FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Auswahl und Anwendung von Modellierungssprachen für multidisziplinäres Engineering von Produktionssystemen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering/Internet Computing

eingereicht von

Lukas Kathrein, BSc
Matrikelnummer 01325082

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Stefan Biffl
Mitwirkung: Projektass. Dipl.-Ing. Dr.techn. Dietmar Winkler
apl. Prof. Dr.-Ing. habil. Arndt Lüder

Wien, 22. Jänner 2019

Lukas Kathrein

Stefan Biffl

Modeling Language Selection and Application for Multi-Disciplinary Production Systems Engineering

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering/Internet Computing

by

Lukas Kathrein, BSc
Registration Number 01325082

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Stefan Biffl
Assistance: Projektass. Dipl.-Ing. Dr.techn. Dietmar Winkler
apl. Prof. Dr.-Ing. habil. Arndt Lüder

Vienna, 22nd January, 2019

Lukas Kathrein
Stefan Biffl

Erklärung zur Verfassung der Arbeit

Lukas Kathrein, BSc
Neuhaus 164, 6552 Tobadill

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 22. Jänner 2019

Lukas Kathrein

”All we have to decide is what to do with the time that is given us.”

Gandalf

Danksagung

Wenn ich nun auf die finale Version dieser Diplomarbeit blicke, so ist einiges darin nicht abgebildet, was ich im Laufe der Arbeit gelernt habe. Dies ist auf die hervorragende Betreuung von Univ. Prof. Dr. techn. Stefan Biffl, Projektass. Dr. techn. Dietmar Winkler und apl. Prof. Dr.-Ing. habil. Arndt Lüder zurückzuführen. Sie haben mir sehr viel, weit über die hier abgebildeten Resultate beigebracht, keine Fragen abgewiesen und mich an ihren Visionen, welche jetzt zum Teil in der hier vorliegende Arbeit enthalten sind, teilhabenlassen.

Ich will ganz bewusst diese Arbeit meinen Eltern widmen. Ihr habt mir immer alle Türen im Leben offengehalten, so dass ich sie nur noch durchqueren musste. Für eure nichtendende Unterstützung und die Möglichkeit, im Leben zu machen, was ich will und mir Freude bereitet, ein herzliches Danke!

Daneben will ich aber auch meinen beiden Schwestern danken. Sie haben mir gezeigt, was man mit harter Arbeit und Zielstrebigkeit im Leben alles erreichen kann.

Ein besonderer Dank gilt auch meiner Freundin, welche auf so manch romantische Stunde verzichten musste, damit ich meinen Wissensdurst stillen konnte.

An meine geliebten Freunde: Von Tag eins wart ihr mir immer eine große Stütze und gemeinsam haben wir viel erreicht. Ihr wart immer für mich da und wart meine mentalen ‚Fechtpartner‘ und seid es zum Glück auch heute noch. Ich bilde mir nicht ein, dass ich heute auf die großen Erfolge im Studium oder privat zurückblicken könnte, wenn ich euch nicht gehabt hätte. Vielen Dank dafür!

Acknowledgements

When I look at this thesis now in its final form, there is a lot that I have learned but which is not represented here. This is because of the superb support that I have received from my advisors: Univ.Prof. Dr.techn. Stefan Biffl, Projektass. Dr.techn. Dietmar Winkler and apl. Prof. Dr.-Ing. habil. Arndt Lüder. I have picked up many new skills on how to deliver the best possible outcome and you never turned down a question or discussion point, to clarify the vision you had in mind, which is now in parts this thesis. Thank you for that.

I want to dedicate this work to my family, especially my parents, who have always given me the opportunity to pursue the things I want in life and have unbelievable support of me. Further, I want to thank my two sisters, who were role models on what is possible in life when you put in hard work and dedication.

To *my precious* girlfriend, who had to spare some of our time so, that I could pursue my craving for knowledge.

To my dear friends: From day one you have been a great support and together we achieved many great things. You always were and are my mental sparring partners and challenge me to new ways of thinking. I do not believe that I would be here now and could look back on such great accomplishments without you. Thank you!

Kurzfassung

Im Engineering von Produktionssystemen sind viele verschiedene Disziplinen involviert. Beispiele dafür sind Elektrotechnik, Mechanik und Informatik. Über den Engineering Prozess hinweg ergeben sich verschiedene Phasen und Domänenexperten sind hier oft voneinander abhängig. Diese Interaktionen zwischen Arbeitsgruppen wird als multidisziplinäres round-trip Engineering bezeichnet. Um die Vision der Industrie 4.0 zu verwirklichen ist ein wichtiger Punkt die Repräsentation von Produkt, Prozess und Ressourcen (PPR) Beziehungen. Viele Stakeholder sind allerdings nur auf ihre eigenen Domänen fokussiert und bilden ”Informationssilos”.

Dadurch, dass PPR Wissen oft vernachlässigt wird, zielt diese Diplomarbeit darauf ab, diese Lücke zu verkleinern. Eine gute Lösung hierbei soll in der Lage sein, einen Engineering Prozess hinsichtlich PPR Wissen zu analysieren und dieses auch auszudrücken. Bereits bestehende Lösungen sind hierzu nicht in der Lage, da sie entweder zu breit oder zu domänenspezifisch nur auf eine Rolle fokussiert sind. Nichts desto trotz bilden diese Lösungen einen guten Startpunkt für neue Lösungsansätze. Existierende Modellierungssprachen sind auch nicht in der Lage PPR Wissen auszudrücken, bieten allerdings Einblick in mögliche Adaptierungen.

Methodisch baut die Arbeit auf bekannten Konzepten wie einer Literatursuche oder dem Design Science Cycle auf. Die Evaluierung wird mittels Interviews und Konzeptentwicklungen in einer echten Produktionsorganisation durchgeführt.

Erstes Resultat dieser Arbeit ist eine Kombination von Ansätzen um einen Engineering Prozess zu analysieren. Resultat ist hier ein Ansatz für die Analyse von Engineering Prozessen und Indikation von PPR Wissen durch Erweiterung des BPMN 2.0 Standards.

Weiters werden mögliche PPR Modellierungssprachen identifiziert, ausgewählt, verglichen sowie adaptiert. Die Adaptierung und Erweiterung des VDI 3682 Standards ist hierzu das Resultat.

Um das gefundene PPR Wissen auch speichern zu können, untersucht die Arbeit mögliche Anforderungen an Speicherlösungen. Hier bilden Fallbeispiele und Gruppen von Daten neue Einblicke bezüglich den Anforderungen.

Alle Ergebnisse der Arbeit verbessern die Möglichkeiten PPR Wissen zu modellieren und zeigen Abhängigkeiten von Arbeitsgruppen auf.

Abstract

In *Production Systems Engineering* there are many different disciplines like mechanical, electrical or software engineering involved. Throughout the engineering process, different phases can be identified where the domain experts often depend on results from one another. The interaction between the different disciplines is identified as *multi-disciplinary* round-trip engineering environment. To meet the vision of Industry 4.0 a key factor is the representation of relationships between the Product, Process and Resource (PPR). However, many stakeholders only focus on their discipline, forming an information silo.

Due to the fact, that PPR knowledge is important but often neglected, this master thesis aims at addressing this issue. A criterion for a good solution is to be able to investigate engineering processes in regard to PPR knowledge and express this newly found knowledge through a modeling language. Existing solutions lack either the depth of analysis or are to narrow on one discipline to achieve this aim. However, useful insights into designing a treatment to overcome this limitation can be used from existing works. Existing modeling languages also have no means to represent PPR knowledge, build however a good foundation for possible extensions.

The thesis first proposes a combination of approaches for a new engineering process analysis, by following the design science cycle with an integrated case study. Outcome is an engineering processes analysis method and representation of PPR knowledge through an adapted BPMN 2.0 notation.

Second, a literature survey is used to select, benchmark and propose adaptations for a PPR specific modeling language. The main result here is, that the VDI 3682 standard presents the best building block and allows for an easy extension.

To also be able to persist PPR knowledge, the thesis investigates how a possible solution could look like. The outcome of this aspect provides useful insights based on current use cases and how system architecture and data/knowledge models represent requirements for a persistence solution.

The solution approach is evaluated with a real-world production systems engineering organization and includes a proof of concept evaluation and interviews with domain experts across multiple disciplines. All results improve the explicit representation of dependencies between workgroups, highlighting improvement potential and express PPR knowledge.

Contents

CHAPTER 1

Introduction

Production Systems Engineering (PSE) organizations have the goal to create manufacturing systems, that are capable of producing products, with high throughput and simultaneously meet high-quality standards. Additionally, PSE organizations cannot fall back on existing, standardized or generalized solutions but have to engineer tailored solutions for their customers depending on the projects [83]. In PSE organizations many different roles like mechanical, electrical and software engineering are present [53][69]. Each domain has highly qualified personnel and focuses on different aspects of a manufacturing system. This leads to a multi-disciplinary engineering (MDE) environment, which is described in detail for example in [8]. In such environments the domain experts are involved in a Round-Trip Engineering (RTE) process. Furthermore, many different interfaces between workgroups exist and need to be maintained. These interfaces play a key part in knowledge transfer and exchange to uphold quality standards and meet customer requirements. Customers often impose limits regarding resources in the form of time, money or production rates. Modeling languages, in either textual or graphical form [31], and their defined rule sets allow addressing exactly the issue of conveying information and knowledge through (engineering) processes.

Many PSE organizations focus on solving challenges related to the machinery, also named *resource*, as most domain experts come from an engineering background. A challenge however is that often important relationships between *Product, Process, Resource (PPR)* [71] are not explicitly modeled. This lack of explicit PPR knowledge leads to insufficiently covered production plans, resulting in unstructured manual reworks and a high risk of bad quality parts. All individual parts of the PPR concept are interrelated. For producing a product is it necessary to execute processes that are executed by a resource. A resource has the capabilities to execute processes and is responsible for the quality of the product. The processes describe the capabilities of resources and are executed to create products.

As the term *PPR knowledge* will often be used in this work, a short description is given here before more background is delivered in chapter 2. The knowledge hierarchy from

Rowley [63] builds a good starting point to define the term PPR knowledge. Knowledge expresses concepts and provides applications of underlying data and information models. In the context of this work, PPR knowledge includes a) success-critical attributes such as parameter settings for production processes or configurations on production resources and b) relationships such as constraint dependencies between the PPR elements.

This chapter motivates the research (see subsection 1.1), describes in more detail the problem context 1.2, presents the aim of the work 1.3 and finally discusses the remaining structure of this work 1.4.

1.1 Motivation

Lüder investigated in [46] different engineering processes. In total, eleven different engineering processes are presented. This highlights, that engineering processes highly dependent on the domain they are in, company size, or the involved domains. Further, the authors point out that there is simply no way of generically investigating an engineering process. Works from both the business community like Business Informatics (BI)/Information Systems (IS) and engineering communities who focus on PSE, present possible approaches on how to investigate engineering processes. However, the results are not suitable for a MDE process where complex interactions and highly trained personnel are at work. One issue that makes it hard to analyze MDE processes is that nearly every project has its own structure. Another issue is based on how domain experts think in MDE PSE organizations. Many domain experts are focused on intra process improvements and do not consider other roles that have tasks in common or are dependent on their work-results. This leads to so-called *information silos* [61]. Workgroups do not optimize their interfaces to other engineering experts for possible coordination and collaboration tasks but are only focused on their task executions and optimizations.

It could be argued that in Software Engineering (SE), these problems have already been solved with approaches like DevOps [86], SCRUM [72] or test-driven development [6]. However, in PSE is it not so easy to rapidly test a system under construction as there are often long engineering cycles present, domain experts do not consult daily with each other, and the involvement of risky hardware make it nearly impossible to apply these concepts which are known to BI/IS domain experts. To further illustrate the problem and also motivate why an explicit investigation of an engineering process with a modeling language selection and application is needed, an example for missing PPR knowledge will be presented. The product from a PSE context is in SE code produced by developers. It does not matter if this code is a small script or a big integrated graphical user interface for a complex shopping system. A (staging) environment as it is known from [35], that is executing the produced code can thus be seen as the production process. An equivalent to a mechanical resource is for example an interactive development environment. These environments are used by developers to produce code. Another example for a resource might be a web server executing the code for the user interface (the product). The (staging) environment executes the code according to the capabilities of the resource.

In a SE context, the absence of PPR knowledge, which is often lost due to missing communications and insufficiently covered engineering processes, can be translated with the risk of miscommunicating, for example, non-functional requirements. For the example, the assumption is made that in a project, the requirements of throughput or security are not communicated to the development team. Only at the end of the project does the client voice the issue of insufficient throughput. At this point in time, it is nearly impossible to add complex encryption algorithms or authentication mechanisms.

For the PSE domain is it thus an enormous challenge to address applying process analysis and improvements, as they are a) not used to these approaches and b) the context and involvement of complex PPR knowledge make it hard to rapidly prototype. Engineering stakeholders mainly focus on improving their respective fields of work and thus the issue of a MDE process analysis and further PPR knowledge representation through multiple domains remains unsolved. Even though these two mentioned problems do have a great impact on the overall engine ring process, the experts do not prioritize them very high. The lack of Business Process Analysis (BPA) and Engineering Process Analysis (EPA) methods that are suitable for investigating an engineering process, motivate a combination of both worlds and introducing possible solution approaches from a SE context into a PSE environment and adapting existing solutions from both communities. The main contributions of this work thus are the investigation of MDE process analysis, representing the outcome of such an approach, and further to use modeling languages to represent PPR knowledge. The contributions are motivated and part of ongoing research presented in [9].

1.2 Problem Statement

The previous section gave an introduction and motivation into the problem space and why it is important to research possible solutions in a MDE environment. This section now presents, in more detail, a common problem for MDE with the help of an illustrating use case: *fragile product*. Before diving into the use case, figure 1.1 will be described as the flow and execution sequence of an engineering process is tightly bound to the use case.

In figure 1.1 there are several distinct elements present. Four different actors are present: the *production process planner (PPP)* (red), the *production system planner (PSP* (green), the *automation engineer (AE)* (blue), and the *production process optimizer (PPO)* (orange). The first two roles, production process planner and production system planner are part of the basic planning phase, which is the first phase of each engineering project and visualized in the upper half of figure 1.1. The automation engineer is part of detail planning, the second phase, and the production process optimizer is responsible for smooth operations of the resources under construction, the last phase, depicted in the lower half of the image. Knowledge handovers between these roles are done via a *resource knowledge base*, which provides the capabilities to convey resource information like structure, parameter settings and composition of the physical machinery under construction. The resource knowledge base is visualized in figure 1.1 in the middle as a rectangle.

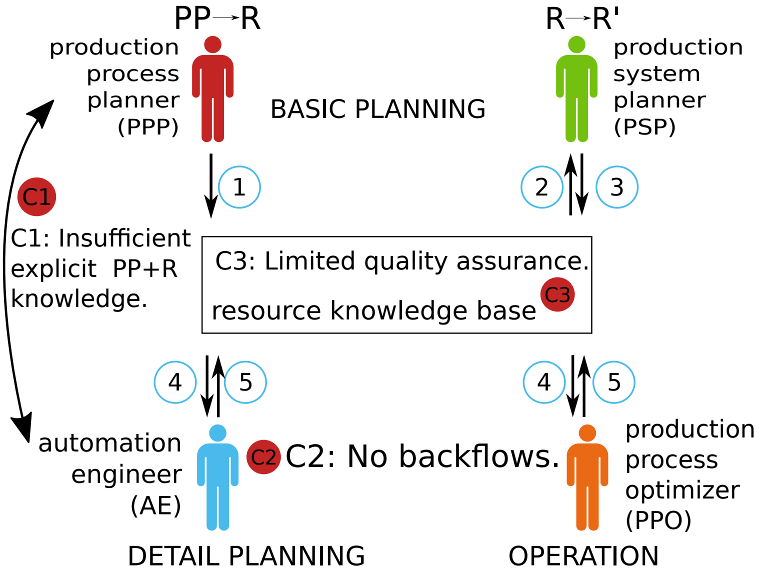


Figure 1.1: Identified key challenges for this research.

To illustrate the sequence and discuss the remaining few elements of figure 1.1, the use case *fragile product* is now presented. A customer requires the engineering and manufacturing of a highly automated production system. This system is responsible for producing fragile products in an integrated assembly line at the customer's site. The customer creates plans of the product and describes characteristic properties like length, width, weight, etc. and hands this information over to the PSE organization, which is responsible for engineering and creating the system.

In the PSE company the production process planner receives all information concerning the product lifecycle, process specifications and resource limitations or requirements. The main goal of the production process planner is to create a first initial design of a production system with the according production processes. Information form the customer often contains aspects of process and product information and is transformed into resource knowledge. This is indicated in figure 1.1 by the annotation above the production process planner *PP->R*. The production process planner makes an offer to the customer and if the offer is accepted the previously created new resource knowledge is handed over, via the resource knowledge base, to the production system planner. This is indicated by the blue circle numbered one.

The production system planner receives only the resource knowledge aspect out of the three PPR aspects and creates basic variants of the system, indicated by the blue circle number two and the annotation above the production system planner P->P'. After creating many variations and choosing one that seems fitting for the given project constraints, the resource knowledge is again stored in the resource knowledge base and the basic planning phase is finished (circle number three).

In parallel, indicated by the two circles, number four, the automation engineer and production process optimizer start their work. Each role creates discipline specific plans and optimizations based on the previously created basic plans. In the example of the fragile product, both roles agree on using a high-throughput transportation system, which is capable of transporting the individual parts under construction between the working cells of the machinery. The transportation system is chosen to fulfill the requirements of the customer regarding the parts per minute throughput. However, during the first operation of the system, the high acceleration of the transportation system damages many of the fragile product parts, leading to unacceptable quality metrics and many bad parts.

This is because the information about the fragility of the product was not conveyed throughout the engineering process, which is the first challenge this work wants to address. The challenge *C1: Insufficient explicit representation of PP+R knowledge in the PSE process*, is represented in figure 1.1 with a red filled circle. The production process optimizer, seeing that many parts are damaged starts uncoordinated, manual communications resulting in an extra effort and unplanned rework. This avoidable rework endangers the projects overall success, as the available time for the manufacturing of the production system is very limited.

Challenge two in figure 1.1, *C2: Backflows insufficiently covered in PSE RTE*, addresses the issue, that it is not possible for the automation engineer or production process optimizer to bring knowledge back into the engineering process. There is currently no reuse of engineering knowledge and with a high possibility, the next project will be executed in the same way. In the example, it would be good for the next project to indicate, that the given requirements are not alignable with the chosen transportation system.

The third challenge *C3: Limited support for quality assurance of PPR in PSE RTE*, focuses on supporting better quality assurance in engineering processes. The situation of the *fragile product* use case could have been avoided if the resource knowledge base would be able to also convey information about the production process or the product under assembly and their relationships to the production system.

In the presented scenario it can be seen that the missing information of PPR knowledge and the lack of traceable design decisions can have great negative impact on the engineering process and its outcome. This high-risk execution is identified through challenges one to three in figure 1.1. The negative impact could have been avoided by the simple means of providing a parameter setting in form of an integer and conveying why this setting has been chosen. However, the basic planners often do not know how great of an impact their decisions have on future engineering phases, which is why in most cases PPR knowledge is not conveyed in the first place. Further, is it currently for all domain experts not possible to adequately store or query this information, even if the basic planners would share their knowledge. This is why on a high level the question can be asked: *How can the PSE process be improved with a better PPR knowledge representation?* The next section, the aim of the work, presents contributions and possible answers to this high-level research question.

1.3 Aim of the Work

The previous two sections 1.1 and 1.2, gave motivation and described the problem the research wants to address in more detail. This section now focuses on presenting the key contributions and the overall goal of this work.

In figure 1.2, the three corresponding contributions to the previously introduced challenges are presented. As this figure is structured the same way as the previously introduced figure 1.1, the structure and elements will not be discussed again.

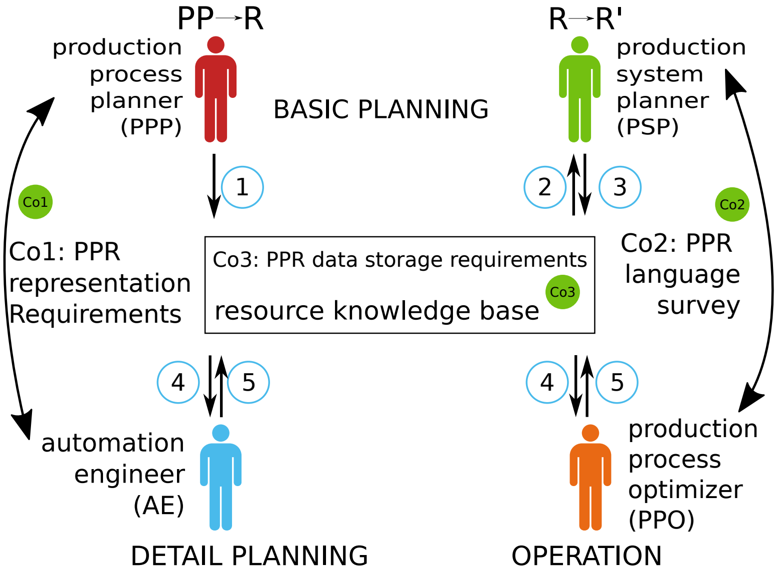


Figure 1.2: Key contributions this work presents.

The first contribution, *Co1: PPR representation and requirements* in the green filled circle on the right hand side of figure 1.2, addresses the missing PPR knowledge representation through the engineering process. The double-headed arrow right next to the contribution circle indicates that it is a goal to allow the flow of knowledge between basic planning and detail planning and operation. The contribution *PPR representation requirements*, addresses the issue that it is not clear for basic planners what impact their design decisions have and further that they have no means to actually express their knowledge in a structured way. Co1 thus focuses on a) visualizing the impact of design decisions through eliciting and representing an engineering process and the flow of knowledge and b) on finding requirements that need to be met to be able to express PPR knowledge.

