Methods

Behrang Ashtari Talkhestani*, Tobias Jung, Benjamin Lindemann, Nada Sahlab, Nasser Jazdi, Wolfgang Schloegl and Michael Weyrich

An architecture of an Intelligent Digital Twin in a Cyber-Physical Production System

Eine Architektur für einen intelligenten Digitalen Zwilling in einem Cyber-Physischen Produktionssystem

https://doi.org/10.1515/auto-2019-0039
Received March 19, 2019; accepted August 2, 2019

Abstract: The role of a Digital Twin is increasingly discussed within the context of Cyber-Physical Production Systems. Accordingly, various architectures for the realization of Digital Twin use cases are conceptualized. There lacks, however, a clear, encompassing architecture covering necessary components of a Digital Twin to realize various use cases in an intelligent automation system.

In this contribution, the added value of a Digital Twin in an intelligent automation system is highlighted and various existing definitions and architectures of the Digital Twin are discussed. Flowingly, an architecture for a Digital Twin and an architecture for an Intelligent Digital Twin and their required components are proposed, with which use cases such as plug and produce, self-x and predictive maintenance are enabled.

In the opinion of the authors, a Digital Twin requires three main characteristics: synchronization with the real asset, active data acquisition from the real environment and the ability of simulation. In addition to all the characteristics of a Digital Twin, an Intelligent Digital Twin must also include the characteristics of Artificial Intelligence. The Intelligent Digital Twin can be used for the realization of the autonomous Cyber-Physical Production Systems.

In order to realize the proposed architecture for a Digital Twin, several methods, namely the Anchor-Point-Method, a method for heterogeneous data acquisition and data integration as well as an agent-based method for the development of a co-simulation between Digital Twins were implemented and evaluated.

Keywords: Architecture of Digital Twin, Cyber-Physical Production System, Intelligent Digital Twin, Artificial Intelligence

Zusammenfassung: Der Digitale Zwilling im Kontext von Cyber-physischen Produktionssystemen gewinnt zunehmend an Bedeutung. Dementsprechend werden verschiedene Architekturen für die Realisierung des Digitalen Zwillings entwickelt. Es fehlt jedoch eine klar definierte Architektur, die alle notwendigen Komponenten des Digitalen Zwillings einschließt, um verschiedene Anwendungsfälle in einem intelligenten Automatisierungssystem zu realisieren.

In diesem Beitrag wird der Mehrwert eines Digitalen Zwillings in einem intelligenten Automatisierungssystem hervorgehoben und verschiedene bestehende Definitionen und Architekturen des Digitalen Zwillings werden diskutiert und bewertet. Anschließend wird eine Architektur eines Digitalen Zwillings und eine Architektur für einen Intelligenten Digitalen Zwilling mit den erforderlichen Komponenten vorgeschlagen, um Anwendungsfälle wie „Plug and Produce“, „Self-x“ und „Predictive Maintenance“ zu realisieren.

Nach Ansicht der Autoren erfordert ein Digitaler Zwilling in jedem Fall eine Synchronisation mit dem realen System, eine aktive Datenerfassung aus dem realen Umfeld und die Fähigkeit zur Simulation. Zusätzlich zu diesen Merkmalen des Digitalen Zwillings benötigt ein intelligenter Digitaler Zwilling auch Eigenschaften der Künstlichen Intelligenz.

***Corresponding author: Behrang Ashtari Talkhestani,** Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70550 Stuttgart, Germany, e-mail: behrang.ashtari@ias.uni-stuttgart.de
Tobias Jung, Benjamin Lindemann, Nada Sahlab, Nasser Jazdi, Michael Weyrich, Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70550 Stuttgart, Germany, e-mails: tobias.jung@ias.uni-stuttgart.de, benjamin.Lindemann@ias.uni-stuttgart.de, nada.sahlab@ias.uni-stuttgart.de, Nasser.jazdi@ias.uni-stuttgart.de, michael.Weyrich@ias.uni-stuttgart.de
Wolfgang Schloegl, Siemens AG, Digital Factory Division, Product Lifecycle Management, Digital Engineering Automation Designer, Gleiwitzerstraße 555, 90475 Nuremberg, Germany, e-mail: schloegl.wolfgang@siemens.com

Zur Realisierung der vorgeschlagenen Architektur des Digitalen Zwillings werden verschiedene Methoden, namentlich die Anker-Punkt-Methode, ein Verfahren zur heterogenen Datenerfassung und Datenintegration sowie ein agentenbasiertes Verfahren zur Co-Simulation zwischen Digitalen Zwillingen, vorgestellt und evaluiert.

Schlagwörter: Architektur eines Digitalen Zwillings, Cyber-physische Produktionssysteme, Intelligenter Digitaler Zwilling, Künstliche Intelligenz

1 Introduction

Current industries are increasingly faced with the challenge of delivering more customized products in shorter production times. This in turn requires them to respond by having flexible production systems, which can be efficiently reconfigured as needed [1]. With shortening lifecycles and increasing market demands, a constant improvement of current systems is taking place. In the context of Cyber-Physical Systems (CPS), a Digital Twin of a production system can address the challenge of making systems easily and quickly reconfigurable by applying it for realizing and testing reconfiguration scenarios in a simulative environment. Recommissioning of the production system therefore takes less time and enables a higher system availability.

Cyber-Physical Systems are an integration of digital data and cyber methods with physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops, where physical processes affect computations and vice versa [2].

The Digital Twin is a virtual representation of a physical asset in a Cyber-Physical Production System (CPPS), capable of mirroring its static and dynamic characteristics. It contains and maps various models of a physical asset, of which some are executable, called simulation models. But not all models are executable, therefore the Digital Twin is more than just a simulation of a physical asset. Within this context, an asset can be an entity that already exists in the real world or can be a representation of a future entity that will be constructed.

In the context of Industry 4.0, a CPPS contains different CPSs, which are physical components having their own intelligence and communication capabilities. The Digital Twin (DT) of the CPPS is a composite of many individual DTs of different CPSs. These DTs communicate with each other and can exchange data and information. A Digital Twin of the production system can simulate and test various scenarios for reconfiguration of the production system in terms of interdisciplinary aspects such as reliability, energy consumption, process consistency, ergonomics, logistics and virtual commissioning, etc. Hence, with the help of the DT, interdisciplinary simulations of different DTs can be carried out throughout the lifecycle of the CPPS. The different simulation models that interact with each other, form a co-simulation of the entire system and show characteristics of the CPPS e. g., the behavior, function, etc.

As a use case, process steps required for reconfiguring an existing production system, can be described with a Digital Twin. Reconfiguration in general is described as a modification of an already existing product or production system instance to meet new requirements [3]. For this use case, the new requirements for the production system and possible changes will be investigated based on the Digital Twin, which supports the test of different reconfiguration scenarios. As an example, industrial robots as well as control units can be programed offline and for each scenario, virtual commissioning of the system will be tested. This way, the ramp up process is shortened and the production system is back in operation in less time, resulting in a higher availability as well as a higher reliability with less failure risk [4]. Other use cases of DT are i. a. predictive maintenance, optimization, plug-and-produce, self-x, code-generation, flexible production, consistency check etc. These use cases show that the Digital Twin has application benefits during the operation phase and not only during the engineering process, as the case so far [5]. To enable this, the Digital Twin can be made available throughout the entire lifecycle of the product and production system through its containment in a Cyber Layer, as shown in Fig. 1.

Figure 1 shows the Digital Replica, the Digital Twin and the Intelligent Digital Twin of a physical entity in a Cyber-Physical Production System. These are based on a literature research regarding the definition and architecture of the Digital Twin. In the next chapters this literature research will be discussed in detail.

In the Cyber Layer, each physical asset consists of its models and data. These models and data combined form a Digital Replica of the asset. If this Digital Replica is equipped with the three characteristics: **synchronization with the physical asset**, **active data acquisition** and the **ability of simulation**, the definition of a Digital Replica is transformed into a Digital Twin. Here, an Intelligent Digital Twin consists of all the characteristics of a Digital Twin as well as **artificial intelligence** to realize an autonomous system. The Intelligent Digital Twin can therefore implement machine learning algorithms on available models and data of the Digital Twin to optimize operation as well as continuously test what-if-scenarios, paving the

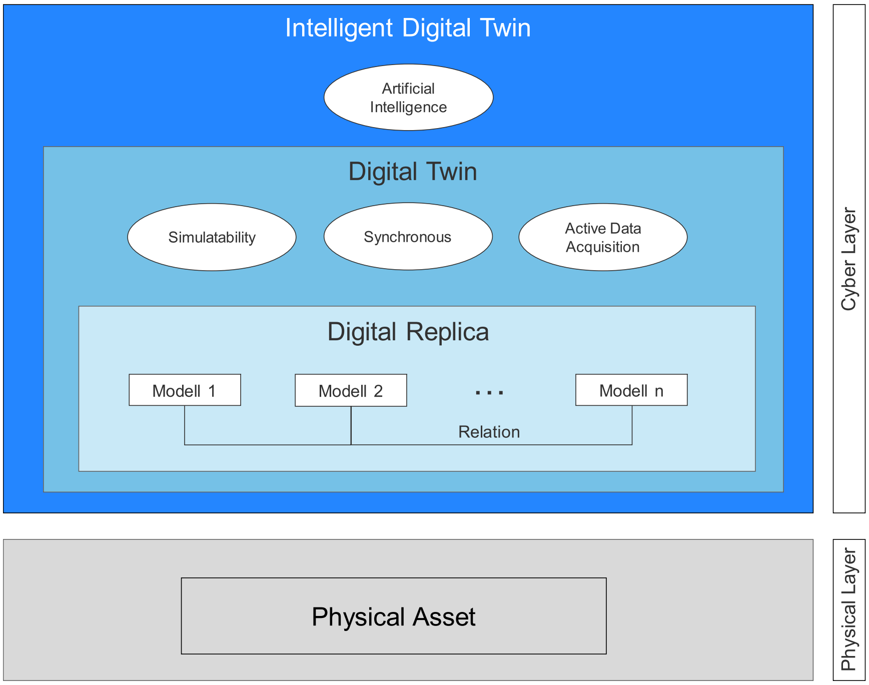


Figure 1: Digital Twin and its innovations in Cyber-Physical Systems System.

way for predictive maintenance and an overall more flexible and efficient production through plug and produce scenarios. Furthermore, having an intelligent Cyber Layer expands the Digital Twin with self-x capabilities such as self-learning or self-healing, facilitating its inner data management as well as its autonomous communication with other Digital Twins.

To show how such Cyber Layer and its' elements can be developed, first the Digital Twin architecture will be introduced.

