Chapter 7
Product/ion-Aware Analysis of Collaborative Systems Engineering Processes

Lukas Kathrein, Arndt Lüder, Kristof Meixner, Dietmar Winkler, and Stefan Biffl

Abstract Flexible manufacturing systems, as a vision of Industry 4.0, depend on the collaboration of domain experts coming from different engineering disciplines. These experts often depend on (interdisciplinary) results from previous engineering phases and require an explicit representation of knowledge on relationships between products and production systems. However, production systems engineering organizations, which are set in a multidisciplinary environment, rather than focusing on process analysis and improvement options ranging over multiple disciplines, focus mostly on one particular discipline and neglect collaborations between several workgroups. In this chapter, we investigate requirements for the product/ion (i.e., product and production process)-aware analysis of engineering processes to improve the engineering process across workgroups. We, therefore, consider the following three aspects: (1) engineering process analysis methods; (2) artifact and data modeling approaches, from business informatics and from production systems engineering; and (3) persistent representation of product/ion-aware engineering knowledge and data. We extend existing work on business process analysis methods and BPMN 2.0 to address their limited capabilities for product/ion-aware process analysis. We evaluate the resulting contributions in a case study with domain experts from a large production system engineering company. We conclude that an improved

---

L. Kathrein (✉) · K. Meixner · D. Winkler
Christian Doppler Laboratory for Security and Quality Improvement in the Production System Lifecycle (CDL-SQI), Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria
e-mail: lukas.kathrein@tuwien.ac.at; kristof.meixner@tuwien.ac.at; dietmar.winkler@tuwien.ac.at

A. Lüder
Otto-v.-Guericke University/IAF, Magdeburg, Germany
e-mail: arndt.lueder@ovgu.de

S. Biffl
Institute of Information Systems Engineering, Technische Universität Wien, Vienna, Austria
e-mail: stefan.biffl@tuwien.ac.at

S. Biffl et al. (eds.), *Security and Quality in Cyber-Physical Systems Engineering*,
https://doi.org/10.1007/978-3-030-25312-7_7

product/ion-aware knowledge representation facilitates traceable design decisions as foundation for better quality assurance in the engineering process.

Keywords Production systems engineering · Product-production process-production resource (PPR) relationships · Engineering process analysis · Engineering knowledge representation · PPR knowledge persistence requirements

7.1 Introduction

Production system engineering (PSE) organizations pursue the goal of creating (automated) manufacturing systems satisfying the requirements toward time and cost while meeting quality criteria imposed by customers or standards. In addition, PSE organizations need to tailor their solutions for their customers (Wiesner and Thoben 2017). The insufficient representation of important *relationships between the product, the production process, and production resources* (PPR) (Schleipen et al. 2015) in the PSE process can increase the risk of poor quality and unanticipated costs during the operation phase of an automated manufacturing system. Even though PSE organizations build on experienced domain experts, surprisingly, PPR relationships are not explicitly modeled by default throughout the PSE process.

The relationship of product, production process, and production resource can also be expressed in an *information systems engineering* (ISE) or software engineering (SE) context (Humphrey 1995). The product is equivalent to code produced by developers, which can be anything from a small script to an integrated graphical user interface for an application. In SE, it is considered a best practice to test code early with explicit test setups that closely represent the production environment (Beck 2003). (Staging) environments (Humble and Farley 2010) executing the code can thus be seen as the equivalent of a production process, which executes according to the capabilities of a resource. The concept of a production resource can be expressed for example with web servers or interactive development environments (IDE), which are used by a developer producing/executing code as the product. The risk of miscommunication in PSE translates as follows to the software engineering context: If nonfunctional requirements, such as performance or security, are not communicated to the developers, it may be hard or impossible to add these requirements later on to code or production environments. To address these challenges, the ISE and SE communities have developed methods like SCRUM (Schwaber and Beedle 2002), DevOps (Zhu et al. 2016), rapid prototyping, or test-driven development (Beck 2003).

PSE is conducted in a multidisciplinary environment (Biffl et al. 2017; Jäger et al. 2011), involving, above others, disciplines like mechanical, electrical, and software engineering (Moser et al. 2010; Schafer and Wehrheim 2007). Further, PSE is usually more complex than information systems engineering due to risky hardware, which cannot be rapidly tested and has often much longer feedback cycles than software systems. In addition, it is, most of the time, simply not possible to build a whole (physical) test system that reflects the imagined production system. These factors make it harder to engineer and test the target system. Domain experts tend to deal with these challenges by focusing on their discipline-specific contributions, and may consider product or production process aspects only implicitly throughout the engineering process. This domain-centered view often leads to information silos (Rilling et al. 2008), where workgroups do not optimize their interfaces to other engineering experts for collaboration or coordination. The need to collaborate closely in all stages of the development process in a multidisciplinary engineering environment is critical (Paetzold 2017) for project success. Working in silos increases the risks of miscommunication and loss of access to essential knowledge during the PSE process and the operation phase of a production system.

In this chapter, we build on and extend previous research (Kathrein et al. 2018, 2019). We focus on the capability for the analysis and improvement of multidisciplinary engineering processes that exchange knowledge between workgroups. We are interested in the product/ion (i.e., product and production process)-aware analysis of engineering processes as there is significant potential for improvement in the collaboration and coordination of PSE workgroups by considering and explicitly representing PPR knowledge.

Based on the knowledge hierarchy (Rowley 2007), we define the following terms for further use. An *engineering artifact* is a document, in a digital or non-digital form containing data. These artifacts are potentially hard to process for machines and might contain data. The term *data* refers to all kinds of symbols, ranging from simple text to more complex data, like drawings in proprietary software tools. Data has, however, an underlying data model, which is described using datatypes. An example would be a simple table where each column defines the basic datatype, like integer, for the rows, or a graph, defining which objects are nodes and what the semantics, expressed by edges, are (Sabou et al. 2017). *Engineering information* defines the stakeholder groups that have access to the engineering data, how the underlying data can be processed and gives insights into what, who, where, and when questions. Finally, *knowledge* expresses concepts and provides applications of the underlying data and information models. For this chapter we utilize the PPR concept (see Sect. 7.2) to define PPR knowledge. We further define the term *PPR knowledge* to express (a) success-critical attributes, such as parameters for production processes or configurations for production resource and (b) relationships between products, production processes, and production resources, such as constraint dependencies.

We illustrate the PSE process with the simple *use case: fragile product*, as the use case highlights common challenges in the engineering process and the current situation in many engineering organizations. We assume that a customer requires a production system for producing a fragile product. Therefore, the customer creates plans of the product and its characteristics and hands them over to a PSE company. In the PSE company, a *basic planner* receives the product lifecycle documents provided by the customer and specifies the production process and system according to the product requirements. Throughout the engineering tasks, the basic planner transforms product and process knowledge into resource knowledge, resulting in first sketches of the manufacturing system. A team of *detail planners* then takes over and

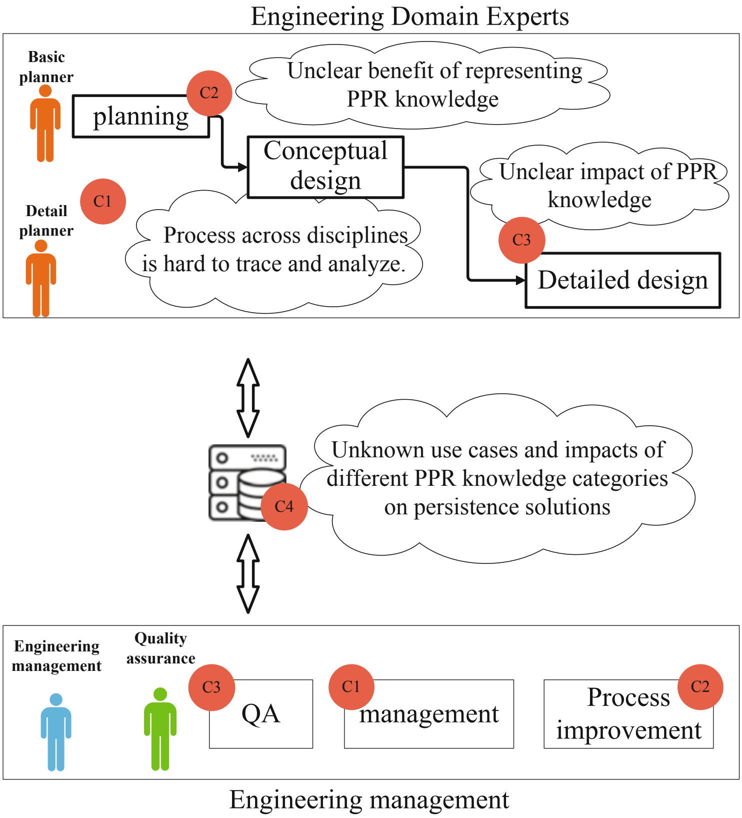

Fig. 7.1 Common challenges in an engineering process

derives discipline-specific detailed plans from the specifications for constructing and operating the production system. This includes a high-throughput transport system, which is required to meet the customer's specifications of parts per minute produced. Unfortunately, during operation of the system, the high acceleration of the transport process damages fragile product parts. This flaw of the production system has many negative effects, such as: extra efforts in rework, uncoordinated communications, and high risk of project failure. These effects all could have been avoided if the missing explicit PPR knowledge on product fragility would have been conveyed in the specifications of the basic planner to the detail planner.

Figure 7.1 illustrates the described engineering process on a high level including the involved stakeholders with their respective challenges. The domain experts for basic and detail planning (orange), represent the operational part of the engineering process, whereas the engineering management with the engineering manager (blue) and quality assurance (green) are more concerned with process planning and improvements.

Figure 7.1 depicts several of the challenges in the use case fragile product, which we describe briefly.

C1. The Engineering Process Between Discipline-Specific Workgroups Is Hard to Trace and Analyze In PSE, workgroups traditionally focus more on intra- than on interprocess improvements. The collaboration of multiple workgroups originates from project needs. Over time, the workgroups may evolve, with new team members joining or team members leaving for another project. Figure 7.1 indicates this through the absence of process/task boundary, which would clearly allow identifying which stakeholder is responsible for which task. There is also no formal process that guides the cooperation or collaboration spanning over multiple disciplines. For the domain experts, this lack of a formal process description makes it hard to trace design decisions throughout the engineering process.

