Chapter 4
Product Lifecycle Management Challenges of CPPS

Detlef Gerhard

Abstract In the chapter *Product Lifecycle Management (PLM) Challenges of CPPS,* data and information management issues arising from the advanced use of modern product development and engineering methods are addressed. These advanced methods are required for engineering processes of smart systems and individualized products with high complexity and variability. Emphasis is put on challenges of the life-cycle oriented information integration of products and the respective *Cyber-Physical Production Systems (CPPS)*. Furthermore, the chapter addresses data and information management problems coming from integration of the use and operation phase of products and systems in terms of forward and backward information flows.

Keywords Product lifecycle management (PLM) • Cyber-physical production systems (CPPS) • Information management • Model based systems engineering (MBSE) • Digital Twin

4.1 Introduction

Gill (2010) coined the term Cyber-Physical Systems (CPS) around 2006 and describes them as *"physical, biological, and engineered systems whose operations are integrated, monitored, and/or controlled by a computational core. Components are networked at every scale. Computing is deeply embedded into every physical component, possibly even into materials. The computational core is an embedded system, usually demands real-time response, and is most often distributed"*. Cyber-Physical Production Systems (CPPS) is a special term that depicts the introduction of the concept of CPS in the production domain in order to make production processes in general or production systems in particular "smarter"; this can be seen similarly to concepts in other domains, e.g. smart mobility, smart home, smart grid.

D. Gerhard (✉)
Mechanical Engineering Informatics and Virtual Product Development (MIVP) Research Group, Technische Universität Wien, Getreidemarkt 9/307, 1060, Wien, Austria
e-mail: detlef.gerhard@tuwien.ac.at

© Springer International Publishing AG 2017

S. Biffl et al. (eds.), *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, DOI 10.1007/978-3-319-56345-9_4

CPS as the entity of "smartness" combined with physical processes and objects that uses the *Internet of Things* (IoT) as a communication platform form CPPS in the sense of production value-added chains.

Within the domain of production of goods and products including associated services, CPS based technical systems have to be taken into account twofold: On the one hand side, products themselves are incorporating CPS concepts and on the other hand production facilities for product manufacturing and assembly as well. Hence, with the introduction of CPS, processes of product development, production system development, production (including production system commissioning) and product use (including maintenance, repair and overhaul processes) move closer together, are even strongly interlinked. This truly indicates the given complexity.

Product[1] Lifecycle Management (PLM) is the general concept to consistently create and manage all information related to products (systems/components). In particular, engineering information linked to corresponding engineering and production processes, as well as operation and usage phase is addressed. The major aim is to generate a sophisticated information basis for business and value generation based on products or systems to be produced. This concept comprises aspects of management, organization, and IT solutions and is typically realized with different types of business information software applications, e.g., Product Data Management (PDM), Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES), and Maintenance/Service/Asset Management.

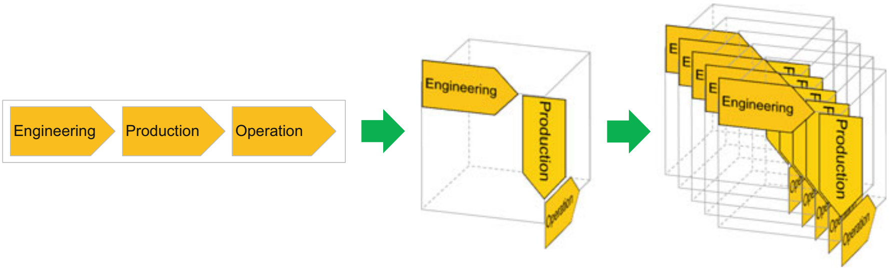
PLM focuses on three major phases: *engineering, production, and operation* of products. The engineering phase reaches from conceptual design of a product including all components up to detailed engineering. Complex technical systems, such as CPPS, are developed with an extensive utilization of different engineering tools and methods. In particular, the model based approach to product development called Model Based Systems Engineering (MBSE) is on the rise. MBSE depicts *"the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases"* (INCOSE 2007). This leads to many interlinked models from different so called authoring tools (e.g. Computer Aided Design CAD, simulation, software engineering) representing various required engineering domains, which have to be managed and maintained. With an increasing complexity of engineering projects and associated models, significant emphasis has to be put on interoperability and the ability to capture the semantics of data in order to be able to efficiently interface different systems and build tool chains. In industrial applications, predominantly PDM systems cover information management tasks of the engineering phase. This system category also supports engineering processes in the sense of workflow management. For this task, many procedural models for dividing this phase into several subsequent steps are commonly used in industry in order to realize product development processes in

[1] The term "product" is used synonymously for any kind of consumer product, machine or technical system in general, which requires a development and engineering process.

a systematic way, e.g. (VDI2221 1993) or (VDI2206 2004). Particularly, release and change management processes together with configuration management and versioning of information captured in documents and models is supported.

During production, the transition from the conceptual and virtual world to the materialization of a product with its parts and components takes place. This phase typically also starts with conceptual tasks, such as production system engineering and operations planning, e.g. Numerical Control (NC) and Programmable Logic Controller (PLC) programming. Commissioning of a production system is the transition to operation. Further, it involves all resource and production planning tasks resulting from order processing and additionally the respective control functions. This is a fundamental difference to the engineering phase. Whereas engineering focuses on generic product definition without consideration of capacities and resources, production focuses on the specificity of a single item instance or lot released for production with given constraints in terms of dates and resources (personnel, machines, material, etc.). Therefore, production can be seen as a lifecycle phase that is "orthogonal" to engineering (VDI2219 2002). The engineering phase does not stop with the beginning of the production phase. It rather continues creating further releases of a product reflecting improvements, variants, derivatives, etc. According to the corresponding hierarchical structure for industrial automation (IEC62264-3 2014), ERP systems are the category of IT systems that cover the customer orders processes by orchestrating all company's activities like commercial, financial, purchasing, logistics, production, etc. (Ben Khedher et al. 2011). Hence, they are predominantly used in industrial applications for planning, controlling, and information management purposes of the lifecycle phase production on level 4 of IEC62264 functional hierarchy. Furthermore, on level 3 specialized MES systems or ERP manufacturing operations management modules cover the required IT functions, e.g. for detailed planning or production data collection. Manufacturing Execution Systems (MES) are used to deal with detailed production planning and control tasks. MES is much closer to the shop floor activities and therefore it requires more specific production related information because of its shorter planning intervals.