1.1 Digital Twin

The concept of using "Twins" originates from NASA's Apollo program (National Aeronautics and Space Administration), where at least two identical spacecrafts were built to reflect the conditions of the spacecraft during a mission in outer space. The space vehicle on the ground is referred to as "Twin". The Twin was extensively used for training during flight preparation. During the mission, the ground-based model was used to simulate alternatives, where the available flight data was used to reflect the flight conditions as accurately as possible and to assist astronauts in orbit in critical situations [6, 7]. The term "Digital Twin" was first introduced into NASA's integrated technology roadmap. According to [6], a Digital Twin is an integrated simulation model of a vehicle or system that uses the available physical models, sensor updates, fleet history, etc. to mirror the life of the corresponding flying Twin. In [8] the Digital Twin was introduced as a continuously updated, digital model of a product or production system that contains a comprehensive physical and functional description of a component or system throughout the life cycle.

In [9] several definitions of Digital Twins are compared and evaluated. The conclusion is that a multitude of definitions of the Digital Twin exist. But in most of the reviewed definitions, a Digital Twin is an as realistic as possible, equivalent digital representation of a physical asset, which is always in synch with it. Additionally, it is possible to run a simulation on the digital representation to analyze the behavior of the physical asset. The authors of [9] give their own definition of the Digital Twin, which is based on the commonality of the definitions, extended by the ability of the digital representation to automatically alter and control the physical asset.

Based on a literature research, the authors present several definitions of the Digital Twin:

A Digital Twin is a digital representation of a physical asset with the following features:

- A Digital Twin has to be a digital representation of a physical asset, including as realistic as possible models and all available data on the physical asset.
- The data has to contain all process data, acquired during operation as well as all organizational and technical information created during the development of the asset.

- A Digital Twin has to be always in synch with the physical asset.
- It has to be possible, to run a simulation on the Digital Twin of the behavior of the physical asset.

Additionally, the following aspects are also required in the opinion of the authors to realize the full potential of the Digital Twin:

- a unique ID to identify the Digital Twin
- a version management system to keep track of changes made on the Digital Twin during its life cycle
- interfaces between Digital Twins for Co-Simulation and data exchange
- interfaces to the tools, in which the models are executed
- an interface to other Digital Twins for co-simulation.

Those aspects are described in more detail later on.

Based on this definition of the Digital Twin, a definition of the Intelligent Digital Twin is made: in addition to the required aspects of the Digital Twin, the Intelligent Digital Twin has to have capabilities of artificial intelligence. With those capabilities, it is possible to draw conclusions from the data and simulations of the Digital Twin and to influence, optimize or control the physical asset without the need of human intervention.

1.2 Evaluation of available Digital Twin architectures based on the above definition

So far, only a few architectures for Digital Twins have been presented in literature. Malakuti and Grüner name four architectural aspects of a Digital Twin; these are internal structure and content, APIs and usage, integration, and runtime environment [10]. The aspect of internal structure and content deals with the following aspects:

- A meta-model for the Digital Twin with its internal models and their relations.
- Possibilities to connect to legacy tools.
- Possibilities to modularize the content of the Digital Twin and to expand the content.
- Standards to be used in defining the content of the Digital Twin to enable the exchange of information.
- Possibilities to translate existing information into these standards.

The aspect of APIs and usage illuminates the interfaces of the Digital Twin and aspects related to the Digital Twins such as cloud-to-device communication or access authorization to information of the Digital Twin.

The aspect of integration requires:

- An identification mechanism for unambiguous identification of the real asset.
- Mechanisms for identifying new real assets in a system, linking them to their Digital Twin, and synchronizing the Digital Twin with the real asset.
- A mechanism for combining several Digital Twins into a Digital Twin of a system.

In addition, it is demanded under the aspect of the runtime environment that possibilities are created for storing and distributing the content of the Digital Twin, for exchanging and synchronizing the content across company boundaries, for secure access to the content and for communicating information from the Digital Twin to the real asset. However, this is only a discussion of a collection of requirements, without concrete implementation proposals being provided. How the individual aspects have to look conceptually or how they can be realized is not discussed. Reference is only made to the Platform Industry 4.0 Administration Shell, which also takes up some of these aspects, but does not discuss all of them [https://www.plattform-i40.de]. The project BaSys4.0 has implemented all these aspects, but no aspects of artificial intelligence are considered in this architecture.

Haag and Anderl define the Digital Twin as a digital representation of the properties, states and behavior of a real asset, realized through models and data [11]. However, no concrete architecture is given here and aspects of artificial intelligence are not considered either.

Boschert and Rosen, on the other hand, do not define a concrete architecture of the Digital Twin, but limit themselves to the demand for relevant models and data and leave the concrete design of the Digital Twin to the developer [12].

Another approach is the Asset Administration Shell. It focuses on a semantic understanding of the physical assets, so that the physical assets can interact with each other, but it does not consider aspects of artificial intelligence. Additionally it is not able to synchronize the Digital Twin with the real world based on physical changes [13].

Further representations of the Digital Twin are usually limited to a collection of models of the real assets and do not deal with data, communication or required aspects of artificial intelligence.

None of the, in the literature presented architectures fulfill all of the above mentioned aspects of the Intelligent Digital Twin. For this reason, an architecture based on the requirements described above is proposed. And as most of

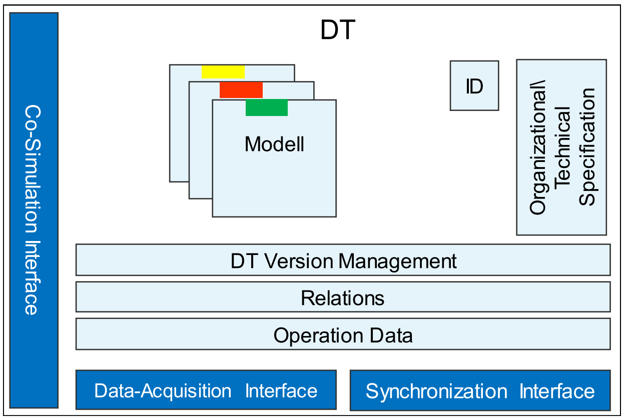

DT
Co-Simulation Interface
Modell
ID
Organizational\ Technical Specification
DT Version Management
Relations
Operation Data
Data-Acquisition Interface
Synchronization Interface

Figure 2: Digital Twin architecture in Cyber Layer.

them also only request some of those aspects, but do not give a proposal on a concept or a realization, more detailed concepts on the aspects of the Digital Twin are given and a realization and evaluation is shown.

2 Proposed architecture of a Digital Twin in the Cyber Layer

In Fig. 2, the properties of a Digital Twin in its architecture are presented. The architecture of a Digital Twin consists of a unique ID in the cyber world, models and associated interfaces to tools, models' version management, the operation data of the physical asset, organization and technical data of the asset, information about its relations to other DTs and an interface to communicate with other DTs as well as an interface to communicate with the real world. In the following each of these aspects are discussed.

ID

This property defines a unique Digital Twin with a unique ID of an asset. With the help of this unique ID, the data and models of the Digital Twin are stored as a module on a database containing all data and information and can be called any time during engineering or reconfiguration. This obviously supports modularity in the context of modular system engineering.

DT Models and corresponding interfaces to tools

A Digital Twin encapsulates the interdisciplinary models of an asset. For example, CAD models, electrical schematic models, software models, functional models as well as simulation models etc. Each of these models are created by specific tool during the engineering process of a Digital Twin. Here an important feature are the interfaces between the tools and their models. Tool interfaces are used to provide interaction between models. For example, the models can be updated or reversioned during the entire life cycle or domain-specifically simulated with the aid of different inputs.

DT Version Management

The Digital Twin of an asset should not only contain current models, but also all generated models during the entire lifecycle. This supports efficient engineering during reconfiguration and expandability throughout the lifecycle. DT Version Management accesses all stored versions of the models and their relations. This allows the old version to be called up any time at the request of an engineer, taking into account the circumstances during engineering or reconfiguration, and to switch to the current version.

Operation data

In order to accurately reflect the behavior and current state of the asset, the Digital Twin must contain current operation data of the asset. This can be sensor data, which is continuously streamed and recorded, as well as control data, which determines the current status of the real component, also recorded over the entire lifecycle. New orders and other business data can also be stored here. This is realized using the interface for data acquisition, which is a structural component within the DT. The term operation data in Fig. 2 represents a database which stores and processes this type of data. Algorithms could work with this data both online and offline to improve the understanding of the asset and to further refine the corresponding models.

Organization and technical data

This data contains information about the physical asset, such as when it was produced, by whom it was produced, designed, developed, etc., with which equipment it was produced, when and by whom it was commissioned, etc. Additionally, all documentations created during the lifecycle of the physical asset are also saved. This includes documents created during the development phase, like requirement specifications, design layouts, etc. and also documentation during operation, like maintenance reports.

Relations (to other DTs)

In a Digital Twin, the current interdisciplinary models that are related to other Digital Twin models are stored, for

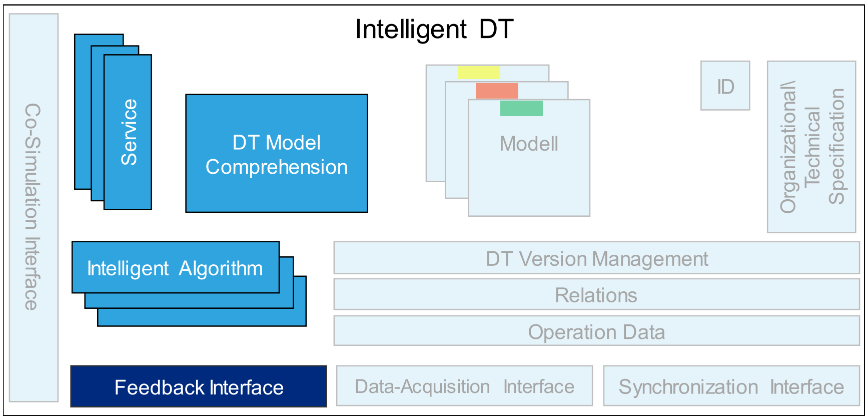


Figure 3: Intelligent Digital Twin architecture in Cyber Layer.

example: instance-instance relations, inheritance, parent-child relations and so on in order for the whole system to remain consistent, these relations must be stored within the architecture of the Digital Twin, otherwise the whole system, containing many Digital Twins, becomes inconsistent.

Co-simulation interface

An interface for communication with other Digital Twins is needed to obtain more than a pure image of reality and to allow a manual what if simulation for the system user. For example, a data exchange can enable multidisciplinary co-simulation in the Cyber Layer. This can be used to simulate the process flow of the entire production system in the real world.