C2. Unclear Benefit of Representing PPR Knowledge Domain experts, who hold a lot of information like the basic planner, are unaware of who would benefit from sharing PPR knowledge. In the described use case this would be the case with the knowledge about the fragility of the product. This knowledge is available to the basic planner through the specifications from the customer. However, the basic planner does not convey this information to the detail planner. In Fig. 7.1, there is no outgoing knowledge from planning into conceptual design. The engineering management again lacks knowledge about the existing knowledge and how it is represented, conveyed, and transformed through the engineering process. This lack of representation makes it also impossible for a quality assurance stakeholder to track or improve engineering artifacts or identify possible reuse scenarios, leading to an improved engineering process.

C3. Unclear Impact of PPR Knowledge Because domain experts do not know what benefit explicit PPR knowledge has (challenge 2), domain experts also do not externalize or document design choices based on product requirements or product design decisions. The product engineer responsible for these decisions simply does not know what impact his decisions might have in the later phases of the engineering of the production system or the operation. In Fig. 7.1, we illustrate this by the two separate "silos" for domain experts and engineering management. The engineering management is not able support the domain experts with this knowledge because they are not aware of project-specific outcomes with possible positive or negative impacts. Explicitly representing PPR knowledge would help both, domain experts and engineering management, to facilitate the analyses of such impacts and highlight dependencies between workgroups that have interfaces for coordination and collaboration. Quality assurance stakeholders have no means on how to improve an engineering process, because they do not know positive or negative impacts that possible new solution approaches might have.

C4. Unclear Use Cases with PPR Knowledge Categories That Require Persistence For software developers, who design and adapt engineering tools for engineering process, it is not clear which are the primary use cases that define requirements for persisting PPR knowledge. Furthermore, it is not clear which categories of PPR data and knowledge exist that may have an impact on the design of data persistence solutions. Addressing the challenges C1–C3 with PPR knowledge representation is not sufficient as the PPR knowledge is not necessarily efficient to search or reuse. For example, engineering managers would require means to query persisting PPR knowledge on project-related information, such as the overall production rate or the percentage of goods with poor quality of projects that include fragile products.

To address challenges C1–C4, we investigate in this chapter a *product/ion-aware engineering process analysis* (PPR EPA) method, based on and extending Kathrein et al. (2018, 2019), resulting in a graphical visualization of the engineering process, classified engineering artifacts and engineering workgroups as a *product/ion-aware data processing map* (PPR DPM). We also investigate use cases to derive requirements for persisting PPR knowledge. The following research questions address these challenges.

RQ1. What Are Main Elements of a PPR EPA Method? To address this research question, we investigate existing EPA solutions and their elements, from both information systems/business informatics and production systems engineering communities. The outcome of this RQ allows identifying buildings blocks for reuse in a new PPR EPA as well as limitations and gaps that a new approach should fill.

RQ2. What Are Main Elements of a PPR DPM Method and Notation? Through applying a PPR EPA, we derive a visualization of the overall engineering process. Because this newly designed artifact is success-critical for the overall application of the PPR EPA, we investigate the main elements that are common, for example, in business process representations from again business informatics and productions systems engineering. In this chapter, we try to close the gap between standard business process representations and extensions that are custom to the PPR DPM approach.

RQ3: What Are Primary Use Cases That Require the Persistence of Different Categories of PPR Knowledge? To address this research question, we build on the use cases coming from RQ1 and RQ2 to elicit primary use cases that stakeholders face in the engineering workflow related to persisting PPR knowledge. The use cases focus on different categories of PPR knowledge present throughout the engineering process and help to define high-level requirements for PPR knowledge persistence.

Main contribution of the conducted research in this chapter allows both ISE and SE as well as PSE communities to gain insights into the other domain. These insights highlight common ground for further research and possible approaches, applicable in both communities, and motivate future research.

The remainder of the chapter is structured as follows: Sect. 7.2 summarizes related work on process analysis approaches, business process notations, and data storage design options. Section 7.3 motivates the research questions and the research approach. Section 7.4 introduces the main elements for the PPR EPA method and PPR DPM artifact, and the treatment designs. Section 7.5 presents the case study conducted with domain experts in a large PSE company. Section 7.6 evaluates the proposed artifacts from RQ1 and RQ2. Motivated by Sects. 7.5 and 7.6, Sect. 7.7

presents PPR knowledge persistence aspects. Section 7.8 discusses the research findings and their limitations and Sect. 7.9 concludes and outlines future work.

7.2 Related Work

This section summarizes related work on product/ion awareness (PPR), on approaches for engineering process analysis, and on notations for representing the analysis results.

7.2.1 Product/ion Awareness in Multidisciplinary Engineering

Technical systems are often distinguished into products and production systems (Biffl et al. 2017). The reason a company exists is often because of its products, that is, products are created in a value-adding process to make profit by selling them (Stark 2015). A production system, however, focuses on creating the products by combining suitable production factors (ElMaraghy 2009). Materials, work-in-progress parts, and production resources (machines) are the most prominent production factors. The product and production system, therefore, have strong dependencies. Schleipen et al. (2015) coined the product-process-resource (PPR) concept for the relationships between products and production systems based on the production process.

This concept of PPR helps to answer questions about the application of engineering data and information and thus, derived from Rowley (2007), is the main building block for the term PPR knowledge used in this chapter.

Figure 7.2 illustrates the relationships between the PPR aspects. We describe the elements of Fig. 7.2 based on the fragile product use case, introduced in Sect. 7.1. The product the customer commissioned contains fragile parts and requires several processes like, gluing, pressing, and transportation. The product has special requirements regarding the transport process, namely, the acceleration of the conveyor belt. Furthermore, the fragile product is processed on an industrial

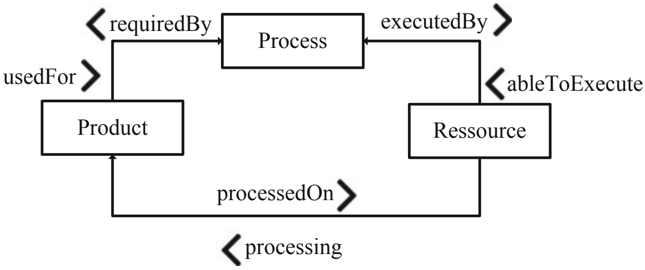

Fig. 7.2 Product-process-resource (PPR) relationships

machine (resource). The link between product and resource has also requirements. For example, the pressing force applied after gluing the fragile parts must range between one and two kilo newton. The resource provides the capabilities that a process needs to be executed with, closing the triangle of Fig. 7.2.

All three concepts can be composed of inner elements, meaning, for example, that a product consists of multiple product parts that are assembled together and make up the final product. This nesting of elements can be described with a pen consisting of the outer shell, the refill, the spring mechanism, and so on. Furthermore, all three concepts of product, process, and resource are interlinked, meaning that they form a graph-like structure, where nodes represent the individual PPR elements and the edges represent links between the individual concepts or hierarchies.

The VDI 3682 standard (VDI 2005) introduces this concept of recursive composition of individual concepts, like the pen example. The standard is further the only visual representation form that has three distinct elements to express, product (parts), processes, and resources.

Other concepts like the ISA95 standard (International Electrotechnical Commission 2003) indirectly allow representing the PPR concept, but are more concerned with describing the interfaces between enterprise resource systems (ERP) and manufacturing execution systems (MES). The goal of the ISA95 standard is to better describe and transfer production order relevant information into the manufacturing system. Furthermore, the standard originates more from batch processing and not so much from discrete manufacturing, which is the primary focus of this chapter. Thus, we do not further consider this option for a solution in this research.

AutomationML (AML) was developed as *glue for seamless automation engineering* (Drath 2009) and uses XML concepts to represent topologies, geometries, as well as behavioral and logical data for production resources. AML became standardized in the open source IEC 62714 standard (International Electrotechnical Commission 2013) and enables representing PPR knowledge as hierarchies with various concepts. Furthermore, AML concepts can be used to model PPR knowledge as a hierarchy of internal elements and linking between the different concepts.

7.2.2 Engineering Process Analysis Methods

To be able to analyze engineering processes and follow the task execution across several workgroups, it is necessary to analyze existing engineering processes on (a) an overview-level of the workgroups and their relationships and (b) detailed analyses of exchanged artifacts and data that identify dependencies between workgroups. These two viewpoints represent the foundation of improving the engineering process between workgroups.

Rosenberger et al. (2018) presents a business process analysis (BPA) method, which determines and defines activities in need of a business context. The presented approach executes a context elicitation, defining contextual functionalities, which in traditional project-based development models is often not done, or simply too much effort. The identified different contexts for various workgroups do not have any implications on other contexts, which makes it hard to use in an engineering process analysis.

To balance exploration and exploitation thinking in a BPA method, Santos and Alves (2017) proposed a three phase BPA, methodologically built on literature surveys, expert opinions, and a case study, all in accordance with the design science cycle from Wieringa (2014). Through the detailed analysis, the results from Santos and Alves allow to identify detailed execution steps, exchanged documents, and a big picture structure of the business process. However, the result does not investigate interfaces between workgroups, as they are predefined and already part of the case study.

Vergidis et al. (2008), who classified several existing business process analysis methods and techniques, highlighted that only a handful of them allows further detailed analysis, or process improvements, which go beyond generic stakeholder, tasks, or input/output artifact identification.

BPA methods allow to easily represent a big picture of a business or engineering process; however, many methods do not consider individual disciplines, interfaces between workgroups, or how the overall collaboration could be improved. The analysis of engineering processes spanning over multiple workgroups requires not only the analysis of the overview on relationships and coexistences of workgroups, but also a more detailed, fine-grained analysis of individual engineering disciplines with specific exchanged artifacts.

On the side of production systems engineering, Jäger et al. (2011) identify the need to "systematically model the engineering workflow, which would allow a deeper knowledge of different engineering aspects and to improve the views of each discipline on the engineering objects." The approach chosen by the authors starts by identifying engineering artifacts and backtracking these artifacts to stakeholders that they belong to. This approach allows the consideration of cause-and-effect analysis in engineering processes, but does not identify interfaces between workgroups and how these could be improved by investigating the engineering artifacts. The process is also driven mainly by engineering documents and not the processes executed by domain experts.

The VDI 3695 standard (VDI 2010) defines the concept of an engineering organization, which conducts its business on a project basis. The engineering organization is further characterized by carrying out the following consecutive engineering activities, depicted in Fig. 7.3: acquisition, planning, realization, commissioning. Such a high-level segmentation of an engineering process does not depict stakeholders, their activities, or artifacts involved. Due to the lack of detail,

Fig. 7.3 Project-related phases identified by the VDI 3695 guideline (VDI 2010)

it is not possible to identify any interfaces that might exist between workgroups and could be the basis for further analyses. The guideline does also not consider how to improve an engineering process but rather gives rough directions that could be taken to improve the overall engineering process.