Again orthogonal to the previous phases engineering and production of the lifecycle, there is the operation phase of a product. Each produced single item instance is used or operated differently after production and shipment. This holds true for consumer products, e.g. household appliances, which are produced in a considerable lot size of identical items as well as for customized one-of-a-kind special purpose machines. Hence, in terms of product information management requirements, each item instance has to be treated separately, sometimes even components of the item instance. Processes that have to be supported during operation phase are maintenance (service), repair, and overhaul (MRO) processes, depending on the type of product. On the one hand side, these processes imply new service orders and generate business processes, which have to be managed and supported. This is typically done with software modules of ERP systems or special service management software tools. On the other hand, there is a link to the upstream product lifecycle phase. In particular, tracking of production steps or



i) Main Phases of Product Lifecycle

ii) Orthogonality

iii) Multiplicity

Fig. 4.1 Different views (schematic) on product lifecycle phases

engineering tasks, relevant for a situation that occurs during operation is required. At this point, the orthogonality of the different phases becomes obvious since very heterogeneous types of information have to be linked and dealt with: For instance, information about a particular product in operation with information of a production lot of a particular component and corresponding generic engineering information.

Figure 4.1 schematically shows the different views towards the product lifecycle, as indicated in the previous paragraph. Part (1) depicts a rather simplified view on the three main phases, (2) emphasizes on the aspect that the different phases have to be seen as orthogonal to each other, and (3) indicates, that the orthogonal phases of the product lifecycle are characterized by massive multiplicity and different maturity status. The latter imposes even increased complexity in terms of chronological interdependencies of products and components in development, production or operation as well as their associated processes.

In order to derive challenges and requirements for PLM in the context of CPPS, it is necessary to analyze product engineering and production processes one level deeper. In particular, it is required to distinguish different product types, production concepts, and production types, since this distinction is the basis for the required product information management approach. As already mentioned in Chap. 1, four main different production concepts, reflecting the procedure during order processing, can be differentiated (Schuh 2006): Make-to-Stock (MTS), Assemble-to-Order (ATO); Make-to-Order (MTO); Engineer-to-Order (ETO). This differentiation of product types and their production concepts is necessary to express the degree of dependencies among the product components and the production system. Below, aspects that typically apply to the different concepts are outlined and described in terms of main characteristics for PLM.

MTS determines a production concept that is typically applied for consumer products, which are produced in larger volumes (series or mass production) without major variants that have to be managed. Such products as for instance hand machining tools are in general not subject to extensive after sales services. Therefore, the application of CPS within those products, particularly to collect usage and operation data, is still an exception. Nonetheless, usage phase data could be collected indirectly based on customer feedback, but collected usage data in the operation phase is not directly fed back to the previous phases or used for maintenance business creation. MTS products are developed without customer orders based on market research. Due to the high production volume, a specialized production system has to be engineered in parallel. This production system respectively the engineering process, can be seen as an ETO product or system itself. Therefore, many companies producing mass products on the one hand side, are also producers of production facilities with separate business units. In order to effectively manage both processes, the two different information flows have to be tightly linked together (Gerhard and Lutz 2011).

The ability to directly optimize the production process based on process data that is collected and analyzed in real-time, is the major goal of CPPS engineering in ETO production processes, not so much CPS based adaptability and re-configurability of production systems due to the large production volumes. Furthermore, a major objective of improvement is to shorten commissioning and ramp-up phase because this can save significantly time and money. Virtual Product Engineering (VPE) methods are widely adopted in industry, i.e., the complete description of complex technical systems and their characteristics as computer model together with integration and optimization of IT tools for domain-spanning, multi-disciplinary information management.

Virtualization of production system engineering, often referred to as *"Digital Factory",* is not as widely adopted in industrial applications but truly on the rise since there is a lot of potential for speeding up time-to-production. From product geometry and material properties defined in engineering design, there is a direct link to required manufacturing operations and programming of NC machining tools in case of material shaping operations. In addition, clamping devices, fixtures, jigs, chucks as well as quality assurance and measurement devices have a direct geometrical link to products or parts of products themselves. Therefore, release and change processes of products and production system items are closely interlinked. The Digital Factory approach goes even beyond this and includes additionally the virtual representation of assembly (ergonomics), intra-logistics, and material handling processes. With respect to the chronological order, two main stages have to be considered, firstly engineering of a production system (typically an ETO process, concurrently started at a certain maturity status of product engineering) and secondly production system operation (including ramp up). The aim is to build a so called *"Digital Twin"* or "C*yber Twin"* during production system engineering, i.e., computerized companions of physical artefacts that can be used for various engineering purposes and use data from sensors to represent their near real-time status, working condition, position, etc. (Tuegel et al. 2011). A production system—as generally most complex technical systems—faces constant changes and adaptions during operation due to maintenance and improvement procedures. In addition to the Digital Twin representing the results of the development engineering processes, an up-to-date Digital Twin representation is the goal during operation phase. This way, all possible changes and modifications can be verified in advance and also tracked. Furthermore, a Digital Twin of each component can be used for capturing condition monitoring records and synthesizing future steps to provide

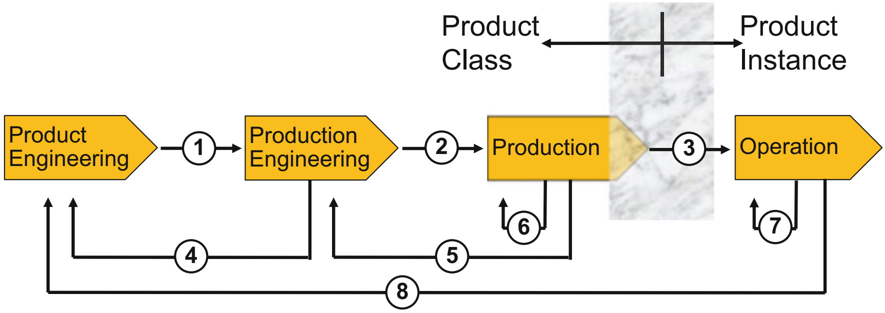


Fig. 4.2 Mayor information flows—example ETO system

self-awareness and self-prediction (Lee et al. 2015). To realize this, a fundamental distinction has to be realized within the respective PLM solution. Whereas all engineering processes take only product classes into account, for each Digital Twin representation, a product instance representation is required representing runtime and operation. This is depicted in Fig. 4.2 below. The different information flows are detailed in a later paragraph. For simplification reasons, phases are shown in a strictly subsequent manner though they are partially overlapping and concurrent.

