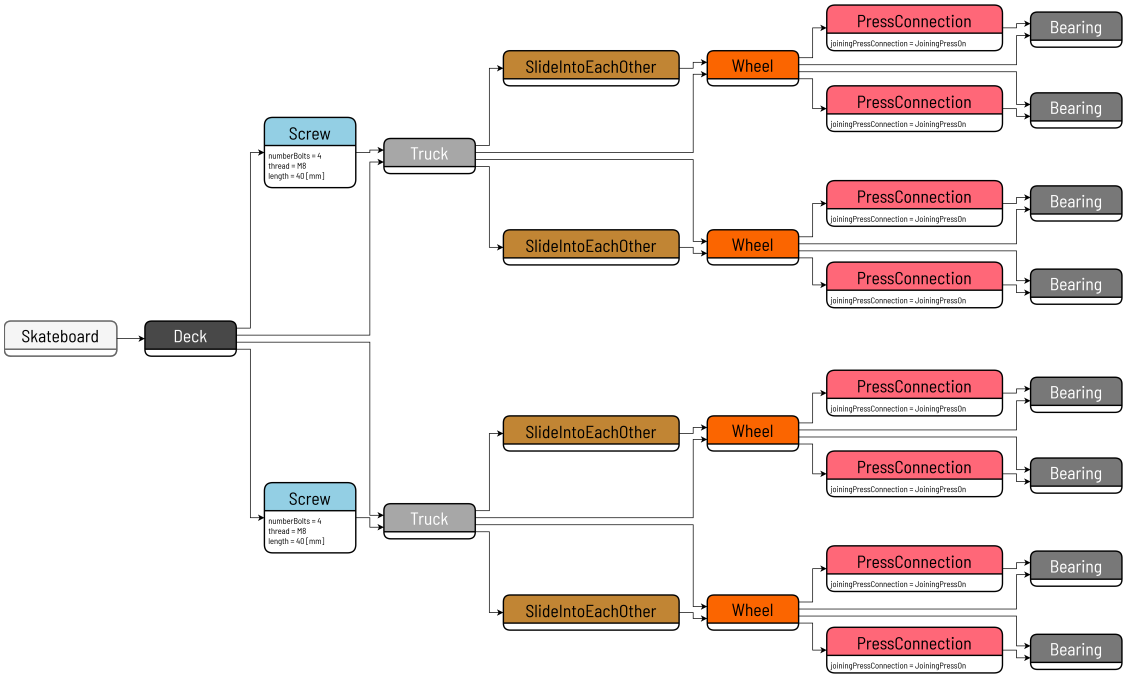


Figure 8. Enriched product model of the skateboard as a knowledge graph.

The enriched product model can also be understood as an interface to the more detailed steps of the assembly planning.

3. Assembly Planning

Assembly planning involves determining the sequence of assembly operations and the resources needed to assemble a product from its modules and components. It describes the process of organising the assembly, i.e., determining the order in which modules and parts should be assembled (the assembly sequence) and the required resources for each assembly operation. The generation of feasible assembly sequences remains a challenging research problem due to the following reasons: the number of potential assembly sequences exponentially grows, some sequences may not be physically executable due to unsatisfied physical constraints, and the geometry of objects can be highly complex [52].

The optimal sequence can be determined using a number of different methods. It is important to consider these methods because the number of possible sequences increases exponentially with each additional combination. In the case of the skateboard, which consists of $p = 15$ parts, this corresponds to a number of 1,307,674,368,000 sequences. Even if we ignore all sequences that are the inverse of another sequence, there are still 653,837,184,000 sequences. It should be noted that the majority of these sequences would not work or would only be feasible with great technical effort. Various methods have been developed to reduce this number to a minimum and to filter out the most important sequences [53–55].

In practice, the starting point is often the assembled module, which is then disassembled. A feasible disassembly sequence always corresponds to a working assembly sequence in reverse order. A common method is to first identify the mating surfaces between components. This can be performed by collision detection, the distance between components, or the structure of the model (as exemplified in Figure 8).The data may be processed to

create an adjacency matrix. The resulting matrix depicts the interconnections between all components. The generated adjacency matrix may be employed as a foundation for the derivation of an interference matrix. The resulting matrix provides information regarding the components that are in contact in terms of their joining direction during the assembly process. Consequently, the assembly sequences can be derived [54]. It is likewise possible to make use of machine learning methodologies [55].

In the case of assemblies comprising a limited number of components, it is feasible to construct a Petri net that maps all sequences directly from the adjacency matrix. The corresponding sequences can be extracted from this using the enumeration algorithm, for example. Furthermore, an assembly priority graph can be derived from the adjacency matrix by enriching the boundary conditions. It is recommended that the assembly be initially analysed at the assembly level, and subsequently, the assembly sequence for the respective assemblies be determined. This is essential as the number of potential combinations increases exponentially with the level of detail.

In order to determine the number of similar assemblies, a width-based search is carried out in which the model is first simplified and then grouped. The width-based search performed on the existing graph, which functions as an enriched product model, is carried out depending on the requirements for fully and partially similar cases. A fully similar case would be defined by identical components (in terms of geometric characteristics and properties) with the same joining technique. Partially similar cases would be characterised by components that are similar in many, but not all, properties and whose joining techniques also have a large number of identical properties. As a result of this search, five assembly sets ("AssemblySet"—a collection of parts that require similar assembly operations) are identified, as shown in Figure 9.