Second contribution *Co2: PPR representation; language survey and definition* uses the output of the first contribution and investigates currently available modeling languages for representing PPR knowledge. The main goal of this contribution is to investigate how a transparent and structured representation of PPR knowledge could look like, enabling the representation and transformation of PPR knowledge through a MDE process. The second contribution also focuses on allowing the knowledge flow between the first phase, basic planning and later phases: detail planning and operation. Co2 is engaged in researching existing approaches for knowledge representation in terms of modeling languages, what their capabilities and limitations are and to then compare the found existing solutions and investigate their suitability for PPR knowledge modeling.

The last contribution this work wants to make, *Co3: PPR knowledge persistence and management* focuses on finding use cases and requirements that should be fulfilled by a possible PPR knowledge base. In the previous section 1.2 it was already highlighted that the resource knowledge base is responsible for many problems that are currently present in a MDE process. By addressing the issue of limited quality assurance, the research wants to lay groundwork for investigating possible persistence solutions. These possible solutions need to meet the requirements of elicited use cases and need to be able to work with different kinds of groups of engineering data. Possible new solutions should thus support the MDE process and all involved domain experts and stakeholders across all different engineering phases.

The main expected results are:

- An approach to investigate a MDE process, focusing on PPR knowledge transfer
- A visual representation of the found (PPR) knowledge of a MDE process
- A set of requirements for expressing PPR knowledge characteristics which need to be expressed in a possible PPR modeling language solution
- Benchmarking results of existing solutions against the elicited requirements
- A possible adaptation of the best fitting modeling approach if not all requirements can be met out of the box for expressing PPR knowledge
- Use cases, groups of different data and requirements that highlight characteristics of PPR knowledge in an engineering process. Focus is here the aspect of persisting knowledge.

1.4 Structure of the Work

As chapter 1, now introduced the context, problems, and aim of the work the remaining structure will be described.

Chapter 2 is divided into multiple sections. The first section introduces the background on PSE and the general information needed for the context of the research. MDE will then be presented in the second section, providing insights into how a MDE process/environment looks like based on existing works. Third, process analysis methods from both communities namely BI/IS and PSE will be presented. The third section focuses on presenting individual strengths and limitations of existing works and carve out building blocks for this work. In section four of the second chapter, different modeling approaches for representing the outcome of process analysis as well as PPR knowledge will be presented. Chapter 2 closes by introducing existing persistence solutions and their respective application areas.

All methodological approaches and the respective Research Issues (RI's) with Research Questions (RQ's) will be presented in chapter 3. Each individual RI, has one or multiple RQ's, but each RI has one methodological approach that is followed to answer the RQ's.

In chapter 4 a detailed approach for a PPR EPA method will be presented. The chapter investigates which common concepts of literature can be used and what requirements make up a good solution to then present adaptations and evaluate them.

Chapter 5, is concerned with how PPR knowledge can be represented in a MDE process. The main focus of this chapter is to find requirements of PPR knowledge representations, eliciting existing solutions and their capabilities and then proposing adaptations and evaluate these adaptations for a possible PPR language.

Chapter 6 is then focused on use cases, groups of data, and requirements for persisting PPR knowledge for further use. All of the three result chapters are structured according to the methodological approach that they follow, which is presented in chapter 3.

In chapter 7, each individual RI is discussed and possible limitations that the research faces are presented. Finally chapter 8 gives a conclusion and presents an outlook for possible future work.

CHAPTER 2

Related Work

Chapter 1 introduced the context of the research, Production Systems Engineering (PSE), with the different involved disciplines and stakeholders. The previous chapter also identified key challenges which are present in a multi-disciplinary engineering (MDE) environment, ranging from PSE specific to more general Software Engineering (SE) and modeling problems.

This chapter presents related work from the relevant research areas and builds an important building block for the solution approach. Chapter 3 will already make use of here described concepts, their strengths and limitations as well as use this knowledge to further motivate the research issues with Research Questions (RQ's).

Section 2.2 provides more background information on the context of the thesis, PSE, with a special focus on the Product, Process, Resource (PPR) concept. Section 2.1 defines the term MDE, and discusses it with related work. Section 2.3 discusses known process analysis methods, from both Business Informatics (BI)/BI and PSE backgrounds. Section 2.4 focuses on different aspects of modeling. First, known approaches for modeling engineering processes and artifacts are discussed, followed by options to model PPR knowledge and express possible use cases with a language. Section 2.5 closes this chapter with related work on PPR knowledge persistence options.

2.1 Multi-disciplinary Engineering

Multi-disciplinary engineering can be defined as the involvement of different disciplines in an engineering process [8][37]. Examples of involved disciplines are electrical, mechanical or software engineering domain experts. Each discipline further typically applies individual tools, methods and data models to achieve their specific goal they pursue in the respective engineering phase.

SE depends in a multi-disciplinary engineering environment heavily on specific data and plans from a wide range of engineering domains [53]. A challenge identified by Moser et al. [53] for SE lies in the insufficient semantic model integration between domain experts in various disciplines. Also is the knowledge which should be expressed explicitly, often embodied in various ways like terminologies, people, processes or methods [50], amplifying the challenge of insufficient knowledge representations. This challenge can be seen, as different but partly overlapping terminologies are used in the different disciplines. Such a distributed and nonuniformed knowledge base impairs the collaboration of different work groups and can even be harmful if one expression, meaning two different things, is used by two domain experts but they do not mean the same underlying concept. To address this challenge Moser et al. [53] propose a semantic mapping, focusing on providing links between data structures from engineering tools and systems to support the exchange of information between the tools.

Another challenge in a MDE environment is, if different disciplines hand over their artifacts, and build on intermediate results, a technical dependency is created between these disciplines [37]. Jäger proposes for this an approach, that identifies first the artifacts and then investigates their life-cycle, resulting in a process model including stakeholders and their tasks. This approach will be closer inspected in section 2.3.

Schafer and Werheim [69] see it as a crucial point that different roles need to cooperate during the engineering phases and let go of the "throw it over the wall" approach, where the creator of knowledge sees himself no longer responsible for artifacts once they are handed over to a new discipline. This approach of not caring for once developed artifacts, concerns the disciplines of PSE like it does SE. In SE, developers often do not see themselves responsible for what happens to their code once it is written, testing and especially deployment of the code is often not their primary focus. SE, has in recent years tried to overcome this situation, by more and more relying on the DevOps approach [86]. DevOps, a set of practices, intends to reduce the time between the writing of code until this code is committed and run on a production system. Even though DevOps has some technological challenges that need to be solved, in many organizations the cultural adaptations pose a larger barrier to overcome. Zhu et al. [86] also point out that the change from a normal to a DevOps based development cycle is not always smooth, due to the revolutionary nature of the changes introduced into the multi-disciplinary organization.

2.2 Production Systems Engineering

This section presents first some general background on PSE, before presenting a more detailed description of PPR.

2.2.1 PSE concepts

PSE can be roughly divided into two main categories: batch and discrete manufacturing. The first one is batch processing. A batch is one atomic unit, meaning that it can not be further distinguished. An example for a batch could be one liter of a compound mixture from different chemical substances. The batch cannot be further distinguished once it is in its final form and is seen as a whole. The second category, where this work also focuses on, is discrete manufacturing [16]. Contrary to batch processing, is it possible to distinguish between individual parts or products. For example, can each individual car from a factory be identified and has its corresponding serial number. Further, could each car be divided again into most of its sub-parts. Discrete manufacturing is characterized by either producing low volume but high complexity, or producing high volume with lower complexity. The processes in discrete manufacturing are also not continuous in nature [16], but can be split up into several different processes where each process can have a different production rate and be started or stopped individually. The remaining of this work mainly focuses on concepts applicable to discrete manufacturing.

As discussed in the problem description, two major disciplines are involved in the planning and construction of production systems namely basic and detail planning.. The first discipline concentrates on the *basic planning* whereas the second discipline is more involved in *detail planning* and *operation*. In basic planning, the main focus lies on the product and process knowledge to construct a resource capable of executing the processes and manufacturing the products. The second discipline covers detail planning which focuses more on the resource part. It is obvious that there should be a methodological and structured exchange of knowledge, allowing the disciplines to better coordinate their work packages and collaborate when a multi-disciplinary approach is needed. In [56] it is stated that there exists no sufficient solution as of yet which enables a reliable exchange of knowledge between *basic planning* and *detailed planning*. Paetzold points also out that it is required to have clear procedures in the design of the development process of production systems and that they should follow some form of a standard.

A more detailed view on the engineering phases can be found in the VDI 3695 standard [78]. The standard defines as a basic concept the Engineering Organization (EO). An EO is defined as an organization that conducts its business on a project basis and each project has several distinct phases. All phases are run through sequentially in exactly this order: acquisition, planning, realization, and commissioning. Figure 2.1 depicts the project-related phases according to the VDI 3695 standard.

The VDI 3695 does not identify any stakeholder or engineering tasks that are present in the individual phases but is more concerned with process improvements. The proposed improvements range from unified data views, a common project related execution

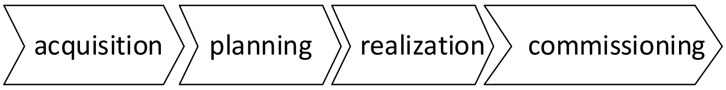


Figure 2.1: Project-related phases according to VDI 3695 [78].

framework to general aspects like supply chain management. Even though the standard presents many improvement possibilities for an EO, no concrete means are presented on how an organization could improve its current situation and reach a new and optimized state. The VDI 3695 standard, is often used as basis for further extensions like for example Schäffler does in [73]. Schäffler et al. [73] build on the standard and use the four perspectives: activities, human resources, toolchains and information exchange to investigate engineering processes.

In [46] Lüder et al. identify besides the VDI 3695 standard, ten other engineering processes. Each of these processes has its own sequence of engineering phases and differs in some way from other existing approaches. This situation shows, that the engineering process is very dependent on the industry an EO is working in and further, that even in one industry organizations do not follow one standard engineering process for production systems.

In [46] it was argued, that the engineering, as well as construction of production systems is very dependent on the EO. Further does the engineering process require a lot of time and highly trained personnel with specialized skills and knowledge. This fact is to some degree owed to the variety of tools used by the different roles to support their tasks. During the engineering process, different artifacts are created. Hundt [36] pointed out the challenges for inter-operable toolchains and identified the three major philosophies that evolved over time. These philosophies are one-for-all, best-of-bread and integration framework. Each philosophy imposes different requirements regarding a data model, different data exchange methodologies and technologies and different software systems [36]. Further is the successful exchange between different tools also dependent on passing on the semantics of produced artifacts and not only focusing on syntax aspects. The artifacts that are conveyed through an engineering process need further be suitable for all involved stakeholders and workgroups across the several involved engineering disciplines. The optimization in the *detail planning* for example requires the analysis of quality assurance data and also the knowledge of requirements coming from the *product* and the *process*. In [8] the transfer of engineering knowledge throughout the engineering process is identified as a requirement for the data model, which needs to be extensible and able to accommodate linked data across the currently isolated disciplines.

Lüder et al. [48] identify the interest of Mechatronical Units (MU) as important. Using mu in an appropriate and systematic way can support the engineering process. This, however, requires a clean definition and structuring of mu, which Lüder et al. [48] describe in their work. Motivation for researching mu is the reduction of costs throughout an engineering process. Production systems engineering companies try to minimize costs resulting from the engineering process. One approach is to interlink different engineering activities. This goal is hard to achieve because first there are information silos, where domain experts work and focus on intra discipline performances instead of multi-disciplinary approaches, and second, the manufacturing systems need to be flexible with respect to products and applied technologies. Great potential is identified in using libraries consisting of mu. These libraries are structured and contain the most relevant information about the mu needed throughout an engineering process and allow also an exchange through different toolchains. One key aspect identified in [48] is the importance of a common exchange format, that is processable by the involved engineering tools.

Schafer and Werheim [69] see the highest priority of modeling in a uniform specification of the discrete and continuous parts of an advanced mechatronic system. Such a specification should support the work of all disciplines that are involved in the engineering process. A model also should include the nested component hierarchies, communication structures and behaviours of components including behavioural specifications.

2.2.2 Product-Process-Resource (PPR) Concept

Technical systems are often distinguished into products and production systems [8]. The reason a company exists is often that of its products, i.e., products are created in a value-adding process to make profit by selling them [75]. A product can be something tangible and robust like a car, or more fragile like a glass panel, but also intangible products exist for example repair services for a car or a print service for documents. Production systems however, are oriented on creating the products by combining suitable production factors [24]. A production factor can be anything from raw materials, work-in-progress pieces, to complete production resources. An example of a work-in-progress piece is, for example, an unfinished handlebar, that is transported to another processing station for finalization. The most prominent production factor, however, is the production resources, all machines that are involved in a production process like robots for welding or gluing, but also conveyor belts for transport processes or quality check modules including cameras and physical measurement machines.

There exist strong dependencies between the product and the production system. The product needs the production system to be created on it. Production systems define boundary conditions that influence the possible properties of a product. The two concepts of product and production system are linked via the production process and its capabilities which are executed to create the product on the production system [8].

Schleipen [71] coined the Product-Process-Resource concept for the relationships between products and production systems based on the production process. It is established by the product with its product characterizing parameters, *product* components (Bill of Material) and the related production *processes* (Bill of operation), the production *resources* executing the production processes with their characteristic parameters and the production process with its characteristic parameters. The product defines in its Bill of Operation all relevant processes that it requires for the manufacturing process. The production processes are defined by the capabilities of the involved production resources. Each production resource processes a set of products. The processes are used for the production of products and are executed on a resource. This forms a triangle as depicted in figure 2.2. The interlinking of the PPR concepts forms a graph, where the nodes represent the individual parts from PPR and the edges the here described dependencies and links. The PPR triangle will shortly be discussed with the already introduced use case: *fragile product*, from section 1.2.

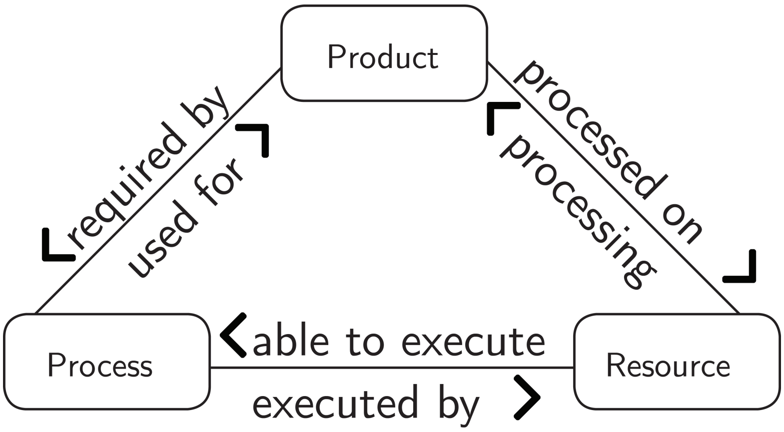


Figure 2.2: Product - Process - Resource (PPR) triangle.

A customer commissions the mass manufacturing of a product, which has fragile parts in it. Due to the complexity of the product, several processes like gluing, pressing and transport are required. Because of the fragile property, the product limits the acceleration speed of the underlying transportation system, for example, a conveyor belt, can actually perform. This makes not only the link between product and process but already shows that product properties can have limiting effects on processes. The product assembly itself is executed by production resources like robots, creating a link between product and resource. The resources provide special capabilities like gluing or welding, so they define the set of processes possible, closing the PPR triangle.

In figure 2.3 the concept of linked trees is visualized, by the use of a car. As can be seen, each different concept of product, process, and resource has its own tree. In each tree, the hierarchies of for example product parts or processes is depicted. The hierarchies can also be nested, as can be seen in the product tree example. The car consists of a chassis which in turn consists of a base plate and the car body. However, the hierarchies are not standing alone but are interlinked. This is visualized through the red lines from product to process and through yellow lines with arrows at each end for the process to resource relationship. Each process has multiple dependencies. For example, the welding process has as inputs the base plate and the car body. The dependency between the product and process tree also allows insights into the overall creation process of the production system, because the two inputs are hierarchically parts of the whole chassis.

It can be thus derived that the chassis is created by a welding process, where the two individual chassis parts are inputs. The same dependencies can be seen for the gluing process, which has the window and chassis as input and later the chassis and window are pressed together so that the window does not fall out of the chassis. Lastly, the engine and wheels are screwed onto the chassis.

On the right side of figure 2.3, the resource tree is depicted which is used to execute the processes. The dependencies, in yellow, give insights on how the process is executed and realized in the production system. For example, the welding process is executed by a specific laser welding cell. So in this dependency relationship, the process depicts the concept and is more abstract, whereas the resource becomes through iterative engineering more and more concrete until one cell is constructed and installed for process execution.

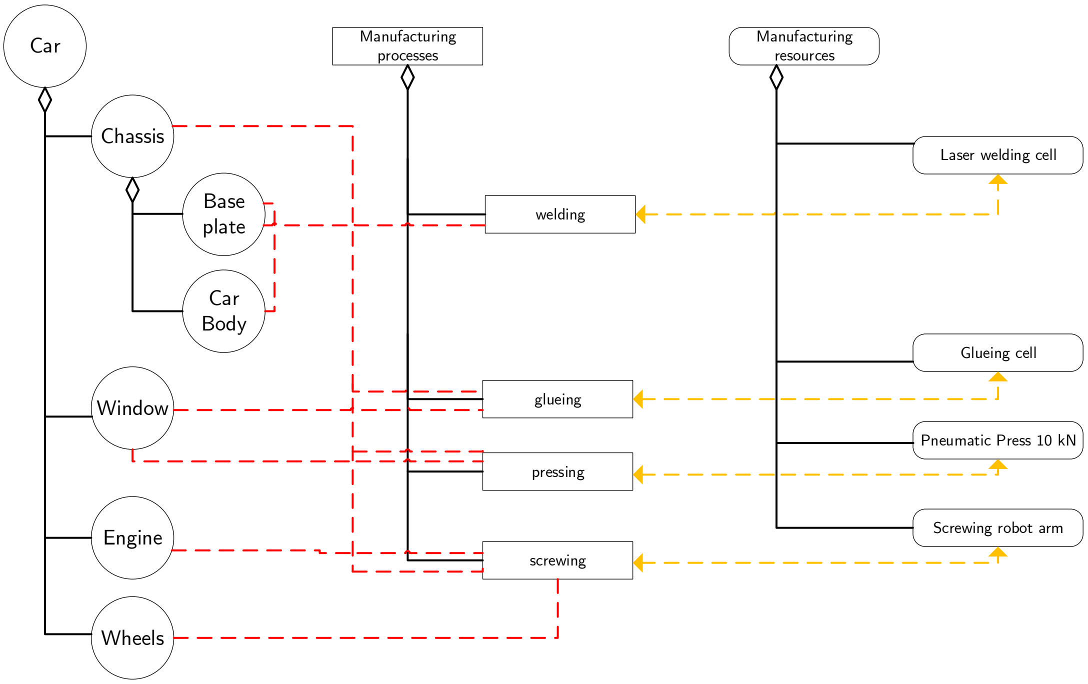


Figure 2.3: Example depicting Product - Process (PPR) - Resource interlinked trees.

Through the more practical example depicted in figure 2.3, can it also be seen that the interlinking of the trees results in a graph representation. The individual concepts of product, processes, and resources can be translated to nodes of the graph. Individual relationships like the hierarchies are then in a graph approach edges or arcs, however, it is important to note at this point, that the relations are of different types and require thus annotations in the graph to identify the different concepts of relationships.

The PPR concept is not only discussed by product engineers who want to describe the manufacturing of *products* with an executable *process* [36], but it is also subject to complexity management, where the complexity of product variants is investigated [57]

[66]. PPR is also of special interest in agent-based systems, as discussed from Lüder in [85]. Lüder emphasizes the importance of breaking down long-term goals into smaller goals. These smaller goals make it easier to adapt and migrate to new solutions which get developed. Balalaie [4] investigates the extended DACS approach from [85]. In that work, it is stated that there is a trend towards more flexible and smaller systems. These migrations make it possible to better adjust the systems in regard to the field and infrastructure they are in. This, in turn, allows for a better design with improved flexibility and leads away from fixed, monolithic solutions. A requirement to meet such a flexible design is the ability to retrieve data from different sources [10] [13].

2.3 Process Analysis

For analyses concerning engineering processes and to be able to follow the execution of tasks across several workgroups, is it essential to analyze existing processes, that the domain experts follow.

To maximize the outcome of a process analysis method, concerning a multi-disciplinary engineering environment, two aspects need to be taken into account: big-picture and detailed approaches. First, an overview of the process under investigation should be analyzed. A high-level analysis allows identifying different workgroups and their interfaces, where potential for improvement in terms of collaboration and coordination is possible. Also due to the fact, that engineering processes tend to be very complex due to their nature, is it a good approach to start from a higher level overview and then go into more detail. The second aspect a process analysis method should consider are detailed analyses. If only high-level overviews are constructed by a process analysis method, then there is the risk of missing out on important details, that are needed for improving and analyzing the underlying process.

Business Process Analysis (BPA) tend to provide a good overview of a process, whereas Engineering Process Analysis (EPA) methods focus more on intra discipline improvements and analysis. The following two subsections describe each approach in terms of building blocks for this work and limitations that will be address.