Whereas MTS and ETO both define the end points, ATO and MTO are located in the center of the production concepts spectrum. ATO is oriented closer to MTS since this concept describes preproduction of standard products with manufacturer-specific variants. MTO is oriented closer to ETO since this concept describes production of standard products with customer-specific variants that are partly composed of pre-defined components and partly made up newly created components. There are two main drivers, that have major influence on production: The combination of IT and "ordinary" products leads to *"Smart Products"* with embedded systems providing an added value to both, customers and producers. Producers, in particular, can extend ordinary products to *Product Service Systems (PSS)*, as described in Chap. 3. Smart products require extensively multi-disciplinary engineering of the product itself as well as the production system and furthermore intelligent backend information technology for supporting the use phase. The trend towards individualized products with customer specific requirements is moving production towards mass customization and lot-size-one concepts.

These concepts for production processes are mainly addressed with CPPS approaches, though CPPS approaches for ATO and MTO production have mainly different goals compared to MTS production. Within MTS production, the product is rather invariant. The production process can be adjusted and optimized with respect to a single product. Short commissioning and ramp-up plays a vital role as well as optimization of the whole process utilizing sensor data and machine feedback with respect to completion, quality or errors in a direct control loop. This goal is mirrored in the *"Overall Equipment Effectiveness" (OEE)* Key Performance Indicator (KPI) coined by Nakajima (1982). This KPI particularly reflects on the measures for optimization of mass production. OEE is beneficial in high-volume and highly automated process-based manufacture where capacity utilization is a high priority. Deployment of OEE in low-volume job shops is not very beneficial (Charaf and Ding 2015).

ATO and MTO production concepts demand extensive flexibility in manufacturing and assembly processes and a by far greater collaboration of product engineering and production system engineering. Variant rich and customized products require particularly configuration of product structures and linked manufacturing and assembly operations. Production according to these concepts typically is realized by manufacturing shops providing different NC machining tools for special operations and flexible manufacturing cells for small and medium batch production of parts. The major goal for CPPS approaches in this case is to realize an intelligent, resilient and self-adaptable system of interconnected machines and manufacturing cells capable of producing customized products with a high degree of automation and thereby at competitive costs. In this scenario, in addition to manufacturing operations, automated material handling plays a vital role. This insight is not new (Sethi and Sethi 1990) but nonetheless still an issue that is currently not sufficiently tackled using computer aided engineering methods (Seibold and Furmanns 2015).

With a PLM view on CPPS in ATO and MTO production environments, again engineering, production, and operation/use have to be taken into account and adequately supported in different ways. In the conceptual stage, based on the product definition, the different manufacturing and assembly steps have to be planned using a variety of CAD methods. This is done in general on a product class level though partially single items have to be tracked on instance level. Production engineering in this case is reduced to operations planning and NC programming, i.e., an existing set of machining tools and material handling systems has to be customized and set up in the sense of a flexible manufacturing shop but a special production system does not have to be engineered and build for this kind of products. Afterwards, during execution time of the respective production orders of parts and assemblies (typically in smaller batches), the required production information has to be provided and actual production data has to be captured. ERP and MES system build the runtime environment for production execution and respective information management, top-down in the sense of planning and target values as well as bottom-up in the sense of shop floor data collection. Production execution is the transition from the virtual to the real product and also the transition from the product class view to the product instance view. Each of the produced product instances during its use phases is operated in a different way and under certain environmental conditions. Nowadays, many of the more sophisticated products are equipped with embedded systems and are capable to a certain extent to capture usage information. One prominent example for such products are cars. They are equipped extensively with controllers and embedded systems. In addition to the functions vital for operating the car, they provide capabilities for capturing data that can be used for classical maintenance and service processes, even for new services supporting drivers or car holders in everyday tasks, navigation, etc.

4.2 State of the Art and Challenges of PLM in the CPPS Context

As indicated above, the vision of CPPS depicts production facilities consisting of smart machines that are connected and able to connect ad-hoc with smart products and objects in order to autonomously exchange information, trigger actions and control each other. CPPS concepts are based on the "Internet of Things (IoT)" concept coined 1999 by Ashton (2009), i.e., every physical object, machine, product or object has a virtual representation. Gunes et al. (2014) elaborates on the following challenges to CPS: Interoperability, Predictability, Reliability, Sustainability, Dependability, and Security. These have also to be taken into account in engineering and PLM for CPPS. Interoperability has several aspects: combining and incorporating heterogenic components of technical systems scaling in size, throughput or other dimensions. Predictability reflects on accuracy of intended outcomes in terms of behavior that autonomous systems show based on inferencing and reasoning in particular contexts. From an engineering point of view, this leads to challenges with respect to robust and stable performance of a technical system, i.e., predictability of a guaranteed behavior and reliant operation. Correct functioning, availability, safety, and maintainability during operation has to be assured, especially if CPPS are self-adaptable or reconfigurable according to changing contexts and dynamic tuning. Particularly, issue tracking based on the right product lifecycle information management becomes difficult since the origin of many issues might be software based. This aspect also leads to security challenges and questions of integrity or reliability, i.e., if privacy and confidentiality of information can be guaranteed and if information is correct and trustful.

In addition to conventional automation and control technology for production facilities, the aim of CPPS is to design "smart" systems that embody so called self-x capabilities (e.g., self-configuration, self-organization, self-optimization) in order to be able adapt autonomously to unforeseen states on machine level as well as non-intended situations on production system level due to failures, lack of material, etc. Even though autonomous interaction on micro-level is intended and required for the implementation of advanced production concepts in modern environments, predictability and controllability of the whole production system on macro-level has to be assured. Machine operators as well as production planners in charge need to have control over the production system. A supplying company of a machine tool, a flexible production cell or a material handling system has to guarantee certain levels of function and behavior. Therefore, behavior and logic needs to be represented using model based descriptions.

Consequently, the challenges of PLM approaches for CPPS are threefold:

- Processes and methods to support systematic multi-domain engineering
- System and information modelling (model representation)
- Information management, particularly data linkage and data analytics.

4.2.1 Processes and Methods

CPPS are complex technical systems characterized by networked structures, nonlinear behavior and means multi-causation, multi-variability, multi-dimensionality, interdependence with the environment, and openness. Therefore, product design as well as production engineering tasks have to be addressed in a systematic way. Complexity in this particular case has the following facets (in causal order):

- Product and production system complexity
- Process and organization complexity
- IT landscape and tool complexity.

Complexity of products is caused by multiple instances and variants of a base product to meet requirements with respect to customization demands and differentiation of the target markets. Besides the mechanical components, nearly all products consist of electronics, embedded systems with sensors and actuators and have firmware/software driven controllers. Because there are many different domains of expertise involved in engineering tasks, process complexity also increases through dissemination over locations (countries, cultures) and distribution within the supply chain (organizations). Collaborative engineering processes require even more extensive use and support of advanced IT systems. The diversity and dynamics of the relationships between project partners, manufacturers, vendors, and suppliers leads to highly sophisticated IT landscapes with docents of data formats, representing different semantics.