Lüder et al. (2012) build upon the presented VDI 3695 standard. The outcome of Lüder et al. (2012) is a more detailed engineering process analysis, which focuses on individual workgroups, their tasks, and a description of engineering artifacts, but with no special focus on PPR knowledge representation. In this approach, it is also not considered how multiple workgroups could better work together for an improved coordination and collaboration in the engineering process. Further, Lüder et al. (2018) investigated common challenges regarding the multidisciplinary aspect of a data exchange process across several workgroups. The authors highlight the importance of an engineering process analysis method that allows the investigation of engineering processes with engineering artifacts and possible dependencies.

The analyzed literature reveals similarities in how the analysis methods of business or engineering processes are conducted, but differ in their focus and results. While BPA methods tend to focus more on the big picture, EPA methods focus more on intra workgroup analyses. A gap that can be identified in both disciplines concerns analysis regarding engineering knowledge exchange between workgroups. Exchanges between workgroups are often the source of missing PPR knowledge, a risk already in traditional production systems engineering, much more for considering flexible manufacturing according to the Industry4.0 vision.

7.2.3 PPR Knowledge Representation in Process Analysis

The previously presented BPA and EPA methods gather a lot of data that needs to be processed in some form. Both communities have different approaches to (graphically) represent the knowledge which is present in an engineering process. This knowledge often contains PPR knowledge aspects and thus, the following existing approaches will be investigated according to their possibilities to represent PPR knowledge and classify data and processes.

IDEF0 (Force 1981; Presley and Liles 1995), for example, is widely used in the engineering domain (Zhang et al. 2010) and provides an overview on processes, their inputs/outputs, controls, and stakeholders. The system analysis standard has only very few distinct elements, namely, arrows and boxes. This limited number of different concepts makes it easy for nonexperts to pick up the modeling approach, but makes it hard to express more complex situations, which would require a richer expression language. For example, is it hard to follow one specific input to output transformation through a large IDEF0 model, because possible other input and output arrows are indistinguishable from each other.

Lüder et al. (2012) introduced a more detailed but not so visual approach, by representing gathered engineering knowledge in tables. This approach allows for a very detailed classification and division of knowledge, but does however become cumbersome to work with when the number of different tables, referencing each other, increases.

Event-driven process chains (EPCs) (Scheer 1998), BPMN 2.0 (Allweyer 2016), or the UML standard (Fowler et al. 2004) are all well-known options to model business processes. Merunka (2017) pointed out that the UML standard has no means to represent product and process knowledge in either one or several combined diagrams. EPCs, extended with data, resources, time, and probabilities are called extended EPCs (eEPC) (Scheer 1998). Both eEPC and BPMN 2.0 are widely used for modeling business processes and have incorporated many similar concepts. Extended EPCs require a more explicit annotation of organizational units for each engineering task, while BPMN 2.0 uses swim lanes for a more compact visualization.

Khabbazi et al. (2013), Huang et al. (2017), and Merunka (2017) proposed the combination of multiple modeling concepts, which should allow overcoming limitations that individual notations have. Even though such a combination allows for a more flexible and detailed notion of processes, the complexity of models also increases for stakeholders, who would like to analyze the underlying models. None of the mentioned authors named the concept of explicitly modeling data and process flows; we use in this chapter the term *data processing map* to express the combined representation of processes with documents.

Unfortunately, PPR knowledge, its flow through an engineering process, or dependencies between tasks and artifacts are not directly expressible in any of the languages discussed in this subsection. The languages do however build a good foundation for closing this gap, by using f. e. BPMN 2.0 and then build custom extensions to express PPR knowledge.

7.2.4 PPR Knowledge Persistence

In this chapter, we use the term PPR knowledge for success-critical attributions, like parameter settings of production resources, of each of the concepts as well as the interrelationships between the individual parts of PPR based on Schleipen et al. (2015). These attributions for product, processes, and resources in combination with the relationships formed between the three concepts need to be represented to allow persistence and retrieval.

We further use the term persistence not as strictly defined as it is in the database community, but we express with it the application of persistence solutions to store PPR knowledge. This can include several different underlying technologies. A designer of persistent PPR knowledge storage should consider established persistence approaches, such as relational databases, NoSQL databases, and AutomationML files, as these fit well to general characteristics of PPR, which essentially are graphs consisting of linked trees in the individual PPR aspects as described in Sect. 7.2.1.

Relational databases have been successfully applied to for persisting business data since the 1970s and gained considerable production experience (Nance et al. 2013). The approach that centers on tables, columns, and rows has been a clear choice for many data-intensive storage and retrieval applications (Vicknair et al. 2010). Relational databases are in general very efficient unless the data is strongly interlinked with many relationships leading to a large number of joins (Vicknair et al. 2010) that reduce access efficiency. A key success factor for relational databases is the fixed structure of each table, which allows for indexing and for using the goal-oriented query language SQL (Date and Darwen 1997). Unfortunately, engineering artifacts often do not follow a predefined fixed structure and may vary from project to project, or depend on customer-specific practices.

NoSQL technologies address this limitation using flexible data models to store schema-less models (Siddiqa et al. 2017). PPR knowledge accumulates in an engineering process and expresses product, process, and resource information as well as the interrelationships in a high number of many-to-many relationships and is to some extent hierarchically structured, which fits NoSQL characteristics presented by Vicknair et al. (2010). Therefore, the available knowledge may also vary depending on project or customer, and thus requires a flexible schema, which is easily changeable, adaptable, and maintainable. NoSQL is not a single solution, but has four major design differentiations to consider for designing an application. These options are key-value, column-oriented, document, or graph databases (Siddiqa et al. 2017). PPR knowledge with its attributions and relationships fits could fit well to a graph-based approach (Vicknair et al. 2010).

Fowler and Sadalage (2013) coined the term polyglot persistence, for using several data storage languages and technologies, each for the use cases it fits best. Nance et al. (2013) pointed out that it is not necessary to make a choice between relational or NoSQL databases but to use both as is seen appropriate. A polyglot data storage approach could help to overcome the requirements of engineering artifact storage by following a “best-of-breed” approach. The solution of polyglot storage requires expertise in several languages and technologies, making the design more complex to understand, implement, test, and operate. Therefore, a key question is what requirements can be derived from use cases and how a sufficiently powerful yet simple design for PPR knowledge persistence might look like.

AutomationML (AML) does not only provide means to express PPR concepts, but also allows representing production systems in XML-like formats. Furthermore, is it possible to represent PPR knowledge for data exchange and logistics storage in AML for small production systems. However, AML files can rapidly grow in size, which may be hard to process efficiently even for medium-sized production systems. Production systems with 5000–10,000 signals may take up 20–50 MB of AML text for its representation, depending on the set of discipline-specific views in the data model.

7.3 Research Questions

By following the design science cycle presented from Wieringa (2014), we address the challenges introduced in Sect. 7.1 by deriving the following research questions for improving the product/ion (i.e., product and production process)-aware analysis of engineering processes.

RQ1. What Are Main Elements of a PPR EPA Method? To address this research question, we build on Kathrein et al. (2018, 2019) and consider the strengths and limitations of approaches from business process analysis and from engineering process analysis to identify promising candidate methods for adaptation and extension. We extend Kathrein et al. (2019) with valuable lessons learned regarding the PPR EPA. We apply a case study design (Runeson and Höst 2009) to elicit what main elements a PPR EPA method needs. These elements need to focus on the design and elicitation of a product/ion-aware engineering process analysis (PPR EPA) method and thus make it possible to identify and collect data on the engineering process. Through focusing on PPR knowledge expression, the EPA method allows to analyze where relevant PPR knowledge is required, created, or lost. From the main elements identified, we derive requirements for a notation to represent the needs and capabilities to represent PPR knowledge.

RQ2. What Are Main Elements of a PPR DPM Method and Notation? Based on Kathrein et al. (2018, 2019), we describe how a product/ion-aware *data processing map* (PPR DPM) can look like. The extended elements serve as foundation for the analysis of gaps regarding PPR knowledge representation in the engineering process. The result of RQ2 highlights elements that are crucial to be able to express in PPR knowledge in an engineering process with the interaction of tasks and engineering artifacts. We follow the design science cycle (Wieringa 2014) and validate both treatments of RQ1 (PPR EPA) and RQ2 (PPR DPM) artifacts, in the context of a case study.

RQ3: What Are Primary Use Cases That Require the Persistence of Different Categories of PPR Knowledge? We use the case study approach from the work of Runeson and Höst (2009) to also investigate common use cases that occur in the engineering workflow and further expand the stakeholders to include software engineering domain experts. These experts, in combination with interviews from RQ1, help to elicit the primary use cases, allowing to derive requirements and different categories of PPR knowledge. The outcome of this RQ allows a three-tier layering of (1) use cases, (2) functions like reuse and search, and (3) persistence technologies like databases. From such a layered outcome, future research and possible new stakeholders can focus on representing PPR knowledge more permanently and make it query-able.

7.4 Product/ion-Aware Analysis of Engineering Processes

This section addresses the limitations of both business process analysis (BPA) methods, such as *context-aware process analysis* and *A2BP* (Rosenberger et al. 2018; Santos and Alves 2017) and engineering process analysis (EPA) methods, such as *mechatronic engineering EPA* and *technical dependency mining* (Lüder et al. 2012; Jäger et al. 2011). We introduce the main elements of a multidisciplinary PPR EPA method (RQ1) as well as the main notation elements of a PPR DPM (RQ2). The goal of the PPR EPA is to focus on product/ion-awareness and have a repeatable process resulting in a PPR DPM. Paetzold (2017) identified the need for a clear and standardized design process, which is connected to the development process and allows efficient and effective work execution. We present in Sect. 7.4.1 requirements for an artifact evaluation, in Sect. 7.4.2 the design of the treatment PPR EPA method, and in Sect. 7.4.3 the design of the treatment PPR DPM artifact proposing an extension of BPMN 2.0 with PPR knowledge elements.

7.4.1 Requirements for PPR Engineering Process Analysis

Following Wieringa (2014) through the design science cycle, this section presents contribution arguments for the PPR engineering process analysis (PPR EPA) and for the PPR data processing map (PPR DPM). A contribution argument is: “an argument, that an artifact, that satisfies the requirements, would contribute to a stakeholder goal in the problem context” (Wieringa 2014). In our case we present the following two sets of requirements, based on Kathrein et al. (2018, 2019), that have been derived from use cases with the involved stakeholders in the case study. The first set of requirements addresses RQ1, the PPR EPA, while the second set focuses on RQ2, the PPR DPM. The requirements are strongly driven by the goal of representing PPR knowledge and are suitable for multidisciplinary PSE organizations and follow the PSE phases basic planning, detail planning, and operation.