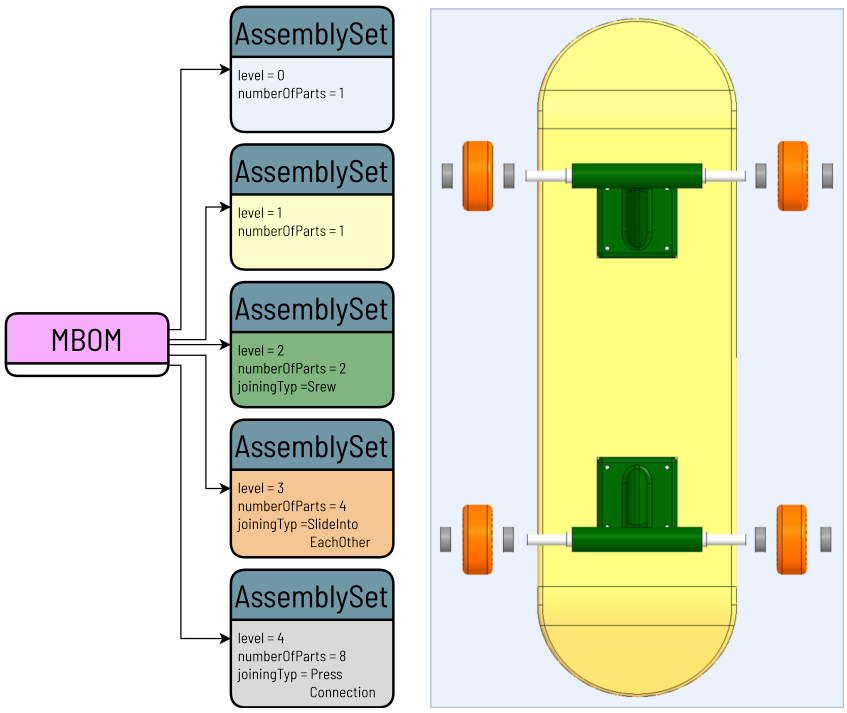


Figure 9. Assembly sets of the skateboard. (**Left**) Knowledge graph. (**Right**) A 3D CAD model.

The transformation of the product model into an assembly model is used to encapsulate the product and to transform it into a model suitable for assembly planning. Process planning is performed consistently from the previously created model, using the assembly sets as the starting point. The main processes, shown in Figure 10, are then derived from the assembly sets.

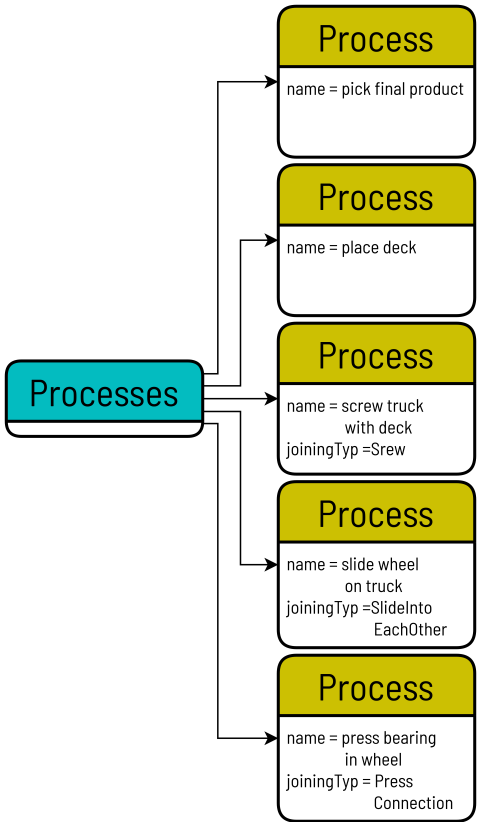


Figure 10. Process planning data model as a knowledge graph.

The main process is a set of sub-processes required to complete an assembly to the point where another assembly sequence can be started. Depending on the components used and the joining technology employed, the main processes are supplemented by sub-processes and handling steps. An example is shown in Figure 11.

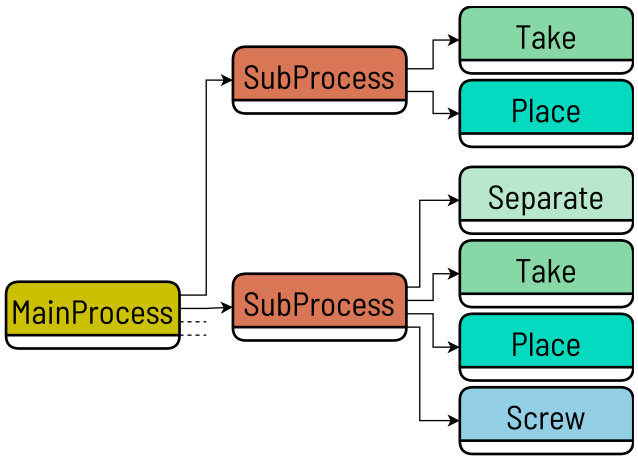


Figure 11. Partial section of the process planning for mounting the axle on the deck as knowledge graph.

The initial assembly sequence is derived from the structure of the data model, as shown in Figure 4, and the resulting assembly sets. These comprise a level that reflects the levels of the model. The assembly sequence begins with the lowest level, which, in the case of the skateboard, represents the pressing in of the layer. The correctness of the assembly sequence is validated by collision detection.

The formal description of the main and sub-processes delineates the full range of handling and joining techniques. These describe the minimum steps that must be carried

out in order for the assembly step to be completed successfully. Based on the aforementioned information, an initial estimation of the cycle time can be made. Subsequently, the cycle time is estimated for each individual handling step and for each joining technique. In this context, two distinct approaches are employed. In this approach, tabular values derived from the "Methods-Time-Measurement" [56] method are employed. The present analysis focuses on the temporal recording of handling steps. The analysis focuses on handling steps employed in manual assembly, with some values applicable to automated assembly. The determined value is multiplied by a correction factor, which serves as an adjustment variable. This factor is adjusted iteratively by comparing the tabular value with the subsequent actual value from the virtual commissioning and adjusting it accordingly. For all other handling steps not included in the table, a simplified approach is used in the calculation.