CPPS, have to be engineered and designed within a multi-domain environment of virtual product development tools. Outcomes of such a development process determines corridors of operation and limitations to autonomous behavior of a CPPS. The product and system development process today is well supported with different tools for e.g. software engineering, CAD, electronic design automation, and simulation. Integration of the domain specific processes for mechanical design, electric/electronics engineering and software development is the main challenge in creating complex technical systems as CPPS in a robust and reliant way. Especially, there is a gap within the information flow between early phases, i.e., engineering design, and later phases, i.e., production (Gerhard and Weilguny 2008). Both phases are in general well supported by different IT tools but the integration and information flow between is still missing the required level of maturity. This problem particular increases with the trend to customization and individualization of products and product service systems. CPPS require a systematic approach towards the different engineering design tasks. Requirements engineering methods help to identify the core of given challenges and further guide through the development process (Cheng and Atlee 2007). Modern CAx systems offer extensive functions to solve geometric modelling and design tasks, but for multi-domain modelling, there is always an information loss when exchanging data between different models.

By expanding the range of functions and system limits, smart products and production systems have more complexity than conventional products. Existing development methods and state of the art tools are not fully suitable to support the specific characteristics and requirements of communication-capable, autonomous decisive CPPS. Methods and a tool chains for the coherent virtual representation of a production system being in operation (product manufacturing) and linked to the product development process are required. In the context of CPPS, this is necessary in order to be able to rapidly respond to required changes and simulate the new behavior of a customized system in advance. Typically, engineering design processes end with the completed definition of a product ready for production and ready to be utilized after production. The requirements of future technical systems such as CPPS go one step further. System functionality from production and operation phase has to be captured by corresponding system models as well, though the semantics of information models in engineering design and production still is quite different. Only in this way, it is possible to collect data from operation and use phase and utilize this models for simulation and optimization of the production process or use phase of a product. Different simulation models are required to capture e.g. cycle times, output or quality data of production systems. Having those models in place, different scenarios of operation can be simulated (e.g. maximum speed vs. average speed or eco mode) in order to find an optimal operation behavior in terms of time, material or energy consumption, friction, loss, wear, etc.

Since PLM in particular focuses on engineering processes, one major challenge is to find a comprehensible way to support engineering design and development process of CPPS on a methodical level. The question is if it is a feasible approach to enhance or adapt the procedural VDI 2206 V-model in terms of cascaded systems of systems modelling. Nattermann and Anderl (2010) for instance proposed a W-model approach for systematic engineering of adaptronic systems. This model suggests the use of a special data management layer, which provides not only a central control of the records of all disciplines but is also able to analyze the discipline-specific records and to synchronize across disciplines. This data management system should be capable to capture state and behavior of the system under development at any time and to ensure compatibility of the discipline-specific components and subsystems.

There is a direct linked to the research questions formulated in Chap. 1: How can model-based methodologies support information creation and processing in the different life cycle phases of a CPPS and how shall several disciplines in product and production system engineering be linked to support the engineering of flexible and self-adaptable CPPS? The question is how virtual engineering support for an integrative CPPS hardware/software co-design, verification, validation, and testing can be realized, given the multitude of different tools and methods. Particularly not tackled adequately so far are methods and technologies that support the links between product, production technology, and production systems engineering, i.e., horizontal, vertical and life cycle integration within production systems and digital links between engineering and operation phases.

4.2.2 Model Representation

The central concept embodied in multi-disciplinary engineering and model-based design is that the 3D product model is the most appropriate vehicle for delivering all of the detailed product information necessary for downstream processes and operations to perform their portion of the product creation (Quintana et al. 2010). CAD models are enriched with explicit and implicit knowledge which needs to be extracted, formalized and managed for re-use in different contexts. However, extracting knowledge encapsulated in CAD models remains a challenge and does not cover at all the complex systems engineering requirements. (PROSTEP 2015) gives a comprehensive over relevant standards for the different procedural steps of the VDI 2206 V-model, but the V-model more or less stops at the start of production and does not cover production and operation.

Nonetheless, for information modelling in engineering processes, there have been considerable developments within *STEP—STandard for Exchange of Product Model Data* (ISO 10303 1994). ISO 10303 is an international standard for the description of physical und functional features of product data. STEP interface definitions and data formats allow data exchange between different CAx systems and aims to represent all data of the whole product lifecycle of a product. In addition to geometric data, this data can e.g. comprise production planning information, bills of materials, simulations, design studies, and much more. To deal with all kinds of lifecycle data, ISO 10303 consists of an extensive collection of so called *Application Protocols* (AP). Each AP is adapted for a special purpose, for instance ISO 10303-242 *"Managed Model Based 3D Engineering"* provides relevant data models merging AP203 and AP214, which are currently the most implemented for CAD data exchange between existing commercial CAD systems. AP 239 *"Application Protocol: Product Life Cycle Support"* furthermore includes the representation of a product through life including product requirements and their fulfilment, the identification of the configuration of a product for a given role, and the specification of effectivity constraints applied to configuration of a product. AP 233 specifies the representation of systems engineering data and defines the context, scope and information requirements for various development stages during the design of a system.

For modelling lightweight representations of geometry together with *product manufacturing information (PMI)*, the newer JT standard (ISO 14306 2012) is interesting to consider. To capture model information beyond geometry, e.g., requirements, logic, function, and physics, System Modeling Language (SysML) (see also Chap. 2 and SYSML 2007) is adopted increasingly. This virtual representation of artefacts is a means to integrate and organize the multitude of models necessary to describe all the aspects of the system with the aim to support interoperability between the domains and their data. Since many different disciplines are involved, complexity handling in engineering and manufacturing (Tolio et al. 2010) is the main issue that is tackled with these standardization approaches.

Self-x functionalities in the sense of smart systems or applicable for a smart factory are functionalities take into account the context of (a group of) CPPS and react accordingly to this context, facilitating the adaptive autonomic behavior of CPPS. Going more into detail, research questions of Chap. 1 contain the challenges of how to define or model self-x functionalities of CPPS as well as means for capturing context, behavior and state of artefacts? Furthermore, strategies and algorithms for modelling and simulation of anticipatory system behavior and layer-crossing integration of self-x actions have to be developed. Monostori (2014) states in a comprehensive survey the following R&D challenges for CPPS (among others): For context-adaptive and (at least partially) autonomous systems, methods for comprehensive, continuous context awareness as well as for recognition, analysis and interpretation of plans and intentions of objects are necessary. Furthermore, the development of new methods is required, which support the fusion of the real systems with the virtual representation in order to reach the goal of an intelligent production system which is robust in a changing and uncertain environment.