Interface for the synchronization of models and relations

A physical asset and its physical relations with other assets (such as wiring, physical fixation position, etc.) can be changed very often during its lifecycle. Here an interface is necessary for synchronizing interdisciplinary models and their relations in a Digital Twin.

Interface for data-acquisition

Via this interface operation data can be transferred and recorded by the DT. In an Digital Twin, it is important that the simulation runs dynamically in parallel to the real world, i. e., the current data collected by sensors in the real world must build the actually dynamic behavior of the asset in a simulative environment.

The DT, in the form described above, does not have its own intelligence to analyze the relationship between items, to react to the environment changes and to learn from the inputs. These characteristics increase the quality and functionality of the DT. The higher level of the DT is the Intelligent Digital Twin which will be described in the next section.

3 Proposed architecture of an Intelligent Digital Twin in the Cyber Layer

As already mentioned, an Intelligent Digital Twin is one level higher than the Digital Twin in a Cyber-Physical Production System. An Intelligent Digital Twin, using the entire system’s actual Digital Twin, enables services such as optimization of the process flow, automatic control code generation for newly added machines in the context of plug and produce and predictive maintenance using stored operation data in the DT throughout the lifecycle. To realize this, additional components are required to equip the Digital Twin architecture with intelligence. As shown in Fig. 3 new components, being the DT model comprehension, intelligent DT algorithms, services and an extra interface for communicating with the physical asset must be added to the architecture of a Digital Twin to make it intelligent. These parts will be elaborated flowingly.

Digital Twin model comprehension

To dynamically synchronize the Digital Twin with the physical asset throughout the entire lifecycle, an element is needed to understand and manage all models and data. Accordingly, the DT model comprehension in the architecture fulfills this purpose by storing information of the interdisciplinary models within the DT and its relations to other DTs. The DT model comprehension realizes a standardized semantic description of models, data and services for a uniform understanding within the DT and between DTs. Technologies to implement such a standardization can be OPC UA or OWL.

Algorithms for the intelligent DT

An Intelligent Digital Twin has two major capabilities regarding the processing of acquired operation data. Firstly, it is able to apply appropriate algorithms on the data to conduct data analysis. The algorithms extract new knowledge from the data which can be used to refine the models of the Digital Twin e. g., behavior models. Hence, an Intelligent Digital Twin is capable of providing assistance to the worker at the plant to optimize the production in various concerns. Secondly, an Intelligent Digital Twin is able to incrementally improve its behavior and features and thus to steadily optimize the above-mentioned assistance to the worker. This is based on and closely tied to the first capability due to the fact that the data analytics results are directly incorporated into the system and models in particular. Hence, an Intelligent Digital Twin can provide assistance in different use cases such as process flow, energy consumption, etc.

There are two use cases for co-simulation in the Intelligent Digital Twin. Firstly, the optimized combination and process chain between DTs can be realized by parameterizing the existing models in relation to other DTs in a co-simulation environment. Based on the results of this simulative environment, the Intelligent Digital Twin can trigger a parametrization of physical assets. The other use case is the processing of Digital Twin capabilities to optimize individual products. For instance, to determine the optimal system parameters and to predict the corresponding quality of products (‘Predictive Quality’). As a consequence, the amount of degraded products can be minimized leading to an increased quality of the manufacturing process [14]. These two use cases are example scenarios for “what if simulation” in intelligent Cyber Layers.

Other algorithms deal with automated code generation, for example through service-oriented architecture approaches for real machines based on the new requirements. This allows approaches such as plug and produce and flexible production facilities to be realized. Other algorithms can provide a simulation-based diagnostic and prediction service through data analysis and knowledge acquisition, with an emphasis on predictive maintenance. The following examples of intelligent algorithms can be noted: „algorithm to product failure analysis and prediction“, “algorithm to optimization and update of process flow”, “algorithms for generating a new control program for the system based on new requests”, “algorithm for energy consumption analysis and forecast” and “Service of user operation guide” etc.

To present a detailed example regarding the applied algorithms within a use case, the investigation of historical process data of production plants to predict future maintenance intervals or to maximize the availability of the plant (‘Predictive Maintenance’) is described in the following. To extract a model from or find correlations within operation data, unsupervised learning techniques such as k-means clustering or auto encoder networks with LSTM cells can be applied on time series data. In case of k-means clustering with sliding windows, the learned time-sensitive cluster structure is used as model for the system behavior. This circumstance allows for instance the detection of anomalies and the prediction of failures. To do so, a distance metric that considers the current point in time is applied on a test data set of currently acquired data and the cluster centers of the trained model. Anomalies in the test data set are detected by defined time-dependent limit violations to the cluster centers as well as the emergence of new, previously non-existent clusters. Thus, the slinking emergence of failure can be predicted based on the frequency of anomaly occurrences and their intensity of deviation [15].

Services

An Intelligent Digital Twin of an asset should include all the functionalities that the asset can perform in the real world on various tasks. These functionalities are stored in the Intelligent Digital Twin as different services. An example for such a service is a robot in the production line, which can cut, drill, glue etc. These functionalities can then be used by other Intelligent Digital Twins for various purposes. For example, the services of an Intelligent Digital Twin can be used within a what-if simulation in communication with other Digital Twins to make the best decision for reconfigurating a system based on new requirements and available resources. Or the Intelligent Digital Twin of a product can manage the production of the real asset by searching and finding suitable services of production systems. However, an important point with the Intelligent Digital Twin is that these services can be continuously expanded, as the real asset changes.

Here a challenge is the semantic description of those services, as a standardization is needed, so all Intelligent Digital Twins can understand and use the services of other Intelligent Digital Twins. One possibility to realize such a standardization is the use of the Industry 4.0 Asset Administration Shell [16].

Feedback interface

In an Intelligent Digital Twin, an interface is needed that enables it to automatically and dynamically access the real asset. This “feedback interface” must enable the transfer of data to the physical assets that can be understood from real assets using semantic technologies. Thus, this interface can be represented as a communication channel or return channel between the DT and the asset for dynamic parameterization and control of the physical asset. With the aid of this channel, autonomy in the Cyber-Physical Production system can be enabled and the Intelligent Digital Twin can control the real asset.

4 Usage and challenges of the Digital Twin of a Cyber-Physical Production System

To acquire the benefits of the Digital Twin, four aspects are ought to be considered; one aspect is the development of the Digital Twin, which can be model-based to contain models of different domains along with simulation models and data from other disciplines generated during the engineering process. For example a Digital Twin consists of interdisciplinary and interrelated models (CAD, structure, behavior, function model, etc.) in the domains mechanic, electric and software. These models are created during the engineering phase of the CPPS by different tools addressing different aspects. A challenge hereby is achieving efficient interdisciplinary model integration to maintain consistency between models [17] (**Integration of Interdisciplinary Models**).

The second aspect for the employment of the Digital Twin is its synchronization throughout the lifecycle of a Cyber-Physical Production System (**Synchronization of Reality and Virtuality**). Since simulations cannot reflect the environment entirely, many changes and adjustments have to be made during commissioning and operation of a production system, therefore, for the employment of the Digital Twin its synchronization with the physical system as of commissioning is essential and needs to be automatic and systematic.

The third aspect is the interaction between Digital Twins, both for the purpose of co-simulation and operation data exchange. The operation data exchange is only needed to realize the Intelligent Digital Twin and only co-simulation is needed as an interaction to realize the Digital Twin. Operation data exchange can be used for machine learning applications, where one Intelligent Digital Twin optimizes its learning results by using information provided by its adjacent Twins. In case of co-simulation, both a co-simulation to reproduce the actual system in the Cyber Layer, to analyze the system and to execute what-if co-simulations to simulate various scenarios are possible use cases. But the conclusions of the co-simulation are not drawn automatically, the co-simulation only assists the user (**Interaction between Digital Twins**).

The fourth aspect is an active data acquisition. The real-time as well as the historical operation data must always be made available to the Digital Twin. This requires a mechanism for integrating and processing the real operational data within the Digital Twin. Active data acquisition means that the operating data must also be made available for simulation models of the Digital Twin without time delay so that the reality can be represented in the Cyber Layer by the Digital Twin (**Active data acquisition**).

In order to meet these challenges, firstly the state of the art for the integration of interdisciplinary models in a Digital Twin based on the consistency between tools and their models will be examined. Afterwards, the Anchor Point Method for synchronizing interdisciplinary models of the Digital Twins based on an implemented and evaluated assistance system will be presented. Additionally, a concept to integrate operation data from heterogeneous data sources will be introduced to extend the capabilities and enable the Digital Twin to conduct data analysis. Finally, for finalizing the Digital Twin in the Cyber Layer in CPPS, an agent-based, dynamic co-simulation between Digital Twins will be presented and evaluated.

4.1 Existing mechanisms for the integration of interdisciplinary models

To highlight the importance of the integration of interdisciplinary models for creating a Digital Twin, a literature review has been conducted, yielding two distinguishable basic model-exchange mechanisms to integrate the interdisciplinary models in a Digital Twin: a central system topology information of the system and decentralized Digital Twins of each mechatronic component in the system.

Central system topology information

This mechanism for achieving cross-disciplinary model exchange is using system topology information of the product. In this case a production system and a created product in this system can both be seen as a product. This mechanism can be achieved by specifying a neutral data format for exchanging domain-specific models between vendor-different tools in a standard. An example of this approach is realized by Automation Markup Language (AutomationML). AutomationML is a data format based on the eXtensible Markup Language, XML. In AutomationML, the asset models are mapped with CAEX as top level format, which consists of relations, interfaces and references between all assets in the system topology. The geometry and kinematics mapped via COLLADA and behavior with the help of PLCopen XML [18].

Furthermore, the operational data of the physical asset must be integrated into the central system topology information.

For this purpose, for instance the operational data of an asset can be connected by OPC-UA to the CAEX architecture of the AutomationML [19]. As an XML-based data exchange format, AutomationML joins the group of open formats such as PLMXML and STEP AP242 [20].

Another example of this mechanism is provided by product lifecycle management platforms, PLM platforms, where model data from interdisciplinary domains can be referenced and exchanged in a structured way. PLM platforms work by integrating interdisciplinary models into a single IT system. It is based on structured data exchange in different domains, which makes it possible to integrate and manage heterogeneous models into a single IT system. Such single IT systems have to be able to integrate numerous tools and their created models in the context of engineering, management and service through their extensive functionalities, workflows and interfaces.

The models stored on the PLM-Platform in interdisciplinary system topology information, called bill of product, equipment and process, which are referenced to each other. This technique realizes a centralized Digital Twin of the entire system, consisting of multiple sub-Digital Twins and their relations. On PLM-platforms data of different domain database elements is exchanged by managing heterogeneous database formats and elements with a meta-database on a single IT-system. This PLM-Platform also combines interfaces to multi-disciplinary engineering tools. Two library elements of different domain databases with different structures belonging to the same mechatronic component can then be joined together [8].