The calculation of t for the handling steps *take* and *place* is described in Equation (1). The product of the two factors $c_l(l_{max})$ and $c_m(m)$, the longest edge of the bounding box, l_{max}, and the average speed of the robot, v_{robot}, give the value of t:

$$t = \underbrace{c_l(l_{max})}_{\text{length heuristic}} \cdot \underbrace{c_m(m)}_{\text{mass heuristic}} \cdot l_{max} \cdot v_{robot} \tag{1}$$

It is assumed that the distance that the component has to cover can be described as a function of its dimensions (see *take* and *place*). For this purpose, the factor $c_l(l_{max})$ was defined, the value of which is calculated according to the definition given in Equation (2):

$$c_l(l_{max}) = \begin{cases} c_{l0} & \text{if } 0\text{ mm} < l_{max} <= 50\text{ mm}, \\ \vdots & \vdots \\ c_{ln} & \text{if } \dots \end{cases} \quad \text{where } n \in \mathbb{N}_0, \quad c_{ln} \in \mathbb{R} > 0 \tag{2}$$

The factor $c_m(m)$ defined in Equation (3) represents a correction factor that takes the weight of the component into account:

$$c_m(m) = \begin{cases} c_{m0} & \text{if } 0\text{ kg} < m <= 0.1\text{ kg}, \\ \vdots & \vdots \\ c_{mn} & \text{if } \dots \end{cases} \quad \text{where } n \in \mathbb{N}_0, \quad c_{mn} \in \mathbb{R} > 0 \tag{3}$$

The parameters $c_{l1}, ..., c_{ln}$ and $c_{m1}, ..., c_{mn}$ were freely selected at the start of the process. Following the method described above, a virtual commissioning is carried out. Based on the results obtained, the factors $c_l(l_{max})$ and $c_m(m)$ are adjusted iteratively. The aim of this procedure is to ensure that the result of the early estimation continuously adapts to the actual cycle time. However, verification using sufficient data is still required. If the application of the present approach does not lead to the desired results, it is possible to train a machine learning model with a sufficient database, which can then be used to predict the cycle time.

The calculation of the total t_{cycle} is described in Equation (4):

$$t_{\text{cycle}} = \sum_{i=1}^{h} t_i \quad \text{(with } h \text{ number of handle and joining steps)} \quad \textit{where } h \in \mathbb{N}_+ \tag{4}$$

Based on this modelling, an initial estimate of the cycle time is possible, as shown in Figure 12.

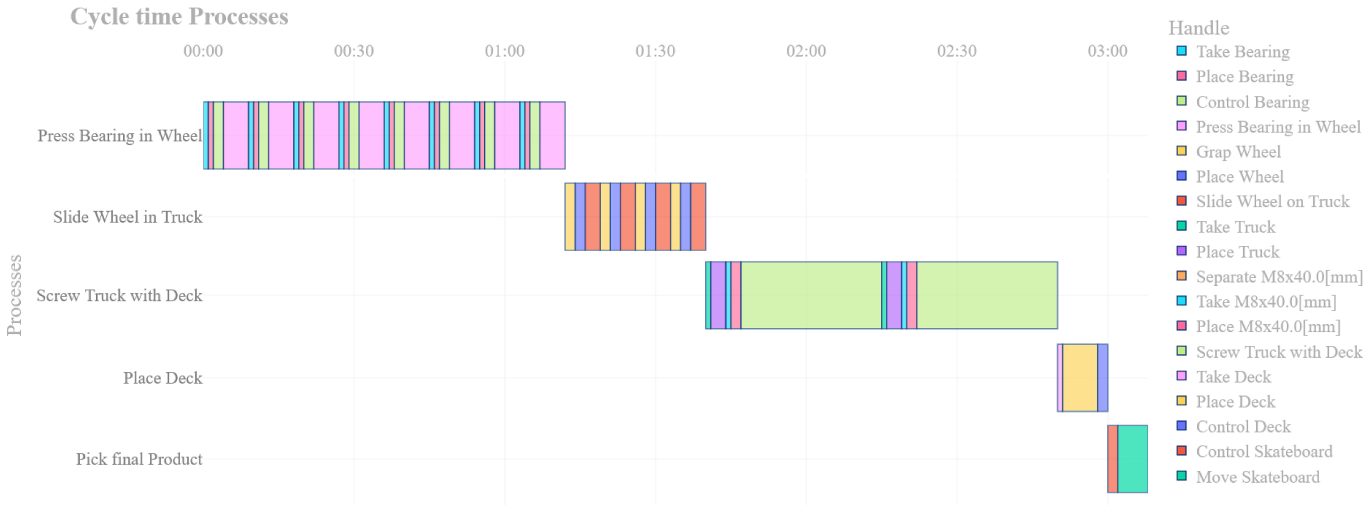


Figure 12. Cycle time diagram of the main processes.

As part of the process optimisation, the cycle time is compared with the specified requirements. If the determined cycle time fulfils the specified requirements, production planning can continue. If the specified cycle time is exceeded, the sequence is optimised. For this purpose, the main processes are split up, for example, to parallelise processes. As part of each optimisation, a check is carried out to determine whether the resulting outcome meets the specified requirements. This applies in particular to the cycle time, the available number of multi-axis robots, and the available space. The optimisation is terminated as soon as all requirements or the termination criteria are met.

Based on the assembly sets, a sensible assembly sequence could be generated via M2M transformation, which can serve as a basis for the further steps of the production planning.

4. Production Planning

In this paper, the term "production planning" is used to describe the steps following the assembly planning, i.e., planning the concrete transport of parts in the assembly system, generating the 2D assembly system layout, generating the geometry of the assembly system with all non-human resources, and generating the movement path of resources (e.g., robot code). In the given project, the descriptions of the assembly system could be generated by taking into account the cycle time and the requirements. These describe the processes that are carried out in a cell, as well as the resources required for this, such as robots, holding devices, and conveyor belts. The above descriptions enable the derivation of a 2D layout, which is shown in Figure 13.