4.2.3 Information Management and Integration

In industrial applications today, product development processes, production planning processes, and order based production planning and control are still to a large extent disconnected. This holds in particular true for data generated during the use phase of products. For CPPS and smart production, a closed loop information management is crucial, spanning the whole lifecycle from product concept and design to production system planning, to order management and production, and finally to product operation or usage. The international standard IEC 62264 (IEC 62264-3 2007) defines models and transactions for the integration of ERP and MES. Its main objective is the integration of business planning and logistics systems to manufacturing operations management systems. While ERP systems operate in time frames of months, weeks and days, the detailed production planning is done using much shorter periods like shifts, hours, minutes, seconds and even sub-seconds. VDI 5600 guideline (VDI 5600 2007) offers a problem-oriented description of MES and its application potentials. The main tasks of MES like detailed scheduling and process control, equipment and material management, etc. are defined and the role of MES for enterprise processes is highlighted. Similar to IEC 62264, VDI 5600 contains recommendations for the interface management between MES and machines/terminals/sensors on the manufacturing level. Both standards mainly address the so called *Automation Pyramid* from shop floor to top floor in a vertical integration direction, respectively along the production value chain in a horizontal integration direction. The integration along the product lifecycle from very early engineering design stages to production and operation stage or vice versa is not covered.

The main challenge is that generated data and relevant information at the various stages of the product lifecycle is quite different in terms of the three orthogonal aspects of data storage, processing, and analysis (3Vs of Big Data), i.e., volume, variety, and velocity (Laney 2001). Particularly, there is a variety of formats capturing different semantics, not just relationally structured data representing different models but unstructured content. Content that is tagged with metadata and hierarchical file system data. In engineering design, information about the product in general is created, in early phases there are abstract models, later more tangible and concrete content. In later stages of the product lifecycle, data or information flows can be highly inconsistent with peaks and the meaning can also change over time. The generated information is to a large extent unstructured and stochastic but not model driven. Data or information flows can be highly inconsistent with peaks and the meaning can also change over time. The more an integration of the different information aspects over the product lifecycle is possible the more efficient processes become. The use of cognitive computing approaches, ontologies and semantic technologies in the integration of data, information, as well as knowledge throughout the complete design cycle of a product has the potential to substantially improve both the product and associated development processes (Welp et al. 2007).

Forward and backward information flows have to be distinguished. Design engineers usually have a good understanding of the product they are developing, but any approach to integrate information required in later process stages (e.g. environmentally relevant information or production relevant information) into early product development and design stages often fails since it adds to workload or complexity of the process. Ostad-Ahmad-Ghorabi et al. (2013) tackle this issue introducing ontological approach to set up primary parameters systematically for particular product categories driving e.g. the environmental performance of products. The aim of ontological approaches in general is to enable the management and (re-)use of heterogeneous data along the product development process. Different information and different views concerning a product structure facilitates the work of each phase. Yet, the lack of contextual relationships within the structures avoids linking the correct data between them. Therefore, similar information must often be entered multiple times and a cross comparison between the information from other domains or an automated comparison of the various activities in concurrent engineering processes become virtually impossible. With the use of semantic technologies and ontologies the continuity of information can be achieved through a coherent semantic structure associated with views on different areas of the development process (Gerhard 2012). Ontologies serve as a neutral or intermediate layer, in which semantic web technologies can be used to build queries or filters that provide specific data of heterogeneous sources. Ontologies are thus of great importance when encoding design knowledge as well as integrating software systems for facilitating semantic interoperability (Chandrasegaran et al. 2013).

Semantic technologies also play a vital role in data analytics. Smart factories collect vast amounts of data from different sources: Product design data such as bill-of-materials (BOM) and CAD-files; production process data such as CAM-files, machine scheduling and QC measurements; logistics data including demand forecasts; and data from a multitude of sensor constantly monitoring machine parameters. Currently, ERP and MES systems used in manufacturing operations do not adequately mine this data to identify useful patterns and draw conclusions for operations or engineering processes. This is because current data mining techniques are typically not suitable for time series data and therefore, are of limited use in making predictions (Gröger et al. 2012). Additional data objects or information model enhancements of existing software tools or standards are required to capture and interface engineering as well as run time data of complex CPPS so they can be indexed and retrieved for reuse across different products or projects. This comprises versioning and means to align different (multi-domain) development paths. Still the research question remains, e.g. if it is possible to find algorithms to analyze data patterns from manufacturing data for re-use at different stages of the product creation process, transforming information from data to knowledge level.

4.3 PLM Forward and Backward Information Flows in CPPS

With the above introduced contents and challenges, it is clear that an enhanced use of CPS in products and production systems imposes new approaches to look at the way product related models and documented information have to be managed along the product lifecycle. As stated before, the lifecycle phases cannot be seen just as subsequent stages but the orthogonality has to be acknowledged.

IT systems and software solutions for engineering information modeling and management represent “virtual” product and production engineering information linked to a product class as well as order based production planning and real time data of the production process as well as individual usage data. In other words, they have to cover a complex patchwork of different views in terms of functionality and models semantics, i.e.,

- Development/engineering, planning, production, operation/use
- Requirements, features/working principles, logic/behavior, geometry/shape
- Structure of products (systems), modules/components and parts/elements
- Mechanics, hydraulics, pneumatics, E/E, control/software
- Manufacturing, assembly, testing, packaging, transportation
- Customer, supplier, service owners/operators
- Building/infrastructure, energy.

A unified and coherent model description of all necessary information (in a knowledge domain as CPPS) is unrealistic. Requiring all applications to share a common standardized data model to be truly integrated is not a feasible solution. Hence, data linkage in a federated manner, semantic technologies, and cognitive computing approaches can be seen as enablers to introduce new agility and expanded scope to enterprise applications, such as for instance:

- Automated extraction of metadata to transform unstructured data into a fully classified resource and synthesize it with existing structured data

- Enrichment of structured data with qualitative data from vast “unstructured” sources like sensor-captured production data or usage data from emails, blogs, chats, and social Web pages
- Identification of embedded meanings and relationships within and across resources through data analytics
- Natural language processing to interpret imprecise requests and offer spelling corrections, close matches, and related content
- Creation of innovative and tailored apps that seamlessly merge content and functionality from diverse sources such as databases, mapping services, and WWW resources.