RQ1: Main Elements of a PPR EPA To identify the main elements needed for a good solution of a PPR EPA, we present requirements for capabilities of the product/ion-aware PPR engineering process analysis (PPR EPA).

Identification of PPR Knowledge The product/ion-aware PPR engineering process analysis should allow identifying PPR engineering knowledge, for example, product knowledge in initial product drawings coming from the customer, process knowledge conveyed through specifications regarding the transport system.

Process Analysis with PPR Knowledge The PPR EPA method should analyze and focus on: the creation of PPR knowledge in an engineering process, the flow of PPR knowledge through the engineering process, and an indication where relevant PPR knowledge may not be carried on. One example path could look like this: First, production process sequences are created based on process knowledge. Second, a layout for the production system is created with the help of resource knowledge. The process knowledge is not carried on from the first to the second step. Lastly, in step three an offer is submitted to the customer, only conveying resource knowledge.

Identification of PPR Knowledge in Interdisciplinary Interactions The PPR EPA method should allow identifying where engineering disciplines interact with each other, for example, handover phases of project responsibility including artifacts, such as the change from basic to detailed planning where all artifacts are handed over to a new team.

RQ2 Main Elements of a PPR DPM The following set of requirements is motivated by how to represent PPR knowledge in an engineering process after the PPR EPA has been conducted, and what main elements of a PPR DPM visual representation is needed.

PPR-Specific Visual Elements The PPR DPM should provide specific elements for the concepts used in the PPR EPA, including visual elements for roles, tasks, the priority a task has regarding PPR knowledge, artifacts, and the PPR knowledge aspects they contain.

Iterative Refinement It should be possible with the PPR DPM to start with small initial models, only representing the most vital engineering process tasks per discipline, and gradually and iteratively expand the models. With each iteration, the context for collecting more detailed workflows can be expanded and refinements of PPR knowledge classifications of the process steps with stakeholders can be executed.

Process Overview The PPR DPM should provide an overview of the engineering process, including: the involved disciplines with their respective process executions, engineering artifacts and their flow throughout the process, interfaces between workgroups and the sequence that engineering tasks are executed in.

7.4.2 A Product/ion-Aware Engineering Process Analysis Method

To address RQ1, and the limitations of existing business process analysis (BPA) and engineering process analysis (EPA) methods, we identify in this subsection the main elements for a multidisciplinary engineering analysis (PPR EPA). Our approach represents a repeatable two-step process (see Fig. 7.4), resulting in a visual product/ion-aware data processing map (PPR DPM).

Figure 7.4 provides an overview on the steps and tasks of the PPR EPA method. We revisited the proposed PPR EPA from Kathrein et al. (2019) and now present a more detailed description regarding the PPR EPA including some lessons learned. The involved stakeholders are *engineering domain experts* (orange), *engineering management* (blue), *quality assurance* (green), and the new role *EPA facilitator* (red).

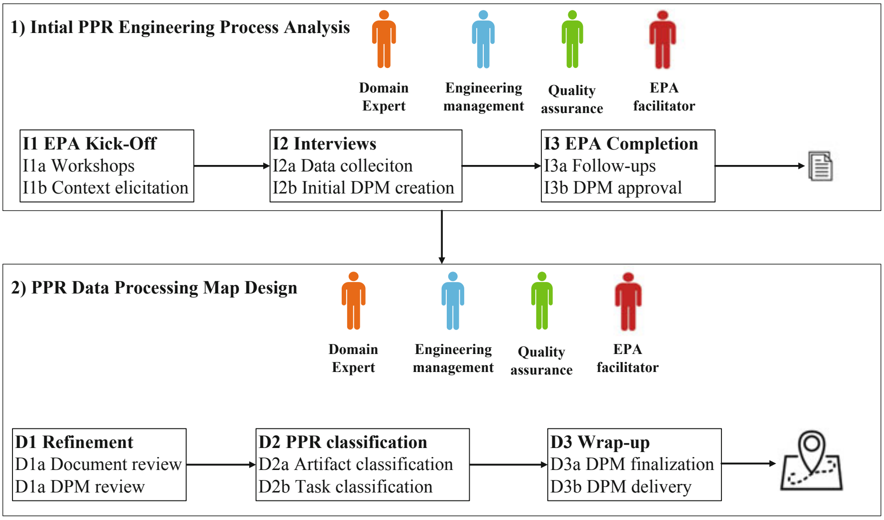

Fig. 7.4 PPR EPA method elements/phases/tasks, based on Kathrein et al. (2018)

The newly introduced role of the EPA facilitator conducts interviews with domain experts and stakeholders, creates initial models for a possible PPR DPM for grading with the domain experts, and holds workshops. This role is similar to the model integrator presented in Fay et al. (2018). All other stakeholders provide insights into their work and are driven to improve the engineering process and optimize existing potential like manual reworks of engineering artifacts due to proprietary engineering tool data formats. The individual tasks of the two phases will be described presently. All tasks prefixed with an "I," represent tasks from the initial PPR EPA phase, and tasks with the "D" prefix correspond to design tasks of the PPR EPA focusing on the PPR DPM.

Phase 1. Initial PPR Engineering Process Analysis starts with initial knowledge about the project under investigation. Outcomes of this phase are interview documentation as notes and audio recording, exemplary files for engineering artifacts, and an initial data processing map depicting a first high-level engineering process.

EPA1 EPA Kick-Off

I1a Workshops. All stakeholders take parts in one or several workshops, stating their role and position that they will play in the PPR EPA.

I1b Context Elicitation. During workshops stakeholders and researchers outline the context of the engineering process under investigation.

Outcome of I1 are documents describing the context, goals, requirements regarding the PPR EPA and PPR DPM and first (hand-drawn) sketches of a DPM.

EPAI2 Interviews

I2a Data Collection. Holding interviews with domain experts allows collecting representative data that is used in a typical engineering project. All captured data should be relevant and put in context to which domain expert and specific task they belong.

I2b Initial DPM Creation. Researchers acting as EPA facilitators elicit PPR knowledge from the domain experts and use this knowledge for an initial PPR classification of engineering artifacts, which results in a first initial DPM.

Outcome of I2 are detailed interview notes and recordings, as well as the initial DPM as basis for further detailing. In regard to Kathrein et al. (2019), we revised the interview task to also contain the initial DPM creation, which allows for a more timely early draft version of a DPM; it is important to not let too much time go by between data collection and initial DPM creation.

EPAI3 EPA Completion

I3a Follow-Ups. The initial DPMN is reassessed, and possible open questions can be discussed with the domain experts. This step is especially important, because it is not guaranteed that the same domain experts will be available in later phases.

I3b DPM Approval. By revisiting domain experts, the modeled initial DPM is either approved or modified to express the engineering process. We propose this additional step as a lesson learned from Kathrein et al. (2019). An early initial approval with the domain experts makes clear that the ground truth for any further work is set and will not be changed.

Outcome of this step is the final basic version of the DPM, representing the basis for further refinements.

Phase 2. PPR Data Processing Map Design is concerned with refining the existing data processing map, and classifying all gathered input data according to PPR and detailing the engineering process model.

DPM1 Refinement

D1a Document Review. All internal data objects (like interview notes) and external data (like engineering artifacts) are investigated more closely and described for following PPR classifications.

D1b DPM Review. The existing basic model is reviewed, potential gaps, notation mistakes and to coarse or detailed tasks are identified and then modeled to represent the as-is engineering process, with references to documents, as closely as possible.

Outcome is a more detailed DPM, identifying engineering artifacts and a data catalogue for easier lookup of exemplary artifacts and data.

DPM2 PPR Classification

D2a Artifact Classification. With the input from F1 Refinement, all engineering artifacts are classified according to product, process, or resource (PPR) knowledge.

D2b Task Classification. All tasks that are present in the PPR DPM are classified regarding their need for PPR knowledge. If so, it further classifies how important PPR knowledge is for a successful execution of the task, including an indication which aspect of PPR is currently available and what additional information would improve the engineering task.

The outcome of this step is the classified DPM, according to PPR.

DPM3 Wrap-Up

D3a DPM Finalization. The PPR DPM is reviewed and all EPA facilitators have a last chance to make small changes to the artifact.

D3b DPM Delivery. The final version is presented to the stakeholders and domain experts and delivered to them for further use.

Outcome is the PPR DPM and all documentation that was accumulated over the course of the PPR EPA.

7.4.3 A Product/ion-Aware Data Processing Map Notation

To address RQ2, and be able to express the knowledge gathered from Sect. 7.4.2, the PPR EPA, we explored business and engineering process analysis notations like UML, BPMN 2.0, or eEPC. In Kathrein et al. (2019), we presented an extension to the BPMN 2.0 standard, which we apply in Fig. 7.5. BPMN 2.0 was chosen because it has already many elements needed to represent business or engineering processes, like events, tasks, documents, gateways. BPMN 2.0 is also a bit “cleaner” than EPCs as it does not require annotating each task with an organizational unit but provides swim lanes to express workgroups.

Our extensions allow to label document content regarding product (P), process (P′), or resource (R) knowledge, as well as to indicate the importance a task has

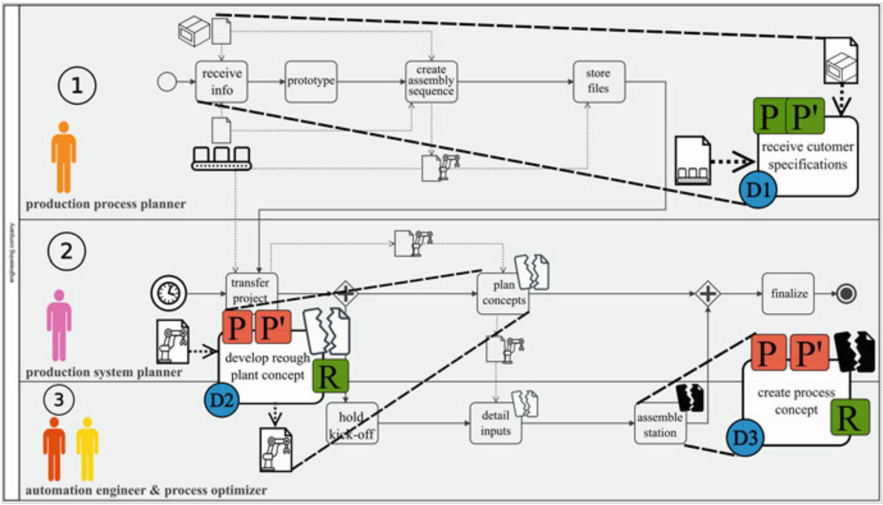

Fig. 7.5 Product/ion-aware PPR Data Processing Map, based on Kathrein et al. (2018)

regarding PPR knowledge. In Fig. 7.5, tags D1 and D2 highlight the use of such a classification. In D1, artifacts containing product (coming from the top) and process (coming from left) information are depicted. In D2, a resource-centered artifact serves as input and another resource containing artifact is created. The individual documents are also graphically distinguishable through annotations in the middle: a package for a product (tag D1), conveyor belt for a process (tag D1), and a robot arm for a resource (D2). This addition to the BPMN 2.0 standard builds the foundation for describing and analyzing a PPR knowledge flow through the engineering process. From this extension possible analyses can be derived, such as where PPR knowledge is created, transformed, or lost.