In a final step, the 3D assembly system model is derived from the 2D layout (Figure 14). A minimum configuration can be derived by considering the main and sub-processes. The configuration encompasses all the resources indispensable for the comprehensive execution of the principal process. Such equipment may include, for instance, multi-axis robots, tables, and front conveyors. Subsequently, the principal processes are conveyed to assembly systems, wherein the cycle time has been optimised and considered in advance. In the subsequent phase, the previously identified systems are integrated with the resources of the minimum configuration. The working area of the assembly system is defined on the basis of the robot employed. It is imperative that, within the context of the current configuration, all sub-processes are conducted within the specified workspace. This entails the utilisation of the assigned resources. Moreover, a transport route must be established between the workspaces to ensure the secure and efficient transportation of parts.

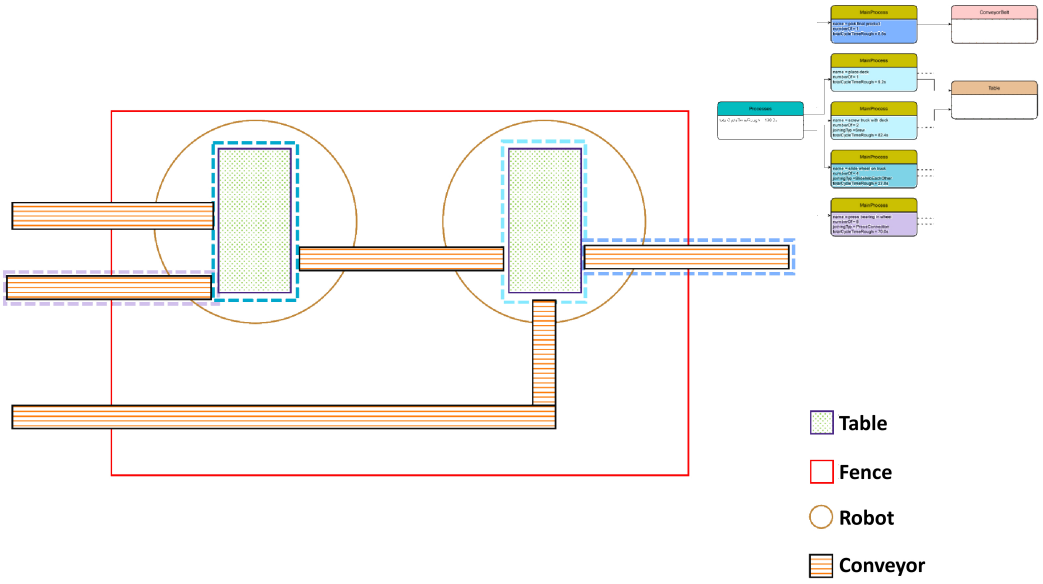


Figure 13. Two-dimensional layout of the system.

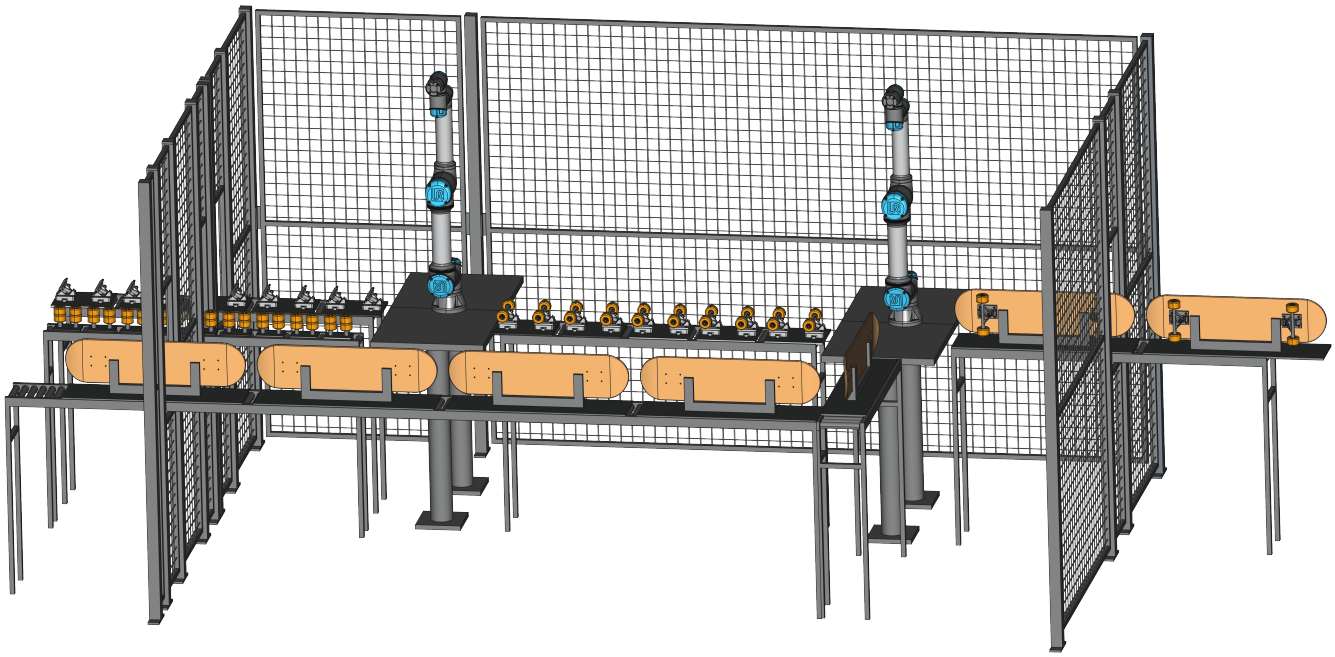


Figure 14. Three-dimensional assembly system model.