2.3.1 Business Process Analysis Methods

Rosenberger et al. [62] propose a context-aware BPA method, analyzing contextual requirements and defining contextual functionalities. Up until now, the elicitation of contexts for different disciplines was always accompanied by high efforts and try and error approaches for the developers. Standard approaches further do not meet the requirements imposed by [62] of context-aware systems, which is why the authors propose a three-phase process. The first phase is concerned with the determination of activities. Here a special focus lies on the identification of activities which can be enhanced by the presence of context and if they can be enhanced the question of how is tried to be answered. The second phase defines processes and system behavior if a context is present. Goal is it to reach consensus between all stakeholders. Possible changes triggered by process executions also need to be taken into account, as they may affect the stakeholders workflow. The third and last phase is context elicitation, where the actual context is elicited. For this, three well-known approaches are presented: category-based, model-centered and user-centered. The approach from Rosenberger et al. [62] is also able to handle different levels of abstraction, as it might be the case that different contexts are or activities range over multiple layers of abstraction. The identified different contexts for different work groups do not have any implications on other contexts, which makes it hard to use in an engineering process analysis, as PPR knowledge flows through all groups in the engineering process and they depend on the outcomes of each other. In such an environment, it is more the case of one shared context than of multiple independent contexts. The approach also lacks any visualization of the document flow through the process under investigation, which is a key requirement for analyzing any kind of multi-disciplinary environment. Further, are there no possibilities to identify interfaces between work groups, that are crucial for process analysis and improvements.

Santos and Alves [68] aim at balancing exploration and exploitation thinking in BPA methods. Explorative thinking focuses on crafting process visions that all involved stakeholders see them as an ideal and through intrinsic motivation, they try to achieve this future, improved state. Such an improvement path is executed via a sequence of transition states and thus rendering the current process obsolete. Exploitative thinking is in contrast concerned with the development of new processes, based on current shortcomings Ambidextrous organizations have capabilities to manage both analytical and intuitive thinking and thus encompasses two profoundly different features of businesses [68]. The approach from Santos and Alves builds methodologically on literature surveys, expert opinion interviews and follows the design science cycle from Wieringa [82]. Santos and Alves also propose a three-phase approach, just like the previous approach from Rosenberger et al. [62]. Each of the following phases is built up of several activities, tasks, techniques, and expected results. *Phase one* is concerned with the planning of the analysis, where the analysis team is defined, an understanding for the business environment is created, the scope of the analysis is defined and the process analysis plan is created. *Phase two*, execution and analysis, follows a waterfall like sequence. In the emphasis task, special focus is given to learning about the process activities, actors, clients, and present resources. The second task defines, organizes and categorizes knowledge which has been elicited up until this point. This includes the current as-is model of the process. In the third task ideas, which are related to process improvement, are generated and discussed. Task four closes phase two with prototyping ideas and selected improvement approaches. *Phase three*, closing and analysis, refines created documents and organizes ideas, insights and the to-be model. Through the detailed analysis, the results from Santos and Alves [68] allow to identify detailed execution steps, exchanged documents and a big picture structure of the business process. A limitation in this approach, however, lies in the strict forward flow of information and data, that is analyzed and all interfaces between stakeholder groups are well defined. This already known definition of interfaces, documents and tasks are defiantly ideal if possible, however in engineering processes under focus from this work, these interfaces are not known in advance. The approach is also concerned with high-level processes and documents, as was already identified as usual in the section introduction for BPA methods. Such a generalization might be harmful when analyzing complex tasks and processes, as crucial execution paths might be overlooked.

There are many more approaches for BPA and Vergidis et al. [80] have classified many of these existing BPA methods. The authors present the following major categories: performance evaluation, validation, verification, observational analysis and simulation. All categories contain several approaches but each one is limited, for example through high resource commitment in form of time and money, or through only very limited analysis support. Even though classical BPA methods allow the identification of stakeholders, their interest and executed process steps with inputs and outputs, these approaches do not have a special focus on engineering tasks and often are not suitable for multi-disciplinary engineering tasks. Only a handful of the existing approaches allows for further detailed analysis or process improvements, going beyond generic stakeholder, task or input/output artifact identification. In [80] it is also stated that for example, the simulation approach has problems with the interrelationship among process components. A further shortcoming of classic business process analysis methods is, that for the creation of a model, domain expertise is needed to have a clear vision of what data and in which level of detail the data should be collected.

Liu et al. [45] use for their BPA method an artifact centered approach. The from the authors presented solution approach focuses on using business artifacts for the detection of business processes. This strategy is motivated by the fact, that business processes often consist of a huge amount of business processes, where only a subset is relevant for further analysis. To filter out unwanted details and processes in an early phase Liu et al. simply follow business artifacts through their life-cycle from the creation, modification, dissemination and possible deletion. Similar approaches can be found for example in [19] and [34]. Even though these approaches allow to focus on the essential processes that are interlinked with artifacts, do they not work well if the artifact life-cycle and the executed process steps of a stakeholder differ greatly. For example, coordinating and collaborating tasks in a multi-disciplinary engineering environment might be lost, because they do not directly interact with artifacts. These tasks, however, are very important and have high importance to be elicited by a BPA.

2.3.2 Engineering Process Analysis Methods

Jäger et al. [37] 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.*" This is motivated by the fact, that in engineering processes parts of the work are done in collaboration from different domain experts and work groups, while other parts are done sequentially. A special focus on sequential execution lies on the dependencies, that are created when (intermediate) output results from one discipline serve as input results for another discipline. A major problem identified by Jäger et al. is, that a lot of necessary information on the actual engineering objects is not recorded or documented. If there is documentation available, it is very global and high-level and does not suit all disciplines. For example, mechanical engineers think more in spacial terms while automation engineers think in Programmable Logic Controller (PLC) code. In such a case it is not sufficient to have documentation on a global level, because this might lead to a situation where both domain experts depend on each other and the different semantics of common concepts do more harm than good, due to the missing finer documentation. To overcome some of these limitations presented in [37], the authors propose an approach, that backtracks from engineering documents to the involved stakeholders with their engineering processes and tasks and finally results in a cause and effect diagram. The cause and effect diagram allows following one specific document through its life-cycle and the engineering process. Even though this approach yields a strong connection between engineering artifacts, domain experts and their executed tasks, the interfaces between several domains is not considered or visualized in the end. Further is it not possible to see the priorities of tasks, that they have regarding (intermediate) results, because the approach is document driven and loses some vital process information bits.

The VDI 3695 standard [78] has already been discussed in this chapter (see section 2.2.1), with a focus on the general PSE background. Here the standard will shortly be discussed in terms us applicability for an EPA method. The fact that the VDI 3695 standard only identifies the four phases: acquisition, planning, realization, and commissioning, with no stakeholders involved, presents a problem of detailed EPA. It is not possible to gain any detailed insights from this high-level segmentation, excluding involved stakeholders, engineering tasks and most importantly engineering artifacts. Another drawback here is, that the interactions and hand-overs between individual phases are not described more in detail. As already stated in 2.2.1, does the standard not consider concrete improvement steps that could be taken by an EO.

Lüder et al. [47] and Schäffler et al. [73] both extend the VDI 3695 standard and present a more detailed EPA. The approach allows identifying individual workgroups, their tasks and also a description of engineering artifacts. In both works, there is no special focus on PPR knowledge representation, even though this could be easily incorporated. Through the presented EPA, an EO can develop engineering methodologies, identify risks based on toolchains, human involvement, information exchange and identify meaningful reuse capabilities of engineering artifacts. All of these possible outcomes are based on a four perspective approach, that includes functional, behavioral, organizational and informational aspects. The results of the investigation are in both works [47][73] represented in tables, structuring all the detailed information that was gathered. However, a shortcoming of this approach is that it does not consider how multiple workgroups could better work together and how interfaces between these workgroups look like.

2.3.3 Summary of Business Process Analysis and Engineering Process Analysis Approaches

The presented literature from both BI/Information Systems (IS) and PSE reveal many similarities on how the analysis methods are conducted. In many cases, the identification of stakeholders, work groups, artifacts, and executed tasks is done in a similar manner. Both approaches, high-level representations from BPA and more detailed approaches from EPA, are necessary and are needed to combine the best of both worlds to investigate the engineering process of an EO in more detail. However, the foci and outcomes of the two different disciplines differ greatly. BPA methods tend to be more focused on a big-picture like approach, while EPA methods are more concerned with the inner workings of workgroups and intra process improvements. The main research gap in both disciplines lies in the absence of focus on exchanged engineering knowledge between work groups. Even though the importance of identifying and optimizing interfaces between workgroups is known [37], little to no concrete outcomes represent interfaces between workgroups. This lack of focus on exchanges between workgroups can often be the source of missing PPR knowledge. The absence of such representations highlights already a risk in traditional PSE and is much higher when considering flexible manufacturing systems according to requirements from the Industry 4.0 vision or other related work like [48].

2.4 Process Modeling

This section presents first different existing process modeling approaches 2.4.1, to then investigate how some of these visual languages could be used to represent the outcomes of an EPA 2.4.2. The last subsection discusses possibilities how to model and express PPR knowledge.

2.4.1 Modeling approaches

There exist many different modeling approaches and languages. Many focus on different areas of knowledge representation [80] and each has its strengths and limitations. In this subsection, the most relevant languages, to either express the outcomes of an EPA or to represent PPR knowledge, will be investigated. Textual or mathematical models will not be considered, as they are not the main area under investigation in this work.

General requirements and elements needed for expressing the outcomes of an EPA method are: Tasks and a control-flow with possible logical operators, manipulating the control-flow, need to be present, to be able to express a sequence of engineering tasks that a domain expert executes. It should also be possible to express the multi-disciplinary environment of the engineering process. Documents or artifacts must be represented as individual concepts, which interact with tasks through a document-flow.

For expressing PPR knowledge is it crucial to have different concepts for product-process-resource, for an easy understanding and semantical differentiation. It is also here important to have control-flow elements and possible operators. Key capabilities are a) structuring elements hierarchically b) interlinking the concepts of PPR for expressing a graph like structure.

The languages under investigation, in alphabetic order, are: Business Process Model and Notation 2.0 (BPMN 2.0), Extended Event-Driven Process Chains, Formal Process Description (FPD), IDEF0, Petri Nets, Sequential Funciton Charts (SFC) and Systems Modeling Language (SySML)-Activity Diagram (AD). The choice for exactly these languages is based on discussion with domain experts and practitioners, as they represent the most promising approaches for expressing PPR knowledge.

Business Process Model and Notation 2.0

The BPMN 2.0 standard, aims at providing a standard notation to model business processes and have an easily understandable language for experts and non-experts alike [3]. This goal should allow business analysts to have an intuitive yet expressive enough notation to also express more complex process semantics. The main diagram beside, conversion, collaboration, and choreography is concerned with modeling processes. Process diagrams, in a very minimal form, as they will be used in this work consist of: *processes/tasks*, *documents*, *control flow*, *sequence flow*, *start and end events*, *gateways* and *swim lanes*. All of these elements are depicted in figure 2.4

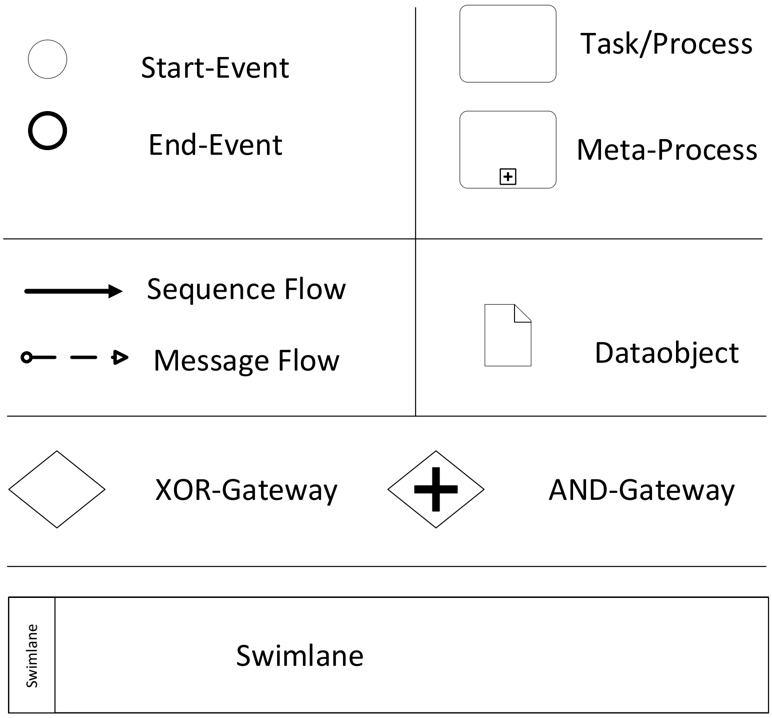


Figure 2.4: Non-exhaustive list of BPMN 2.0 elements [3].

BPMN 2.0 has further elements derived form the basic concepts presented in figure 2.4. For example exist different events like message receive, time based events, error events. Further, can all these elements be annotated as either throwing or catching. It is also possible to specify a certain task type like manual, reoccurring, receiving or sending. These more advanced concepts are especially useful for process enactment, as there are several BPMN 2.0 engines available that are able to interpret the underlying XML format, that is produced by a BPMN 2.0 diagram.

Even though BPMN 2.0 has documents and an explicit document-flow, the concepts are held very basic and generic. For usage in a EPA is there further research needed, if the basic concept of documents is sufficient or if more advanced interaction models are needed.

For PPR modeling BPMN 2.0 lacks the distinction of the three basic types and does not provide any interlinking between these concepts. An approach that could circumvent this lack of explicitly could be to use swim lanes and documents as proxies for products and resources.

Extended Event-Driven Process Chains

The extended EPC (eEPC) approach is similar to the previously introduced BPMN 2.0 notation, and provide a flowchart-like notation which allows defining business processes, involved stakeholders in form of organizational units and documents [70]. In contrast to BPMN 2.0, the eEPC approach requires annotating each individual task with an organizational unit, making the diagrams more heavyweight and in larger examples more complex. In figure 2.5 a small example of an eEPC diagram is given.

Red hexagons, *events* like receive goods, goods are ok, goods are not ok, either trigger functions or are a result of a function. Each eEPC diagram starts with such an event and also ends with an event. Green squares with rounded corners, *functions*, perform specific actions like check goods or ship goods back to the seller. Events and functions are connected via arcs, representing the control flow. In the control flow, there can be circles, connectors, present. Common logical connectors allow the following operations: AND, OR and XOR. The blue rectangle with white text, *bill of landing* in figure 2.5 represent data or documents present for the function. The yellow ellipses, *organizational units*, can be used to depict stakeholders or workgroups in a process.

The eEPC model is easy to understand even for non-experts, through the use of different visual elements for different concepts. Also, has eEPC many concepts present that are useful to represent an outcome of a BPA or EPA. However, it is not possible to express PPR knowledge or classify documents regarding their content. These two limitations make the approach not so suitable for the aims of this work.

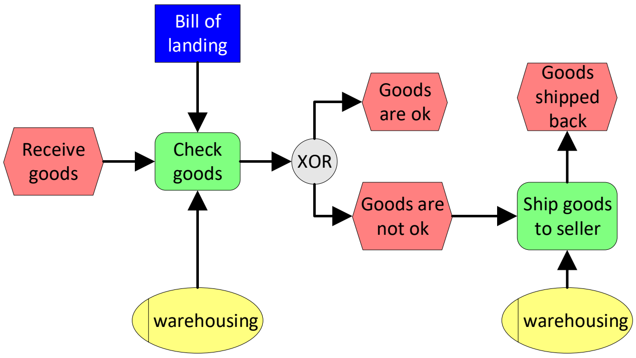


Figure 2.5: Extended Event-Driven Process Chain example.

Formal Process Description - VDI/VDE 3682

The FPD was developed to model technical processes in engineering terms and throughout the life cycle of technical systems [79]. The core element and concept is the technical *process*. A process transforms a state *ante* into a *post* state. *States* are according to the VDI/VDE 3682 standard, to general and thus the terms *products* and *energy* are used. Both elements, products, and energy, are represented with different graphical elements. Products are represented using *circles*, while energy is depicted as *diamonds*. Products and energy are connected to processes via directed arcs and represent the flow. *Technical resources* are represented using rectangles with rounded corners, which are used for executing a process. The link between technical resources and processes is indicated via double-headed arrows and dotted arrow lines. In the FPD is it also possible to model system boundaries, with large dashed frames, surrounding the system under consideration. An example using all these elements is depicted in figure 2.6.

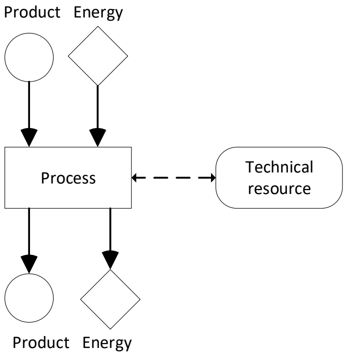


Figure 2.6: Simple graphical process description in FPD based on [79].

The FPD is the only modeling language that actually provides different elements for product, processes, and resources. Not only is it possible to have different elements for

PPR, but the concepts are also interlinked as was described in section 2.2.2, depicting a directed graph, where the nodes are the PPR elements and the edges support the semantics on how these nodes stand in relation to each other. Further is it possible to create meta-systems in the FPD. These meta-systems are just like meta-processes in BPMN 2.0, and allow to decomposition a process into smaller pieces, allowing modeling on different hierarchy levels. One drawback from the FPD is that there are no gateways for creating control-flow decisions. This issue was addressed in [17], where the authors proposed notations to express parallelism and exclusive operations. Even with this extension the FPD lacks explicit gateways that are clearly and visually distinguishable from the normal control-flow. This approach is a good basis for PPR modeling, but not so good for representing the outcome of an EPA as there are no documents or organizational units present.

IDEF0

To model manufacturing systems, many approaches are present. The IDEF0 approach, developed by the U.S. Air Forces [26], assists in organizing system analysis and provides effective communication means between the process analyst and the customer[30]. The IDEF0 approach tries to achieve this goal by a simplified graphical notation. The main visual elements of an IDEF0 diagram are rectangles depicting *activities/functions* and directed arrows which can be used for forking and joining and representing: control, input, output or mechanisms/resources. In figure 2.7, the basic elements and syntax of an IDEF0 diagram can be seen.

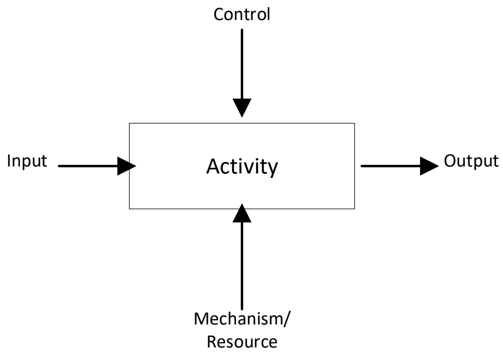


Figure 2.7: IDEF0 syntax for the basic building block of an activity.

The system analysis standard has not only a limited number of distinct graphical elements but also very few distinct elements. This makes it easy for non-experts to catch up the notation and understand diagrams even while they are drawn. But especially due to the fact that IDEF0 focuses on a simple syntax and semantic, run diagrams based on this notation into problems when they are scaled to several tens or hundreds of activities. It is, for example, hard to follow one specific input through a complex process and see all the transformations that this input undergoes. The models become further "crowded" with arrows joining and leading to activity boxes, making it hard to pinpoint specific details. Also is the lack of different elements a problem to express PPR knowledge because the arrows are not distinguishable graphically and only textual annotations are not expressive enough. Thus, the approach is neither suited for modeling EPA outcomes nor to model PPR knowledge, as for that many concepts are missing.

Petri Nets

Petri Nets are based on a mathematically defined foundation and provide a graphical notation in form of a directed bipartite graph. The petri net graphs describe discrete event systems and can thus be used to model business processes [77][59]. In [59], system modeling with petri nets is described as follows: Events are mostly modeled with transitions (squares). Pre- and postconditions of these events are then expressed by places and are only then considered as fulfilled if the places (circles) contain any tokens (black filled circles). Directed arcs connect each transition with all the places that represent postconditions for the event under consideration. Figure 2.8, represents such a petri net with the initial marking $m_0 = (1, 0, 0)$, meaning that only the place p_1 has an initial token present, fulfilling the precondition of t_2 and allowing that transition to fire. The firing of t_2 would result in a marking $m_1 = (0, 1, 1)$, which would fulfill the preconditions for t_1 and also for t_3, resulting in the marking of $m_0 = (1, 0, 0)$.

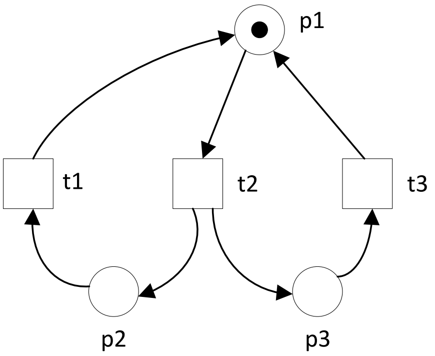


Figure 2.8: Petri net example based on [59].

Petri nets are often used for formal verification or token engines, as they are well defined and easy to implement. However, even in a small example of a process, petri nets can get very complicated and non-experts find them hard to understand. In terms of expressing PPR knowledge, petri nets could be used for verifying a process flow. However, there are no concepts for products or resources which makes it hard to fully use this approach. Also to model PPR knowledge in an EPA method, is not possible as documents or organizational responsibilities are missing from the approach.

Sequential Function Charts (SFC)

SFC diagrams are define in part four of the IEC 61131-3 standard[20] and can be used to structure and drive PLC [32] [12]. A typical application domain for this language is PSE, as it is able to control industrial processes where security is a key challenge. The main concepts of a SFC diagram are *steps* and *transitions*. Each diagram has exactly one initial step that "starts" the execution of the model. Following the steps are *actions* which are essentially instructions to programming languages or other SFC diagrams. These actions enable the execution and layering of SFC diagrams. The diagrams also have the concept of *guards*, which are conditions regulating the execution of actions or steps. A simple example of a SFC diagram can be seen in figure 2.9.