Modularization

In the Modularization mechanism a module encapsulates all data and models related to a component in interdisciplinary domains. The core goal of encapsulation is that an encapsulated module is thus a system with less complexity than the entire system [21]. In this approach for exchanging models, a semantic network can be created using ontologies to model intra-model as well as model-to-model relations. Another related approach is the object-oriented one, where objects of any domain model can be modeled as well as their associations to objects within and outside of a specific domain by encapsulating their data in the form of object attributes [22]. For this purpose, the ontology languages RDF and OWL can be used to semantically describe models and their dependencies as well as to formulate rules and conclusions [23].

4.2 Evaluation of the mechanisms

In this chapter, two mechanisms for model-data exchange within the Digital Twins of a system have been presented. These mechanisms are applicable considering the complexity of the system as well as its design requirements. For the development of the Digital Twin it is necessary that model-data exchange is referenced in a way that is coherent and comprehensible for external systems to use.

For the realization of the Digital Twin architecture and its three main characteristics, synchronization, active data acquisition and co-simulation ability, it is necessary that the modeled data and their relationships can be easily adapted. For the scope of this research the modeling of Digital Twin on the basis of the central systems topology information approach is chosen. Thereby the Anchor Point Method will be proposed to synchronize multidisciplinary models of Digital Twin, including interrelations and static data. To synchronize the operational data of the real asset with the Digital Twin, a cloud-based approach to data acquisition and integration using semantic technologies will be proposed. Finally, an agent-based co-simulation approach for the simulation of heterogeneous models of the Digital Twins in the Cyber Layer will be presented. These concepts as well as the realization of them will further be elaborated in the following chapters.

5 Synchronization of a Digital Twin during the entire lifecycle of a production system

As mentioned, during the operation phase within the lifecycle of a CPPS, any occurring changes in the physical system should be fed into the Digital Twin, so that it is always synchronized to the current state of the CPPS. For synchronizing models, this means that model data as well as their relations to other data should be synchronized. This requires manual effort as well as exact knowledge of the changes and how they should be adjusted within the digital models. Therefore, a methodology is needed that meets the following requirements:

- Automated detection of change scenarios in the real world and dealing with missing information.
- Automated detection of interdisciplinary dependencies and consistency checking of changes in mechanics, electrics and software domain.
- Automated adaptation and change management in the models of the Digital Twin.

For this purpose, various approaches will be discussed based on the presented requirements in the context of synchronizing the Digital Twin.

5.1 Existing approaches to synchronizing Digital Twin's models

Regarding (partially) automated change detection in an existing production system, there are two main approaches:

- process monitoring to identify changes in a plant to synchronize behavior and geometry models with the real plant [24]. The process monitoring method is based on passively capturing data from the communication network of the industrial communication system, then filtering and categorizing this data, transferring process values into a simulation and synchronizing the simulation and the process [24]. In this approach, the signals associated with a system component in a communication network contain the components identifiers, such as MAC, IP address or device names.
- multi-view 3D object recognition from a point cloud and change detection [8]: In this method, two different cloud point data are captured by scanning procedures from different times of the physical production system. To detect changes, the captured data is automatically compared with each other and the difference is detected.

The following approaches are available for (partially) automated cross-domain relational analysis between models in mechanical, electrical and software domains:

- Static control software relationship analysis is a method for detecting modules and their dependencies in software domain. Application areas for static analysis span from model checking [25] to detecting modularities [26] and effort estimation for software evolution to modernize existing machines and plants [27]. In order to detect interdisciplinary relations of changes made in the automation domain, a study by [28] has been conducted.
- There exist several methods and frameworks for analyzing cross-domain, inter-model relations of a system during engineering. One of which is the combined application of SysML as a modeling approach and semantic technologies to analyze changes in a system [29, 30]. Another method is a knowledge-based framework with an integrated rule-relationship diagnosing consistency failure and dependencies in mechatronic models during the engineering process of the system [31]. A third method is ontological-based for consistency check, by which a consistency failure in multidisciplinary engineering can be detected [32].

In addition, the following approaches can be mentioned for (partially) automated cross-domain change management for the adaptation or reversion of old partial models in the Digital Twin in order to ensure that the entire Digital Twin remains consistent:

- The integration of a change management system for multidisciplinary engineering throughout the life cycle of the system based on an AutomationML server [33], or application of product lifecycle management systems with a service-oriented architecture for coupling various tools [34, 35]. Both methods facilitate the search and editing of Digital Twin submodels.

Finally, there are some approaches that deal with the automated creation of the Digital Twin. These approaches assume that no Digital Twin is available during the engineering process. They build their Digital Twins directly based on existing systems:

- generation of a CAD-Model in a mechanical domain, current geometry model, by applying 3D laser scan based on point cloud technology and 3D-tachymeter [36], or matching and mapping between the existing structural model of the mechanical equipment, called Bill of Equipment (BOE), and active devices, which can be read from the system network [37].

It can be concluded from the approaches mentioned in the study that there is no approach that enables automated, cross-domain synchronization of the models of the Digital Twin in the domains of mechanics, electrics and software with a real automated production system.

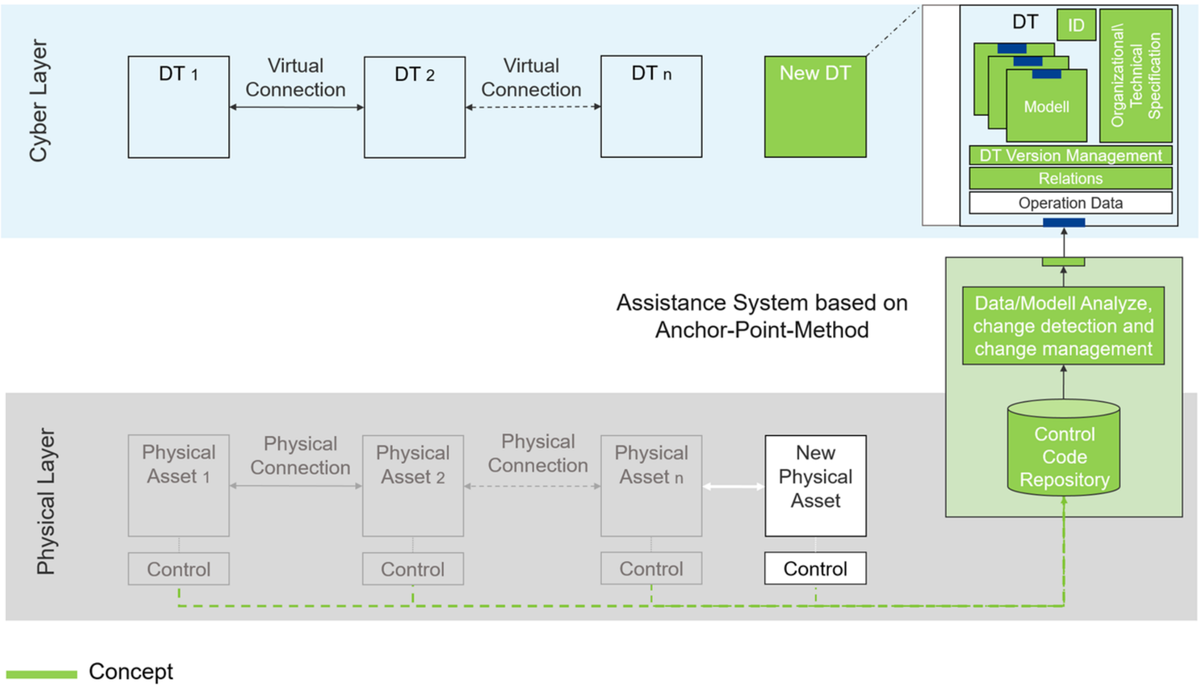


Figure 4: Anchor-Point-Method for synchronizing interdisciplinary models and associated relations of Digital Twin.

5.2 Anchor-Point-Method for synchronizing the models of Digital Twin

To synchronize cross-domain models of the Digital Twin, the Anchor Point Method can be of help when detecting occurring changes in the physical production system as well as analyzing the relations of the changed components within it. In this method, the current programmable logic controller code of the PLC-controlled system is considered as the data source of the current state of the system.

This method enables a subsequent adaptation of the changes in the interdisciplinary system topologies and control models also partial adaptation of ECAD- and CAD-models within Digital Twin, so that the Digital Twin of the system remains consistent. The Anchor Point Method has been presented in previous works of the authors and can be referred to in [5, 38, 17].

Anchor points are static data within a component's domain model in the system topology information. This can be for example geometry data of a sensor or the electrical circuit model, control function model and the signals of this sensor within the electrical domain and software domain. These anchor points have relations within the internal domain model, such as the cable connection of the sensor in the electrical domain, within other domains, this sensor has geometry data in the mechanical and a function block in the software domain. For the synchronization of domain models, these anchor points must be identified as well as their relations to anchor points in other models.

For the scope of this paper, an application specific assistance system based on the Anchor Point Method is realized. At the time of the commissioning of a manufacturing cell, mechanical, electrical as well as the software models in the Digital Twin are synchronized to the physical asset. The software models are denoted by the control program written on the PLC to achieve a desired process such as welding or drilling. These processes are achieved via mechatronic components installed in the system and communicating with the controller via signals defined in the control program. The current control program as well as the defined signals can serve as a reflection to the currently used mechatronic components and their role in achieving the desired industrial process. Considering different versions of the control program, one at the time of commissioning and the other version being a more recent one can give insight as to whether any physical reconfiguration has taken place on the manufacturing cell. Knowing whether any changes to the physical components have taken place is an important step for adjusting the digital models, which highlights the importance of the Anchor Point Method in detecting changes and synchronizing models. As shown in Fig. 4, the assistance system has the goal of detecting occurring changes and analyzing their relations, therefore assisting responsible en-