This is necessary in order to perform data specialization tasks in forward direction (early lifecycle phases to late phases) and data generalization tasks in backward direction (vice versa).

Forward integration of engineering information in the sense of “Design For X” (DfX) is the concept comprising all endeavors towards making the right decisions in the product development process on basis of sufficient and universally applicable knowledge basis. The aim is to take into account impacts that decisions in early phases of the product development process have on later phases. Particularly, concepts of Design for Manufacturing, Assembly, and Service are relevant for CPPS and in many companies in place in order to ensure high quality at optimized cost and time efforts in the production or operation phase. Forward integration nowadays means predominantly manual processing of data, e.g. using CAD/CAM data in order to extract required information for operations planning. PDM systems support these tasks to a certain coverage since they provide easy access to required information but they do not provide assistance in terms of e.g. supporting operations planning. Especially the inherent semantics of product defining data to be used at later stages is still a weak point. Therefore, to a large extent, operations planning relies on the experience of planning engineers. Knowledge that can be derived from past projects and tasks is not taken into account systematically in many cases and the potential of intelligent knowledge re-use is not addressed. In the forward direction new questions arise since products become more sophisticated integrating embedded IT systems and/or IoT technologies.

In the backward direction, feedback information from usage and operation phase collected on a single item or instance basis, which is in general less structured needs to be aggregated and generalized to be used earlier phases, i.e., engineering design or production, in the sense of knowledge management. Backward information integration in terms of PLM also requires new approaches in order to leverage opportunities and adequately support processes related e.g. to PSS. The semantics of information models in engineering design and production still is quite different. Instance based information from the use phase of a product has to be captured, generalized and mapped to product class information of product engineering or production engineering phase in order take benefit in terms of knowledge management. In the operation phase of a product or machine (maintenance and support phase), the inherent task of evaluating if design requirements are met is to take a close look at the performance of the product and actual use. A closed feedback loop is the idea that the output is looked at with respect to a desired goal and then the inputs are modified in order to change the output to close the gap between what the output is producing and what is desired. Feedback loops can be direct and internal or indirect and external to a system. If a product does not meet its design requirements in actual usage or if actual usage surfaces additional requirements, respective information should be fed back into a base of knowledge so that engineering can understand the gap between the requirements they thought could be fulfilled and what actually occurred. Conclusions can be drawn and requirements for future versions of the product can be adapted.

Particularly for the backward information flow, it is important to distinguish the type and the instances of an information object over its lifecycle, i.e., to have unique identifiers both in the digital (virtual) and in in the physical world.

- Instantiation of product data: For each product in the field, there has to be a separate instance of product data created. This dedicated instance will be maintained over the lifetime of the product, to stay up to date, even when parts are changed during maintenance (e.g. for long life products like machining centers).
- Instantiation of usage data: Data collected during the usage of products have to be associated to the specific instance of the product instead of its generic model.

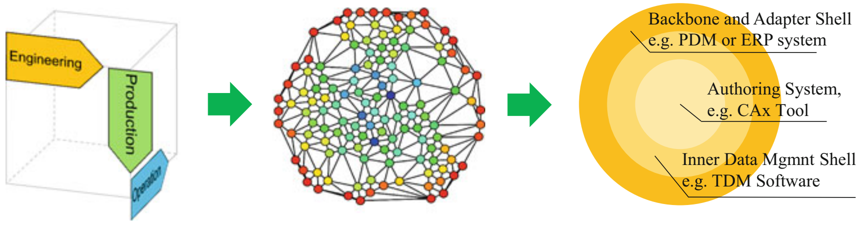
Figure 4.2 of the previous section gives an example (ETO production concept) for the different information flows. Briefly the main information flows can be depicted as follows:

1. Engineering design data are used to generate production system and operations planning data
2. Planning data serve as the basis for production control (target values)
3. Actual data of production are the basis for product operation/use including MRO processes
4. Feedback data to improve the product
5. Feedback data to improve production system and operations planning
6. Actual data for direct optimization of the production process (target-performance comparison)
7. Actual data for direct optimization of the operation and MRO support
8. Actual data for improving and further developing the product.

Benefits resulting from the forward and backward information integration are to a large extent company and use case specific. For each use case, the first step is to figure out who benefits from the delivered information, and therefore, in what form and where the information has to be presented, e.g. is the information already necessary/useful in the production planning phase, or is it important later in the physical production process. For example, simulated milling operation times stored in the PDM system can be used to calculate target times for operations planning, or assembly instructions can be displayed on terminal screens in manual assembly lines. After identification of the required data, suitable approaches for data structures and data processing have to be defined. Concerning backward integration, it is important to identify, which data from the MES or other data sources is accessible and useful for feedback in the PDM data backbone. That can be raw data (e.g. from sensors) or already processed data (e.g. key performance indicators of machines). It has to be clarified, which data has an impact to the generic data in the PDM system, and how information can be viable created out of all the collected data. For example, if production introduces new cutting inserts for milling operations resulting in higher duration of the tool and less tooling time in a production process, this information can be fed back to operations planning. This means that raw production data has to be mined and analyzed with respect to the deviation of planned and actual values taking into account possible outliers.

A software architecture comprising so called authoring tools for different engineering tasks on the one hand side and comprehensive engineering and business information management software systems on the other hand side has to take application diversity into account. Ther e are ongoing research activities to investigate and develop an approach for multidisciplinary life-cycle oriented information integration in systems engineering within the *Open Services for Lifecycle Collaboration* (OSLC) working group of OMG (OSLC4MBSE 2013). The focus of major activities still is in the engineering domain, but engineering, production and usage have to be treated holistically in the context of CPPS, which goes beyond this viewpoint. Especially, many different concepts have to be mixed and supported with software tools, like e.g. model based engineering, document oriented information flows, time related data and time series processing, location based data and geospatial processing, database oriented transaction handling and posting entries to ledger accounts. Hence, a principle IT system architecture needs to support a strongly federated network approach of nodes performing particular tasks, in which the nodes themselves follow an approach that can be compared to an onion with a shell like structure incorporating the concept of microservices. Microservices is an emerging trend in the cloud era: briefly *“microservices are small, autonomous services that work together*” (Newman 2015) to achieve a common or requested functionality. Similar to the Service Oriented Architecture (SOA) approach, microservices are independently deployable, small, and modular services that communicate loosely coupled over HTTP protocol typically through REST APIs with simple semantic standards that can map to any data model using JSON as a data exchange format. This concept also supports, that contextual information from third party systems can be provided via a persistent linked data layer that overcome system and organizational boundaries. Figure 4.3 below depicts the rationale behind a principle IT system architecture suitable for PLM in the context of CPPS. As stated before, the different phases of the product lifecycle have to be seen orthogonal to each other.