In the initial stage of the process, all resources are positioned within the designated workspace. Subsequently, an optimisation process utilising a packing algorithm is employed. Once the individual assembly systems have been generated, they are positioned globally and connected to each other using transport systems. Subsequently, a delineating structure, such as a fence, is erected around the assembly systems. The final result is a two-dimensional representation of the assembly cell, as illustrated in Figure 13. The requisite resources and layout dimensions are associated with each principal process.

In the final stage of the process, the 3D assembly system model is derived from the 2D layout. In order to achieve this, the enriched line models depicted in Figure 13 are employed, with due consideration given to the pertinent requirements. Such information was provided during the automated generation process. For example, data regarding the height of the fences were incorporated into the line model. The height of the fences is determined either

by the requirements or, in the absence of a defined requirement, by the maximum height of the workspace. Subsequently, the 3D geometry is generated automatically, taking into account the aforementioned information and the 2D layout. The individual handling steps and the respective positioning are employed to derive a robot code, which is then utilised to assess the configuration. In the event that the assembly is not feasible, a return is made to the 2D layout, and the configuration is re-optimised. Conversely, if the assembly can be carried out as required, the automated design process is completed and the system can be exported to the standardised AutomationML data format.

The proposed design architecture enables the generation of simulations (see [57]) and of a digital twin of the assembly system.

5. Digital Twin

The culmination of the models created in an integrated product and assembly system development process can be employed as a dynamic simulation of the assembly system in the form of a digital twin. In general, a digital twin is understood as a virtual dynamic representation of a physical system with automatic and bidirectional exchanged data [58]. It is important to note that a full-scale digital twin (in contrast to a digital shadow) contains data of the complete product life-cycle [18]. The continuous flow of M2M transformations described in this paper leads to a unified digital twin (see Figure 2). In Figure 15, the architecture of employing this digital twin is shown.

For the given field of application, the Design Cockpit 43 [40] serves as a specialised editor to design a semantic model of a product and its assembly system. This system generates outputs in the form of ontologies and product and assembly system geometry, structure, and behaviour. The coded information is transformed into a knowledge graph. This knowledge graph can be enriched with performance data from an operating assembly system (execution—past assembly runs) and can interact with artificial intelligence (AI) and explainable artificial intelligence (XAI) [59] components in the digital twin framework [60]. The knowledge graph also allows for the execution of detailed simulations.

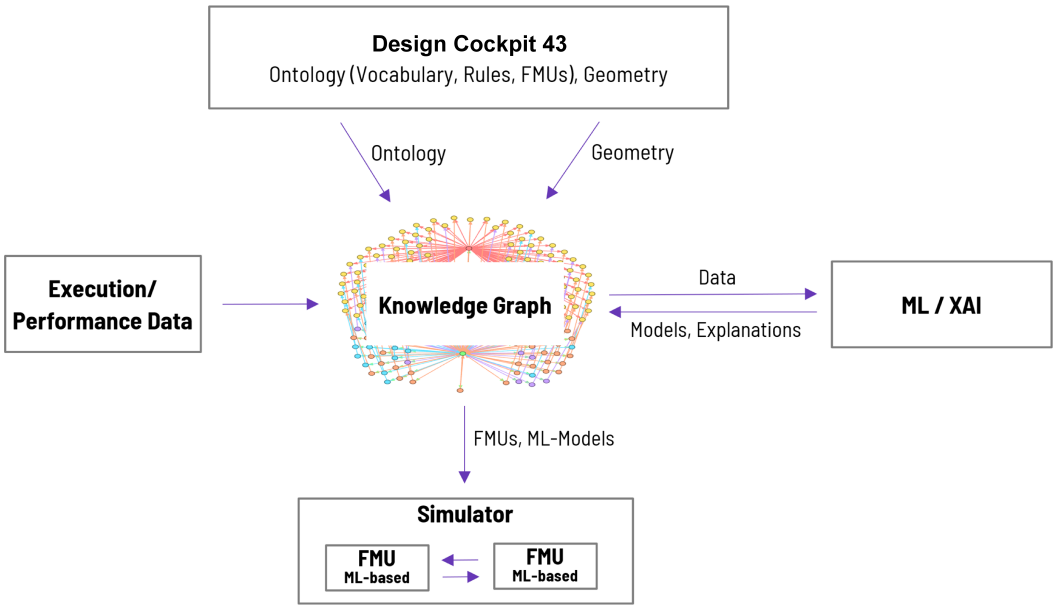


Figure 15. Digital twin architecture [60].

In order to allow for flexible simulations of the system behaviour, the behaviour is divided into so-called functional mock-up units (FMUs) [61]. In this kind of co-simulation, each FMU allows us to simulate an independent dynamic behaviour in the form of an input–output relation [62]. In order to model this relation, systems of ordinary differential

equations (ODEs) are usually employed [63]. However, in the given framework, machine learning (ML)-based FMUs are also possible (see Figure 15 and also [64,65]). In this case, the input–output relation (simulating the dynamic behaviour) of the FMUs can be deduced from assembly system performance data.

As pointed out in Section 1, the step-wise enriched product and assembly model together with the geometric and kinematic information can be the basis for a digital twin; consequently, the different M2M transformations are essential steps in the digital twin creation (see Figure 2).

6. Conclusions and Outlook

The scientific core of this paper is the proposal of a consistent digital framework for the integration of product structure and geometry, assembly sequence, and assembly system structure, geometry, and behaviour. This framework is based on the application of graph-based design languages and allows for an automated design of a complete assembly system for a given product. The application of this framework leads to two machine-executable V-models for product development and assembly system development in the scope of MBSE. Unlike traditional MBSE approaches that rely on document-based process management, this framework uses abstract models, which can be automatically compiled into detailed product models and assembly system models.