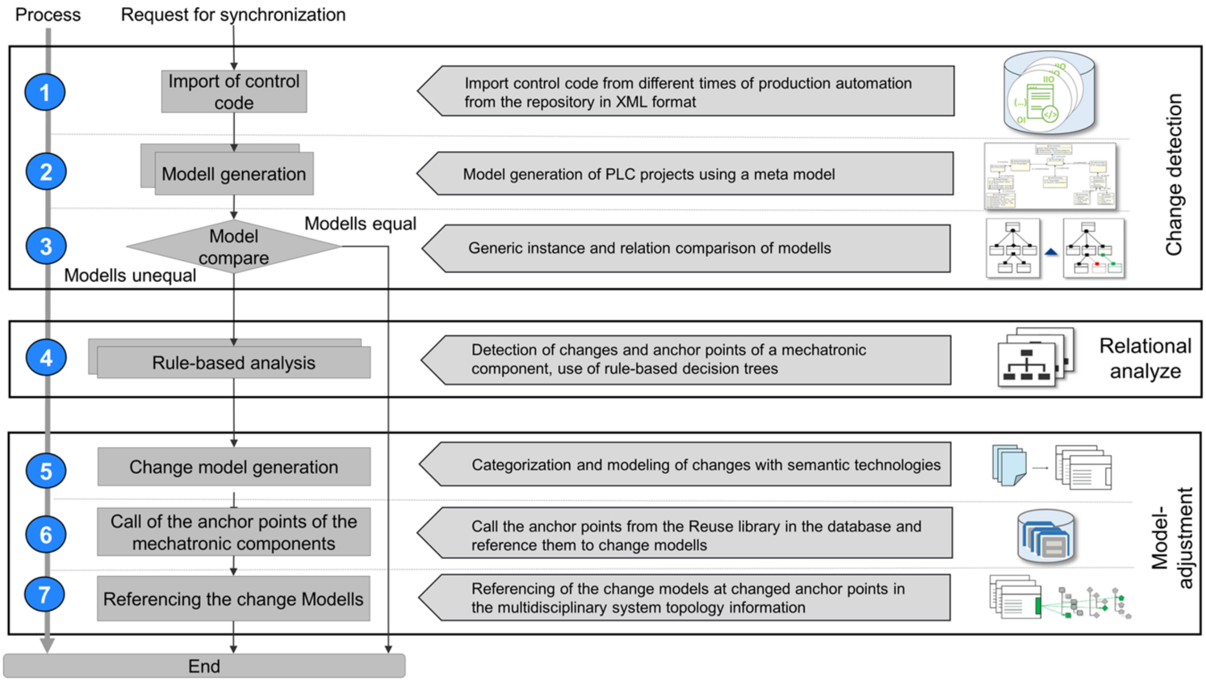

Process
Request for synchronization
1
Import of control code
Import control code from different times of production automation from the repository in XML format
2
Modell generation
Model generation of PLC projects using a meta model
3
Model compare
Modells equal
Modells unequal
Generic instance and relation comparison of modells
Change detection
4
Rule-based analysis
Detection of changes and anchor points of a mechatronic component, use of rule-based decision trees
Relational analyze
5
Change model generation
Categorization and modeling of changes with semantic technologies
6
Call of the anchor points of the mechatronic components
Call the anchor points from the Reuse library in the database and reference them to change modells
7
Referencing the change Modells
Referencing of the change models at changed anchor points in the multidisciplinary system topology information
Model-adjustment
End

Figure 5: Overview of the process steps of the Anchor Point Method.

gineers in knowing the data needing adjustments within the digital models. In addition, the assistance system will directly address anchor points in interdisciplinary topologies of the central Digital Twin through SOA libraries of the PLM System and reference or adapt changes that have occurred in the Digital Twin.

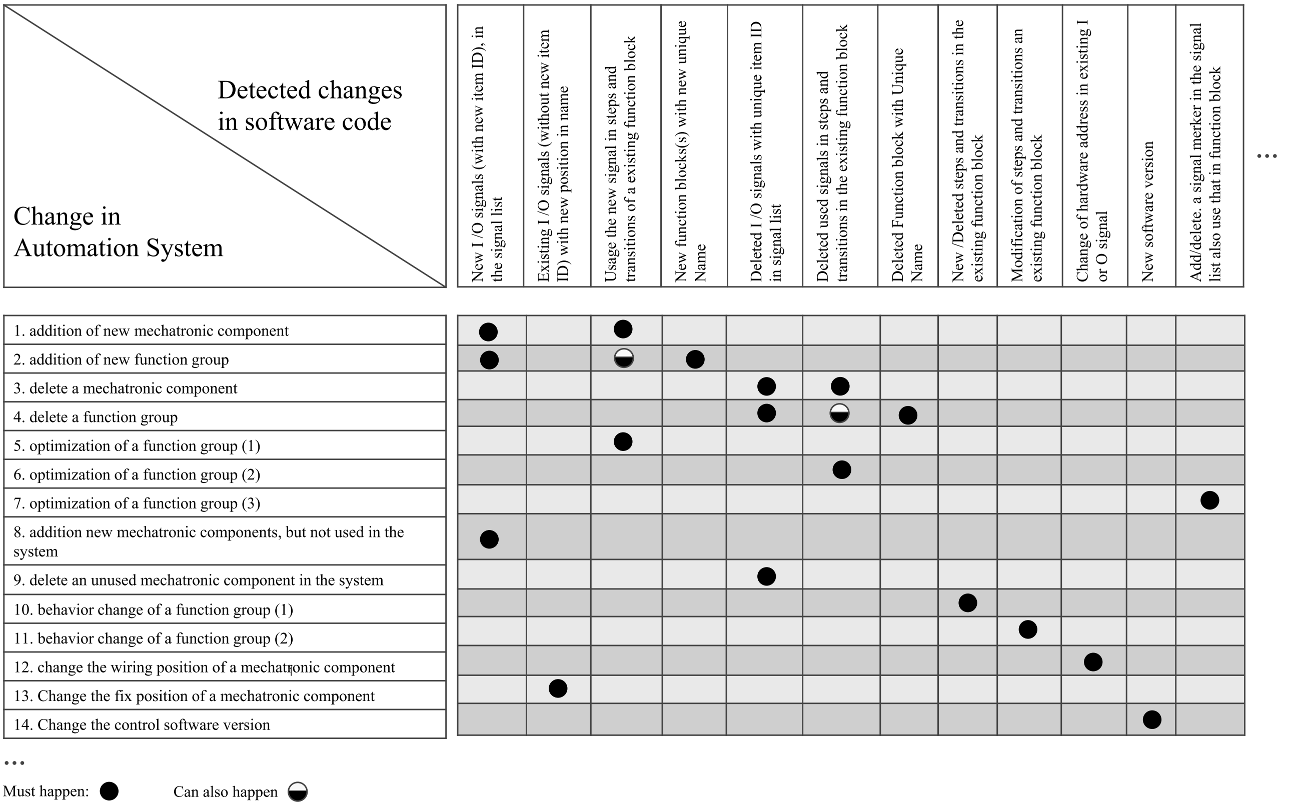
5.3 Overview of the Anchor-Point-Method

To synchronize the Digital Twin, the Anchor-Point-Method consists of three main steps, automated change detection, relationship analysis and change management. In Fig. 5 a flow chart of the process steps of the Anchor Point Method is shown. In the first stage, 2 versions of the control code of the system are imported from a repository, where versions are stored during different points in time that is one after system commissioning has taken place and the second point is the current system control code. The storage of the versions and the trigger for the request of change detection is out of scope for this paper. The second and third stage of the flow chart describes the formalization of the control code based on a central metamodel, enabling instance-instance model comparison of the control codes.

If any changes within the compared models are detected, the fourth stage is entered, which triggers a further detection of change scenarios and their interdisciplinary dependencies within the entire system. Change scenarios have been defined in a rule table, which can be seen in Fig. 6. This table contains a set of change scenarios, which may occur to the physical manufacturing cell and how they can be detected within the control code. The scenarios are grouped together indicating the change type and its trace in the control code. The scenarios include the adding or removal of a mechatronic component, the adding or removal of a function group consisting of mechatronic components, the rewiring or repositioning of a mechatronic component or changes in the logic of the control program. After change and dependencies detection in the physical system, the corresponding models of the Digital Twin should be adjusted. The fifth stage is the automated generation of so called change request model based on engineering change management approaches using semantic technologies.

Here the term semantic technologies are used to emphasize that the change request models that are created in anchor point method must be comprehensible for system topology information and multidisciplinary tools that use these models. Therefore, the change request model must be built with the same structure as System Topology Information.

The sixth stage is the conclusion of cross-domain changes to change request models. Here, based on the anchor points of the changed components in the software domain and the detected change scenario from the rule table, further related anchor points from the electrical and me-



Detected changes in software code / Change in Automation System	New I /O signals (with new item ID), in the signal list	Existing I /O signals (without new item ID) with new position in name	Usage the new signal in steps and transitions of a existing function block	New function blocks(s) with new unique Name	Deleted I /O signals with unique item ID in signal list	Deleted used signals in steps and transitions in the existing function block	Deleted Function block with Unique Name	New /Deleted steps and transitions in the existing function block	Modification of steps and transitions an existing function block	Change of hardware address in existing I or O signal	New software version	Add/delete. a signal merker in the signal list also use that in function block	…
1. addition of new mechatronic component	●		●										
2. addition of new function group	●		◒	●									
3. delete a mechatronic component					●	●							
4. delete a function group					●	◒	●						
5. optimization of a function group (1)			●										
6. optimization of a function group (2)						●							
7. optimization of a function group (3)												●	
8. addition new mechatronic components, but not used in the system	●												
9. delete an unused mechatronic component in the system					●								
10. behavior change of a function group (1)								●					
11. behavior change of a function group (2)									●				
12. change the wiring position of a mechatronic component										●			
13. Change the fix position of a mechatronic component		●											
14. Change the control software version											●		
…													

Must happen: ● Can also happen ◒

Figure 6: Rules table for rule-based change detection in automation systems.

chanical models as well as function block models can be accessed within a library. Therefore, a reuse library is necessary, which stores all the components' interdisciplinary models (3D-CAD, circuit diagrams, signals and function blocks etc.) applied in the system under a unique ID.

Once the respective components are retrieved from the library, anchor points and their relations can be allocated and encapsulated in the change request models. Finally, these encapsulated change models must be referenced to the changed anchor points in the system topology information and made known to engineers responsible for editing these changes using domain-specific tools.

For an efficient model change request, which takes place in steps 5 to 7, a service-oriented architecture, SOA can be applied. Within a SOA, all the mentioned steps can be carried out automatically.