In all phases, many different software tools (depicted as dots of the network in the respective colors in Sect. 2 of the figure) generate information that is linked to another portion of information, e.g. structured or unstructured information, simulation models of the engineering phase as well as sensor data of the production phase that cannot be captured in models. At the outer area of the network (colored in red), there are even dots representing system boundaries or transition to other



i) Orthogonal Phases ii) Network of microservices iii) Shell structure of microservice

Fig. 4.3 Principle IT system architecture for PLM in the context of CPPS

domains, e.g. smart grid energy management systems. The example given in Sect. 3 of Fig. 4.3 reflects the viewpoint of mechanical engineering: A CAD system as core of required software functionality for a particular process task is wrapped in a Team Data Management (TDM) environment with high integration depth supporting collaborative engineering work. The TDM system again is enveloped in a PDM or ERP backbone. On this level the microservice approach comes into play, i.e., communication to other services through defined API leads to a federated approach of exchanging required information within the given plethora of systems. Beyond the expressed example, the same holds true for different engineering domains, e.g. a CAE system wrapped in a specialized simulation data management tool or a software engineering tool which organizes the software development work within source code management environment. Section 3 of Fig. 4.3 could also be colored blue or green in order to represent a microservice of the production or operation phase or even in mixed colors if there is no clear assignment possible. The major differences are the type of generated or captured information and the point in time, which the information represents. Nonetheless, information of the three different life cycle phases is truly interlinked.

Incorporating a microservices concept likely leads to a situation, that instead of less large and established enterprise applications suddenly a landscapes with a variety of small, fast-changing services emerges, which all have to be configured, managed, and monitored. This issue can be tackled using a so called container technology like *"Docker"*. Docker uses *"containers"*, which capture everything that is needed to run a chosen software (e.g. code, runtime, system tools, system libraries, binaries, dependencies, etc.). Docker containers represent one encapsulated unit of functionality to the "external world". In this way, it is assured that the code will run in any selected environment the same way (Mouat 2015). Docker containers are rather lightweight in comparison to hypervisor techniques, since virtualization is done on operating system level, encapsulated from the rest of the host system. The feasibility of this concept is underpinned by the fact that most of large public clouds have made their systems compatible to Docker (e.g. AWS Elastic Beanstalk, Google AppEngine, IBM Cloud, Microsoft Azure, Rackspace Cloud). With this support and adoption, Docker will probably become the most prevalent system used to create cloud applications (Matthias and Kane 2015).

4.4 Summary and Outlook

Digital and real worlds merge. The development of products that are networked within their operational environment and the support of service-oriented business models requires the linkage of traditional product data with the digital shadow of the delivered product configuration (Cyber Twin) as well as the use and association with data of production and operation. A shift from divided designs of physical systems, control subsystems and software architecture to integrated and optimized design can be observed with respect to the process of product and production systems engineering. Concerning operation of production systems, human and information-centric operation moves to highly-automated, autonomous, and coordinated frameworks. Engineers have to be better supported in their development work through system-spanning information links, and assistant functionality that utilizes advanced information analytics and cognitive computing approaches. Thus, IT system strategies supporting operation and product lifecycle information management are changing from centralized to federated, decentralized, and configurable approaches.

Previously, the focus was on the modeling of all necessary artifacts and preparation of all necessary documents for the design and manufacture of a system. In the field of mechanical design, the result virtually consists of a complete digital mock-up on product class level. In the software domain, the result of development activities is a static program code, which e.g. evaluates captured sensor data and system status and possibly performs actuator actions or executes user interaction. With CPPS, modeling of systems in operation as well as the continuous documentation of all MRO operations and changes in the operation is necessary. From the mechanical engineering viewpoint, a functional mock-up is required on product instance level. In the software field, adaptive program code (for example, PLC, CNC) to map self-x functionalities.

In particular, the role of PLM in the context of *Smart Systems, CPPS and Industrie 4.0* approaches will change dramatically towards product information management on a single item instance basis. Today’s PDM systems are complex technical IT systems that require considerable effort for customizing and implementation in specific contexts of manufacturing enterprises. System deployment and operation of a PDM system are complex and costly issues and not only few projects heavily struggle. The networking of products and services on the Internet of Things (IoT) raises the question whether ordinary PLM approaches are not completely overburdened and will become redundant with *Linked Open Data, Big Data or self-learning systems*. PLM approaches have to be adopted in order to support the companies optimally in their *digital transformation* processes.

The benefit of a particular PLM solution heavily depends on the processes to be supported within a company and therefore on the production concept, e.g. ETO or MTO. After sales and customer service play an increasingly important role in the context of PSS. Customized and individualized products have to be produced in smart factories incorporating CPPS approaches with the goal to keep required efforts low. Data from the operation phase of the products has to be linked with the engineering data. Necessary are on the one hand side PLM concepts that support multi-disciplinary development of products with high degree of integrated control technology and software and on the other hand side methods and system functions for cross-domain engineering collaboration. Nonetheless, PDM systems have to be able to manage complex product configurations, including electronics and software and also depict changes in the configuration during operation. Therefore, PLM solutions need to support data structuring approaches that go beyond centrality of the traditional *Bill of Material (BOM)* concept coming predominantly from mechanical engineering focus.

Monolithic approaches with one single leading system for PLM in general do not meet the given demands, particularly because engineering is done to a large extent collaboratively in joint ventures together with development partners embedded in a supply network. A modular IT architecture with best of breed solutions ensures a flexible and user-friendly working environment. It is essential that PLM solutions are dynamically adaptable reflecting the ongoing changes of data models together with the process changes in the organizations. Similar to the way the world-wide web works, federated approaches that link the distributed digital models are required for the product related information management.