The advantage of this approach is the realisation of a central, multi-domain model in the form of a knowledge graph, which can also be the basis for co-simulation based on functional mock-up units (FMUs) and for certain forms of machine learning or explainable artificial intelligence. A central element was the generation of a sensible assembly sequence; several possibilities for the challenging endeavour were explored. The resulting sequence is based on predefined assembly sets and takes into account the relevant necessities and possibilities of the assembly process. The approach is universal and can be used in the production of components for the automotive industry, their assemblies, and entire vehicles. Future research work will be aimed at even more complex products and more diverse assembly systems with human beings for certain assembly steps. Integration with AI methods is a further promising field for further research.

Author Contributions: Conceptualisation, T.S., M.T., R.S. and S.R.; methodology, T.S., M.T., R.S. and S.R.; validation, T.S.; investigation, T.S.; writing—original draft preparation, T.S. and R.S.; writing—review and editing, M.T. and S.R. All authors have read and agreed to the published version of the manuscript.

Funding: Parts of the described research were funded by the Carl Zeiss Foundation in the scope of the project AI-based digital twin (KI-basierter digitaler Zwilling, KIDZ).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study. Requests to access the datasets should be directed to markus.till@rwu.de.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2D	two-dimensional
3D	three-dimensional
AI	artificial intelligence
CAD	computer-aided design

DFMA	design for manufacturing and assembly
DT	digital twin
EBOMs	engineering bill of materials
FMU	functional mock-up unit
GBDL	graph-based design language
HiL	hardware in the loop
M2M	model-to-model
M2T	model-to-text
MBOMs	manufacturing bill of materials
MBSE	model-based systems engineering
ML	machine learning
ODE	ordinary differential equation
PDM	product data management
PLM	product life-cycle management
XAI	explainable artificial intelligence

References

1. He, L.; Ni, Y.; Ming, X.; Li, M.; Li, X. Integration of bill of materials with unified bill of materials model for commercial aircraft design to manufacturing. *Concurr. Eng.* **2014**, *22*, 206–217. [CrossRef]
2. Demoly, F.; Dutartre, O.; Yan, X.T.; Eynard, B.; Kiritsis, D.; Gomes, S. Product relationships management enabler for concurrent engineering and product lifecycle management. *Comput. Ind.* **2013**, *64*, 833–848. [CrossRef]
3. Layer, M.; Neubert, S.; Stelzer, R. Introducing a multipliable BOM-based automatic definition of information retrieval in plant engineering. *Proc. Des. Soc.* **2024**, *4*, 413–422. [CrossRef]
4. Cinelli, M.; Ferraro, G.; Iovanella, A.; Lucci, G.; Schiraldi, M.M. A network perspective on the visualization and analysis of bill of materials. *Int. J. Eng. Bus. Manag.* **2017**, *9*, 1847979017732638. [CrossRef]
5. Enomoto, A.; Yamamoto, N.; Yamamura, Y.; Sugawara, Y. Process knowledge integrated assembly sequence planning for control panel. *Int. J. Autom. Technol.* **2020**, *14*, 6–17. [CrossRef]
6. Gräßler, I.; Wiechel, D.; Pottebaum, J. Role model of model-based systems engineering application. In *IOP Conference Series: Materials Science and Engineering, Proceedings of the 19th Drive Train Technology Conference (ATK 2021), Aachen, Germany, 9–11 March 2021;* IOP Publishing: Bristol, UK, 2021; Volume 1097, p. 012003.
7. Pulm, U.; Stetter, R. Systemic mechatronic function development. *Proc. Des. Soc.* **2021**, *1*, 2931–2940. [CrossRef]
8. Stetter, R.; Pulm, U. Strategies and Methods for the Fault-Tolerant Function Development of Multi-Domain Systems. *Appl. Sci.* **2024**, *14*, 11646. [CrossRef]
9. VDI/VDE. *VDI/VDE 2006: Development of Cyber-Physical Mechatronic Systems (CPMS)*; Beuth: Berlin, Germany, 2020.
10. Graessler, I.; Hentze, J. The new V-Model of VDI 2206 and its validation. *At-Autom.* **2020**, *68*, 312–324. [CrossRef]
11. Eisenbart, B.; Gericke, K.; Blessing, L.T.; McAloone, T.C. A DSM-based framework for integrated function modelling: Concept, application and evaluation. *Res. Eng. Des.* **2017**, *28*, 25–51. [CrossRef]
12. Prager, F.; Tröster, P.M.; Albers, A. The C&C^2 Approach as a Thinking Tool in Mechatronic Systems. *Procedia CIRP* **2024**, *128*, 704–709.
13. Striffler, N.; Voigt, T. Concepts and trends of virtual commissioning–A comprehensive review. *J. Manuf. Syst.* **2023**, *71*, 664–680. [CrossRef]
14. Rueckert, P.; Goetsche, A.K.; Tracht, K. Assembly workshops in virtual reality as an integral part of the assembly planning process. *Procedia CIRP* **2024**, *126*, 793–798. [CrossRef]
15. Unreal Engine. Epic Games. Cary, North Carolina, USA. 2025. Available online: https://www.unrealengine.com/ (accessed on 12 February 2025).
16. Unity. Unity Technologies. San Francisco, California, USA. 2025. Available online: https://unity.com/ (accessed on 12 February 2025).
17. Fertsch, M.; Stachowiak, A.; Oleśków-Szłapka, J. Innovations-Changes in the Environment of the Production Planning Process in Enterprises. In Proceedings of the International Scientific-Technical Conference MANUFACTURING, Poznan, Poland, 14–16 May 2024; Springer: Cham, Switzerland, 2024; pp. 201–212.
18. Trauer, J.; Schweigert-Recksiek, S.; Engel, C.; Spreitzer, K.; Zimmermann, M. What is a digital twin?–definitions and insights from an industrial case study in technical product development. In Proceedings of the Design Society: DESIGN Conference, Online, 25 October 2020; Cambridge University Press: Cambridge, UK, 2020; Volume 1, pp. 757–766.
19. Javaid, M.; Haleem, A.; Suman, R. Digital twin applications toward industry 4.0: A review. *Cogn. Robot.* **2023**, *3*, 71–92. [CrossRef]