Further, we provide *PPR knowledge requirements*. These requirements are expressed by (a) annotations of P, P′, and R surrounding the task outline (see Fig. 7.5 tags D1, D2, and D3), and (b) white/black broken documents, if the task misses at least one of the PPR aspects (see Fig. 7.5 tags D2 and D3). The annotations of P, P′, and R indicate what information the task currently receives (colored in green) and what information would additionally be needed but is missing (colored in red). In Fig. 7.5, tag D1, for example, requires and receives product and process information; in tags D2 and D3, the same information and resource information is needed, but only resource information is received. This leads to the red coloring of the product and process annotations. The white broken document highlighted in Tag D2 indicates that for a task execution it is important to receive PPR knowledge; however, the execution is not hindered if this knowledge is not present. This annotation allows indicating which tasks could be executed more efficiently or with better quality if additional PPR aspects, like parameter settings, were present. However, the knowledge can be derived, even if this is not time-efficient. Black broken documents, such as in tag D3, indicate that the role cannot execute this task properly if PPR knowledge is absent. It is absolutely crucial for the task execution to have PPR knowledge present or otherwise will run into efficiency, quality, or cost issues. In a situation where PPR knowledge is crucial, it is not possible for the domain expert to derive this knowledge, make assumptions about settings, or start a communication process.

We evaluate the proposed extensions for the PPR DPM notation, with a case study conducting the proposed PPR EPA (see Sect. 7.5).

7.5 Case Study

We conducted a case study following Runeson and Höst (2009) to evaluate the proposed approaches PPR EPA (RQ1) and the PPR DPM (RQ2). Researchers took the role of the EPA facilitator, which is described in Sect. 7.4.2. The EPA facilitator followed the proposed PPR EPA, executing each task with domain experts. We collected data on the existing engineering process as well as representations of PPR knowledge in the current setting. All domain experts voiced their needs regarding the PPR EPA and how the PPR DPM should look like to better support their work packages.

Study Subject The case study on the proposed *engineering process analysis* (EPA) method was conducted with domain experts at a large production system engineering and manufacturing company. The company focuses on discrete manufacturing systems and can be seen as representative for systems engineering enterprises that conduct their business on a project basis. The company did not consider PPR knowledge at the point of the case study. The case study for collecting data on the PPR EPA method and on the PPR DPM notation spanned over nearly 2months from the initial kick-off to the final version of the data processing map and the final feedback from the involved stakeholders. In the case study, six domain experts, five stakeholders for the engineering process, and three software engineering stakeholders were interviewed. This allowed us to execute the PPR EPA and model the PPR DPM, as well as gather input for data storage requirements, which will be presented in Sect. 7.6. In the context of this case study, one project, focusing on one manufacturing system, was investigated. This means, that the production system and all engineering processes focused on one product, with a set of processes and adequate selected resources for the execution.

Study Execution We followed the PPR EPA approach presented in Sect. 7.4.2 by starting with a project kick-off, consisting of workshops that helped elicit the context. This first step allowed the company stakeholders to introduce their work field context, context, and current problems to the three researchers, who took on the role of the EPA facilitator.

Following the kick-off, each domain expert and stakeholder was interviewed separately for 2 h. The interviews followed a funnel approach (Runeson and Höst 2009), meaning that the question started broad, for example, regarding context and general responsibilities, and became more detailed later, concerning individual work aspects.

Breaks after the interviews allowed creating the initial DPM (Step I2b in the PPR EPA), and collecting feedback from the domain experts. On a separate day, the team completed the EPA with follow-ups, a small presentation of the DPM model, and a check if all needed exemplary documents were given to the researchers for phase 2, the design of the PPR DPM.

All gathered information was reexamined, reviewed, and ordered for easier retrieval. The gathered artifacts were carefully classified regarding the information on the product, process, or resource; an example can be seen in Table 7.1.

The classification builds on a mapping proposed by Hundt and Lüder (2012), who map between different engineering phases and engineering artifacts, such as electrical or mechanical plans, which are present in the detailed engineering phase. In addition, we reexamined the identified engineering tasks and expressed their requirements for PPR knowledge as *no need*, *important need*, or *crucial need*. Figure 7.5 illustrates a representative part of the final version of the PPR DPM.

Table 7.1 Classification of engineering artifacts and PPR knowledge, based on Kathrein et al. (2018)

PPR EPA concept	Collected data
Stakeholder	Domain expert engineering
Process step number	1
Process step name	Receive customer product life cycle management documents
Input artifact name	Product variations
Description	The artifact provides a mapping of which individual parts are used in which product families and created on which part of the production resource. The knowledge is usually stored in an excel document
Product relevant knowledge:	Individual parts used in the product Mapping from part to product family Product name given by the customer Identification numbers from the customer for the individual parts
Relevant process knowledge	None
Relevant resource knowledge	The mapping between which part is created, or processed on which resource part
Output artifact name:	No output artifact is created

The *production process planner* (light orange and swim lane number one), starts each individual project. He receives product and process information from the customer, presented in detail tag *D1*. From the product and process information he is the one to create first new resource knowledge and convey this to the next role. The problem here is that the product and process information is not transported alongside the resource knowledge.

The second stakeholder, the *production system planner* (purple, swim lane number two), receives the resource knowledge and holds an internal kick-off meeting for all other involved workgroups (indicated by the clock symbol). Tag *D2* depicts that for the development of rough plant concepts the production process planner needs PPR knowledge, but only receives the R part.

In swim lane number three, the *automation engineer* (dark orange) and the *production process optimizer* (yellow), work in parallel. Each domain expert delivers a more detailed view regarding the system under construction. For the creation of process concepts, tag *D3*, the workgroups are in need of PPR knowledge but again only receive the R part. For the domain experts it is crucial to receive all possible knowledge and through manual uncoordinated communication with other domain experts, the automation engineer and production process optimizer try to get hold of additional information. The execution of this task is thus highly risky, due to missing PP knowledge, and can lead to unsupported decision making and in later phases to bad quality.

7.6 Evaluation of PPR EPA Visualizations

This section reports on a comparison between the outcomes of different data processing map notations in an initial feasibility case study (Runeson and Höst 2009) with domain experts at a large multidisciplinary systems engineering company.

We evaluate in this section (a) the visualization of engineering processes currently used at the company, discipline-specific EPC workflows, (b) a standard BPMN 2.0 model, and (c) in Sect. 7.4.3, the proposed PPR extensions to the BPMN 2.0 standard.

The evaluation was conducted in an engineering company that creates custom, project-based, automation systems. We conducted interviews with the engineering manager as well as involved domain experts that gave feedback for the parts that were relevant for them. All interviewees could rate the approaches regarding usability, usefulness, and effort based on a 3-point Likert scale (+, 0, −). "+" indicates fulfillment of the criterion, "0" represents neutral fulfillment of the criterion and "−" indicates disagreement that the approach fulfills the criterion (Table 7.2).

The current approach at the company, using EPC workflow diagrams in selected workgroups, is not very usable due to a high level of detail, and changes always imply high rework efforts. The approach is only useful to a limited number of people conducting intra process optimizations.

A standard BPMN 2.0 model was rated usable because it is easy to understand and has concepts like tasks, swim lanes, and documents. The overall creation and adaptation effort was rated adequate as well. However, the standard BPMN 2.0 model is not useful for any PPR-related analyses, due to missing classifications regarding engineering artifacts.

The last approach, the product/ion-aware BPMN 2.0 model, was rated overall very positive. It is as useful as the standard version of BPMN 2.0, but has a much higher usefulness due to the classification of PPR knowledge in engineering artifacts. This classification has a minor drawback and needs a bit more effort to work with than for example the standard BPMN 2.0 model.

The case study results reveal that our proposed approach of extending a well-known standard, in this case BPMN 2.0, allows breaking out of the existing "information silos" that exist in the engineering company. Also, it is much simpler and more useful to classify engineering artifacts regarding PPR knowledge and use these insights. We also learned from the case study and the evaluation that it is a

Table 7.2 Evaluation results, based on Kathrein et al. (2018)

Approaches—>criteria	Current DPM approach: Discipline-specific EPC workflows	Standard BPMN 2.0 model	Product/ion-aware BMPN 2.0 model
Usability	−	+	+
Usefulness	0	−	+
Effort	−	+	0
Overall DPM quality	−	0	+

good first step to represent PPR knowledge explicitly in the form of a PPR DPM, but that it is also vital to investigate possible PPR knowledge persistence solutions. For the involved domain experts, it is not enough to exchange PPR artifacts but they have the need to query and reuse PPR knowledge currently represented in the artifacts. This need is based on use cases that occur in the engineering process and are drivers for further research. In Sect. 7.7, we introduce primary use cases that are relevant for PPR knowledge persistence.

7.7 PPR Knowledge Persistence Use Cases and Data Categories

To address RQ3, we built on the case study presented in Sect. 7.5 to gain insights into the current persistent representation of engineering knowledge. We interviewed three team leaders of software engineering projects responsible for the development of engineering tools, for production machine programming, and for data mining.

PPR Persistence Use Cases The following use cases describe and motivate requirements of software systems that use the PPR knowledge persistence system as foundation for deriving technology requirements.

UC1 Product/ion-Aware Engineering Tool Support

Advanced engineering tool functions based on PPR knowledge, such as checking whether the characteristics of a production process fit the characteristics of the product to be produced, require a *programmable interface* to PPR knowledge. The stakeholders in the engineering process phases have both common and different needs.

UC1a. Basic Engineering. For designing the production process, the basic engineer requires the definition and access to mapping of product parts to process steps characteristics, which are currently stored in excel tables providing only poor possibilities to execute this task. For identifying a set of useful resources for a specified product feature, the basic engineer requires the access to mapping of product features to production resource characteristics. For finding and comparing promising production process variants, the basic engineer requires the capability to discern between the desired process (customer requirements or product manager of a family of similar systems) and the possible process variants (a) derived from a product specification or (b) derived from the set of resource components and their combinations. For reusing PPR knowledge in a family of products or production systems, the basic engineer requires the capability for variant management in a PPR context.