References

Ashton, K.: That ‘internet of things’ thing. RFiD J. **22**, 97–114. http://www.rfidjournal.com/articles/view?4986 (2009). Retrieved 25 Jul 2016

Ben Khedher, A., Henry, S., Bouras, A.: Integration between MES and product lifecycle management. In IEEE International Conference on Emerging Technologies and Factory Automation (ETFA’11), pp. 8–13. Toulouse, 2011

Chandrasegaran, S.K., Ramani, K., Sriram, R.D., Horvath, L., Bernard, A., Harik, R.F., Gao, W.: The evolution, challenges, and future of knowledge representation in product design systems. Comput. Aided Des. **2013**, 204–228 (2013)

Charaf, K., Ding, H.: Is overall equipment effectiveness (OEE) universally applicable? The case of Saint-Gobain. Int. J. Econ. Fin. **7**(2), 241–252 (2015)

Cheng, B.H.C., Atlee, J.M.: Research directions in requirements engineering. In: Proceedings of Future of Software Engineering, pp. 285–303. IEEE Computer Society, Washington (2007)

Gerhard, D.: The role of semantic technologies in future PLM. In: Fathi, M. (ed.) Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives, pp. 157–170. Springer, Berlin (2012)

Gerhard, D., Lutz, C.: Rechnerunterstütztes Konfigurieren und Auslegen kundenindividueller Produkte. ZWF Z. wirtschaftlichen Fabrikbetrieb. **3**, 103–104 (2011)

Gerhard, D., Weilguny, L.: Applied feature technology – review of developing a generic solution facilitating data-consistency and enabling knowledge-based engineering. In: CIRP Design Conference 2008, Twente, NL, 2008

Gill, H.: Cyber-physical systems – beyond ES, SNs, and SCADA. In: Presentation in the Trusted Computing in Embedded Systems (TCES) Workshop. http://repository.cmu.edu/cgi/viewcontent.cgi?article=1724&context=sei (2010). Retrieved 25 Jul 2016

Gröger, C., Mitschang, B., Niedermann, F.: Data mining-driven manufacturing process optimisation. In: Proceedings of the World Congress on Engineering, vol. III, pp. 1–7, 2012

Gunes, V., et al.: A survey on concepts, applications, and challenges in cyber-physical systems. KSII Trans. Internet Inf. Syst. **8**(12), (2014)

IEC 62264-3: Enterprise-control system integration – Part 3: activity models of manufacturing operations management. Beuth, Berlin (2014)

INCOSE: International council on systems engineering: systems engineering vision 2020 INCOSE-TP-2004-004-02 ver. 2.03. http://oldsite.incose.org/ProductsPubs/pdf/SEVision2020_20071003_v2_03.pdf (2007). Retrieved 25 Jul 2016

ISO 10303: Industrial Automation Systems and Integration – Product Data Representation and Exchange. International Organization for Standardization (ISO), Geneva (1994)

ISO 14306: Industrial Automation Systems and Integration – JT File Format Specification for 3D Visualization. International Organization for Standardization (ISO), Geneva (2012)

Laney, D.: 3-D data management: controlling data volume, velocity and variety. http://blogs.gartner.com/doug-laney/deja-vvvue-others-claiming-gartners-volume-velocity-variety-construct-for-big-data/ (2001). Retrieved 25 Jul 2016

Lee, J., Bagheri, B., Kao, H.A.: A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. Elsevier Manuf. Lett. **3**, 18–23 (2015)

Matthias, K., Kane, S.P.: Docker Up & Running, Shipping Reliable Containers in Production. O'Reilly Media, Sebastopol, CA (2015)

Monostori, L.: Cyber -physical production systems: roots, expectations and R&D challenges. Variety management in manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems. Proc. CIRP. **17**(2014), 9–13 (2014)

Mouat, A.: Using Docker, Developing and Deploying Software with Containers. O'Reilly Media, Sebastopol, CA (2015)

Nakajima, S.: TPM tenkai, JIPM Tokyo. https://en.wikipedia.org/wiki/Seiichi_Nakajima (1982). Retrieved 25 Jul 2016

Nattermann, R., Anderl, R.: Approach for a data-management-system and a proceeding-model for the development of adaptronic systems. In: Proceedings for the ASME International Mechanical Engineering Congress & Exposition (IMECE), Vancouver, 2010

Newman, S.: Building Microcervices, Designing Fine-Grained Systems, p. 18. O'Reilly Media, Sebastopol, CA (2015)

OSLC4MBSE: Working Group. http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-oslc:oslc4mbse_working_group (2013). Retrieved 25 Jul 2016

Ostad-Ahmad-Ghorabi, H., Rahmani, T., Gerhard, D.: Forecasting environmental profiles in the early stages of product development by using an ontological approach. In: Abramovici, M., Stark, R. (eds.) Smart Product Engineering, pp. 715–724. Springer, Berlin/Heidelberg (2013)

PROSTEP: White Paper Datenmanagement für Smart Systems Engineering (Smart SE). ProSTEP iViP (2015)

Quintana, V., Rivest, L., Pellerin, R., Venne, F., Kheddouci, F.: Will model-based definition replace engineering drawings throughout the product lifecycle? A global perspective from aerospace industry. Comput. Ind. **61**(5), 497–508 (2010)

Schuh, G.: Produktionsplanung und -steuerung Grundlagen, Gestaltung und Konzepte, 3rd edn. Springer, Berlin (2006)

Seibold, Z., Furmanns, K.: Dezentrale Koordinationsmechanismen für Multifunktionalität und Wiederverwendbarkeit. In: Bauernhansl, et al. (eds.) Industrie 4.0 in Produktion, Automatisierung und Logistik, pp. 1–17. Springer, Berlin (2015)

Sethi, A.K., Sethi, S.P.: Flexibility in manufacturing: a survey. Int. J. Flex. Manuf. Syst. **2**, 289–328 (1990)

SYSML: The SysML specification, v 1.0. http://www.sysml.org (2007). Retrieved 25 Jul 2016

Tolio, T., Ceglarek, D., El Maraghy, H.A., Fischer, A., Hu, S.J., Laperrière, L., Newman, S.T., Váncza, J.: SPECIES – co-evolution of products, processes and production systems. CIRP Ann. Manuf. Technol. **59**(2), 672–693 (2010)

Tuegel, E.J., Ingraffea, A.R., Eason, T.G., Spottswood, S.M.: Reengineering aircraft structural life prediction using a digital twin. Int. J. Aerospace Eng. **2011**, Article ID 154798 (2011). http://dx.doi.org/10.1155/2011/154798. https://www.hindawi.com/journals/ijae/2011/154798/

VDI 2206: VDI-Guideline 2206 – design methodology for mechatronic systems. Beuth, Berlin (2004)

VDI 2219: VDI-Guideline 2219 – information technology in product development – introduction and usage of PDM systems. Beuth, Berlin (2002)

VDI 2221: VDI-Guideline 2221 – systematic approach to the development and design of technical systems and products. Beuth, Berlin (1993)

VDI 5600: VDI-Guideline 5600 – manufacturing execution systems (MES). Beuth, Berlin (2007)

Welp, E.G., Labenda, P., Bludau, C.: Usage of ontologies and software agents for knowledge-bases design of mechatronic systems. In: Proceedings of the 16th International Conference on Engineering Design (ICED 07), Paris, 2007