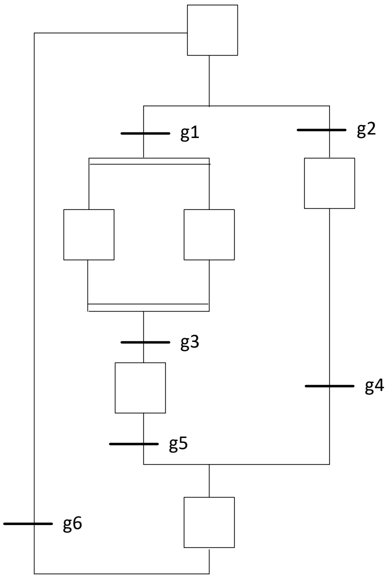


Figure 2.9: SFC example based on [12].

The top square in figure 2.9 represents a step, whereas the remaining squares are all actions. Transitions are the vertical lines connecting the actions. As already described is the execution of an action based on conditions, guard g1 to g6 in the example above. if the guard g1 is fulfilled, the two actions are executed in parallel, incorporating the convergence and divergence concept. Divergence means the branching of one execution path into multiple transitions, thus allowing for parallelism. Convergence is then the opposite concept, namely joining previously forked branches and synchronizing the execution again.

It is evident that SFC diagrams focus mainly on the execution of PLC code. Even though the approach comes from a PSE community, does it not provide any useful or necessary elements to be able to represent the outcome of EPA approaches. For modeling PPR knowledge, can SFC diagrams also not be used because there are no concepts for products or resources.

Systems Modeling Language - Activity Diagram

SySML as it is described in [28], is a general-purpose, graphical modeling language for representing systems that include a combination of hardware, software, people, facilities or natural objects. SySML is a child language of the Unified Modeling Language (UML) 2 standard and as such has a great overlap with its parent. Nearly 80 % of the two languages are identical in syntax and semantics. However, SySML extends some new diagram types which are not present in the UML 2 standard. These new diagrams are Requirement and Parametric diagrams. Block Definition diagrams, Internal Block diagrams and AD have been modified by SySML.

An AD is essentially a directed graph, where the nodes represent the activities and the edges represent the control-flow. The control and optional data-flow represent the sequence in which activities are executed. Optionally, an AD can have AND or XOR splits, as well as parameters for input/output modeling of activities as well as pre- and postconditions. SySML extends on this basic definition and introduces control as data, continuous systems, and probabilities. The control as data extension allows to cancel or abort an already executing activity, which is not possible in UML 2. The concept of continuous systems introduces restrictions on the rate at which entities flow along the edges. Probabilities can be annotated at gateways and thus influence the execution paths taken.

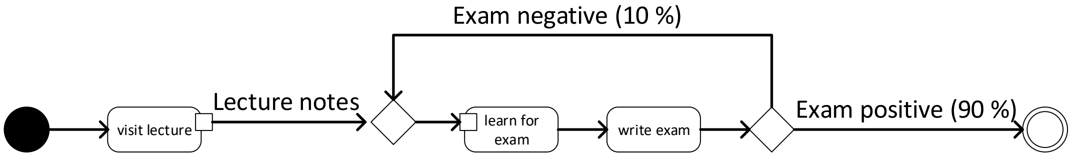


Figure 2.10: SySML Activity-Diagram with probabilities and data objects.

SySMLAD has many concepts that can be used for modeling an engineering process. The document flow can be visualized via pins and input/output parameters. The central element of an activity is present and the control-flow can be manipulated with the use of gateways and also probabilities.

SySMLAD could also be used to express and model PPR knowledge. Activities can express processes, input/output parameters can be used to express the products and their flow. A resource concept is missing to express basic PPR knowledge. Further, there is a clear distinction and division of the PPR concepts missing, meaning that products could only be modeled by the use of activities and not as a standalone first-class citizen.

2.4.2 Data Processing Map Visualizations

Section 2.3 presented methods from both BI/IS and PSE disciplines which are focused on gathering information on the engineering process. Both communities have developed different approaches to process qualitative knowledge in form of text into more visual notations. This subsection now focuses on these approaches an how they can be used to represent engineering artifacts. The engineering artifacts which get elicited from a BPA or EPA method, often contain knowledge on PPR information. Thus it is a requirement for existing approaches to be able to represent or at least classify which engineering artifact consists or contains which aspect of PPR knowledge.

The previously described IDEF0 approach [26][60] is commonly used in the engineering domain to express dependencies or task sequences of a process [41][44]. The basic semantics and syntax allow expressing input and output artifacts as well as controls and stakeholders. To be able to model an engineering process, IDEF0 provides a solid basis. However, when the models grow and more and more involved stakeholders need to be represented which more complex process execution paths, IDEF0 lacks more advanced concepts like swim-lanes or explicit gateway notations. Already depicting a few ten tasks and scaling these presents a barrier for modeling and also makes the models hard to understand.

Lüder et al. [47] and Schäffler et al. [73] presented besides an EPA approach also implicitly a notation form. To classify and organize the gathered knowledge, the proposed approaches make use of structured tables, referencing each other if need be. This allows for a fine-grained and detailed representation of engineering knowledge, however, working with a multitude of different table forms for multiple stakeholders becomes cumbersome very fast.

Eventdriven Process Chains (EPC) [70] and BPMN 2.0 [3] as well as the UML standard [27] are well known options provided by the BI and IS communities. All of these approaches provide similar capabilities [84][65][29], and differ mostly in the use of certain concepts like explicit or implicit representations of logical operators. Through this fact, choosing an appropriate representation often then comes down to preferences and the understanding of involved non-experts. Merunka [51] investigated the UML standard and found out that it has no means to represent product or process knowledge, neither in one nor in several combined diagrams. EPC, extended with data, resources, time dimensions and probabilities are called eEPC [70]. BPMN 2.0 and also eEPC are often used in the BI domain for modeling business processes. Both concepts have incorporated similar approaches. While eEPC is more explicit and requires annotating for example organizational units for each involved task, provides BPMN 2.0 an aggregation of tasks per domain through the swim-lane concept.

To overcome limitations of individual languages, different works like:[40], [33] and [51] have proposed combinations of languages. These combinations follow a "best-of-breed" approach, where the most suitable concepts are used for the task at hand. The proposed approaches do provide flexible and detailed notions of processes, however, the complexity of these models increases with each new modeling concept introduced and combined. Through these combinations and the increase in complexity, non-experts have a high learning curve to master before they are able to understand the depicted processes and knowledge.

PPR knowledge representation in an engineering process is of no concern for existing visual representations. The flow through an engineering process, dependencies between artifacts and tasks or interfaces of stakeholders are never explicitly modeled or represented. The languages do however build a good foundation which can be used to close this gap. One approach could be to use BPMN 2.0 and then build custom extensions to express the missing concepts. This was already done in [15] with a focus on the health care domain. The approach and domain can easily be mapped to a PSE context, as both domains involve process sequences, document flows and are from nature multi-disciplinary. For expressing PPR knowledge in BPMN 2.0, the limited options of BPMN 2.0 for representing interactions and documents in general need to be overcome, to allow possible classifications an prioritization of PPR knowledge.

In all the presented literature up until now, no one named the concept of treating engineering artifacts besides engineering processes as a first-class citizen. In this work the explicit and equivalent position of engineering artifacts to processes is named *data processing map*. The need for explicit and domain-specific representations of data and processes is already motivated in [56] and is thus one focus of this work.

2.4.3 Product, Process, Resource (PPR) Modeling

All three concepts of PPR can be composed of inner elements, meaning that a product can consist of multiple product parts that are assembled together and make up the final product. An example would be a pen consisting of the outer shell, the refill, the spring mechanism and so on. The VDI 3682 standard [79], introduces this concept of recursive composition of individual concepts. The standard is further the only visual representation that has three distinct elements to express, product (parts), processes and resources and will be described in more detail in section 2.4.

The PPR concept is applied in several different ways, when focusing on the production system life cycle, including the production system engineering phase. Several approaches like Domain engineering [38], SkillPro [58] and Plug-and-Produce [71] allow the modeling of production processes that are required to create a product. These approaches allow the identification of appropriate production resources for the execution of processes and finally the creation of the product. The selection of these resources can either be done at production system engineering time or production system runtime. In [52] the dependency between products and production systems, over their linking processes, is utilized to analyze the complexity of these mutual dependencies. Such analyses could possibly allow methodologies for better complexity management and thereby risk reduction within product and production system co-engineering.

These examples highlight the broad scope of PPR based approaches. Nevertheless, there are only limited modeling means especially developed to handle PPR knowledge in the production system life cycle. As in most cases in [25] existing engineering tools and their internal data structures are applied for PPR modeling. Explicit notations are given for example with the already mentioned VDI 3682 [79] and further within the ISA95 standard. The ISA95 standard [21] indirectly allows to represent the PPR concept but is more concerned to describe the interfaces between an Enterprise Planning System (ERP) and a Manufacturing Execution System (MES). The goal of the ISA95 standard is to better describe and transfer production order relevant information into the manufacturing system, producing the individual parts. Further, comes the standard more from batch processing and not so much from discrete manufacturing, which we focus on in this chapter. Thus, we do not further consider this option for a solution in this research.

2.5 Product, Process, Resource (PPR) Knowledge Persistence

This work builds upon Rowley [63] and the proposed wisdom hierarchy to define the term *PPR knowledge*. Before defining knowledge, the terms of an engineering artifact, data and information should be discussed. Engineering artifacts are the simplest representation form present in an engineering process. Artifacts can, but must not be in digital form, containing data and possibly also knowledge. However, artifacts are not easily or automatically processable. Data, defined via a data model, refers to different kinds of symbols ranging from PROSA text to complex 2/3 D drawings of engineering machinery and also tables containing product to product family mappings. The data model also defines data types that are applicable for certain attributes, an example is a table where the columns define data types like integer and possibly value ranges like the positive numbers. A PPR data model could define which elements in a graph can be represented as nodes and which different edges exist, defining syntax and semantic. Engineering information is one level higher in the wisdom pyramid and defines who has access to underlying data. Information can, like data, be processed with machines and to some degree even algorithmically. On the top of the wisdom pyramid is finally knowledge. Knowledge is backed by underlying concepts and provides applications for the engineering information and data PPR knowledge uses the already described PPR concept form Schleipen [71] in section 2.2.2. Further is PPR knowledge used to express a) all success-critical attributes like parameter settings for production machines or process configurations and b) relationships such as dependencies between PPR concepts or hierarchies in the individual PPR aspects.

The term persistence is not as strictly defined as it in the database community. Persistence expresses more the application of persistence solutions to be able to store and later retrieve PPR knowledge without making any restrictions regarding the actual implementation in one, or multiple technologies. This section discusses the most relevant and promising approaches to store PPR knowledge namely: relational databases, NoSQL databases, and AutomationML (AML) files. Subsection 2.2.1, has already established that PPR knowledge with its concepts and relationships essentially is a graph or a network. Based on this assumption the selected technologies seem adequate for further investigation.

The relational database approach with many different technological implementations, has been applied for many years. From the 1970s relational databases have gained invaluable production experience [54]. Relational databases represent data in a fixed schema, based on tables with columns and rows. The tables represent the data model, where each column has a specific data type and possibly value range. Columns represent individual entries that are identifiable. Many data-intensive storage and retrieval applications have been built around relational databases [81], which is to some part surely the case due to their general efficiency. However, relational databases lose efficiency when the underlying data is strongly interconnected, with many spanning relationships, requiring possibly a large number of joins to retrieve data [81]. SQL [23], a goal-oriented query language for relational databases working on the fixed structure of tables is commonly known and one of the key success factors for this approach. A drawback from relational databases is that they follow the fixed schema approach. Engineering artifacts often do not follow such a predefined fixed structure, as they might vary from project to project, domain expert to domain expert or customer to customer.

One key difference between relational databases and NoSQL databases is, that NoSQL databases allow flexible data models without imposing restrictions on the schema (also called schema-less models) [74]. In [81] Vicknair et al. list several criteria to check if a NoSQL is an appropriate solution. The items to check against are:

- Having tables with lots of columns, each of which is only used by a few rows.
- Having attribute tables.
- Having lots of many-to-many relationships.
- Having tree-like characteristics.
- Requiring frequent schema changes.

As PPR knowledge consists of product, process and resource information, and due to the fact that these concepts are all interconnected, a high number of many-to-many relationships are formed. This fact fits the third criterion from Vicknair et al. and so does also the fourth and fifth one. The individual concepts of PPR are as presented in the VDI 3682 [79] also tree-like, hierarchically structured and as pointed out earlier do the schemes often depend on project or customer specific circumstances.

NoSQL solutions can be categorized into four major categories namely: key-value, column-oriented, document, or graph databases [74].

Key-Value databases are designed for a continuous growth of inconsistent values, that are made up of a key and the corresponding value [74]. A special focus of key-value storage systems lies on fast queries for large volume data. Unlike block storage systems, which store data in large blocks, focus key-value data structures on small data objects which make the whole approach more easily configurable. Examples, where these systems are heavily used are online games, online shopping or all kinds of small web-based systems that have to store small records but in high numbers.

Column-oriented approaches are especially suitable for vertically partitioned, contiguously stored, and compressed storage systems [74]. This kind of NoSQL databases provides eventual consistency, making them a good choice for applications that need high reliability and availability. Examples are data-warehousing, large management systems like libraries, or analyses of aggregated data.

Document-oriented databases are similar to the *key-value* approach and also store their information in such tuples. However, document-oriented databases allow for more complex queries and hierarchical relationships [74]. The absence of a fixed schema does not hinder the indexing of data entries. A key aspect of this approach is, that meta-data is collected, which allows further optimizations and storage as documents. Well-known examples are user profiles on diverse social media platforms.

Graph-based approaches are the best choice to store not only data but also the relationships between these data entries [74]. A key aspect of graph-based databases is their capability to persist complex business processes along with interconnected data and relationships. An example where graph-based databases are used is, for example, a recommendation system, building on the low latency characteristics of such databases.

PPR knowledge, consisting of many attributions and relations, forms a graph-like data structure, which would fit well to a graph-based approach [81].

In both [67] and [54] are arguments made that it is not an exclusive choice when thinking about data storage solutions, but that also combinations of different kinds of approaches can be used. Fowler and Sadalage [67] coin thus the term of *polyglot persistence*, which means that several data stores, with probably differing characteristics, are used for one technical realization of a system. A "best-of-breed" approach could help to meet the requirements of a flexible, yet efficient PPR knowledge persistence solution. As in most cases, the usage and combination of multiple technologies lead to a higher degree of complexity regarding implementation, testing and operation.

AML, allows to express PPR as a concept and further is also capable of representing complex production system dependencies and issues in Extensible Markup Language (XML) like formats. In AML is it besides the PPR knowledge representation also possible to represent this knowledge for data exchange and logistics storage in small production systems companies. As however AML files can rapidly grow in size, it can easily become hard to process them efficiently, even for medium-sized production systems. A medium sized production system in this context consists of five to ten thousand signals, which in an AML file might already result in 20 to 50 MB.

CHAPTER 3

Research Issues

Chapter 1 framed Production Systems Engineering (PSE) as the underlying context for this work with its multi-disciplinary engineering (MDE) environment. In particular section 1.1 motivated a) the need for combining well known process analyses methods from both Business Informatics (BI)/Information Systems (IS) with PSE approaches and b) a visual approach to model Product, Process, Resource (PPR) knowledge. Also motivated by chapter 1 was the identification of groups of data, based on use cases, for PPR knowledge persistence. Chapter 2 provided background information on the context, presented strengths and limitations of existing solution approaches for both Engineering Process Analysis (EPA) methods and modeling languages, as well as current persistence solutions that are valid options for PPR knowledge persistence. The main three main research areas that this work addresses, are in this chapter aggregated into research issues.

Section 1.2 discussed figure 1.1, which visualizes the three main challenges that the work wants to address. Challenge one "insufficient explicit PP+R knowledge" and challenge two "no backflows", can not directly be addressed. To be able to address these challenges, first, an EPA has to be conducted, which is also one major outcome of this work, which is described in section 3.1. The adapted EPA method is derived from the existing research gap, discussed in section 2.3. Both communities BI/IS and PSE do not allow to fully express an engineering process, so that analyses for PPR knowledge can be made.

Section 3.2 addresses the research approach for challenges one and two. There the methodological execution steps, for investigating existing solutions and their capabilities and limitations in regard to expressing PPR knowledge, will be presented. A further research approach is to investigate possible adaptations to existing modeling concepts and how use cases from domain experts can be supported by a possible PPR modeling language.

The third research issue in section 3.3, builds on the identified use cases for PPR modeling and for PPR knowledge analyses. Research in this area focuses on a small set of requirements to persist PPR knowledge and so addresses challenge three, "limited quality assurance", from figure 1.1. Also, a focus is the identification of different groups of data, that are present in a MDE process. Further is it an interest what common use cases are, and what requirements these use cases impose regarding possible data storage technologies.

The here identified research issues will now be presented in more detail with the respective research approaches. The following sections are structured as follows: first, the Research Issues (RI's) including Research Questions (RQ's), are presented and then the research approaches are discussed in more detail. As some RI's and RQ's are similar in their nature, the research approaches are also to some degree similar. This circumstance is why a detailed description, of methodological approaches, is only given for the first time a research principle is presented. Other research issues that are similar in their nature then only reference to the methods already introduced.

3.1 RI-1: Engineering Process Analysis and Representation

Current approaches in BI/IS that perform Business Process Analysis (BPA), strongly focus on the big picture, whereas EPA tend to be very detailed and focused on intra discipline improvements. This situation was already discussed lengthily in section 2.3. An outcome of PSE organizations, to represent a discipline-specific work-flow, is to use extended EPC (eEPC) diagrams. These artifacts, depicting a current engineering process for one domain, try to formalize the standard engineering process but lack any indication of interactions with other work groups. A further problem is, that the creation of such work-flow diagrams does not follow a previous, formalized, standardized, or repeatable process that yields these diagrams. This means, two stakeholders who model their individual discipline are very likely to not use the same concepts for information gathering or depicting their processes. One stakeholder, for example, an engineering manager, is interested in the process of a discipline and would like to have these diagrams at hand. Further he wants to rely on common concepts that all work-flows can be compared. Through the lack of a non-repeatable process the outcomes and artifacts represent often very different quality levels and are not comparable to data flow or interaction analyses. To address this issue, the main question that needs to be asked is:

RQ1a: What adaptions or combinations of business/engineering process analysis methods allow overcoming the limitations of the communities regarding product/ion-aware engineering processes?

Major outcome of this research question is a **repeatable** two-phase process, focusing on product/ion-aware engineering processes (PPR EPA). The proposed PPR EPA, yields mainly documents and detailed process descriptions. The main benefactors of this RQ are stakeholders involved in the engineering management like, for example, engineering managers and quality assurance personnel. Engineering managers are interested in the engineering process, investigations in the current state of project specific engineering processes, and engineering artifact representations. Quality assurance stakeholders also benefit from this outcome, as there is a repeatable process, allowing them to propose and implement process improvements and then compare the previous process with the improved one.

RI-1 also focuses on representing engineering process analysis outcomes. Even though RQ1a, allows a repeatable process, there is only limited visual representation of the engineering process currently available. Also both BPA and EPA methods propose visualizations. However, major drawbacks as discussed in section 2.4 is the missing possibility to represent and indicate PPR knowledge. To address this gap a second question needs to be raised and subsequently answered:

RQ1b: What adaptions or combinations of business/engineering process notations allow overcoming the limitations of the communities for representing stakeholders, processes, and documents that may represent PPR knowledge?

As already mentioned discipline specific eEPC diagrams lack the visualization of interfaces between work groups. Even though BPA and EPA methods do not fully allow to visually represent PPR knowledge, they form a solid basis for extensions to indicate PPR knowledge that flows through an engineering process. Thus, the main outcome of this RQ is an extension to the Business Process Model and Notation 2.0 (BPMN 2.0) standard, and a visual representation of the engineering process PPR Data Processing Map (DPM). The introduced extension focuses on the limited concepts of documents and tasks that are currently available in BPMN 2.0. Tasks are extended by two new concepts. The first one is to indicate what a process needs in regard to PPR knowledge and what he currently receives. The second is a priority level, how critical it is for the task execution to receive PPR knowledge. The extensions for documents focuses on classifying which concept of PPR is represented in an engineering artifact. Main benefactors are the same as for RQ1a, namely the engineering management and also quality assurance.

The outcomes of both presented RQ, are methodologically based on known research approaches. Several approaches are combined to yield these outcomes and are now presented in more detail.

Research Approach

To be able to answer the proposed RQ's in this first research issue, a set of methodological approaches needs to be executed. These execution steps provide guidelines and building blocks where this work can build upon and guarantee a safe and sound research outcome. The proposed research approach is depicted in figure 3.1, an IDEF0 diagram, and will shortly be described.

The four blocks receive inputs (arrows coming from the left-hand side), execute an action and produce an output (arrows leaving on the right-hand side). Coming from the bottom

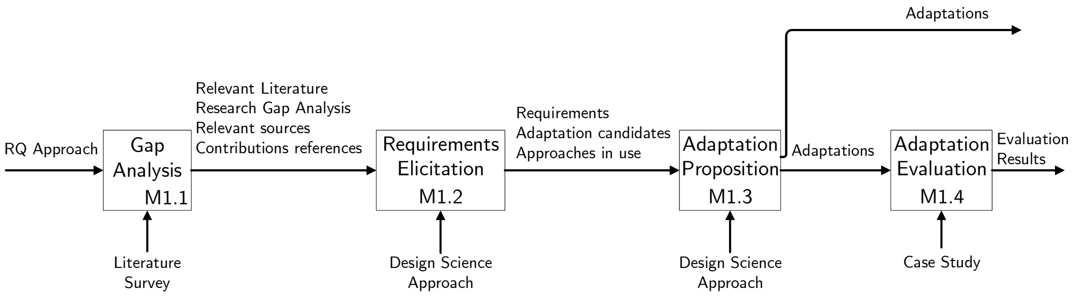


Figure 3.1: Research approach that research issue one follows.

are arrows indicating mechanisms/stakeholders or research methodologies. All blocks have an identifier starting with an M and then the numbers one to four.