UC1b. Detail Engineering. For designing a production system from an early rough sketch to a detailed construction plan, the detail engineer requires the capability to define and enhance the design of a resource from the viewpoint of one discipline and describe design dependencies across disciplines, for example, for machine configurations, which could be stored again in excel files or relational databases. For designing a production system part from reusable components, the detail engineer requires the capability to discern between information on a specific product and on a library of products and resources with detailed information on product and resource types, for example, a tree of motors, electrical motors, and specific motor types and instances. In a PPR context, this resource-specific view shall be linked to product/ion-relevant characteristics. For validating his design decisions, the detail engineer requires traceability of design decisions back to basic engineering by mapping the configuration of the production system parts back to parameters of the product to be produced and the planned production process.

UC2 PPR-Based Run-Time Data Analysis

UC2a Run-Time Process Data Analysis. For comparing the intended (specified) production process to the actual operation process, the production process optimizer requires capabilities for defining and comparing planned and actual production processes. To do this, operational data logs of the resource are needed as well as test data and if possible simulation results.

UC2b Run-Time Data Mining. For better understanding the impact of engineering and operation factors on the production process results, the production process optimizer requires capabilities for data integration and aggregation of production operation data with engineering data. This requirement is based on improvements for (a) the production process and (b) the capabilities of the production system family. For data integration, the production process optimizer requires capabilities for linking operation data to engineering data, for example, mapping of identifiers in data sets coming from a variety of sources like configuration files, operational data, and planned layouts from basic engineering.

PPR Data Category Characteristics The current technology landscape of the company consists of several in-house development tools used in the engineering process and of applications for configuring and analyzing the operation of manufacturing systems. These tools are only focused on expressing resource knowledge, neglecting the potential that a full PPR knowledge base could have. PPR knowledge could be used for expressing (a) success-critical attributes, such as parameters for production processes or configurations for production resource and (b) relationships, such as constraint dependencies, between products, production processes, and production resources. The three major groups identified with the domain experts currently in use areas follows:

1. *Engineering data* is all data that is created during the engineering process, for example, for designing a robot work cell, ranging from engineering artifacts, such as CAD drawings, to data tables, such as Excel files, hierarchically structured product parts, and PPR knowledge, such mappings between processes and resources in the robot work cell. Engineering data structures may differ from project to project and consist most of the time of complex engineering artifacts, objects with attributes, or graphs.

2. *Configuration data* includes data that describes the resource (machinery), such as relationships between production components or configurations or parameter settings for machines and devices. This data can be described and stored in classical table structures, consisting of many primitive values, like integers and strings. Configuration data schemas are rather stable; challenges come from keeping track of the semantics of changes in versions that may differ only in numerical/textual changes and linking these configuration values to outcomes in run-time data files.
3. *Run-time data* consists of all data accumulated during the operation of the manufacturing system. Analyses, logs, quality measurements, and so forth are all representatives of run-time data as foundation for data mining. Run-time data can be characterized as time series data, which is written once and read many times. The underlying schema may change with every new quality metric or sensor added, making it challenging to keep track of the semantics of the collected data.

Although these data categories have very different characteristics, they are often stored in a large relational database, which introduces challenges regarding technical debt, understandability, performance, and maintainability of data definition and access. Through mapping the different characteristics of these data categories into one shared schema many PPR knowledge aspects, like relationships between the individual concepts, might be lost, for example, if there is only a focus on configuration data for resources, there might be no concept for storing process or product-relevant data.

PPR Persistence Requirements From the discussion of these use cases with the software domain experts, we derive the following major requirements for PPR persistence design.

Data Representation for the Different PPR Knowledge Groups UC1 and UC2 target different phases of an engineering process. UC1 focuses on the early engineering phases where the planning and creation of PPR knowledge is the main objective. In these phases, a lot of the configuration data is initially created to be then detailed in later phases. UC2 aims at the run-time perspective of an engineering system, where large amounts of quality data in different forms are accumulated. Due to these different foci of the use cases, it is a requirement for a PPR persistent solution to be able to handle different data groups and their characteristics like fixed schema tables, graphs expressing relationships between PPR concepts, and time series consisting of quality metrics measured by the production system.

Programmable Interface A PPR persistence solution consisting of many different data aspects and data groups has a high potential for reuse, spanning over different disciplines and engineering phases. To avoid the accumulation of technical debt, a PPR persistence solution requires a programmable interface, an *API to the PPR knowledge base*. This API should represent the only entry point for accessing PPR knowledge and possible metadata representations, like for example who or what tool changed which part of the PPR knowledge representation. This requirement is based on the different existing tools present in an engineering company, which all support their individual specialized use case like in UC1, basic versus detail planning, resulting in different engineering artifacts.

Flexibility Derived from the two previous use cases and the different requirements of the data groups is flexibility, also a requirement for a PPR persistence solution. For example, UC1 provides two different views regarding PPR knowledge. In basic planning, stakeholders plan a production process and design the resources. Following this phase, detail planning is interested in the actual and more detailed process and the concrete realization of the design. These two use cases might have different requirements for a PPR knowledge persistence solution, requiring *flexibility* and easy to maintain data model implementations. UC2 also motivates this requirement, because the use case is interested in how the production system performs and how possible optimizations might look like, requiring adaptations to existing solutions and their persistence.

Usability and Usefulness A possible new solution should provide *usability* for the developers that need to work with the new technology and should also be *useful* and provide reusability in similar but different projects. As already identified, the mapping of different data groups into one technical solution may lead to high technical debt; also, this approach does impose many restrictions onto the developers that are responsible for the development of engineering tools. These restrictions can be seen currently in high development cycles and nearly unusable solutions, where even custom-made software leads to a vendor lock-in, making it virtually impossible to adapt a solution. Also these solutions do not provide any reusability in different projects. A new solution thus should focus beyond the PPR knowledge representation on providing useable and useful concepts for domain experts responsible for the technical implementation and maintenance.

Performance The presented use cases derived from UC2 focus on data mining and process data analysis. These use cases impose with increasing data sizes requirements regarding the performance. Performance can be expressed in the time period needed from measuring the quality/run-time data until it is analyzed and ready to provide again insights into the engineering of current or future systems.

Reusability of PPR Knowledge Engineering companies often have similar but not the same requirements regarding production systems and their design. For each new contract the two use cases UC1a and UC1b are executed, requiring the involved domain experts often to start from scratch or reuse, through many years of experience, existing solutions. Even though many products or systems could be classified and aggregated into families of products and production systems, this is not done, resulting in high rework efforts. A new PPR knowledge persistence solution should provide means of *reusability* for the engineering domain experts, providing libraries for reusing already existing PPR knowledge, mappings of (a) product to processes and (b) process to resources. Especially, these mappings often are based on reoccurring requirements from customers or imposed limitations from production resources.

Overall, the use cases revealed important requirements for PPR persistence that are hard to meet with the typical traditional persistence technology mix of (proprietary) engineering artifacts, Excel tables, XML configuration files, and relational databases.

7.8 Discussion

This section reports on a discussion of the overall process execution, observations, and lessons learned and extend Kathrein et al. (2019). It discusses results regarding the research questions introduced in Sect. 7.1 and in detail in Sect. 7.3.

RQ1. What Are Main Elements of a PPR EPA Method? Both business process analysis (BPA) and engineering process analysis (EPA) methods are concerned with investigating an existing process, involved stakeholders, and exchanged artifacts. Whereas BPA approaches like Santos and Alves (2017) and Rosenberger et al. (2018) focus more on the big picture of an engineering process, and do not allow for very sophisticated and detailed analysis (Vergidis et al. 2008), EPA approaches like Lüder et al. (2012), Jäger et al. (2011), and VDI (2010) tend to represent more individual workgroups and their procedures. Our presented approach in Sect. 7.2.2 combines the existing solutions and identifies the main elements in a repeatable two-phase process, resulting in a visual product/ion-aware representation, namely, the PPR data processing map (DPM). The proposed main elements: kick-off, interviews, refinement, and PPR artifact classification were evaluated in a holistic case study (Runeson and Höst 2009).

To support the proposed PPR EPA and execute its tasks, we introduced the role of the *EPA facilitator*. This role mediates the interests of all involved stakeholders and is responsible for choosing the right level of detail of the EPA as well as for choosing an adequate visual representation. In the conducted case study, three researchers took on this role. The execution and enactment of the proposed PPR EPA with its steps provide a first outline of how multidisciplinary engineering processes can be investigated. However, possible open issues that may surface in practice are still open for investigation and should be addressed.

The PPR EPA method allows collecting data, which is passed through the engineering process and records the current engineering process with links to engineering artifacts. A special focus lies on identifying tasks that create, require, or lose PPR knowledge and prioritizing the need of PPR knowledge for certain tasks and stakeholders. All involved stakeholders found the PPR EPA method suitable and useful. The PPR EPA further gave the stakeholders insights into not only their own line of work but also beyond and into other workgroups.

Both, independent investigations of workgroups and a high-level analysis for improvement potential for cooperating and collaborating stakeholders is possible with the proposed PPR EPA and further brings the benefit of explicit PPR knowledge identification.

The proposed process concept can also be used for the identification of technical depth and the identification of necessary security measures. Within the planning phase, information flow and therefore necessary user access privileges for the project can be derived. Furthermore, responsibilities of certain workgroups for certain components can be defined and non-repudiation can be ensured. This can be done either on a system level, or by applying cryptographic measures, which has the benefit of being independent of file- and operating system. A key challenge thereby lies in the nonintrusive support of employees in their daily work, which allows them to execute their work tasks as efficiently as before. One possible solution could allow for "weak" access rights, where users can execute tasks they are not responsible for, based on the engineering process description. Such an overstepping of a security boundary could be allowed, which should, however, be monitored, logged, and traced in a comprehensible way for project members and managers.

RQ2. What Are Main Elements of a PPR DPM Method and Notation? Section 7.2 briefly gave an overview of existing visualization notations for process analysis. In Sect. 7.2.3, we introduced the PPR DPM notation based on the BPMN 2.0 standard. The result is a PPR DPM, allowing a stakeholder to classify engineering artifacts regarding product, process, or resource knowledge and how these artifacts interact with certain engineering tasks.

The main elements from the standard BPMN 2.0 notations are: tasks, gateways, documents, and events. The newly introduced product/ion-aware notation elements are: annotations for documents regarding product, process, or resource knowledge classifications. We extend the task concept by annotating which of the PPR concepts is currently available, as well as further information that would be needed for an ideal task execution. A second extension to the task notation is an importance level, distinguishing important or crucial PPR knowledge dependencies, depicted as white/black broken documents.