5.4 Realization based on an assistance system

For the realization of the described steps of the Anchor-Point Method, an assistance system has been developed. For the formalization of the control codes, represented in a neutral data format being XML, Eclipse Modeling Framework, EMF has been chosen to implement a metamodel. Applying EMF, an XML schema describing exported XML directories of control codes is transformed into an object oriented metamodel representing nodes contained in the XML data as classes. The two control codes imported to the assistance system are then considered as two model instance of the generated meta-model with their accessible objects. This approach enables an efficient implementation of the rule table to conclude the occurring change scenarios and facilitates analyzing relations within the software domain. The assistance system stores knowledge in a rule-based manner. The implementation of the rule table is in the form of if-else decision trees using the Eclipse IDE. Here, it must be emphasized that the change detection based on rule-based decision trees is only possible if a naming convention for signal and function blocks is considered in the control code during the entire life cycle of the production system as well as during changes at different points in time. To implement an interface for creating semantic change request models to the system topologies and tool's models, the assistance system uses the functional interface of the PLM-Platform for creating the change request model. For this purpose, various Java

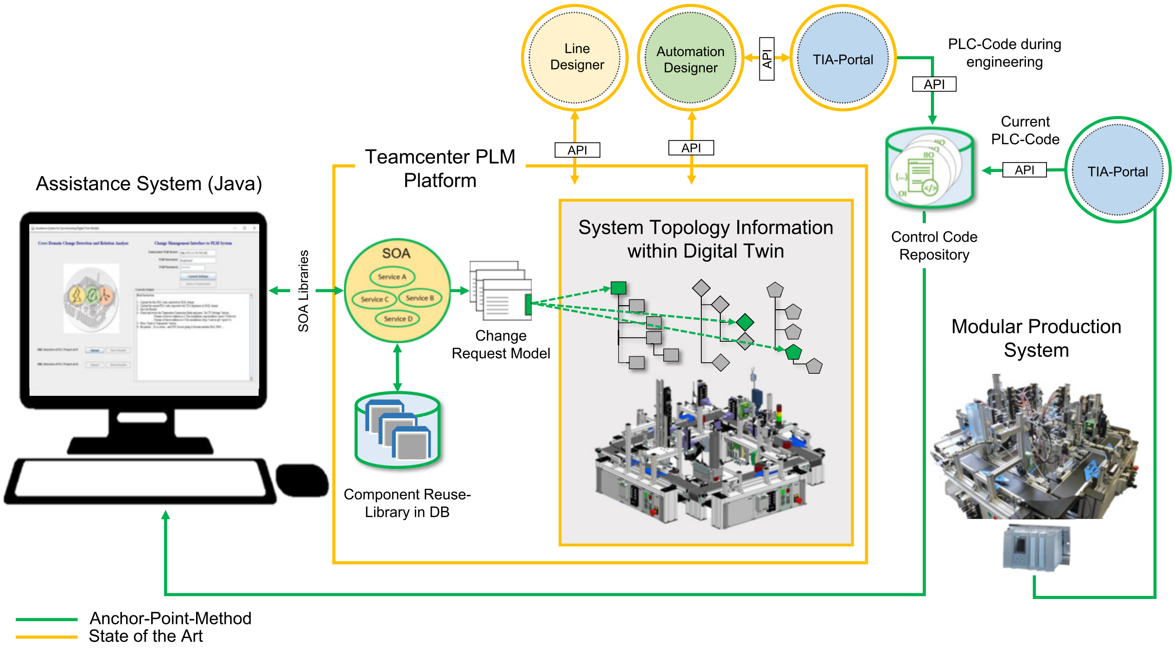


Figure 7: System structure for synchronizing Digital Twin models of a modular production system based on the Anchor Point Method.

classes were implemented which use the SOA libraries of the PLM platform.

5.5 Evaluation of the assistance system based on a modular production system as an experimental prototype

As shown in Fig. 7, to evaluate the Anchor Point Method in a first step, a Digital Twin of an experimental prototype was to be created. For this purpose, a central Digital Twin of a modular production system (MPS) based on system topology information in a PLM-Platform was created. The target MPS contains about 120 mechatronic components and is controlled via a Siemens PLC S7-1500, which uses TIA Portal V14 for the development of the control program. NX-Line Designer, NX-Automation Designer and TIA Portal were used for the multidisciplinary modeling of the MPS Digital Twin in the mechanical and software domain. For the creation of the Digital Twin, these models were merged in a central system topology information on the Siemens Teamcenter PLM platform. In addition, a component reuse library within the Teamcenter PLM platform was created for automated referencing of anchor points of changes in the Digital Twin. In this component reuse library, which is integrated on the Teamcenter PLM Platform, all interdisciplinary component models are stored based on the component's unique item ID, which is applied to encapsulate.

Now to use the assistance system for automated change detection, relationship analysis and change management in system topology information, some physical changes had to be made in the MPS. Here, a change scenario with 32 changes in the real system were carried out and accordingly, new PLC code for the system was written. The PLC projects before the change and after the change were written in S7-Graph language, which a graphical language for programming industrial sequences. These were converted to XML using the TIA-Openness Tool and stored in a control code repository. TIA Openness is an API on the TIA portal, which enables an export of a current control program in the form of neutral XML files. Therefore, the inputs of the assistance system are two XML file directories reflecting the control program of a manufacturing cell at two different points in time and the output is a conclusion of detected changes and their multidisciplinary relations. Finally, based on Teamcenter's SOA libraries, the assistance system will automatically log on to the PLM platform, generate semantic change models with associated anchor points and reference these to affected anchor points in the system topology information. Finally, as evaluation results, all 32 changes made to the system were successfully detected by the assistance system and referenced in the Digital Twin.

As already discussed in detail in previous chapters, with regard to the active data acquisition characteristic in the Digital Twin, the real-time and historical operation data of the real asset must always be made available in the Digital Twin. This requires a concept that makes it possible to systematically collect, analyze and model all operation data until it is ready for use in Digital Twin.

The following chapter presents a cloud-based approach for data acquisition and integration using semantic technologies to synchronize the operation data of the real asset with the Digital Twin.

6 Integration and processing of the real operation data within the Digital Twin

An Intelligent Digital Twin has to be capable of acquiring operation data from a physical asset as well as processing and analyzing this data. This is the basis for the knowledge extraction to transform the Digital Twin into an intelligent one and to provide various assistance functions such as diagnosis and predictive quality. The foundation for this is the acquisition, preprocessing and semantical annotation of operation data. To enrich the Digital Twin of a Cyber-Physical Production System with operation data, the ability to sample process data in a highly accurate manner and to assign it to the corresponding Digital Replicate is a major challenge in the industrial implementation. A problem that has to be tackled in this context is the heterogeneity of data sources concerning data formats, protocols and sampling rates. Hence, a software module is developed that uniformly transfers the collected data to the Digital Twin. This module consists of an individual interface for each data source in the field level and a standardized interface to further process the recorded data within the Digital Twin. The individual interfaces of the connector run on control devices and the standardized interface runs in the cloud [39].

The individual interface basically consists of two major functionalities, namely a filter and an interpreter. The filter scans the protocol structure and extracts data depending on the bus system. It processes protocol specific messages. The interpreter is responsible for the interpretation of the messages and for the transformation into a certain format. The concept to handle operation data and to integrate it to the Digital Twin includes, alongside the homogenization process, a semantic description of the captured data as well as the correct assignment to the corresponding physical components. Consequently, the output of the interpreter is mapped onto a defined information model, e. g., the OPC UA information model. This allows a semantic description of the captured operation data. The measured data is mapped onto the model and either cyclically posted to the standardized interface that runs in the cloud or available as a service within the Digital Twin. If it is available as service, semantically annotated operation data can be requested on demand.

To ensure a consistent assignment of operation data to a physical production system, OPC UA servers can be implemented on the control devices in the field level. The servers implement an address space that describes the topology of the machine and the corresponding condition of the machine components. The address space delivers the context in which the operation data has been generated and is the meta model of the machine. Hence, the data is annotated with the necessary meta data. The servers offer a surveillance service corresponding to the alarms and conditions specification. A continuous surveillance of changes in the address space is realized using subscriptions. Thus, all condition changes of the components and machine status being relevant as context information for the operation data can be detected and annotated as meta data. To ensure the same for physical products, a multidimensional modeling approach is applied.

The approach is based on data cubes as described in [40]. The Jedox Suite is used as tool to realize the online modeling of operation data. In case of a new product being produced on a certain production route, the data integration stack maps the new information on the hierarchical dimension structures. As a result of the mapping process, new elements are generated within the hierarchy for the identified changes, namely the product identifier. The corresponding operation data is assigned to the new subspace of the cube respectively. Hence, the data cube extends itself and adapts changes automatically based on the extracted meta data. The whole integration process is cyclically executed and GPU supported to enable parallel computing. The measured data can be stored in a relational database where long-term data analytics techniques and intelligent algorithms can be applied or forwarded to a real-time database to conduct ad-hoc calculations and online analytical processing. The general concept is depict in Fig. 8.

Evaluation of the data unification concept for the flexible handling of operation data

The presented approach for the acquisition of operation data is applied in the field of metal forming [41]. A model process chain is utilized to evaluate the functionality. The

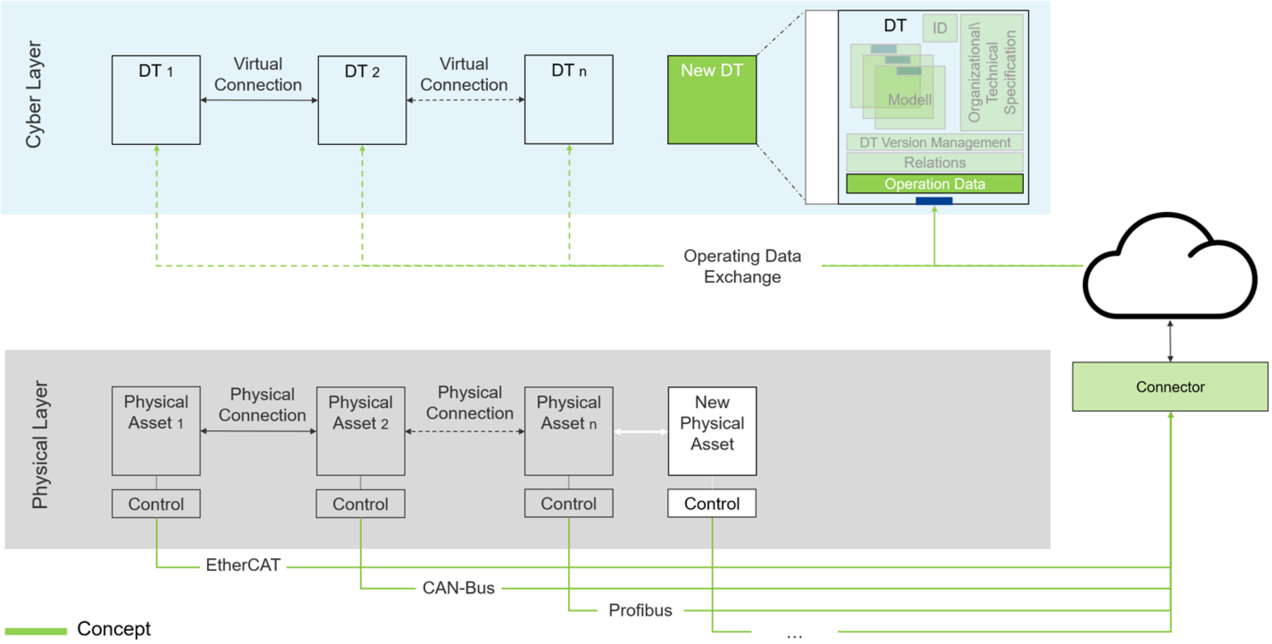


Figure 8: Data unification concept for heterogeneous data sources and processing of operation data within the Digital Twin.

core of the model process is two-step hot forging conducted using a hydraulic press. The production site also includes a heating treatment unit, spray systems for lubrication solvent, a transport robot and quality measurement units. The presented system is developed and tested on the basis of the real bus communication and PLCs of this process chain. The bus communication within the process chain includes the bus systems of EtherCAT, CAN-Bus and Profibus. The press is controlled through a soft PLC, namely TwinCAT 3. The described concept for data integration and processing allows the flexible handling of operation data and the corresponding meta data for all physical assets. Changes within the process chain trigger subscriptions so that the meta data is updated permanently. Thus, changes in the topology of each machine or status changes of components such as sensors can be detected and incorporated in the semantic description of the operation data. This information is available within the Digital Twin and can be requested in real-time.