M1.1 - Gap Analysis
The methodological approach for this research issue starts with a general concept for the RQ's. With this input, a gap analysis can be made, which builds on a literature survey. A **literature survey** is used because the here conducted research addresses existing works and thus the underlying process of how literature got selected and processed throughout the work will be briefly discussed here. The used methodology is based on an already known and widely used standard systematic literature survey. The major steps of this literature search will be discussed and thus will legitimize the selected works.

The underlying process for this literature research is adapted from [5][43][42]. This is an established and well-known research methodology for gaining knowledge on the state-of-the-art and the state-of-the-practice in a specific research area. In this work, the basic principles and guidelines according to this research methodology have been applied. The efficient execution was possible because of the presented process (provided in [39]) on investigating BPA and EPA methods. Figure 3.2 presents the basic adapted steps taken, which are based on [39].

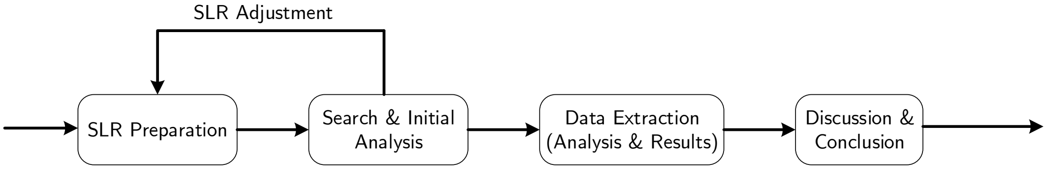


Figure 3.2: Adapted basic literature survey process based on [39].

The basic steps of the literature review taken are as follows:

1. The preparation phase includes a) basic definition of the research questions b) definition of keywords, c) identification of candidate sources, and d) the definition of the protocol.

2. Search and Initial Analysis. This phase includes a first (initial) search and a basic analysis of the findings. Because several sources have not been included in the beginning and the research questions have to be refined and modified, some adjustments have to be implemented in a second loop. Results include the final research questions and a finalized search and analysis process.

3. Data Extraction. The results have been analyzed according to the research questions and the results have been generated.

4. Discussion & Conclusions summarize the main findings and identify future work as a major outcome.

The search process includes Keyword Definition, Search String Definition, a set of sources, and the detailed search process. To answer the two RQ's, different keywords and search strings need to be defined, which will be presented in the according section of the result chapter 4. The set of sources, however, is identical for all literature surveys undertaken, which is why they are presented in table 3.1 below. All of the selected digital libraries were chosen, because most of Software Engineering (SE) conferences are indexed by one of these libraries and also many PSE specific venues publish and index their work there. Due to the fact, that the target venues publish their most relevant work in on of these catalogues, is the set adequately diverse and complete.

Source	URL
CatalogPlus	http://catalogplus.tuwien.ac.at
Scopus	https://www.scopus.com/home.uri
Google Scholar	https://scholar.google.at/
IEEExplore	https://ieeexplore.ieee.org
Elsevier	https://www.elsevier.com/
ResearchGate	https://www.researchgate.net/
Springer	https://link.springer.com/

Table 3.1: Relevant sources for the literature research process.

The first entry in the table 3.1, was the primary starting point for the literature research. The catalogPlus from the technical university of Vienna provides a vast amount of research papers which are hosted across well-known research portals. Scopus is also a search catalog, referencing many different research libraries and providing easy search options based on keywords and terms. Some works were not listed in either the first nor second catalog, which is why Google Scholar was added as a third large search starting point. More detailed analyses were done with the remaining and more specific search libraries of individual publishers.

As many libraries yield similar or the same papers, removal of duplicates was necessary. Through the use of JabRef [1], a literature management system, this step is fairly easy.

Further analysis of the found papers focused on only keeping the most relevant sources for actual reading.

The literature survey yields the following outcomes: relevant literature, list of relevant sources, contributions of relevant references, gap analysis with strengths and limitations of references.

M1.2 - Requirements Elicitation
The results of the gap analysis can now be used as an input for requirements elicitation. From literature an initial set of requirements can already be extracted, however, the proposed outcomes of this research issue need to fit to the individual requirements of the involved stakeholders of an engineering process. This is why a **design science approach** is used, to also elicit the requirements of domain experts and be able to propose adaptations. The design science cycle [82], provides an iterative method, to elicit requirements, design solution approaches and evaluate them. In the design science cycle, the problem investigation is vital to find out who the involved stakeholders are and what goals they pursue. This allows setting a conceptual framework and boundaries for the research. In the case of this work this phase, in combination with found literature, defines the context in which the proposed RQ's are answered. The approach is later on visualized in figure 3.3. Through early workshops and interviews with domain experts, common problems in an engineering process have been elicited so that in a later phase the treatment can be designed. The second phase of the design science cycle, treatment design, is responsible for specifying requirements for a good solution. Stakeholder interviews and existing works are major building blocks for this step, as are investigations of existing treatment solutions.

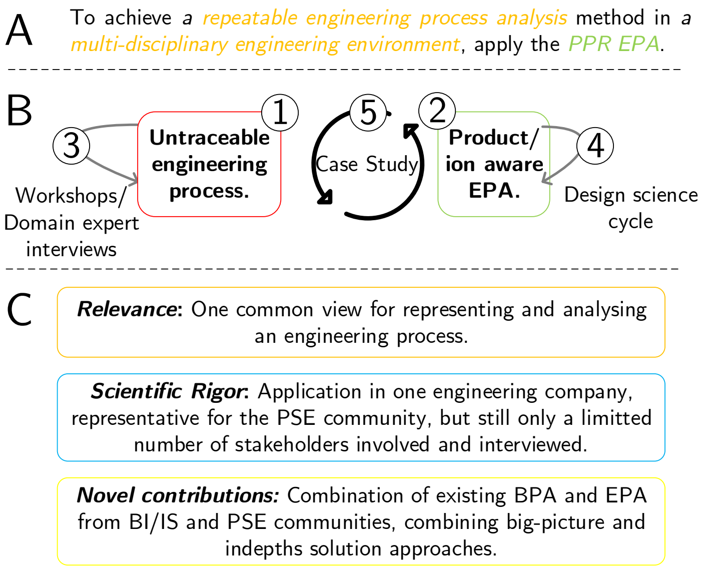
The outcome of the second methodological building block yields a set of requirements, lists of promising approaches of methods that can be adapted and existing approaches used in practice.

M1.3 - Adaptation Proposition
The output from the previous methodological building block M1.2, are now used as input to propose adaptations. The adaptations are based on both literature and elicited requirements. This step focuses on building on already good existing solutions and at the same time adapting these solutions to overcome the found limitations of existing approaches. The adaptations are designed with the requirements of both literature and stakeholders in mind and that new treatments need to provide a benefit for the involved stakeholders.

Outcome here is a set of adaptations to existing solutions that need to be evaluated in the last step. Adaptations focus on both RQ's, namely the PPR EPA so that it is repeatable but still uses existing well-proven approaches, and also adaptations for the visualizations of engineering processes based on BPMN 2.0. The adaptations represent also an intermediate result that can be used on its own for other research areas as proposed here in this work. In the IDEF0 diagram, this is indicated by the second output arrow going to the right-hand side of figure 3.1.

M1.4 - Adaptation Evaluation
The final methodological step is the evaluation of proposed adaptations. Based on the design science cycle [82], and a case study execution [64], the proposed adaptations can be evaluated. The case study approach form [64], was executed in a real-world engineering company and with several domain experts of PSE. The company is specialized in engineering and manufacturing highly automated production systems for discrete manufacturing systems and the involved domain experts all look back on multiple decades of experience and thus know the processes and artifacts very well which are under investigation. Through guided interviews, each domain expert was questioned about his work and also the new treatment could benefit him.

In figure 3.3 below, the design science approach is visualized, using a visual abstract representation based on [76]. The template provided by Storey et al. has three major components: *A) the theory proposed or refined in terms of a technological rule; B) the empirical contribution of the study in terms of one or more instances of a problem-solution pair and the corresponding design and evaluation cycles; and C) the assessment of the value of the produced knowledge in terms of relevance, rigor, and novelty.*[76].



To achieve *a* *repeatable engineering process analysis* method in *a* *multi-disciplinary engineering environment*, apply the *PPR EPA*.

Relevance: One common view for representing and analysing an engineering process.

Scientific Rigor: Application in one engineering company, representative for the PSE community, but still only a limitted number of stakeholders involved and interviewed.

Novel contributions: Combination of existing BPA and EPA from BI/IS and PSE communities, combining big-picture and indepths solution approaches.

Figure 3.3: Visual abstract [76] for research issue one, identifying the main elements of the research.

Used for this research issue, the theory proposed is the repeatable PPR EPA in a MDE environment, which is applied for receiving a repeatable EPA.

In the second building block B), the problem instance (circle number one), is the current situation where the engineering process is not traceable across multiple domains. Second, the contribution to this problem is a product/ion-aware EPA, that addresses this issue.

The approaches used to understand the problem context, are workshops and interviews with domain experts (circles number three and four), as already described in M1.2. To propose a new solution the design science cycle from Wieringa [82], is used and finally (circle five) a holistic case study (M1.4 based on [64], is applied.

Part C) consists of three assessment blocks. The proposed solution is relevant for stakeholders in the engineering management and quality assurance field, as they receive repeatable processes and a common notation to express an engineering process. Scientifically, is the design and evaluation based on one engineering company, which is representative for discrete manufacturing companies. However, only a limited number of use cases and domain experts were involved. The contribution is novel, as it combines both BI/IS and PSE approaches and thus overcomes their respective limitations.

3.2 RI-2: Modeling of Product, Process, Resource (PPR) concepts

The result of RI-1 gives an overview of an engineering process, involved stakeholders, engineering artifacts and engineering processes. The visualization of RQ1b, allows further to investigate PPR knowledge that is present in an engineering process. Currently, available modeling techniques, presented already in section 2.4, have different capabilities to represent processes and involved data aspects. To be able to compare these existing approaches, the following question needs to be answered first:

RQ2a: What are the requirements for modeling PPR concepts in a MDE context for PSE?

Major outcome of this RQ, is a criteria catalog, based on existing requirements. The catalog focuses on two aspects of requirements namely a) generally applicable to process modeling and b) more specific to representing PPR knowledge. The resulting artifact of this RQ is based on an adapted systematic literature survey. Further requirements, which cannot be found in common literature and are unique to the MDE from PSE, are motivated and supported by examples and use cases elicited from interviews with domain experts. Future research in the field of modeling and comparing existing techniques, benefit from this outcome, as it represents an overview and benchmarking foundation that can be re-executed. The outcome of RQ2a, the criteria catalog, allows then to ask the following question:

RQ2b: What are the capabilities and limitations of different existing modeling language approaches considering the different concepts of PPR modeling?

The contribution of this RQ, is another table, this time representing benchmarking results of selected modeling techniques. Investigated modeling languages are: BPMN 2.0, eEPC, Formal Process Description (FPD) (VDI 3682), IDEF0, petri nets, SySML activity-diagrams, and Sequential Funciton Charts (SFC). All of these representations with their strengths and limitations have been presented and discussed in section 2.4. The outcome of this RQ is obtained by comparing the capabilities of modeling approaches with the imposed requirements from RQ2a. Involved stakeholders and future researchers can use this benchmarking result for their needs and select, according to their requirements, a suitable modeling language.

In the context, where this work is placed, stakeholders for MDE in PSE are further interested in a) expressing knowledge identified in RQ1a and RQ1b, and b) make PPR knowledge more explicit. To address this requirement and stakeholder need, a third question has to be answered in this research issue:

RQ2c: How can representative PPR use cases from domain experts be supported by the identified modeling languages?

A set of adaptations for PPR modeling, is the major outcome of this RQ. The two previously introduced outcomes in this RI-2, allow the outcome of this RQ, to be expressed in an appropriate, best-fitting modeling language. In the case of this work this is the FPD. Domain expert interviews, conducted in the case study and following the design science cycle, provide the use cases to evaluate the proposed adaptations. Main benefactors form this outcome are involved stakeholders like, basic planners, engineering managers and also quality assurance personnel. The basic planners, for example, a production system planner, receives a means to express common scenarios and use cases in an appropriate modeling language. Engineering management stakeholders, in turn, receive a basis for future analysis of product/ion-aware engineering knowledge that is present throughout the engineering process.

Research Approach

As already presented in the first research issue, does this research issue also follow a set of methodological building blocks and guidelines to answer the proposed RQ's. The proposed research approach is depicted in figure 3.4, an IDEF0 diagram, and will shortly be described.

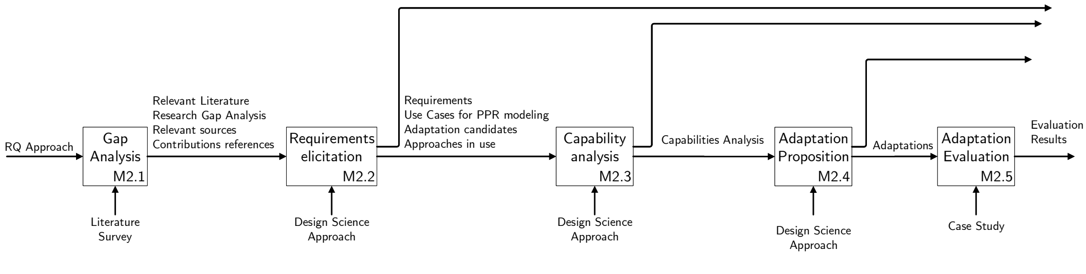


Figure 3.4: Research approach that the second research issue follows.

As discussed with the first research issue, the blocks represent actions that are executed. Input arrows coming from the left and output arrows leaving on the right from each action represent the flow and sequence of the research approach. Arrows at the bottom of the boxes indicate the mechanisms used.

M2.1 - Gap Analysis
Just like in the first research issue, a literature survey is applied to find relevant literature, promising sources, and contributions of the sources. The approach for the literature survey can be seen in detail in figure 3.2 and will at this point not be discussed again, because the same execution steps and major building blocks are used. The research questions for this research issue represent a vital input for the first methodological building block, as it provides a context around the whole goal of the literature survey.

M2.2 - Requirements elicitation
The output from the gap analysis is, in a second step, used to elicit requirements. The design science approach [82] is a vital concept for this execution step. Requirements which are found in this research step are also used and presented to domain experts for discussion and later on treatment design and evaluation. The second RQ is answered by executing this step and is thus lengthily discussed in the result chapter 5. The found requirements and especially the rationale behind the individual criteria can be used for further investigations and evaluations. Thus, the output of this building block is also an intermediary result and can be used for possible research outside of this work. This is indicated by a second arrow leaving the action box M2.2.

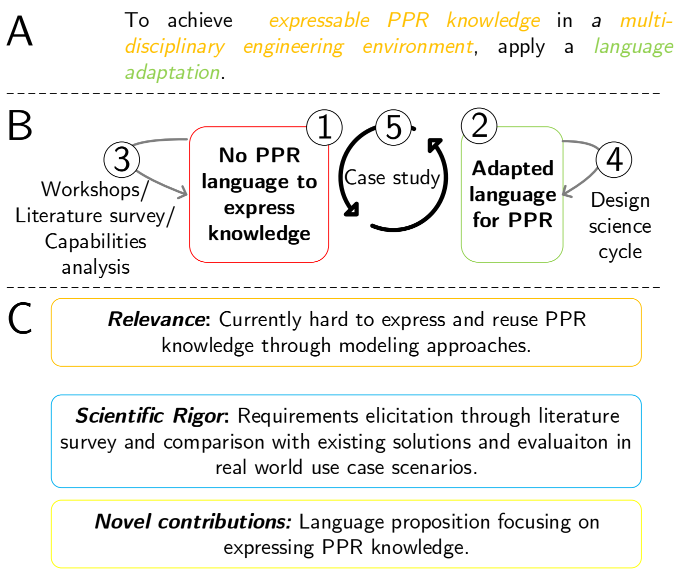
M2.3 - Capability analysis
As there are already many existing modeling approaches available, as was presented in section 2.4, a capability analysis under the guidance of the design science approach is executed. The capability analysis is focused on eliciting what approaches are suitable for modeling PPR knowledge or if they are not suitable, how big the gap is to fill for a PPR modeling language. To answer the RQ, the requirements from the previous step provide valuable information and are thus used for benchmarking the existing approaches. Also, the list of promising approaches and their capabilities are used in this step. The outcome here is a capabilities and limitations analysis of found approaches from practice. Both of the next building block as well as other research outside of this work can build upon this outcome.

M2.4 - Adaptation Proposition
From all the previous steps, promising approaches with their capabilities and limitations are known and benchmarked in a quantitative manner so that an objective comparison can be made. This fourth step is then concerned with adapting the most promising approach(es), to fully meet the requirements to represent PPR knowledge. Also just like in the first research issue, this steps follows the design science approach. The possible adaptations and treatment design are executed under the guideline of the iterative approach from [82]. The proposed adaptations can also be used for further investigations and are thus an intermediary result, again indicated by the second output arrow from the building block M2.4.

M2.5 - Adaptation Evaluation
The last step for this research issue is again an evaluation of the proposed adaptations. A case study approach [64], is used to grade the newly introduced modeling concepts. The evaluation presents first a proof of concept implementation of a possible PPR

modeling language which then, in a second step, is again evaluated with domain experts and stakeholders. Domain experts were again interviewed and asked to grade the new treatment based on several ISO 25010 [7] metrics.

In figure 3.5 a visual abstract [76] of this research issue is presented. The figure has the same elements as already figure 3.3.



To achieve *expressable PPR knowledge* in a *multi-disciplinary engineering environment*, apply a *language adaptation*.

Relevance: Currently hard to express and reuse PPR knowledge through modeling approaches.

Scientific Rigor: Requirements elicitation through literature survey and comparison with existing solutions and evaluaiton in real world use case scenarios.

Novel contributions: Language proposition focusing on expressing PPR knowledge.

Figure 3.5: Visual abstract [76] for the second research issue.

Part A) presents the theory proposed or refined in terms of a technical rule. For this research issue the focus lies on expressing PPR knowledge in the MDE environment.

Part B) depicts the problem instance under investigation, namely the situation that currently no dedicated language is present to visually model PPR concepts, and knowledge. Thus, the solution (circle two), proposes a language adaptation to meet PPR requirements. Workshops, literature surveys and the presented capabilities analyses (building blocks M2.2 and M2.3) are used to elicit the context and approach to meet the problem. The design science approach is used for designing a solution, and lastly, a case study (circle five) is used for evaluation.

Part C) is concerned with the assessment of the value of the created knowledge. In terms of relevance for research communities, is this research issue of interest as it investigates currently non-existing concepts of expressing PPR knowledge. Scientifically are all results based on requirements elicited from well-founded literature and existing research as well as domain expert interviews. Also, the novelty of the research can easily be determined, as a language adaptation and proposition would meet the requirements that are already present for modeling PPR knowledge but are not yet met.

3.3 RI-3: Product, Process, Resource (PPR) Knowledge Persistence

As the two previously presented research issues address challenges one and two, here now the third and last RI will be presented. Focus of this RI is the third challenge of figure 1.1. Based on domain experts interviews, that are concerned with the execution and creation of SE tools and supporting the engineering process, a baseline for the current development and technical architecture was elicited. Up to this point, the major concern was how to elicit and then express PPR knowledge. Now it is also a point of interest on how to make PPR knowledge persistent and allow possible queries and reuse options. The question that needs to be answered for such a PPR knowledge persistent representation is:

RQ3: What are primary use cases that require the persistence of different categories of PPR knowledge?

The already mentioned domain expert interviews, build the basis for the outcome of this RQ. Through the elicitation of use cases, is it possible to find requirements that are based on use cases for PPR knowledge persistence. Further, do the use cases motivate the distinction of several categories of PPR knowledge, which in turn have also an impact on the underlying persistence solution. Stakeholders that benefit from this research are future data curators and technical architects. Data curators get an overview of the primary use cases and classifications of PPR knowledge and can thus derive basic concepts for an underlying data model. Further allows the outcome of this RQ, the data curator to think about distributed data models that need to be interlinked but due to their different requirements, need to be encapsulated into specialized solutions. The technical architects benefit from the outcome through the primary requirements that different classifications of PPR knowledge have for a data storage solution. It is also in the interest of a possible technical architect to investigate these requirements in regard to concrete technologies that support the requirements and possible future functionalities, like search and reuse of PPR knowledge.

Research Approach

The third and last research issue also follows methodological building blocks, which are presented in this section. In figure 3.6 the steps to follow for answering the RQ are presented. Just like in the two approaches before, the diagram is based on IDEF0.

M3.1 - Use Case Elicitation
As first step a use case elicitation is executed. The focus of this step is to provide use cases that are currently present in a PSE organization. It is here the aim to investigate the use cases and focus mor on PPR knowledge persistence aspects than to have some general use case description of an engineering process, as does for example RI1. The design science approach [82], is also here a valuable mechanism to build upon. Outcome of

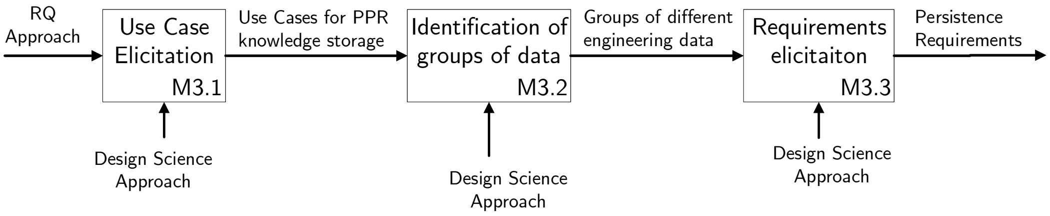


Figure 3.6: Research approach for the third and last research issue.

this step are use cases that are relevant to consider when, in future, designing persistence solutions for PPR knowledge.