By using a well-known and easy-to-use notation, the number of different concepts was minimized, which kept the level of complexity lower than in other approaches like Khabbazi et al. (2013), Huang et al. (2017) and Merunka (2017).

For the application of the new PPR notation, the stakeholders required a little bit of training but evaluated the PPR DPM as usable, useful, and a little bit less effort than the existing eEPC modeling approach.

RQ3. What Are Primary Use Cases That Require the Persistence of Different Categories of PPR Knowledge? From the case study for evaluating the PPR EPA and PPR DPM, we collected use cases on *Product/ion-Aware Engineering Tool Support* (UC1) and on *PPR-Based Run-Time Data Analysis* (UC2) to gain insights into the current technical landscape at the engineering company. These use cases build the first layer of a possible PPR knowledge persistence solution. Combining the insights from the use cases with interviews lead to the identifying of the characteristics of PPR knowledge categories and requirements on how to store and access PPR knowledge.

While the engineering tools currently focus on functions that use production system engineering data, advanced engineering tool functions require capabilities for defining and accessing PPR data and knowledge. The PPR knowledge categories of engineering data, configuration data, and run-time data indicate conflicting requirements for the persistence of mainly engineering artifacts, tables, graphs, and time series data. The requirements for PPR persistence were found hard to meet with the traditional persistence technology, such as repositories for engineering artifacts, structured text, and relational tables and databases. Also these requirements, combined with the PPR knowledge categories, provide functional requirements for the second layer of the PPR knowledge persistence solution. The third layer of the solution can in parts be addressed with the combination of use cases, requirements, and the knowledge gathered from the current situation at the company, but which requires further research.

While relational databases are a good choice for table-based data persistence (Vicknair et al. 2010), repurposing table-based data storage technologies for applications that require rapid changes of schemas or an altogether schema-less data model accumulates technical debt. Siddiqa et al. (2017) argued for the advantage of NoSQL data storage technologies for further flexibility of data definition and analysis in the development and operation phases.

As comparable persistence challenges can be found in business informatics, Sadalage and Fowler (2013) and Nance et al. (2013) pointed out that a combination of relational and NoSQL database technologies could be used for persistence design. However, this means redesigning the existing solution with new concepts and a clean data model leading to risks from data migration and from introducing a persistence design that uses considerably more complex technologies beyond the expertise of the domain experts, who often have an engineering background, but not from engineering large and heterogeneous software systems of systems. Therefore, we see future research work in exploring PPR knowledge persistence designs that allow addressing the use cases elicited in this chapter regarding their strengths and limitations in theory and in empirical studies with typical domain experts.

Limitations As all empirical studies, the presented research has some limitations that require further investigation.

Feasibility Study To evaluate the PPR EPA and the PPR DPM, we focused on specific use cases, which were chosen in cooperation with domain experts from an engineering company. The company is representative in size and domain for systems engineering enterprises, conducting business on a project basis. The focus of the engineering company lies on the manufacturing of production systems, without PPR knowledge management. All of our evaluation results are based on a limited sample of engineering projects, involved stakeholders, as well as different data models. The approach thus did not investigate situations where multiple products or variants are created and how this might affect the overall engineering process. We plan to overcome these limitations by expanding the case study in other domains and application contexts and further investigate possible issues of the PPR EPA that might arise.

Expressiveness of the PPR DPM Notation The notation of the PPR DPM enabled the involved stakeholders of the feasibility study to better express which PPR knowledge concerns are present in engineering documents. The proposed notation is not yet formalized or described and only presents a first visual aspect of ongoing research. There are also more advanced applications and analyses in prospects like constraint modeling or variation modeling. Constraint modeling would require extending the current PPR DPM notation to have an even higher expressiveness at hand, possibly exploiting concepts of ISA 95 (International Electrotechnical Commission 2003) or formal process specification given in VDI Guideline 3682 (VDI 2005). The involved stakeholders have also expressed the desire to model basic variations of products or product families, ranging from simple color adaptations to more complex process and system variations, which would affect the whole manufacturing system.

PPR Knowledge Flow and Artifact Exchange Investigation As mentioned previously, the PPR DPM notation is solely a visual extension to the BPMN 2.0. Even though it is possible to investigate an engineering process regarding the flow of knowledge, our proposed the notation come short regarding concrete dependencies between stakeholders and content of engineering artifacts. It was discussed that domain experts depend on intermediary results of one another; however, in some cases there might be only a partial dependency between a stakeholder and an artifact or one concrete value out of this artifact. The proposed PPR DPM only classifies the artifacts regarding PPR knowledge and does not detail the artifacts very much. This is however addressed in Chap. 8, with several approaches and methods to investigate such data logistics dependencies across several domains.

PPR Knowledge Persistence Use Cases and Requirements We collected and analyzed the use cases and requirements with domain experts at a single company. While we expect these use cases and requirements to be relevant for a wider application context, the focus on one company introduces bias that should be addressed by extending and validating the use cases and requirements with researchers and domain experts from a wider and representative set of data sources.

7.9 Conclusion and Future Work

The work environment of domain experts in systems engineering organizations is characterized by many different, collaborating disciplines and, from project to project changing of personnel. In such a multidisciplinary environment, many workgroups focus solely on improving their own local processes, tools, and methods. Little to no thought is given on how improvements of engineering interfaces for better collaboration and coordination could look like. This mindset leads to information silos, where only the bare minimum effort is fulfilled to have a working project collaboration.

The domain experts of systems engineering organizations also tend to focus more on the technical aspects of a system and product or process aspects are often neglected. This one-sided view on the PPR concept bears the risk of not communicating crucial parameter settings and endangering the project success and operation phase, as was described in Sect. 7.1 with the use case fragile product.

In this paper, we investigated a product/ion-aware method for an engineering process analysis (PPR EPA) method, as well as a notation for product/ion-aware data processing map (PPR DPM). Both contributions were based on elicited use cases from the systems engineering domain and should help domain experts, including the newly introduced role of an EPA facilitator, with a systematic repeatable approach to represent PPR knowledge in an engineering process. The introduced PPR EPA method is capable of tracing PPR knowledge throughout an engineering process. A special focus and capability is that tasks can be investigated regarding PPR knowledge requirements. The investigation of engineering artifacts further builds a main building block for analyzing PPR knowledge gaps that are present in an engineering organization and its process. Such analyses are a first step toward closing this knowledge gap.

The PPR EPA method provides the foundations for addressing the characteristics of Responsible Information Systems, such as flexibility, trustworthiness, and security. In respect to security, it allows the EPA to investigate possible security measures based on involved domain experts and their security clearance as well as classified engineering artifacts. Such an investigation finally can be the basis for planning necessary countermeasures and secure the intellectual property of an engineering organization. The EPA specifically addresses the major challenges introduced in Sect. 7.1.

C1. The Engineering Process Between Discipline-Specific Workgroups Is Hard to Trace and Analyze The outcome of the proposed PPR EPA approach visualizes a multidisciplinary engineering process. The visualization allows identifying PPR knowledge flows throughout the engineering process, highlighting tasks that create, transform, or lose PPR knowledge, as well classifying engineering artifacts regarding PPR knowledge aspects. This makes it possible to trace process executions and engineering artifacts through the engineering process. The PPR EPA also identifies interfaces between different disciplines and creating descriptions of which tasks are executed under which responsibility.

C2. Unclear Benefit of Representing PPR Knowledge Through visualizing the different involved disciplines of the engineering process, and further focusing on expressing the importance a task has regarding PPR knowledge, it is possible to analyze the whole engineering process and explicitly express PPR knowledge gaps. This product/ion-aware processing map (PPR DPM) can be analyzed regarding high-risk tasks and estimating the cost and effort it takes to explicitly represent PPR knowledge in engineering artifacts. Through this approach, domain experts see what information is available in which engineering phase and can match this to the actual PPR knowledge they receive and demand to close possible gaps or losses of knowledge along the engineering process.

C3. Unclear Impact of PPR Knowledge The PPR EPA and PPR DPM are able to assess the impact of PPR specific knowledge aspects, leading to considerations as to which PPR knowledge should be explicitly modeled. This is based on expressions regarding engineering tasks that need PPR knowledge for their execution. The PPR DPM addresses this challenge by indicating the priority an engineering task has regarding PPR knowledge. This allows all involved domain experts to identify especially critical tasks and address possible high-risk issues. The PPR DPM also refines the awareness and impact of early design decisions by domain experts.

C4. Unclear Use Cases with PPR Knowledge Categories That Require Persistence To address this challenge, we elicited primary use cases on *Product/ion-Aware Engineering Tool Support* (UC1) and on *PPR-based Run-time Data Analysis* (UC2) and the main PPR knowledge categories: engineering data, configuration data, and run-time. These use cases revealed a range of requirements for PPR knowledge persistence to guide software engineers who design and adapt engineering tools. Unfortunately, these requirements are conflicting and hard to address with traditional relation-based methods and technologies. Therefore, the initial research results on requirements suggest exploring a combination of persistence technologies regarding their technical capabilities to support advanced product/ion-aware use cases and regarding their usability and usefulness in typical application contexts.

Future Work Future work will include further applications and evaluations of the PPR EPA method and the PPR DPM notation in other engineering domains and application areas. Possible evaluations include the execution of the PPR EPA in a second engineering company, to cross-evaluate how the PPR EPA performs and if the found strengths and limitations are comparable between these two case studies, or if there is a bias based on the engineering organizations and their domains that should be investigated. The following aspects are of special interest for future research.

Advanced PPR Knowledge and Artifact Flow Investigation As discussed in the previous section, the presented approach does not provide means to investigate data logistics issues. Chapter 8 presents, however, several options on how dependencies between single values of engineering artifacts and dependencies on a more granular level can be addressed. Thus, in future work the proposed PPR EPA and also PPR DPM should be investigated with regard to how they can be combined with a possible data logistics approach.

Advanced PPR Knowledge Representation To be able to annotate PPR knowledge aspects directly onto engineering artifacts, there is the requirement and need to represent PPR knowledge explicitly in an engineering process. In future, these annotations should not only be visualized but also stored for further processing, analyses, and knowledge queries. The actual representation and storage of PPR knowledge could allow domain experts and stakeholders to move from general artifact representations to specific PPR knowledge aspects, which is also part of the Industry 4.0 vision.

Traceable Design Decisions Through expressing PPR knowledge explicitly, the relationships between the concepts and inherently made design decisions build the foundation for analyzing rationales and give insights into the early phases of an engineering process. Especially, the systems engineer gains understanding on how certain values for operational system parameters were chosen.