After the presentation of the two concepts for synchronizing the models and operation data of the Digital Twin in the Cyber Layer in Chapters 5 and 6, the next important aspect for the realization of the architecture of the Digital Twin is the enabling of an interaction between Digital Twins in the Cyber Layer. It is therefore necessary to present a concept that enables interaction between Digital Twins and their various simulation models within the context of co-simulation.

In order to realize this interaction between Digital Twins, in the next chapter an agent-based co-simulation method for the simulation of heterogeneous models of the Digital Twins in the Cyber Layer will be presented and realized.

7 Enabling co-simulation between Digital Twins

To completely realize the Digital Twin the interaction, namely a co-simulation between Digital Twins has to be realized. Hence, a framework for such a co-simulation has to be provided, which includes the interfaces to the simulation tools, needed to execute the simulations of the individual Digital Twins and a platform for the data exchange, needed for the co-simulation, between the Digital Twins. This platform has to provide both, a possibility to exchange the actual data between the individual simulations and data, needed for the synchronization of the individual simulations, as without a synchronization, causality errors can occur in a co-simulation [42].

The actual data, exchanged by the individual simulations, simulates the real interactions between the assets, represented by the Digital Twins. Those interactions can be assigned either to the communication between the assets over communication technologies like Wi-Fi or Bluetooth or to a physical interaction, like when one asset physically alters another asset. Therefore to realistically simulate the actual system, both the communication technology and the environment for the physical interactions have to be simulated.

To obtain all benefits of the co-simulation, the co-simulation has to be dynamic, meaning it has to be possible to add new simulations during runtime [43]. It also has to be possible to include different simulation tools in the co-simulation, as not all Digital Twins will have the same kind of models [43].

7.1 Co-simulation approaches

[44] researches several approaches for co-simulation, with regard to those challenges. Here, established co-simulation approaches like Functional Mock-up Interface (FMI) and High Level Architecture (HLA) are researched, as well as scientific approaches based on OSGi (Open Services Gateway initiative), OPC UA (OPC Unified Architecture) and software agents. Additionally domain-specific approaches, like CAPE-OPEN (Computer-Aided Process Engineering) were researched, but those are not suitable for the purposes of the Digital Twin, as only domain specific tools can be used and therefore only the Digital Twins of certain domains would be useable for co-simulation.

Co-simulation approach	Dynamic adding of simulations	Possibility of adding different simulation tools
FMI	not met	not met
HLA	partially met	not met
OPC UA	fully met	partially met
OSGi	fully met	partially met
Agents	fully met	partially met

As can be seen, both FMI and HLA cannot be used for a framework for a dynamic co-simulation, as in both cases only a limited number of simulation tools can be used, as both have high requirements on simulation tools and in case of FMI additionally no simulations can be added during runtime. All other approaches, OPC UA, OSGi and the agent concept, meet the challenge of adding simulations during runtime fully and partially meet the challenge of using different simulation tools. The reason for this is, that for each simulation tool which is wished be used, an interface for the data exchange and the synchronization has to be developed. As those three approaches all meet the challenges in the same degree, all of them can be used interchangeably, for co-simulating Digital Twins.

7.2 Agent-based co-simulation approach

In [44] an agent-based approach for a dynamic co-simulation is presented. The use of agents was chosen, as [44] additionally demands the possibility to add intelligence to the co-simulation, but this additional intelligence is not needed for realizing a co-simulation for Digital Twins. The presented approach includes a concept for the interfaces between the agents, which are used to distribute the data between the simulations, and the simulations. Those interfaces are divided into communication interface, used to simulate the communication between the assets, process-oriented interface, used to simulate the physical interaction between the assets, and synchronization interface, used to synchronize the simulations, to avoid causality errors. Those interfaces are further divided into a generic part, which provides functionalities, needed for all simulation tools and is connected to the agent and a specific part, which provides the tool-specific connection to the tool and has to be developed individually for each simulation tool. To add simulations dynamically during runtime to the co-simulation, the ability of agents, to enter an agent system during runtime is used. Additionally the concept of the simulation of the communication between the actual assets is also described in greater detail in [44]. The concept for the process-oriented interface is given in [45]. An concept for synchronization of the simulations is described in detail in [42].

7.3 Adaptation for the Digital Twin

The described co-simulation framework can be adapted to realize a co-simulation for Digital Twins, as it meets all the posed challenges. But the realization of such a co-simulation is not limited to agents, as the agents can be replaced by OSGi-bundles or OPC UA servers, as long as they provide interfaces to the simulation tools and a synchronization for the simulations. In case of the interfaces, the same interfaces can be used for both OSGi and OPC UA and in case of the synchronization, the clock agent has to be replaced by a “clock bundle” or by a “clock OPC UA server”. To realize a co-simulation for Digital Twins, each Digital Twin has to be connected to an agent, a bundle or an OPC UA server.

To realize this connection, an agent, a bundle or an OPC UA server has to be implemented into the co-simulation interface of the Digital Twin. Then this connection can happen automatically, whenever the relations of the Digital Twin towards other Digital Twins change, meaning whenever a Digital Twin is connected to another Digital Twin. Only the synchronization and the additional simulations for the communication and the environment have to be provided, but those are unavoidable, as they are needed for every co-simulation.

7.4 Evaluation of the dynamic co-simulation

For the evaluation, the agent system was developed in Jadex and the simulations of the assets were done in MATLAB/Simulink, OpenModelica and Java. The simulation scenario used for the evaluation was a smart warehouse with climate control. The approach was able to connect the different simulations, namely MATLAB/Simulink, OpenModelica and Java, during runtime. A more detailed description of the evaluation can be found in [45] and [44]. After the publication of those papers the prototype was extended by a simulation done in Unity 3D and a clock agent, which synchronizes the different simulations. We were not able to synchronize the models in OpenModelica, as OpenModelica provides no possibility to influence its internal clock. This shows, that not all simulation tools can be used unlimited, but that useable simulation tools have to fulfil certain requirements, especially for data exchange and synchronization.

8 Summary and outlook

In this paper, based on a literature research, three characteristics of the Digital Twin in a Cyber-Physical Production System were discussed, being synchronization with the real asset, co-simulation capability and active data acquisition. Furthermore, AI was proposed as the main feature of an Intelligent Digital Twin to enable autonomy in the Cyber-Physical Production Systems.

According to these characteristics, distinct architectures of the Digital Twin and the Intelligent Digital Twin have been proposed. The first architecture can be used to assist people in controlling and understanding real assets and their interactions in the Cyber Layer of the CPPS, but the realization of the Intelligent Digital Twin architecture allows for the development of an autonomous system with decision-making capabilities based on its own intelligence in different use cases.

To realize the Digital Twin architecture, the Anchor Point Method is applied to synchronize multi-disciplinary models of a Digital Twin, including inter-relations and static data. To synchronize operation data onto the Digital Twin, a cloud-based approach for data acquisition and data integration using semantic technologies has been proposed. Acquiring operation data enables the use of machine learning to detect anomalies and predict failures. Lastly, an agent-based co-simulation approach to simulate heterogeneous models in the Cyber Layer has been presented. The described concept has been partially realized with two industrial use cases, namely a modular production system as well as a metal forming industrial process to show the transferability as well as the potential an Intelligent Digital Twin has for a Cyber-Physical Production System.

Prospectively, an extended realization of the Intelligent Digital Twin will be taken on to further implement the concept. The Intelligent Digital Twin on the basis of a dynamic data acquisition is partly realized for the metal forming use case but will be extended and optimized within the scope of future research by extending existing machine learning algorithms and enabling a dynamic co-simulation.

Author contributions: Behrang Ashtari Talkhestani and Tobias Jung equal contribution.

Literature

1. L. Ribeiro and M. Bjorkman, “Transitioning From Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges”, *IEEE Syst. J.*, vol. 12, pp. 1–13, 2017.
2. E. A. Lee, “Cyber Physical Systems: Design Challenges”, in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363–369.
3. T. Männistö, T. Soininen, J. Tiihonen and R. Sulonen, “Framework and conceptual model for reconfiguration”, *AAAI99 Work. Config.*, vol. 99, pp. 5–21, 1999.
4. M. Eigner, “IT-Lösungen für den Produktentwicklungsprozess”, in *Neue Entwicklungen in der Unternehmensorganisation*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 211–229.
5. B. Ashtari Talkhestani, W. Schlögl and M. Weyrich, “Synchronisierung von digitalen Modellen”, *atp Ed.*, vol. 59, no. 07-08, p. 62, Sep. 2017.
6. M. Shafto et al., “Modeling, Simulation, Information Technology & Processing Roadmap: Technology Area 11”, *Natl. Aeronaut. Sp. Adm. Washington, DC, United States Am.*, 2012.
7. R. Rosen, G. Von Wichert, G. Lo and K. D. Bettenhausen, “About the importance of autonomy and digital twins for the future of manufacturing”, *IFAC-PapersOnLine*, vol. 28, no. 3, pp. 567–572, 2015.
8. G. PANGet al., “Multi-view 3D object recognition from a point cloud and change detection, Pub. No.: US 2014/0270163 A1”, Pub. No.: US 2015/0254499 A1, 2014.