M3.2 - Identification of different categories of PPR knowledge
Using the use case descriptions from building-block M3.1, this step identifies different groups of data that are common for the use cases. Use cases and valuable insights from domain experts from a PSE company help to understand what different groups of data are present in a PSE organization and how they are linked to the use cases. In this step, thus, is it a focus to identify on top of the use cases groups of data, which possibly have different requirements regarding a PPR persistence solution. The execution of this activity is also guided just like M3.1 by the design science approach [82]. Outcome are groups of data for persisting knowledge in a MDE PSE environment.

M3.3 - Requirements elicitation
The last methodological step is to derive requirements from use cases and groups of data. Just like in previous research issues, requirements represent a vital building block for research. However, this research issue is more concerned with identifying these requirements, as there are currently not available than to implement a treatment that meets these requirements. Based on the design science cycle is it possible to combine the input with also domain expert interviews for an adequate outcome. The main outcome of this RI are requirements regarding PPR knowledge persistence.

In figure 3.7, the research approach is visualized using the approach form [76]. As before the graphic is built on the same concepts.

Part A) focuses on this research issue on providing use cases and data groups under consideration of the MDE tooling landscape by applying domain expert interviews.

Part B) presents the problem under investigation, namely the absence of knowledge on how to store PPR knowledge. Annotated with circle two is the proposed solution, the elicitation of requirements form use cases. Domain expert interviews help to understand the problem and the design science cycle helps designing the solution approach (circles number three and four). Finally, the evaluation is done by interviewing practitioners about the proposed found use cases, data groups and then elicited requirements.

In part C) the proposed research is relevant as it addresses an open issue, namely the storage requirements of PPR knowledge. Based on requirements elicitation from domain

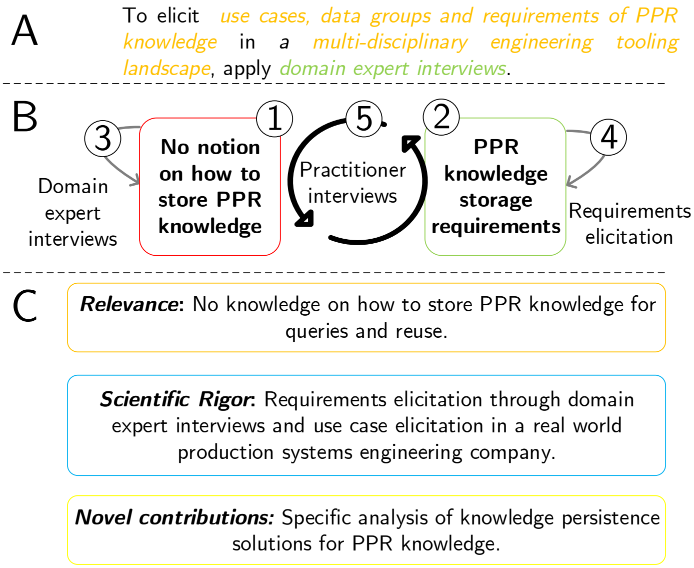


Figure 3.7: Visual abstract [76] with its main elements for research issue three.

expert interviews and the design science approach is it also scientifically well-founded. Finally, does this research approach provide novel insights into requirements regarding persistence solutions.

CHAPTER 4

Engineering Process Analysis

In this chapter, the research issue one will be discussed with input from [9]. The overall focus of the research issue is to investigate engineering processes and possibilities how to represent them. The following two Research Questions (RQ's) will be answered in this chapter, by following the presented methodological approach in section 3.1.

RQ1a: What adaptions or combinations of business/engineering process analysis methods allow overcoming the limitations of these communities regarding product/ion-aware engineering processes?

RQ1b: What adaptions or combinations of business/engineering process notations allow overcoming the limitations of these communities for representing stakeholders, processes, and documents that may represent PPR knowledge?

The chapter is structured as follows: The next section 4.1, and its two subsections present the literature survey approach and outcomes, section 4.2, presents elicited requirements for a good solution, in section 4.3 proposed adaptations will be presented on how to overcome found limitations and to fulfill the elicited requirements. Finally section 4.4 closes this chapter with evaluating the proposed approach with its adaptations.

4.1 Literature Survey

To answer both RQ an adapted literature survey is executed. As already presented in section 3.1, the literature survey has multiple phases, which are equal for all research results. Because of that this section focuses on execution specific results, like: keyword definition, search string definition and the search process execution.

4.1.1 Keyword definition

In this subsection, the used keywords are presented in tables 4.1 and 4.2. The key words are simple and atomic strings that can be combined for a search in a digital library as presented in table 3.1.

PPR - Engineering Process Analysis

process	analysis	business	engineering
redesign	execution	planning	

Table 4.1: Keywords used for the literature survey regarding engineering process analysis methods.

PPR - Data Processing Map

process	visualization	modeling	abstraction
notation	business	engineering	structure

Table 4.2: Keywords used for the literature survey regarding engineering process visualizations.

4.1.2 Search String definition

The search strings are made up of combinations from individual search key words. Not all combinations of the key words do yield a result or would be clever to use, thus only the best combinations are presented in tables 4.3 and 4.4.

PPR - Engineering Process Analysis

business process analysis	business process redesign	engineering process analysis
process analysis	process redesign	engineering process
process execution	process planning	

Table 4.3: Search strings for investigating engineering process analysis methods.

PPR - Data Processing Map

business process visualizations	business process modeling
process abstraction	process notation
engineering process notation	process structure
engineering process visualization	process structure visualization
process structure abstraction	process structure notation

Table 4.4: Search strings for investigating possible visualizations regarding engineering process knowledge.

4.1.3 Search & Initial Analysis

The search strings were executed on the digital libraries, see table 3.1 in the research approach 3.1. All found titles from the search were examined regarding possible contributions and how they fit the goal of process analysis methods. If the title seemed appropriate enough, stating a contribution relevant to this research, the abstract was further read, otherwise, the paper was not further included. From this initial analysis, around 30 papers did get selected.

Removal of Duplicates and Initial (Quick) Analysis

A criterion which had to be met for a paper to be included further on was that it needed a description of how the process analysis was executed and how the results were managed. This criterion was established to exclude early on papers, that are too high-level and do not actually execute a process analysis method. After this step there were 23 papers, that could be used for both Product, Process, Resource (PPR) Engineering Process Analysis (EPA) and PPR Data Processing Map (DPM). The twenty papers are: [78] [15] [19] [25] [30] [33] [34] [36] [40] [41] [44] [45] [47] [49] [46] [48] [53] [56] [62] [65] [68] [73] [80]

4.1.4 Data Extraction (Analysis & Results)

In this step, the papers were read and the most vital information on process execution and representation was extracted. Especially the extraction of execution steps allowed to identify common elements like interviews that are present in most Business Process Analysis (BPA) and EPA methods. An interesting fact is that most papers execute a case study in some form with domain expert interviews, but they differ from the level of depths that is covered. Major differences of the domains Business Informatics (BI)/Information Systems (IS) and Production Systems Engineering (PSE), have been already presented in detail in section 2.3. Just to recap shortly, literature focusing on BPA methods are more high level and allow the identification of interfaces between workgroups, even though they do not focus on this. Approaches coming from a PSE background, are more focused on intra domain improvements and do not allow to represent a big picture. The data extraction step also allowed for an initial set of requirements for a good solution, for both PPR EPA and PPR DPM visualizations. Requirements for both RQ will be presented in the next section in more detail.

4.2 Requirements Elicitation

As a second methodological building block, the design science cycle form Wieringa [82] was used. In [82], requirements are identified as *contribution arguments*, which are: *"arguments, that an artifact, that satisfies the requirements, would contribute to a stakeholder goal in the problem context"*. This means, that if proposed requirements are met by a possible solution, the solution would contribute to the stakeholder goals. Building on [9], the following requirements have been elicited from the stakeholders involved in a case study that represent usual daily process execution paths. The case study spanned over nearly two month at an engineering company in the discrete manufacturing system. Over the course of the case study, several interviews and workshops were held to match expectations and present results. The following two subsections present the elicited requirements.

4.2.1 PPR - Engineering Process Analysis

This subsection focuses on requirements regarding the EPA approach.

Knowledge identification. The PPR EPA should allow identifying engineering knowledge. The identification of PPR knowledge should focus on the life cycle of engineering artifacts. A special point of interest should be the creation, manipulation, and loss of engineering knowledge. These three key capabilities are essential to follow the artifacts and engineering knowledge through the engineering process. An example is: Initial product drawings from the customer are received at the project kick-off as 2D and 3D drawings, which are then stored in a document management system for the later domain experts for further processing.

Knowledge classification. The PPR EPA should allow an explicit classification of the knowledge in to product, production process or production resource relevant information. In combination with the first requirement *knowledge identification*, does this second requirement allow a refinement of engineering artifacts and the knowledge they hold. This is of special interest so that reasoning can, later on, be made if knowledge is currently present in engineering artifacts and if so where, or if knowledge is completely missing and new artifacts need to be designed. An example of knowledge classification are product drawings containing information about the product or requirements regarding time constraints containing critical information about process execution. This would mean that the product drawings artifact contains product and process relevant knowledge.

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. This requirement combines both *knowledge identification* and *knowledge classification*, and places a special focus on the interaction of engineering artifacts with process tasks. 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.

Gradual refinement. The PPR EPA should allow for starting on a high-level, outlining the context and responsibilities of stakeholders, to gradually become more detailed and elicit individual tasks and engineering artifacts. This expansion should be possible until the desired level of detail is reached and the engineering management and quality assurance stakeholders are able to gain fast and efficient insights into the engineering process. An example of this is: A basic planner outlines the start and end tasks of his work and identifies roles that are dependent on his outcomes. The parts between start and finish are at first left out. Later on, the individual engineering tasks are identified, their sequence aligned and engineering artifacts elicited and classified according to PPR.

Multi-disciplinary interaction identification. The PPR EPA should allow identifying where multiple disciplines are communicating and interacting with one another. Such a highlighting of interfaces for collaboration and coordination are essential and present an individual requirement. This interaction identification can be achieved through gradually refining the engineering process and especially placing focus on multi-disciplinary engineering (MDE) interactions. Also are focused questions on the interactions a key requirement to achieve good results. An example is project hand-over phases where the responsibility of the overall project is shifted from one stakeholder to another. These handovers mark collaborations of multiple disciplines from basic and detail planning and are a vital part in engineering processes.

4.2.2 PPR - Data Processing Map

This subsection is now concerned with requirements regarding the visual representation of the outcome of the EPA.

PPR-specific visual elements. For representing PPR knowledge that is present in an engineering process, distinct elements for product, process, and resource should be present. Further is it important to also depict roles, tasks and the priority a task has regarding PPR knowledge. Through this requirement, an easily understandable representation should be guaranteed which allows experts and non-experts alike to uniquely identify elements and discuss them. An example is, that tasks can be distinguished according to their priority regarding PPR knowledge.

Knowledge classification representation. The visualization of a product/ion-aware EPA should classify and also individually represent the engineering artifacts and their respective knowledge. This means, that engineering artifacts should visually indicate which part of the PPR triangle they contain, so that reasoning regarding knowledge creation, manipulation or loss can be made. A direct knowledge classification should allow for easy identification of PPR knowledge and its flow.

Iterative refinement and layered modeling. It should be possible, as with the PPR EPA, that also the visualization the PPR DPM can iteratively be refined. This means that it should be possible to start with a small initial model which does not require high-efforts for an initial draft version. But it should be possible to depict further details and also represent concepts that are layered like meta- and sub-processes which consist of several other processes. This requirement allows having an easily understandable high-level initial model which can be extended and represent also more detailed process execution paths, for detailed analyses of domain-specific executions.

Well known standard. The gathered data from the PPR EPA should be visualized using an established and well-known standard, so that also non-experts have the chance to pick up this standard without requiring high training and introduction phases. Especially in large environments with multiple disciplines and the possibility that in each work group someone else has to visualize the internal executions, a standard helps to unite and unify concepts across multiple stakeholders and domains.

Process overview. The PPR DPM should provide an overview of the engineering process, involved disciplines, interfaces of the multi-disciplinary environment, engineering artifacts and their flow through the process as well as the interaction of engineering artifacts with engineering processes. Especially quality assurance stakeholders are interested in these representations, as they yield great potential to identify optimizations in the engineering process.

4.3 Adaptation Proposition

To address the first research issue, with RQ1a and RQ1b, the design science cycle [82] is also used to propose adaptations to be able to answer these RQ. The research builds on known BPA methods like *A2BP* from Santos and Alves [68] as well as the *context aware process analysis* from Rosenberger [62]. On the side of PSE this work uses inputs from [73] and [47], the *mechatronic EPA* as well as from [37] the *technical dependency mining* approach. These inputs are used to present the outcomes of the next subsection.

Regarding the notation of the outcome of a PPR EPA, the second subsection presents adaptations to the Business Process Model and Notation 2.0 (BPMN 2.0) standard. BPMN 2.0 was chosen because it has already many elements that are needed for representing an engineering process and the respective engineering artifacts. Alternative solutions would be extended EPC (eEPC), but that approach requires annotating each task with the respective organizational unit, which makes it harder to scale the MDE representation that this work wants to present.

4.3.1 PPR - Engineering Process Analysis

As already discussed in section 2.3, both BPA and EPA methods lack crucial elements for representing PPR knowledge in an engineering process. The absence of detail in BPA should be overcome by the EPA approaches, where in return the lack of high-level representations of EPA methods should be balanced with BPA approaches.

In figure 4.1, the result of combining BPA with EPA approaches is visualized and addresses RQ1a. The two-phase approach presents a repeatable EPA, which focuses on identifying and later on classifying and visualizing PPR knowledge throughout the engineering process. In both phases, the following stakeholders are involved: domain experts (orange), engineering management (blue), quality assurance (green) and EPA facilitator.

Domain experts represent a key group for the PPR EPA as they build the foundation of engineering knowledge that needs to be gathered and represented later on. The domain

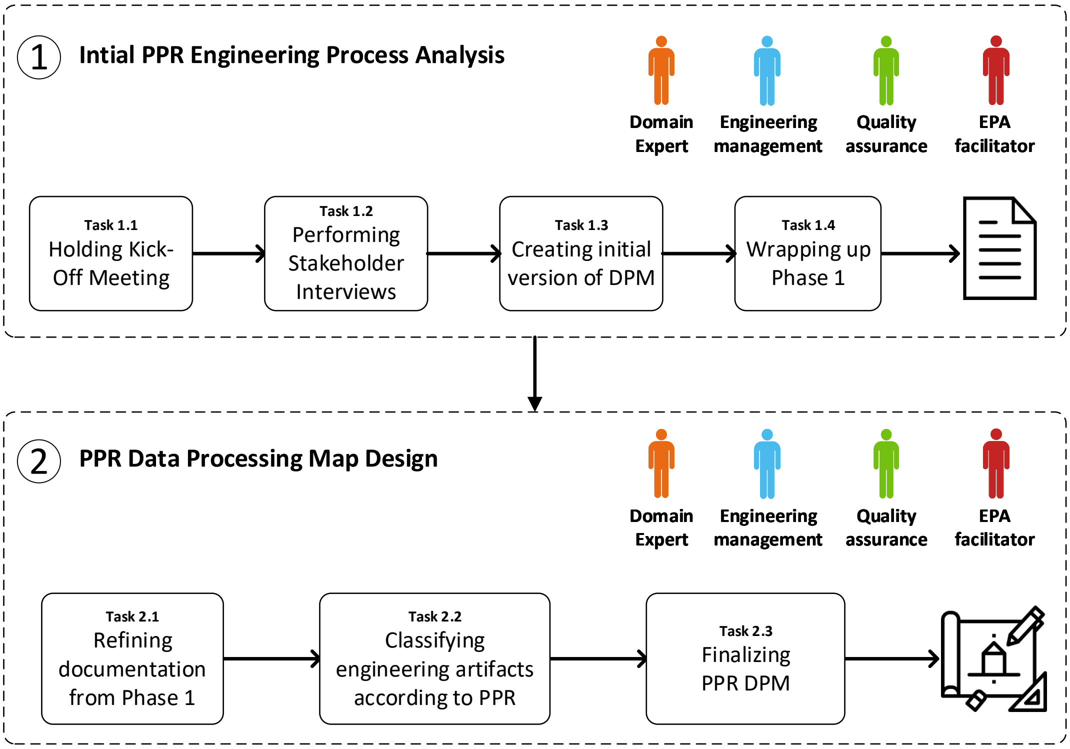


Figure 4.1: Repeatable Product, Process, Resource (PPR) - Engineering Process Analysis, based on [9].

experts are often only interested in improving their individual processes and have little to no interest of general process improvement. This is because the domains often work in parallel and interactions are not so frequent with each other. However, these information silos as motivated in section 1.1, have high risks regarding project success. So it is natural to include domain experts in an EPA.

Engineering management is interested in the overall engineering process. This stakeholder group represents a higher management level, which is interested in a smooth and low-risk execution of engineering tasks. It is them, who are interested in knowing which engineering artifacts are present in the process, or if new artifacts should be designed to increase productivity or output. The engineering managers often have already a good understanding of the overall engineering process and can identify some interfaces between work groups and help to provide an initial big-picture like overview. Also are these stakeholders vital for the creation and adaptation of a visual notation, as they are the main involved personnel to deal with such a representation.

Quality assurance personnel is mainly interested in improving the engineering process as well as the involved and exchanged engineering artifacts. This group needs to be included in an EPA method so that the right level of detail is achieved, as well as crucial questions regarding the possible outcome of the PPR EPA can be clarified. Besides the engineering managers, the quality assurance personnel is also highly dependent on the representation of the engineering process. Through the right adaptations of a PPR notation for a EPA, the quality assurance stakeholders can model as-is and to-be processes and analyze them and even create possible migration paths.

EPA facilitator is a stakeholder who actually executed the PPR EPA. It is this role who manages workshops, interviews domain experts, searches for appropriate visualization methods and models the outcome of the PPR EPA. The EPA facilitator is interested in finding common ground between all other stakeholders and to represent the engineering knowledge that is present. This role creates initial models and holds consultations with other domain experts if the requirements regarding a PPR EPA are met or how adaptations could look like. This role can either be filled by an engineering organization internal person or researchers conducting a EPA for an engineering organization.

Phase 1 - Initial PPR EPA
In the first phase of the PPR EPA, initial knowledge about the engineering process is gathered which helps to outline the context of the engineering process. The outlining of the context is crucial because the main goals and level of depths the EPA wants to achieve are built on this. Also, initial models are created in this phase. Main outcome of this phase are, interview documentations with domain experts and involved stakeholders, representative engineering documents for later classification, an initial DPM without a PPR classification and high-level artifact flows.

Task 1.1. Engineering Process Analysis Kick-Off
The context of the project under investigation is outlined and the main involved stakeholders are identified. Through the context elicitation first, rough plannings for the remaining EPA can be made, for example, initial simple models can be created to decide on an appropriate representation in the later phase two. Outcome of this task is a documentation regarding context, goals and stakeholders as well as a schedule when and where the interviews should take place and with whom. An initial model of a DPM can also be created based on these outcomes, consisting of tasks, events and data elements.

Task 1.2. Interviews
The EPA facilitator interviews domain experts, using interview techniques based on for example: [64]. The domain experts have generally a very deep understanding and can depict their processes very detailed. This is why the context elicitation and goal definition is crucial so that not too much information is gathered which might not be relevant after all. For this, an initial model, representing an anchor point, is a good building block. The interviews start high-level but become more specific regarding engineering tasks and artifacts. Such an approach is called funnel approach, by Runeson in [64]. In this task, representative documentation is represented and the interviews can also include questions regarding these artifacts for a better understanding. Outcome of this task are mainly interview documents and representative engineering artifacts.

Task 1.3 Initial DPM
As the previous task yields a great number of documents, this task focuses on the reassessment of the gathered information and how this knowledge can be represented in an initial DPM. A goal is to represent the engineering process and the flow of engineering knowledge in form of engineering artifacts and to sort and organize the gathered engineering documents. The artifacts at this point are not yet classified, and standard representations of a modeling standard like BPMN 2.0 can be used. The outcome of this task is the initial DPM, consisting only of process tasks and general artifacts as well as an organized structure which is interlinked with the initial DPM.

Task 1.4 Wrap-up
The final task of phase one allows a EPA facilitator to hold council again with all involved stakeholders, as they might not be available in the next phase. Possible open issues regarding gathered knowledge and artifacts from the interviews can be resolved. Also is it possible in this task to gather feedback regarding the initial DPM. Outcome of this task are the interview notes and an initial DPM interlinked with the gathered representative engineering artifacts.

Phase 2 - PPR DPM design
The second phase focuses on detailing, classifying and visualizing the previously gathered information which is mostly of qualitative form. The initial DPM serves as input for further detailing, whereas the interviews and links to engineering artifacts are needed to classify PPR knowledge. Outcome of this phase is a PPR DPM, representing the engineering process, interactions between engineering tasks and artifacts and classifications of engineering artifacts as well as priority identifications of engineering tasks.

Task 2.1 - Refinement
The previously gathered information is reexamined in this task. Too coarse or detailed modeled engineering tasks of the initial DPM can either be split up or modeled more detailed. Outcome of this task is the final version of the initial DPM, which can from here on be used as a single point of truth regarding the engineering process under investigation.

Task 2.2 - PPR classification
All artifacts, input and output, from the individual engineering tasks are classified regarding their information content of PPR knowledge. The insights from the interviews (Task 1.2) are vital for this step, as are detailed interview notes which can be used to resolve unclear issues. The classification regarding PPR knowledge is done in both the visual DPM representation but also attached to the gathered representative engineering artifacts. This two-fold classification allows representing the PPR knowledge on different levels. Visual representations in form of a PPR DPM are easily understood and analyzed, whereas textual descriptions for the artifacts can be more detailed and be used for more sophisticated analyses. Outcome of this task is a DPM with classified engineering tasks and artifacts.