Generation of System Design Aspects From explicitly modeling PPR aspects and having traceable design decisions, it could be possible to derive design parameters from product/ion design decisions and engineering design patterns. Through efficiently deriving system designs and reusing these systems for whole production system families, an engineering company can achieve a considerable business advantage against its competitors.

Exploration of PPR Knowledge Persistence Requirements and Design Options We plan to explore PPR knowledge persistence designs that address the use cases and requirements elicited in this chapter. Possible designs need to be investigated regarding their strengths and limitations in theory and in empirical studies with typical domain experts.

IT Security Considerations The PPR EPA presents a detailed set of documentation regarding the engineering processes currently implemented in an engineering organization. This knowledge allows analysis of data flows across workgroups and could thus be interesting to a potential IT security attacker. Such threats to the integrity of the collected PPR knowledge and further even industrial espionage have to be researched in future work.

Apart from that, an interesting advancement can be the (semi-/fully) automatic detection of intentional/unintentional wrongdoing: the system should recognize if a certain step may result in bad engineering quality. The main challenge here is to recognize such possible results. One approach could be to let people define quality within the context of the project in an early project phase.

In terms of security, a next step for PPR can be to integrate secure software lifecycle processes, such as NIST SP 800-64 (Kissel et al. 2008) or ISO/IEC 27034-3:2018 (ISO 2018) (and future versions).

Acknowledgments The financial support by the Christian Doppler Research Association, the Austrian Federal Ministry for Digital and Economic Affairs, and the National Foundation for Research, Technology and Development is gratefully acknowledged.

References

Allweyer, T. (2016). *BPMN 2.0: Introduction to the standard for business process modeling*. Norderstedt: BoD–Books on Demand.

Beck, K. (2003). *Test-driven development: By example*. Boston, MA: Addison-Wesley.

Biffl, S., Gerhard, D., & Lüder, A. (2017). Introduction to the multi-disciplinary engineering for cyber-physical production systems. In *Multi-disciplinary engineering for cyber-physical production systems* (pp. 1–24). Cham: Springer.

Date, C. J., & Darwen, H. (1997). *A guide to Sql standard* (Vol. 3). Reading, MA: Addison-Wesley.

Drath, R. (Ed.). (2009). *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*. Berlin: Springer.

ElMaraghy, H. A. (2009). Changing and evolving products and systems–models and enablers. In *Changeable and reconfigurable manufacturing systems* (pp. 25–45). London: Springer.

Fay, A., Löwen, U., Schertl, A., Runde, S., Schleipen, M., & El Sakka, F. (2018). Zusätzliche Wertschöpfung mit digitalem Modell. *atp magazin, 60*(6–7), 58–69.

Force, U. A. (1981). *Integrated computer aided manufacturing (ICAM) architecture part II*. Volume IV-functional modeling manual (IDEF0), Air Force Materials Laboratory, Wright-Patterson AFB, Ohio.

Fowler, M., Kobryn, C., & Scott, K. (2004). *UML distilled: A brief guide to the standard object modeling language*. Boston, MA: Addison-Wesley.

Huang, Y., Huang, J., Wu, B., & Chen, J. (2017). Modeling and analysis of data dependencies in business process for data-intensive services. *Communications, 14*(10), 151–163.

Humble, J., & Farley, D. (2010). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. London: Pearson.

Humphrey, W. S. (1995). *A discipline for software engineering*. Boston, MA: Addison-Wesley.

Hundt, L., & Lüder, A. (2012, September). Development of a method for the implementation of interoperable tool chains applying mechatronical thinking—use case engineering of logic control. In *Proceedings of 2012 IEEE 17th international conference on Emerging Technologies & Factory Automation (ETFA 2012)* (pp. 1–8). IEEE.

International Electrotechnical Commission. (2003). *IEC 62264-1 Enterprise-control system integration–Part 1: Models and terminology*. Geneva: IEC.

International Electrotechnical Commission. (2013). *Engineering data exchange format for use in industrial systems engineering – Automation Markup Language AML*. Retrieved from http://www.automationml.org/

ISO/IEC 27034-3. (2018). *Information technology – Application security – Part 3: Application security management process*. Retrieved March 6, 2019, from https://www.iso.org/standard/55583.html

Jäger, T., Fay, A., Wagner, T., & Lowen, U. (2011). Mining technical dependencies throughout engineering process knowledge. *Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference* (pp. 1–7).

Kathrein, L., Lüder, A., Meixner, K., Winkler, D., & Biffl, S. (2018). *Process analysis for communicating systems engineering workgroups; Technical report CDL-SQI-2018-11*, TU Wien. http://qse.ifs.tuwien.ac.at/wp-content/uploads/CDL-SQI-2018-11.pdf

Kathrein, L., Lüder, A., Meixner, K., Winkler, D., & Biffl, S. (2019). Product/ion-aware analysis of multi-disciplinary systems engineering processes. In *Proceedings of 21st International Conference on Enterprise Information Systems, ICEIS* (Vol. 2, pp. 48–60). Setúbal: SciTePress. ISBN 978-989-758-372-8. https://doi.org/10.5220/0007618000480060

Khabbazi, M. R., Hasan, M. K., Sulaiman, R., & Shapi'i, A. (2013). Business process modeling in production logistics: Complementary use of BPMN and UML. *Middle East Journal of Scientific Research, 15*(4), 516–529.

Kissel, R. L., Stine, K. M., Scholl, M. A., Rossman, H., Fahlsing, J., & Gulick, J. (2008). *Security considerations in the system development life cycle* (No. Special Publication (NIST SP)-800-64 Rev 2). National Institute of Standards and Technology.

Lüder, A., Foehr, M., Köhlein, A., & Böhm, B. (2012). Application of engineering processes analysis to evaluate benefits of mechatronic engineering. In *Emerging technologies & factory automation (ETFA), 2012 IEEE 17th conference* (pp. 1–4). IEEE.

Lüder, A., Pauly, J., Kirchheim, K., Rinker, F., & Biffl, S. (2018). *Migration to AutomationML based tool chains –Incrementally overcoming engineering network challenges*. Retrieved January 2, 2019, from https://www.automationml.org/o.red/up-loads/dateien/1548668540-17_Lueder_Migration-ToolChains_Paper.pdf

Merunka, V. (2017). Symmetries of modellingconcepts and relationships in UML -Advances and opportunities. *Lecture Notes in Business Information Processing, 298*, 100–110.

Moser, T., Biffl, S., Sunindyo, W. D., & Winkler, D. (2010, February). Integrating production automation expert knowledge across engineering stakeholder domains. In *Complex, intelligent and software intensive systems (CISIS), 2010 international conference* (pp. 352–359). IEEE.

Nance, C., Losser, T., Iype, R., & Harmon, G. (2013). *Nosql vs rdbms-why there is room for both. SAIS 2013 proceedings* (p. 27).

Paetzold, K. (2017). Product and systems engineering/CA$_*$ tool chains. In *Multi-disciplinary engineering for cyber-physical production systems* (pp. 27–62). Cham: Springer.

Presley, A., & Liles, D. H. (1995). The use of IDEF0 for the design and specification of methodologies. In *Proceedings of the 4th industrial engineering research conference*. Citeseer.

Rilling, J., Witte, R., Schuegerl, P., & Charland, P. (2008). Beyond information silos—An omnipresent approach to software evolution. *International Journal of Semantic Computing, 2*(04), 431–468.

Rosenberger, P., Gerhard, D., & Rosenberger, P. (2018). Context-aware system analysis: Introduction of a process model for industrial applications. In *ICEIS* (Vol. 2, pp. 368–375). Setúbal: SciTePress.

Rowley, J. (2007). The wisdom hierarchy: Representations of the DIKW hierarchy. *Journal of Information Science, 33*(2), 163–180.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering, 14*(2), 131–164.

Sabou, M., Ekaputra, F. J., & Biffl, S. (2017). Semantic web technologies for data integration in multi-disciplinary engineering. In *Multi-disciplinary engineering for cyber-physical production systems* (pp. 301–329). Cham: Springer.

Sadalage, P. J., & Fowler, M. (2013). *NoSQL distilled: A brief guide to the emerging world of polyglot persistence*. London: Pearson.

Santos, H., & Alves, C. (2017). Exploring the ambidextrous analysis of business processes: A design science research. In *International conference on enterprise information systems* (pp. 543–566). Cham: Springer.

Schafer, W., & Wehrheim, H. (2007, May). The challenges of building advanced mechatronic systems. In *Future of software engineering, 2007 (FOSE'07)* (pp. 72–84). IEEE.

Scheer, A.-W. (1998). *ARIS: Vom Geschäftsprozeß zum Anwendungssystem/August-Wilhelm Scheer*. Berlin: Springer.

Schleipen, M., Lüder, A., Sauer, O., Flatt, H., & Jasperneite, J. (2015). Requirements and concept for plug-and-work. *at-Automatisierungstechnik, 63*(10), 801–820.

Schwaber, K., & Beedle, M. (2002). *Agile software development with scrum* (Vol. 1). Upper Saddle River: Prentice Hall.

Siddiqa, A., Karim, A., & Gani, A. (2017). Big data storage technologies: A survey. *Frontiers of Information Technology & Electronic Engineering, 18*(8), 1040–1070.

Stark, J. (2015). Product lifecycle management. In *Product lifecycle management* (Vol. 1, pp. 1–29). Cham: Springer.

VDI 3682. (2005). *Formalised process descriptions*. Berlin: BeuthVerlag.

VDI 3695. (2010). *Engineering of industrial plants, evaluation and optimization, Part 1*. Berlin: BeuthVerlag.

Vergidis, K., Tiwari, A., & Majeed, B. (2008). Business process analysis and optimization: Beyond reengineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C, 38*(1), 69–82.

Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010, April). A comparison of a graph database and a relational database: A data provenance perspective. In *Proceedings of the 48th annual southeast regional conference* (p. 42). ACM.

Wieringa, R. (2014). *Design science methodology for information systems and software engineering*. Berlin: Springer.

Wiesner, S., & Thoben, K. D. (2017). Cyber-physical product-service systems. In *Multi-disciplinary engineering for cyber-physical production systems* (pp. 63–88). Cham: Springer.

Zhang, C., Chen, X., Feng, Y., & Luo, R. (2010, June). Modeling and functional design of logistic park using IDEFO method. In *2010 7th international conference on service systems and service management* (pp. 1–5). IEEE.

Zhu, L., Bass, L., & Champlin-Scharff, G. (2016). Devops and its practices. *IEEE Software, 33*(3), 32–34.