20. Pronost, G.; Mayer, F.; Camargo, M.; Dupont, L. Digital Twins along the product lifecycle: A systematic literature review of applications in manufacturing. *Digit. Twin* **2024**, *3*, 3. [CrossRef]
21. Liang, J.; Lu, Y.; Yin, G.; Fang, Z.; Zhuang, W.; Ren, Y.; Xu, L.; Li, Y. A distributed integrated control architecture of AFS and DYC based on MAS for distributed drive electric vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5565–5577. [CrossRef]
22. Kargar, M.; Zhang, C.; Song, X. Integrated optimization of power management and vehicle motion control for autonomous hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2023**, *72*, 11147–11155. [CrossRef]
23. Liang, J.; Lu, Y.; Wang, F.; Yin, G.; Zhu, X.; Li, Y. A robust dynamic game-based control framework for integrated torque vectoring and active front-wheel steering system. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 7328–7341. [CrossRef]
24. Zhang, H.; Wang, S.; Zhao, X.; Zheng, Z. Path tracking and Handling Stability Coordinated Control of 4WS and DYC for Distributed in-Wheel Motor Drive Electric Vehicle Under Extreme Conditions. *IEEE Trans. Veh. Technol.* **2024**, *73*, 18402–18417. [CrossRef]
25. Jin, L.; Zhai, X.; Wang, K.; Zhang, K.; Wu, D.; Nazir, A.; Jiang, J.; Liao, W.H. Big data, machine learning, and digital twin assisted additive manufacturing: A review. *Mater. Des.* **2024**, *244*, 113086. [CrossRef]
26. Formentini, G.; Boix Rodríguez, N.; Favi, C. Design for manufacturing and assembly methods in the product development process of mechanical products: A systematic literature review. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 4307–4334. [CrossRef]
27. Bouissiere, F.; Cuiller, C.; Dereux, P.E.; Malchair, C.; Favi, C.; Formentini, G. Conceptual design for assembly in aerospace industry: A method to assess manufacturing and assembly aspects of product architectures. In Proceedings of the Design Society: International Conference on Engineering Design, Delft, The Netherlands, 5–8 August 2019; Cambridge University Press: Cambridge, UK, 2019; Volume 1, pp. 2961–2970.
28. Formentini, G.; Favi, C.; Cuiller, C.; Dereux, P.E.; Bouissiere, F.; Jurbert, C. Conceptual design for assembly in aerospace industry: Sensitivity analysis of mathematical framework and design parameters. *Proc. Des. Soc.* **2021**, *1*, 731–740. [CrossRef]
29. Tang, H. An integrated product-process hierarchical modeling method for development of complex assembly manufacturing systems. *Procedia CIRP* **2018**, *76*, 2–6. [CrossRef]
30. Manimuthu, A.; Venkatesh, V.; Shi, Y.; Sreedharan, V.R.; Koh, S.L. Design and development of automobile assembly model using federated artificial intelligence with smart contract. *Int. J. Prod. Res.* **2022**, *60*, 111–135. [CrossRef]
31. Kshatra, D.P.; Akhil, S.; Kiran, U.; Vineeth, Y. Process Design and system Layout for an automobile Manufacturing and assembly plant. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *9*, 440–446. [CrossRef]
32. Arifin, Z.; Prasetyo, S.D.; Prabowo, A.R.; Cho, J.H. Preliminary design for assembling and manufacturing sports equipment: A study case on Aerobic Walker. *Int. J. Mech. Eng. Robot. Res.* **2021**, *10*, 107–115. [CrossRef]
33. Sali, M. Designing an Innovative Modular Platform for Sports Car Using the Generative Design Method. Doctoral Dissertation, University of Bologna, Bologna, Italy, 2024.
34. Demoly, F.; Yan, X.T.; Eynard, B.; Gomes, S.; Kiritsis, D. Integrated product relationships management: A model to enable concurrent product design and assembly sequence planning. *J. Eng. Des.* **2012**, *23*, 544–561. [CrossRef]
35. Enomoto, A.; Hayashi, N.; Inoue, R.; Tsutsumi, D.; Kajita, D.; Nakasu, N. Near-Optimal Assembly Task Sequencing and Allocation Method for Multi-Arm Robot System. In Proceedings of the 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, 26–30 August 2023; pp. 1–7. [CrossRef]
36. Tram, A.V.N.; Raweewan, M. Optimal task allocation in human-robotic assembly processes. In Proceedings of the 2020 5th International Conference on Robotics and Automation Engineering (ICRAE), Singapore, 20–22 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 158–162.
37. Culbertson, P.; Bandyopadhyay, S.; Schwager, M. Multi-Robot Assembly Sequencing via Discrete Optimization. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 6502–6509. [CrossRef]
38. Liu, R.; Chen, A.; Zhao, W.; Liu, C. Physics-Aware Combinatorial Assembly Planning using Deep Reinforcement Learning. *arXiv* **2024**, arXiv:2408.10162.
39. Paehler, L.; Matthiesen, S. Mapping the landscape of product models in embodiment design. *Res. Eng. Des.* **2024**, *35*, 289–310. [CrossRef]
40. Design Cockpit 43. IILS Ingenieurgesellschaft für Intelligente Lösungen und Systeme mbH, Trochtelfingen, Germany. Available online: https://www.iils.de/ (accessed on 18 February 2025).
41. Rudolph, S. Übertragung von Ähnlichkeitsbegriffen. Habilitation Thesis, Fakultät Luft- und Raumfahrttechnik und Geodäsie, Universität Stuttgart, Stuttgart, Germany, 2002.
42. Neumaier, M.; Kranemann, S.; Kazmeier, B.; Rudolph, S. Automated Piping in an Airbus A320 Landing Gear Bay Using Graph-Based Design Languages. *Aerospace* **2022**, *9*, 140. [CrossRef]
43. Zech, A.; Stetter, R.; Rudolph, S.; Till, M. Capturing the Design Rationale in Model-Based Systems Engineering of Geo-Stations. *Proc. Des. Soc.* **2022**, *2*, 2015–2024. [CrossRef]