Task 2.3 - Finalization
As a last step the EPA facilitator creates the final version of the PPR DPM. This version

includes the priorities of the engineering tasks they have regarding PPR knowledge for smooth task execution, as well as classifications of engineering artifacts regarding PPR knowledge. In this outcome, all involved stakeholders, domains, tasks, artifacts and interactions are depicted and presented to the interview partners. The final outcome is then delivered to the stakeholders for further use like analysis and improvements.

4.3.2 PPR - Data Processing Map Notation

To address RQ1b, and to provide a notation that allows representing the gathered knowledge from the PPR EPA, a possible notation will be presented in this subsection. Different approaches regarding modeling languages have been investigated for a product/ion-aware notation, all of which have been discussed in section 2.4. Shortly summarized, are approaches from PSE like IDEF0, easy to understand but lack capabilities to scale or represent different concepts with individual elements. The approaches from BI/IS like BPMN 2.0, eEPC or Unified Modeling Language (UML) are good foundations, however do none allow to classify documents or engineering artifacts and lack the interaction of tasks with documents.

The best fitting solution for extension is BPMN 2.0, as it has many required concepts already built in. Some of these concepts are tasks, events or documents, as they make up most of the elements that are present in an engineering process. Also does BPMN 2.0 provide a large set of additional events and has the capability to model exception paths as well as sub-processes. Both concepts are of interest for further investigations and might be needed for other engineering process as, as are depicted in this work. Due to these reasons BPMN 2.0 was chosen as a foundation for an extended notation to represent the knowledge from a PPR EPA. In figure 4.2 the proposed extensions to the BPMN 2.0 standard are depicted.

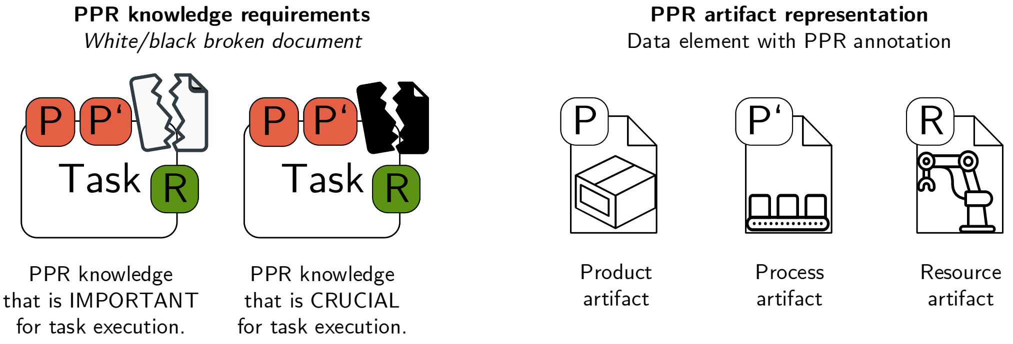


Figure 4.2: Extension to the BPMN 2.0 standard [3] based on [9].

On the left hand side of figure 4.2 are adaptations for the task concept of BPMN 2.0 depicted. There are several new concepts present. The first one is a classification of the task which concept of PPR the task currently receives (colored in green), and which additional information is needed (colored in red). In the figure, the P stands for product, the P' is a process requirement and R stands for resources. The example shows that the task needs all three concepts of PPR, but only resource information is present. A second new concept is to express the importance level a task has regarding PPR knowledge there are either none, a white or a black broken document. If no document is added to the task, then it is not crucial for the task to receive PPR knowledge, and the task can be executed without risk. White broken documents indicate, that for a seamless execution of a task it is *important* to receive PPR information. In case the P, and P' knowledge is not present the task can still be executed but with inconvenience for the domain expert. This means that additional effort in form of manual communication or collaboration is needed. Black broken documents indicate that it is crucial for a task to receive PPR knowledge. If a required concept is not present, then the task is very likely to run into problems and has a high risk of unsatisfactory fulfillment. Such high risks are, for example, major uncoordinated communications across several domains, and that delays of the task execution can endanger the whole success of the project.

On the right side of figure 4.2, the standard document representations of BPMN 2.0 can be seen. The proposed custom extensions are in this case visual representations of a product (a package), a process (a conveyor belt) and a resource (robot arm) inside of the document representation. Additionally to the visual pictograms inside of the BPMN 2.0 documents are again annotations regarding PPR knowledge. The same elements as have been proposed for the task extension are used to indicate if a document contains product, process or resource knowledge. Examples for the annotations are: product drawings contain information regarding the product, artifacts containing requirements regarding cycle times are classified as process artifacts and layout drawings of the manufacturing system contains resource information in form of arrangements and layouts.

To evaluate the proposed adaptations of a PPR EPA and a PPR notation for a PPR DPM, a case study was executed. The results are discussed in the next section.

4.4 Adaptation Evaluation

The previously presented adaptations, need to be evaluated, to be able to see if the newly designed artifact holds up against the imposed requirements. To evaluate these adaptations, a case study approach based on [64], is used, where domain experts and stakeholders as presented in the previous subsection, are interviewed and the gathered knowledge is visualized as a PPR DPM.

The next subsection presents evaluation results, containing interview snippets from the case study and the PPR EPA. The PPR DPM will be presented in the second subsection, where results of the interviews will be presented and discussed. This chapter closes then with an overall evaluation of the PPR DPM, based on a Likert scale [2] with discussion.

4.4.1 PPR - Engineering Process Analysis

The execution of the PPR EPA was executed in the role of the EPA facilitator. The role was already described previously in the adaptation proposition and all the steps of the proposed adapted EPA were executed.

Study Subject
Domain experts in an engineering company, which create highly customized and automated discrete manufacturing systems, were interviewed. Each interview partner has (multiple) decades of experience and knows not only the ins and outs of his role but also of neighboring roles. This knowledge about interfaces with other disciplines was very valuable as it reduced the time needed to focus solely on these interactions. The current situation at the company can be seen as representative for many PSE organizations. Over the years multiple adaptations and improvements to the engineering process have been made. However, each adaptation was focused on improving one discipline increasing the effects of information silos. Only very little thought was given up until now on the collaboration and coordination of multiple work groups across several disciplines.

Study Execution
The case study followed the proposed two-phase approach of the PPR EPA method, executing each task and retrieving the described information.

At first, a kick-off was held, where only the most important stakeholders and only one for each role were present. The small number of involved people helped to elicit the context and keep discussions at a manageable length. Each stakeholder was able to introduce himself and his field of work, as well as a focus of his field of work. These early workshop meetings helped to elicit the context and set boundaries for future interviews.

The **kick-off** and context elicitation allowed to model first sketches of an initial DPM. Only the most vital elements like stakeholders, tasks, and documents were depicted in this early version. However basic this version is it represents a good starting point for the interviews, as the focus and boundaries can be explained to the individual stakeholders. The initial DPM makes it also easier to start off the interviews, as there is already some common ground present.

For every **interview** with a domain expert around one and a half hours were needed. The interviews followed a funnel approach [64], meaning that the questions started very broad, for example *What is your general responsibility in this domain?* and became more detailed like *Please describe the first tasks that you execute in each project in detail, including necessary engineering artifacts.* Through this funnel approach, the EPA facilitator was able to start off, with to some degree, known context information and then more detailed domain specific information. Through the interviews, the in the kick-off elicited context was refined and a new level of detail added. In general, the engineering tasks were identified first and their sequence established. An example question, with an answer, regarding this task set can be seen in table 4.5.

Only in a second step were then the most vital engineering artifacts elicited that are

Stakeholder	Domain Expert Engineering
Question	Can you describe your minimal set of activities which get executed in each project and the order they are executed in?
Answer	1) Receive product life cycle (PLM) artifacts from the customer. 2) Create some cheap and rough paper concepts, representing first thoughts for the project and identifying potential problems. 3) Plan the assembly sequence in more detail with tool support. 4) Create a plant layout – as 2 D drawing also with tool support. 5) Calculate the costs of the whole project based upon the plant layout and chosen assembly resources. 6) If any problems with the layout or arise, or the costs are too high for the customer redo steps 3 – 5. 7) Submit a final offer to the client 8) Transfer all data from the "private" engineering server to another part of the server, so that it becomes accessible for the next phases and disciplines.

Table 4.5: Interview Question from the Product, Process, Resource (PPR) - Engineering Process Analysis, regarding the process execution of a basic engineer.

needed for smooth task execution. If possible for each artifact an example was given to the EPA facilitator. In some cases however, only screenshots were available, as the artifacts were incorporated into a custom-built tool of the engineering company with no other way to get to this representative knowledge.

Breaks between the interviews, and the fact that a basic DPM was already present, allowed the EPA facilitator to adapt and expand the basic model into an **initial DPM**. The creation of an initial DPM, also allowed to receive feedback from the interviewees and to hold the feedback cycle very short. An important note here is, that this version of the DPM only consists of standard BPMN 2.0 elements like swim-lanes, tasks, gateways, and general documents. No classification regarding PPR knowledge is present at this point.

On a separate day, after the interviews, follow-up interviews were held to clear any open issues with the interview partners, as they might not be available for further questioning. In this **wrap-up** task, a small presentation was held and the current outcome in form of the initial DPM and interview notes with representative engineering artifact examples.

The **second phase** of the PPR EPA started shortly after the conclusion of the interview sessions. Main focus of phase two, was to reexamine and then classify the gathered knowledge to be able and to create a final PPR DPM.

In the **refinement** task, all interview notes were reviewed. Corresponding engineering artifacts were organized for easier retrieval. All artifacts were organized according to their domain expert and if it is an input or output engineering artifact. This basic ordering gives already a good overview of the artifact flow and minimizes the time needed for retrieval.

With an organization of gathered knowledge, was it possible to execute the **PPR classification** task. In this task, each engineering artifact and task was reexamined carefully with the gathered examples and interview notes. With the combined knowledge was it possible to classify engineering artifacts, as is listed in table 4.6 for the first process task. A more detailed description of process tasks can be found in appendix B. The classification is interested which stakeholder executes the engineering task, what the task is and where in the overall execution sequence this task is situated. Possible input and output artifacts are additionally covered. A description and classification regarding PPR knowledge, of the artifacts, can also be seen in the table below.

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
Process relevant knowledge	None
Resource relevant knowledge	The mapping between which part is created, or processed on which resource part.
Output artifact name	Same as input, no new knowledge is created, simply forwarded to the next task.

Table 4.6: Detailed description of the first process task in the Product, Process, Resource (PPR) Data Processing Map (DPM).

The classification for the engineering artifacts builds on [36], who proposed a mapping of engineering artifacts to engineering phases. An example could be: electrical and mechanical plans are mapped to the basic planning phase, as these artifacts are created in this early stage of an engineering process.

Through the classification of engineering tasks and artifacts, was the initial DPM extended by PPR knowledge classifications. This extension builds on the previously proposed PPR DPM notation extension for BPMN 2.0. The so created version of the DPM allows to better analyze and see the artifact flow and possible knowledge dependencies between several domains. As this artifact is so essential for the outcome of both RQ, it is presented in detail in the next subsection.

In the **finalization** task, a polishing of the PPR DPM artifact was done, meaning that the artifact was investigated closely to be compliant with the BPMN 2.0 modeling standard. The description of the engineering tasks and artifacts was reviewed and possible ambiguities were removed. Finally, the PPR DPM was presented to all involved domain experts and stakeholders and delivered as a description and representation of the investigated engineering process. This step can also be used as a kick-off, starting an iterative cycle for further investigations.

4.4.2 PPR - Data Processing Map

To evaluate the proposed adaptations for the BPMN 2.0 standard, a PPR DPM has been constructed based on the gathered information from the previous section, the PPR EPA. The creation of the PPR DPM is the main focus of phase two in the PPR EPA, in the PPR classification task.

The DPM uses the basic shapes from the BPMN 2.0 standard namely, start and end events, tasks, arrows indicating relations, parallel and exclusive gateways, documents representing data objects and their relation to the tasks and a timer event indicating that this task is only executed after a time constraint is met.

Newly introduced elements, extensions to the BPMN 2.0 standard, were introduced previously. A short recap of the extensions is, that tasks can be annotated what knowledge from PPR they currently receive and what in addition they need. It is also possible to highlight the importance level of PPR knowledge for tasks as well as classifications of engineering artifacts containing PPR knowledge. Figure 4.3 depicts the PPR DPM, that was constructed in the case study with an PSE company.

The **production process planner**, number one in red, starts each project and receives three input documents from the customer, these are product drawings, product variations, customer specifications. This is indicated in figure 4.3 on the top right corner with tag D1. All documents that this role receives contain either product or process information, which is also what the start task needs, indicated by the two green annotations. The task depicted in tag D1, has no priority regarding PPR knowledge, this can be seen as there are no white or black broken documents, a more detailed depiction of this task can be seen in figure 4.4 on the left side.

The product drawings are 2D or 3D drawings for the end product and the assemblies and also might include explosion drawings, parameters for dimensions, special comments regarding product requirements like resistance. This artifact is usually received as a PDF file and can not be further adapted or be manipulated for reuse purposes. The product variations come in form of excel spreadsheets and describe a mapping between individual parts and the use of these parts in the final product or a variation of this product. Through this artifact the customer can transfer information about individual parts and how they

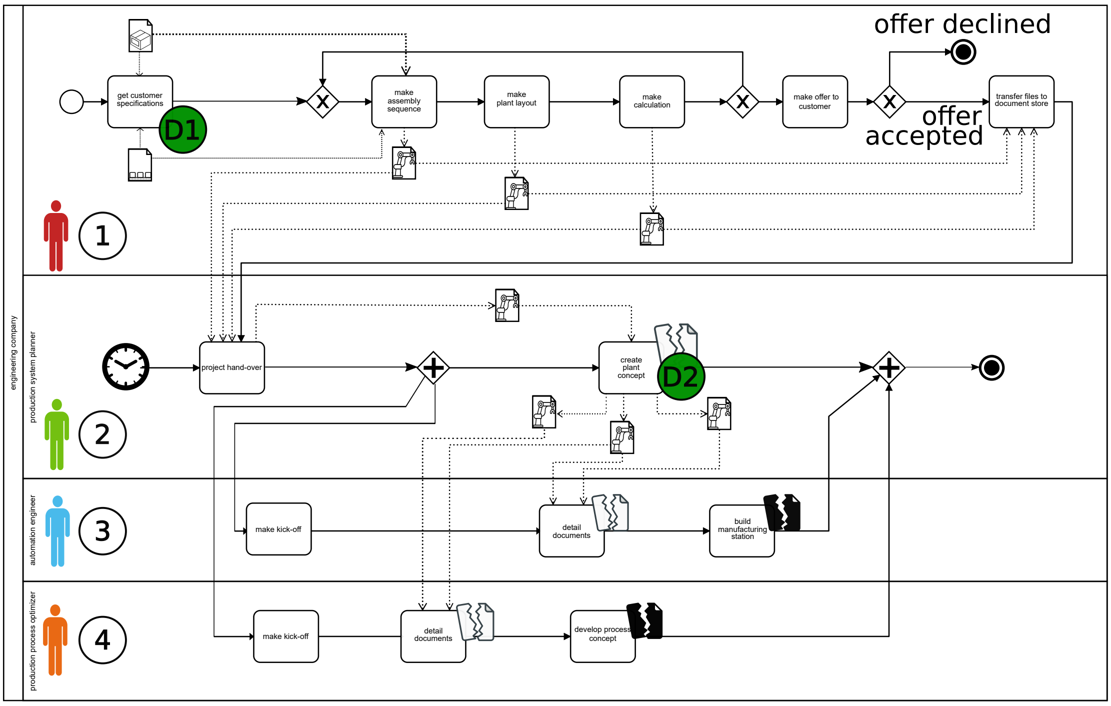


Figure 4.3: Overview of the Product, Process, Resource (PPR) - Data Processing Map.

should be treated and if they should be processed on special production resources like high tech welding lasers. The artifact is usually not further adapted from the production process planner and taken as an input for further planning. The customer specifications are overall project requirements, ranging from timelines to specific cycle times the final production system needs to adhere to. It is also used as a technical contract where the product owner from the customer and the responsible project management team signs. Most of the information is represented in tables, sometimes annotated by hand and finally scanned and sent as a PDF file.

With these three documents, the production process planner starts and creates first hand-drawn sketches on paper and thinks about various alternative forms on how to assemble the final product and what strengths and limitations each alternative might impose on the overall system. For these considerations, the three input documents are used and previous similar projects investigated for input. If needed the next three steps the role executes can be iterated as long as need be. These steps create the assembly sequence for the production system, the plant layout, and the calculation. All these artifacts are needed for further work in the disciplines to come, but only production resource-specific knowledge is incorporated into them. The assembly sequence consists of several parts: screenshots of primitive CAD drawings of parts (3D), cycle time calculations, textual descriptions of the overall project and the production resource and conveyor concepts. All this information is stored in tables and spreadsheets in excel and does follow only a self-imposed format, where the roles adhere to. The plant layout is a 2D drawing with indications of measurements, module orderings and numbers, positions of human-machine interface terminals, doors and special security areas. Measurements of individual parts are only annotated and the whole artifact is exported into a PDF file, which makes it impossible for further manipulation by the next roles. In the calculation artifact all information regarding the production system to construct are weighted against costs. This artifact is a basic calculation with all man-hours, material costs and so forth. The output of this document is stored in internal calculation tools and the company internal Enterprise Planning System (ERP) system. With the price being calculated, an offer is extended from the production process planner to the customer. If the offer is accepted, the role copies their created artifacts from a document server to an internal project specific location where the next roles do receive this information.

A **production system planner**, indicated by number two in green, starts working after the production process planner made the hand over of documents, and all further involved team members from different disciplines have time for a project kick-off – indicated by the timer event. The role receives as input all the output artifacts from the previous role, the production process planner. The production system planner holds at first a meeting with all involved roles for the project, including the production process planner. In this first step the project handover is made, but already under the responsibility of the production system planner. Here is the only official time, when all team members are assembled together and a rational from earlier design decisions can be discussed. With the project transfer completed, the individual teams are concertized and might be expanded a little bit from the initial kick-off. This allows then the automation engineering team and production process optimizer team to also start their work. The production system planner is then responsible to create more detailed planning from the initial layouts received from the production process planner. Indicated in figure 4.3 with tag D2, and in figure 4.4 on the right side, there a more detailed depiction of this task can be seen. Resource information is currently the only knowledge that is conveyed to this task, however, product and process information is also required but missing indicated by the two red annotations P and P'. Further is it important for the task execution to receive knowledge on these two concepts, indicated by the white broken documents. If this information is not present at the given time, then the production system planner starts manual callbacks to other roles and increases thus the project time and often has to do manual reworks. The role creates the following artifacts: detailed concepts for the plant layout, analysis of the critical path in the plant layout and criteria lists for acceptance tests. A CAD drawing represents the more detailed plant layout, which now is enriched with mechanical information and also might get changed due to mechanical limitations in regard to the initial layout.

The other roles, automation engineer and production process optimizer, also have access to this document and fill in their critical aspects of the production resource. To know critical paths and be aware of any issues, the production system planner analyzes the existing layout for possible paths in the system, which take up the highest priorities and/or

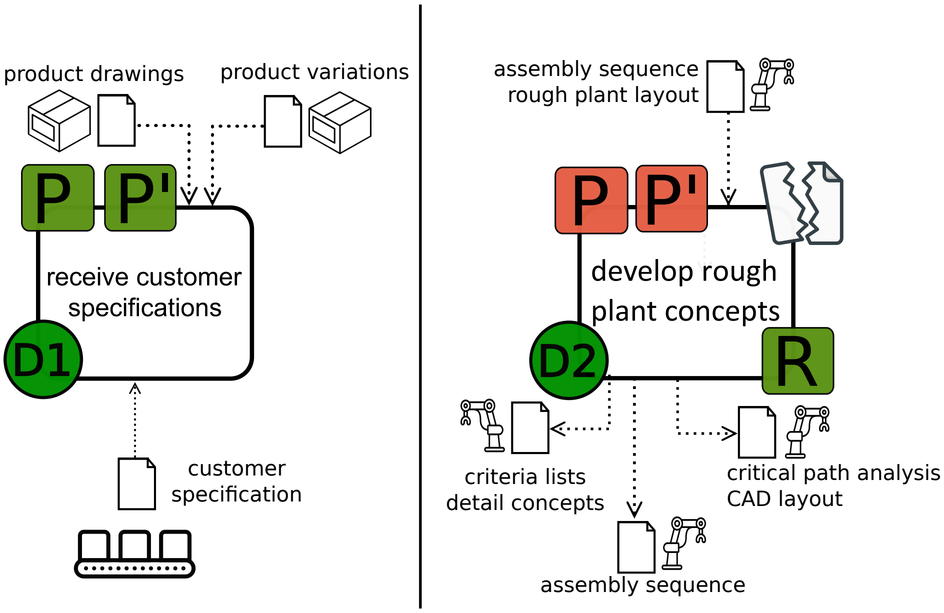


Figure 4.4: Detailed view of tags D1 (left) and D2 (right) from figure 4.3 with the custom BPMN 2.0 notation from figure 4.2.

might take up the most time. This knowledge represents direct process information. The production system planner also organizes meetings, for synchronization purposes with the other, in parallel working, disciplines. In these meetings the criteria lists for acceptance tests are very important. These lists represent the most important aspects of the system under construction and are essential to be met for a successful project. All involved roles can contribute to this list and raise any issues they might run into while their work is ongoing. From a PPR perspective, the list contains mainly resource knowledge and is stored in an excel file format. When all disciplines are finished with their work, it is the responsibility of the production system planner to finish the project in accordance with the customer. This means that the production system needs either a) to be installed on-premise of the customer or b) is being operated on-premise of the company itself. The conclusion of this step finalizes the overall project.