9. W. Kritzinger, M. Karner, G. Traar, J. Henjes and W. Sihn, “Digital Twin in manufacturing: A categorical literature review and classification”, *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
10. S. Malakuti and S. Grüner, “Architectural aspects of digital twins in IIoT systems”, in *Proceedings of the 12th European Conference on Software Architecture Companion Proceedings – ECSA’18*, 2018, pp. 1–2.
11. S. Haag and R. Anderl, “Digital twin – Proof of concept”, *Manuf. Lett.*, vol. 15, pp. 64–66, Jan. 2018.
12. S. Boschert and R. Rosen, “Digital Twin—The Simulation Aspect”, in *Mechatronic Futures*, Cham: Springer International Publishing, 2016, pp. 59–74.
13. BMWi, “Details of the Asset Administration Shell – Part 1 – The exchange of information between partners in the value chain of Industrie 4.0, Version 1.0”, 2018.
14. B. Lindemann, N. Jazdi and M. Weyrich, “Multidimensionale Datenmodellierung und Analyse zur Qualitätssicherung in der Fertigungsautomatisierung”, in *Automation Kongress*, 2018.
15. B. Lindemann, F. Fesenmayr, N. Jazdi and M. Weyrich, “Anomaly Detection in Discrete Manufacturing Using Self-Learning Approaches”, *Procedia CIRP*, 2019.
16. VDI/VDE-RICHTLINIEN, “Sprache für I4.0-Komponenten Struktur von Nachrichten-VDI/VDE 2193”, 2019.
17. B. A. Talkhestani, N. Jazdi, W. Schloegl and M. Weyrich, “Consistency check to synchronize the Digital Twin of manufacturing automation based on anchor points”, *Procedia CIRP*, vol. 72, pp. 159–164, 2018.
18. R. Drath, *Datenaustausch in der Anlagenplanung mit AutomationML*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
19. M. Schleipen and O. Sauer, “Use of dynamic product and process information in a production monitoring and control system by means of CAEX and OPC UA”, *Proc. 3rd Int. Conf. Chang. Agil. Reconfigurable Virtual Prod.*, pp. 662–671, 2009.
20. M. Eigner, D. Roubanov and R. Zafirov, *Modellbasierte virtuelle Produktentwicklung*, 2014.
21. M. H. Landherr, *Integrated product and assembly configuration for the volume assembly of customized products*, 2014.
22. G. Frank, “Durchgängiges mechatronisches Engineering für Sondermaschinen, Dissertation”, Stuttgart: Fraunhofer Verlag, 2016.
23. F. Bellalouna, “Integrationsplattform für eine interdisziplinäre Entwicklung mechatronischer Produkte, Dissertation”, Fakultät für Maschinenbau der Ruhr-Universität Bochum (2009).
24. H. Zipper, F. Auris, A. Strahilov and M. Paul, “Keeping the Digital Twin up-to-date – Process Monitoring to Identify Changes in a Plant”, 2018.
25. S. Biallas, *Verification of Programmable Logic Controller Code using Model Checking and Static Analysis*, July. 2016.
26. S. Ulewicz, S. Feldmann, B. Vogel-Heuser and S. Diehm, “Visualisierung und Analyseunterstützung von Zusammenhängen in SPS-Programmen zur Verbesserung der Modularität und Wiederverwendung”, *VDI Kongress Autom.*, 2016.
27. P. Marks and M. Weyrich, “Assistenzsystem zur Aufwandsabschätzung der Software- Evolution von automatisierten Produktionssystemen”.
28. S. Feldmann and B. Vogel-Heuser, “Änderungsszenarien in der Automatisierungstechnik–Herausforderungen und interdisziplinäre Auswirkungen”, *Eng. von der Anforderung bis zum Betr.*, vol. 3, p. 95, 2013.
29. S. Feldmann, K. Kernschmidt and B. Vogel-Heuser, “Combining a SysML-based modeling approach and semantic technologies for analyzing change influences in manufacturing plant models”, *Procedia CIRP*, vol. 17, pp. 451–456, 2014.
30. K. Kernschmidtet al., “An integrated approach to analyze change-situations in the development of production systems”, *Procedia CIRP*, vol. 17, pp. 148–153, 2014.
31. S. Feldmann, K. Kernschmidt and B. Vogel-Heuser, “Konzept eines wissensbasierten Frameworks zur Spezifikation und Diagnose von Inkonsistenzen in mechatronischen Modellen”, *at - Autom.*, vol. 64, no. 3, pp. 199–215, Jan. 2016.
32. O. Kovalenko, E. Serral, M. Sabou, F. J. Ekaputra, D. Winkler and S. Biffl, “Automating Cross-Disciplinary Defect Detection in Multi-disciplinary Engineering Environments”, 2014, pp. 238–249.
33. S. Makris and K. Alexopoulos, “AutomationML server – A prototype data management system for multi disciplinary production engineering”, *Procedia CIRP*, vol. 2, no. 1, pp. 22–27, 2012.
34. D. Bergsjö, M. Vielhaber, D. Malvius, H. Burr and J. Malmqvist, “Product Lifecycle Management for Cross-X Engineering Design”, in *International Conference on Engineering Design, ICED’07*, August 2007, pp. 469–470.
35. S. Herbst and A. Hoffmann, “Product Lifecycle Management (PLM) mit Siemens Teamcenter”, in *Product Lifecycle Management (PLM) mit Siemens Teamcenter*, München: Carl Hanser Verlag GmbH & Co. KG, 2018, pp. I–XII.
36. W. Walla and J. Kiefer, “Life Cycle Engineering – Integration of New Products on Existing Production Systems in Automotive”, in *Glocalized Solutions for Sustainability in Manufacturing, vol. 82*, no. 3, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 207–212.
37. F. Biesinger, D. Meike, B. Kras and M. Weyrich, “A Case Study for a Digital Twin of Body-in-White Production Systems General Concept for Automated Updating of Planning Projects in the Digital Factory”, in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2018, pp. 19–26.
38. B. Ashtari Talkhestani, N. Jazdi, W. Schlögl and M. Weyrich, “A concept in synchronization of virtual production system with real factory based on anchor-point method”, *Procedia CIRP*, vol. 67, pp. 13–17, July 2018.
39. A. Faul, N. Jazdi and M. Weyrich, “Approach to interconnect existing industrial automation systems with the Industrial Internet”, in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016, pp. 1–4.
40. S.-J. Shin, J. Woo and S. Rachuri, “Predictive Analytics Model for Power Consumption in Manufacturing”, *Procedia CIRP*, vol. 15, pp. 153–158, 2014.
41. B. Lindemann, C. Karadogan, N. Jazdi, M. Liewald and M. Weyrich, “Cloud-based Control Approach in Discrete Manufacturing Using a Self-Learning Architecture”, *IFAC-PapersOnLine*, vol. 51, no. 10, pp. 163–168, 2018.

42. T. Jung and M. Weyrich, “Synchronization of a ‘Plug-and-Simulate’-capable Co-Simulation of Internet-of-Things-Components”, *Procedia CIRP*, 2019.
43. T. Jung, N. Jazdi and M. Weyrich, “A survey on dynamic simulation of automation systems and components in the Internet of Things”, *2017 22nd IEEE Int. Conf. Emerg. Technol. Fact. Autom.*, pp. 9–12, Sep. 2017.
44. T. Jung, P. Shah and M. Weyrich, “Dynamic Co-Simulation of Internet-of-Things-Components using a Multi-Agent-System”, *Procedia CIRP*, vol. 72, pp. 874–879, 2018.
45. T. Jung, N. Jazdi and M. Weyrich, “Dynamische Co-Simulation von Automatisierungssystemen und ihren Komponenten im Internet der Dinge – Prozessorientierte Interaktion von IoT-Komponenten”, in *15. Fachtagung EKA – Entwurf komplexer Automatisierungssysteme*, 2018.

Bionotes

Behrang Ashtari Talkhestani
Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70550 Stuttgart, Germany
behrang.ashtari@ias.uni-stuttgart.de

Behrang Ashtari, M. Sc., works since 2016 at the Institute of Industrial Automation and Software Engineering in the University of Stuttgart as a research assistant. His research focuses on the integrated modeling of the Digital Twin of an automated manufacturing system and on the synchronization of the Digital Twin’s models throughout the entire life cycle of the system. He also works on development of a flexible manufacturing system based on Digital Twin in the Research Campus ARENA2036 “Active Research Environment for the Next Generation of Automobiles”.

Tobias Jung
Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70550 Stuttgart, Germany
tobias.jung@ias.uni-stuttgart.de

Tobias Jung, M. Sc. Is a research assistant at the Institute of Industrial Automation and Software Engineering at the University of Stuttgart. His research focusses on “Plug-and-Simulate”-capable Co-Simulation of heterogeneous automated systems in the Internet-of-Things.

Benjamin Lindemann
Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70550 Stuttgart, Germany
benjamin.Lindemann@ias.uni-stuttgart.de

Benjamin Lindemann, M. Sc. is a research assistant at the Institute of Industrial Automation and Software Engineering at the University of Stuttgart. His research focusses on data-driven process modeling and analysis to optimize quality assurance along high-dimensional, industrial production chains.

Nada Sahlab
Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70550 Stuttgart, Germany
nada.sahlab@ias.uni-stuttgart.de

Nada Sahlab, M. Sc. is a research assistant at the Institute of Industrial Automation and Software Engineering at the University of Stuttgart. Her research focusses on intelligent automation for user-oriented assistance for the elderly.

Nasser Jazdi
Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70550 Stuttgart, Germany
Nasser.jazdi@ias.uni-stuttgart.de

Dr.-Ing. Nasser Jazdi is the deputy head of institute at the Institute for Automation Technology and Software Systems at the University of Stuttgart since 2003. He specializes in the Internet of Things as well as learning ability, reliability, safety and artificial intelligence in industrial automation.

Wolfgang Schloegl
Siemens AG, Digital Factory Division,
Product Lifecycle Management, Digital
Engineering Automation Designer,
Gleiwitzerstraße 555, 90475 Nuremberg,
Germany
schloegl.wolfgang@siemens.com

Dr.-Ing. Wolfgang Schloegl studied manufacturing engineering at the University of Erlangen/Nuremberg, where he subsequently earned his doctorate in the field of simulation in production engineering. Schloegl has been working for Siemens AG since 2003. He currently heads the Digital Engineering department in the Digital Industries operating company. His research focuses on integrating Siemens digital manufacturing systems with automation systems and on the Digital Twin of production.

Michael Weyrich
Institute of Industrial Automation and
Software Engineering, University of
Stuttgart, Pfaffenwaldring 47, 70550
Stuttgart, Germany
michael.Weyrich@ias.uni-stuttgart.de

Prof. Dr.-Ing. Dr. h. c. Michael Weyrich teaches at the University of Stuttgart and is head of the Institute of Industrial Automation and Software Engineering. His research focuses on intelligent automation systems, complexity control of cyber-physical systems and validation and verification of automation systems.