44. Elwert, M.; Ramsaier, M.; Eisenbart, B.; Stetter, R.; Till, M.; Rudolph, S. Digital Function Modeling in Graph-Based Design Languages. *Appl. Sci.* **2022**, *12*, 5301. [CrossRef]
45. Open Cascade, Open Cascade SAS, Guyancourt, France. 2025. Available online: https://www.opencascade.com (accessed on 12 February 2025).
46. CATIA, Dassault Systèmes, Vélizy-Villacoublay, France. 2025. Available online: https://www.3ds.com/products-services/catia/ (accessed on 12 February 2025).
47. Ansys, Ansys, Inc., Canonsburg, Pennsylvania, USA. 2025. Available online: https://www.ansys.com (accessed on 12 February 2025).
48. Abaqus FEA, Dassault Systèmes, Vélizy-Villacoublay, France. 2025. Available online: https://www.simulia.com (accessed on 12 February 2025).
49. Microsoft Excel, Microsoft Corporation, Redmond, Washington, USA. 2025. Available online: https://microsoft.com/en-us/microsoft-365/excel (accessed on 12 February 2025).
50. PTC Creo, PTC Inc., Boston, Massachusetts, USA. 2025. Available online: https://ptc.com/en/products/cad/creo/ (accessed on 12 February 2025).
51. Holder, K.; Zech, A.; Ramsaier, M.; Stetter, R.; Niedermeier, H.P.; Rudolph, S.; Till, M. Model-Based Requirements Management in Gear Systems Design Based On Graph-Based Design Languages. *Appl. Sci.* **2017**, *7*, 1112. [CrossRef]
52. Tian, Y.; Willis, K.D.; Al Omari, B.; Luo, J.; Ma, P.; Li, Y.; Javid, F.; Gu, E.; Jacob, J.; Sueda, S.; et al. ASAP: Automated sequence planning for complex robotic assembly with physical feasibility. In Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, 13–17 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 4380–4386.
53. Abdullah, M.A.; Ab Rashid, M.F.F.; Ghazalli, Z. Optimization of Assembly Sequence Planning Using Soft Computing Approaches: A Review. *Arch. Comput. Methods Eng.* **2019**, *26*, 461–474. [CrossRef]
54. Yu, J.; Yu, J.; Wang, C.; Wang, C. Method for discriminating geometric feasibility in assembly planning based on extended and turning interference matrix. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 1867–1882. [CrossRef]
55. Sınanoğlu, C.; Börklü, H.R. An assembly sequence-planning system for mechanical parts using neural network. *Assem. Autom.* **2005**, *25*, 38–52. [CrossRef]
56. Syska, A. *Produktionsmanagement: Das A–Z Wichtiger Methoden und Konzepte für die Produktion von Heute*; Gabler Verlag Wiesbaden: Wiesbaden, Germany, 2006; p. 188. [CrossRef]
57. Bendowska, K.; Zawadzki, P. Development and verification of a simulation model of an automated assembly line. *Appl. Sci.* **2023**, *13*, 10142. [CrossRef]
58. Trauer, J.; Pfingstl, S.; Finsterer, M.; Zimmermann, M. Improving Production Efficiency with a Digital Twin Based on Anomaly Detection. *Sustainability* **2021**, *13*, 10155. [CrossRef]
59. Došilović, F.K.; Brčić, M.; Hlupić, N. Explainable artificial intelligence: A survey. In Proceedings of the 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 21–25 May 2018; pp. 210–215.
60. Arff, B.; Haasis, J.; Thomas, J.; Bonenberger, C.; Höpken, W.; Stetter, R. Analysis and visualization of production bottlenecks as part of a digital twin in industrial IoT. *Appl. Sci.* **2023**, *13*, 3525. [CrossRef]
61. Larsen, P.G.; Fitzgerald, J.; Woodcock, J.; Fritzson, P.; Brauer, J.; Kleijn, C.; Lecomte, T.; Pfeil, M.; Green, O.; Basagiannis, S.; et al. Integrated tool chain for model-based design of Cyber-Physical Systems: The INTO-CPS project. In Proceedings of the 2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data), Vienna, Austria, 11 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–6.
62. Saft, P.; Pfeil, M.; Stetter, R.; Till, M.; Rudolph, S. Integration of geometry modelling and behavior simulation based on graph-based design languages and functional mockup units. *Procedia CIRP* **2024**, *128*, 310–315. [CrossRef]
63. Elsheikh, A.; Awais, M.U.; Widl, E.; Palensky, P. Modelica-enabled rapid prototyping of cyber-physical energy systems via the functional mockup interface. In Proceedings of the 2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), Berkeley, CA, USA, 20 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 1–6.
64. Rackauckas, C.; Gwozdz, M.; Jain, A.; Ma, Y.; Martinuzzi, F.; Rajput, U.; Saba, E.; Shah, V.B.; Anantharaman, R.; Edelman, A.; et al. Composing modeling and simulation with machine learning in Julia. In Proceedings of the 2022 Annual Modeling and Simulation Conference (ANNSIM), San Diego, CA, USA, 18–20 July 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–17.
65. Falay, B.; Wilfling, S.; Alfalouji, Q.; Exenberger, J.; Schranz, T.; Legaard, C.M.; Leusbrock, I.; Schweiger, G. Coupling physical and machine learning models: Case study of a single-family house. In Proceedings of the Modelica Conferences, Linköping, Sweden, 20–24 September 2021; pp. 335–341.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.