The work of an **automation engineer** in blue, number three, runs in parallel to the production system planner and production process optimizer. The role is responsible for creating and detailing the production system from an electrical point of view. To achieve his results and work properly, the role receives the artifacts from the production system planner. Further domain-specific knowledge is then applied to the individual documents and artifacts that an operative working for the team members in this discipline is possible. If the detailing step of this discipline has no implicit PPR knowledge present, manual callbacks via e-mail or phone calls are executed to receive the information about the project that is missing. Further, if in the last step, the actual assembly of the production system explicit PPR knowledge is missing, there is a high risk that the processes that are executed at runtime from the production resource are faulty and produce bad quality products. In the current state, this can not be avoided and is one of the major reasons why dry runs and test assemblies on-premise are being made.

The **production process optimizer** in orange, labeled number four, also works in parallel to the production system planner and automation engineer. The main focus of this role however is, that: all processes run in minimal time, critical paths throughout the system run smoothly and do not lead to any errors and if possible that optimization's regarding time constraints are made. For this role, the same concepts apply as in the previous role, the automation engineer. If in the detailing step no implicit PPR knowledge is available, the production process optimizer starts to search for this information and apply time and cost inefficient methods to get to the desired results. This step is similar to the depicted detail process step with the tag D2, with the only difference, that in this case, it is critical to receive the required input artifacts/knowledge. The process optimizer starts rework which are due to the fact that again product and process information is needed but missing. The received resource information is a good starting point, however, to enough for a task which critically requires all three concepts of PPR knowledge, indicated by the black broken document. If in the last step, the creation of process concepts, explicit PPR knowledge is missing there is a high risk, that the production system does not perform in an optimal state and in a worst-case scenario, customer requirements are not met.

The final PPR DPM was handed over to the stakeholders for further use in the finalization task of the PPR EPA. After working with the visualization, the engineering manager, quality assurance personnel and involved domain experts rated the visualization based on ISO 25010 metrics [7].

In table 4.7 the evaluation results are presented. The comparison is made with the previously used domain specific Eventdriven Process Chains (EPC) diagrams, a standard BPMN 2.0 model (the initial DPM) and finally the proposed PPR DPM. The metrics were chosen in accordance with the engineering management, because it is they who have to use this artifact later on. All approaches are then graded based on a five-point Likert scale [2], where "++" means complete fulfillment, "+", means fulfillment of the criterion, "0" means neutral, "-" means the approach does not fulfill the requirement and "- -" means no fulfillment at all.

Functional appropriateness measures to what extent the designed artifact allows expressing the engineering environment appropriately. This criterion is important, as it investigates how well the PPR DPM can represent the actual engineering process. But not only the engineering process representation is of interest, also how well the approaches reflect this knowledge and how usable the representation is and stays when scaling an approach.

Learnability measures how easy domain experts and stakeholders are likely to be able to understand and use the concepts represented in the data processing map. The created PPR DPM should reflect a kind of template for future analysis and representations.

These newly created models will then probably not be created by the same personnel as now. This is why it is crucial that also new personnel can easily pick up the used concepts and learn how to model an engineering process with possible PPR classifications.

Performance efficiency measures the level of time and resources required to use the PPR DPM notation as part of the PPR EPA method. Especially during, but also after the creation of a DPM, is it a requirement to not waste any time due to cumbersome notation requirements. Further, are quality assurance managers interested in improving the engineering process, which results in adapting the current version of the PPR DPM. This, in turn, requires, that it is time and resource efficient to adapt an existing solution.

Analyzability measures to what extent future analyses can be conducted based on the PPR DPM. Both engineering and quality assurance managers are interested in the current state-of-the-practice regarding the engineering process. To be able to pick up the current flow of engineering artifacts and task sequences is it a key requirement that a PPR DPM is analyzable regarding these concepts. Also, should future analyses be possible with the current notation.

Approach Criteria	EPC diagrams	Standard BPMN 2.0 model	Product/ion-aware BPMN 2.0 mode
Appropriateness	- -	-	++
Learnability	++	+	+
Efficiency	- -	0	0
Analyzability	+	0	++
Overall quality	**-**	**0**	**++**

Table 4.7: Evaluation of the Product, Process, Resource (PPR) - Data Processing Map (DPM) based on ISO 2510 metrics [7].

The first metric that was evaluated, was functional appropriateness. The results here vary from the worst to the best grade. EPC diagrams, as they are currently used as discipline-specific engineering task descriptions are not seen as appropriateness. This is because they are very detailed but do not allow any analysis of engineering artifacts or knowledge and the respective flow. Further is it not possible to identify interfaces with other workgroups. The BPMN 2.0 approach, only using the standards elements, does not fulfill functional appropriateness. Also, this approach lacks the possibilities to classify PPR knowledge. However, this representation provides already an overview of all involved domain experts, their interfaces and to some degree the flow of engineering artifacts. The proposed and adapted BPMN 2.0 notation for product/-ion aware modeling, is graded to fully fulfill functional appropriateness. The approach allows not only to depict a big picture, and if needed also details, but does it also classify engineering artifacts for a better analysis of creation, modification and possible loss of engineering knowledge.

Second metric, learnability, does not vary so much as the first metric. The EPC approach is assessed with the best possible grade, as it is very simple to pick up and already used in some parts of the PSE organization. As BPMN 2.0 has similar concepts as EPC, the approach is also seen as learnable and has a positive grade. Besides the standard BPMN 2.0 notation, has also the extension a positive grade, as the extensions are not very complicated and build on the basic concepts of the standard, making it easy to pick up and use.

Efficiency for EPC diagrams, was assessed very poorly. The justification from the domain experts and stakeholders was, that it is very cumbersome to change single parts in a diagram. This starts already at the beginning when the part to change needs to be found in a complex diagram, which is a time-consuming task in itself. Changes often also propagate new adaptations, consuming even more time for this approach. Both the standard and extended BPMN 2.0 notation have a neutral grade. Creating an initial model can become time consuming, depending on the level of detail that should be depicted in a diagram. Changes are, however, due to layered modeling with sub-processes, not so complicated or time-consuming. Reclassifications are also easily possible, require however some time and also need to be checked for possible propagation's through the remaining engineering process.

Last metric to grade was analyzability. The current approach of using EPC diagrams is seen to fulfill this requirement. Diagrams following this approach allow to detailed analyses of domain-specific engineering paths and in some degrees allow for process improvements. However, the approach does not allow for any PPR knowledge analysis, as this is not possible in this approach. The standard from BPMN 2.0 is assessed as neutral, and worse than the EPC approach. Only using the high-level depiction of an engineering process, does not justify losing the details form the EPC approach. With the extension to the BPMN 2.0 approach, however, is it possible to fully utilize the potential of engineering analyses. Through the classifications of PPR knowledge and the depiction of the engineering process, does this approach receive full grades and an overall positive remark for this metric.

The current approach of using discipline-specific EPC diagrams, is overall not seen as worthwhile keeping. Even though the approach is easy to learn and has received a fulfillment grade for analyzability, it is not appropriate for representing the overall engineering process. The approach lacks possibilities to represent interfaces with other work groups and is also very time consuming and error-prone. A further limitation is, that there are no PPR knowledge classifications possible.

The standard BPMN 2.0 approach received in the evaluation a neutral grading. This is due to the easy to understand concepts of the standard and the neutral gradings regarding efficiency and analyzability. Both metrics do achieve neutral grades because the standard does require some time to model the engineering process but overall represents a good basis for business analyses. Its strengths of providing a good overview are only damped with the limitations of not providing PPR knowledge classifications and losing the details of discipline-specific work executions.

Last is the PPR EPA approach with the extensions for BPMN 2.0. Overall is the approach rated very good. Especially functional appropriateness and analyzability are seen as much better than the other two approaches. The PPR DPM with its custom notation presents a good starting point for current and future analyses and is also due to its level of detail still appropriate without losing vital information. However, this approach takes also some time, just like the standard BPMN 2.0 approach does. Due to the fact that this approach also uses mostly BPMN 2.0 elements, it was seen as learnable as the previous approach of an initial DPM.

Overall can the results of the evaluation be summarized as follows: The EPC approach has its strengths in representing detailed discipline-specific engineering paths, lacks however in efficiency and appropriateness. Using the standard BPMN 2.0 notation allows for a more efficient modeling of an engineering process, does however not perform very well in terms of analysis and is thus only neutral. Extending the BPMN 2.0 notation as was proposed in this work, uses the strengths of the simple to use and learn BPMN 2.0 notation and introduces concepts, that allows for very good analysis and functional appropriate representations of an engineering process.

4.5 Summary

This chapter focused on investigating engineering processes, their structure, involved domain experts, artifacts and how all of this information could be expressed visually.

In section 4.3, adaptations to existing solution approaches were presented to be able to investigate a multi-disciplinary engineering process. As existing approaches lack in this part, basic combinations of solution approaches yielded a repeatable process, overcoming individual limitations. Also, a notation extension to BPMN 2.0 was proposed to be able to express the found knowledge.

The proposed adaptations were in section 4.4 evaluated. This showed that the designed treatment actually is able to answer the two RQ's, that were the main focus of this chapter. The main result is, that the PPR EPA is seen as a good approach and domain experts and stakeholders are willing to use it, as this artifact represents easy to understand concepts and represent the most vital knowledge from an engineering process. Also, does the proposed approach overcome current limitations in regard of expressing PPR knowledge.

CHAPTER 5

PPR Modeling Comparison

This chapter addresses the second research issue. The area under investigation of the research issue is how PPR knowledge can be expressed through modeling techniques. All research questions, RQ2a - RQ2c will be answered in this chapter.

The chapter has the following structure: First, a literature survey for existing solutions and requirements will be presented in section 5.1. Section 5.2 discusses then found requirements to be able to model PPR knowledge and commonly found concepts. In section 5.3, the capabilities of existing languages, presented in section 2.4, will be investigated resulting in a benchmarking table for further comparison. Based on the results of existing solutions and requirements, section 5.4 presents adaptations to an existing solution to be capable of expressing PPR knowledge. The chapter closes with section 5.5, an evaluation of the proposed adaptations with a proof of concept implementation and expert interviews regarding the proof of concept.

5.1 Literature Survey

To answer the research questions of this chapter, again an adapted literature survey was executed. The literature survey follows the in section 3.1 presented approach. In this subsection the most vital and differing elements like like: keyword definition, search string definition and the search process execution will be presented here again.

5.1.1 Keyword definition

In this subsection, the used keywords are presented in table 5.1. The key words are simple and atomic strings that can be combined for a search in a digital library as presented in table 3.1.

process	modeling	requirements
taxonomy	production systems	domain
cyber physical systems	comparison	classification
approach		

Table 5.1: Keywords used for the literature survey regarding requirements for modeling approaches.

5.1.2 Search String definition

The search strings are made up of combinations from individual search key words. Not all combinations of the key words do yield a result or would be clever to use, thus only the best combinations are presented in table 5.2.

process modeling	taxonomy of process modeling languages
domain specific modeling languages	modeling for cyber physical systems
modeling production systems	requirements for modeling languages
requirements for modeling languages	comparison of process modeling languages
classification of modeling languages	process modeling language approaches

Table 5.2: Search strings used for the literature survey regarding requirements for modeling approaches.

5.1.3 Search & Initial Analysis

The search strings were executed on the digital libraries, and the resulting titles were examined regarding possible contributions and how they fit the goal of process analysis methods. As the goal of this literature survey is to find requirements and possible approaches to model PPR knowledge, was it not possible to only include the title or abstract. In most papers, the abstract has only very limited results, but for this chapter and research, the requirements are needed which are in most cases an integrated part but not a key outcome. This is why in this step all papers have been scanned for tables presenting requirements, listings comparing languages or headings describing some form of requirements elicitation. Papers, that clearly focused on data models, simulation, time modeling or version modeling were excluded as this is not part of the scope of the language. From this initial analysis, around 45 papers did get selected for further analyses.

Removal of Duplicates and Initial (Quick) Analysis

A criterion which had to be met for a paper to be included further on, was that at least one specific criterion could be extracted or was visible. All papers containing complete tables or listings of criteria were directly included. After this step there were around 15 papers, that could be used to extract requirements out of them, all of these works can be found in an individual bibliography, *Criteria Literature*, at the end of this work, after the regular bibliography.

5.1.4 Data Extraction (Analysis & Results)

In this step all papers were thoroughly investigated and analyzed regarding possible requirements they impose on a modeling language selection. Also, comparisons of different approaches were stripped down into requirement parts so that a requirements catalogue could be built. A lot of work was put into the comparison of different but similar sounding criteria and mapping them to one term. This was done to keep the number of requirements at a minimum and to not completely overload a possible requirements representation. Papers that are focused on code generation or modeling in a non-graphical way like textual descriptions in form of PROSA text were also excluded, as PPR knowledge needs to be represented graphically as motivated in section 1.1. All found criteria and their literature pointers, reasons for inclusion, possible examples and mappings from criteria to PPR modeling requirements will be presented in the next section.

5.2 Requirements for Product, Process, Resource (PPR) modeling

This section presents requirements for modeling PPR knowledge and thus answers the second research question:

"RQ2a: What are the requirements for modeling PPR concepts in a MDE context for PSE?"

Main input for this section is the literature survey from the previous section. But also the design science cycle from [82] is used as a methodological building block. Through interviews with domain experts and stakeholders, was it possible to elicit some requirements and establish a context including goals that should be fulfilled by a possible PPR modeling language. Found requirements from literature were thus presented to the interview partners and they were allowed to prioritize the criterion. The priorities are A for high, B for middle and C for low.

Table 5.3, presents all criteria that were seen worth keeping. Additional requirements, elicited through the interviews, were only added if they were not already part of the existing list from literature. The table is constructed as follows: On the very left side, the first column indicates if the criteria are basic criteria that apply to most common modeling languages or if the criteria belong to a PPR specific group. The second column is an identification number. In column number three, the name of the criterion is then presented followed by the priority of the criterion. Columns five to seven present a mapping of criterion to PPR pragmatics, column eight and nine are requirements for structure or behavior. A row should always be read like this, for example, line one: *"The requirement, relation/flow, has a high priority and needs to be fulfilled from a possible PPR language to be able to express product, process, resource and structural information."*

group	Nr.	criterion	priority	product	process	resource	structure	behaviour
basic elements of modeling language	1	relation/flow	A	X	X	X	X	
	2	logical operators	A		X			X
	3	convergence and divergence	A	X	X		X	
	4	function/activity	A		X			X
	5	result/state	A	X	X	X	X	X
	6	additional parameters	A	X	X	X	X	X
	7	scalability/granularity	B	X	X	X	X	
	8	comments	B				X	
	9	organizational responsibilities	C			X	X	
modeling pragmatics for PPR	10	product assembly modeling	A	X			X	
	11	production process modeling	A		X		X	X
	12	production resource modeling	A			X	X	X
	13	expressing consistencies between PPR elements	A	X	X	X	X	X
	14	parent-child relations	B				X	
	15	relations between PPR concepts	B				X	
	16	relations between the same concept	B	X	X	X	X	
	17	hierarchical structuring of PPR	B	X	X	X	X	

Table 5.3: Criteria for Product, Process, Resource (PPR) language selection.

Each individual requirement, its priority, and mapping are described in more detail in Appendix B. Here only a brief overview will be given of the most important to note elements and facts for a better understanding of the context and later on how the benchmarking results of different languages came to be.

The first group of requirements, the *basic elements of a modeling language*, is characterized by the two-thirds of high priority characteristics. This is easily explained, as the modeling of PPR knowledge, needs to also build on solid common modeling concepts. For example is *relation/flow* crucial to track a path across an engineering process or the execution path of a production system. For more complex systems is it required to be able to express *logical operators*, as definitely not all systems are simple sequential flows. Further, when the modeling of execution processes and their order is of relevance, logical operators are needed to depict details in the execution paths and find possible optimizations through the use of different resource combinations. Other requirements like *function/activity* and *result/state*, make up the main elements of a modeling language and express the very common elements needed, independent of target knowledge which should be expressed. *Convergence and divergence* explain that one process can, for example, have multiple input or output products/resources. Also in the first group of requirements are two entries that are rated medium priority, as it is not of the utmost importance to scale-up the PPR examples, and it is more common to detail them than to use such a language to plan whole factory layouts. *Comments* are also only medium priority, as most knowledge should be represented structurally through the use of *additional parameters. Organizational responsibilities* are not rated very important, as little use is seen in using them.

Also, the second group of requirements has more than half of high priority entries. The three most important ones are numbers ten, eleven and twelve, as they build the foundation for PPR modeling and go beyond standard concepts that are present in most languages. All of the three concepts are only then fulfilled by a possible target language if each element can be modeled separately but also in combination. This means that for example, it is possible to model product assemblies without the need for blank or ghost activities that have input/output pins representing the product or object. Additionally to modeling PPR knowledge is it important to express consistencies between these three elements, as currently there are often issues of rework and unstructured communications because some inconsistencies are present. Entries 14 to 17 represent then mainly requirements which are concerned with relationships between the PPR concepts or between the same concept. As seen in section 2.2, the PPR trees are interlinked, and so many relationships are present. Extending this approach, the interviews with domain experts yielded these four individual requirements that should be fulfilled by a possible target language.

As many production systems are focused around the execution of processes and the process view is an important one, is it not surprising that eleven out of the seventeen requirements are needed to express process specific properties. This is only topped by the need to also express structural information. As the modeled information has not only to be visually represented but needs a structured form of representation, is this not surprising. The modeling of product and resource-specific characteristics follows the process modeling and finally, behavioral information has the least amount of mapped criteria.

5.3 Capabilities analysis

This section, the third methodological approach to answer the second research issue, is focused on existing modeling techniques and what their capabilities are. Main outcome is the answer to the third research question:

"RQ2b: What are the capabilities and limitations of different existing modeling language approaches considering the different concepts of PPR modeling?"

The main input for this section is twofold. First relevant and existing modeling approaches from literature and practice serve as a foundation. This foundation is needed to have a baseline of languages that are already used to depict process models and thus provide well-established concepts for modeling PPR knowledge. A second input is the in the previous section presented criteria catalog (see table 5.3). All the found criteria build now a baseline against which the found modeling languages are compared against. The result of this benchmarking is presented in table 5.4 below. The first three columns are the same as previously, grouping, numbering, and name of the criterion. Following this basic frame is a "top" column, which represents the best achievable score if all requirements/criteria would be met. After this benchmark column, the individual found languages are listed and compared against the requirements. The number one (1) indicates that the language fulfills the requirement, whereas the number zero (0) indicates that it is not possible to express the criterion in this specific language. All languages that are now benchmarked have been presented in the related work section 2.4.1.

group	Nr.	criterion	Top	FPD	BPMN 2.0	SySML activity diagram	Petri nets	eEPC	IDEF0	SFC
basic elements of modeling language	1	relation/flow	1	1	1	1	1	1	1	1
	2	logical operators/connectors	1	1	1	1	1	1	0	1
	3	convergence and divergence	1	1	1	1	1	1	1	1
	4	function/activity	1	1	1	1	1	1	1	1
	5	result/state	1	1	1	1	1	1	1	1
	6	additional parameters	1	1	1	0	0	0	0	0
	7	scalability/granularity	1	1	1	1	1	1	1	1
	8	comments	1	0	1	0	0	0	1	0
	9	organizational responsibilities	1	1	1	1	1	1	1	0
modelling pragmatics for PPR	10	Product assembly modeling	1	1	0	0	0	0	0	0
	11	production process modelling	1	1	1	1	1	1	1	1
	12	production resource modelling	1	1	0	0	0	0	0	0
	13	expressing consistency dependencies between PPR elements	1	0	0	0	0	0	0	0
	14	Parent-child relations	1	0	0	0	0	0	0	0
	15	relations between PPR concepts	1	1	0	0	0	0	0	0
	16	relations between the same concept	1	1	1	1	0	0	1	0
	17	hierarchical structuring of PPR	1	1	0	0	0	0	0	0
		sum	17	14	11	9	8	8	8	7

Table 5.4: Benchmarking results for existing languages.

5.3.1 Formal Process Description (FPD)

The FPD fulfills nearly all basic criteria, with the only exception of comments that are not part of the language. It is the only language that actually allows to explicitly model the three different concepts of Product, Process, Resource (PPR) and has for each concept a graphical element. However, there are no capabilities in the language that allows expressing consistency requirements between concepts of PPR knowledge. Further is it not possible to follow parent-child relations which can be seen when processes consist of other sub-processes and the relations are lost or the product modeling on a hierarchical level. All other criteria can be met with the FPD allowing the best result of 14 points out of 17 as a benchmarking result.

5.3.2 Business Process Model and Notation 2.0 (BPMN 2.0)

Focusing on representing business processes, is BPMN 2.0 capable of representing all basic elements of a modeling language. Ranging from relation/flow to functions and comments, all elements are present in this modeling language. Regarding PPR knowledge expression does the approach, however, miss out on crucial parts like explicitly modeling products or resources. Both concepts are not directly representable and do not have their own representations. Further is it not possible to express important relations like parent-child, hierarchical structures or relations between the same concept. BPMN 2.0 achieves a result of 14 out of 17 total points.

5.3.3 Systems Modeling Language (SySML)-Activity Diagrams

The activity diagram extension from SySML, supports many basic requirements. Only additional parameters and comments are missing in the language regarding the basic elements. In regard of the PPR concepts is SySML with Activity Diagram (AD) not very suitable as solution. Neither product nor resource concepts can explicitly be expressed and nearly all other concepts are missing as well. Only the relation between the same concept criterion is fulfilled, as this is a basic concept even to model PPR knowledge. The SySMLAD achieves 9 out of 17 points, resulting in the third place of the overall benchmark.

5.3.4 Petri Nets

Petri nets, as presented in section 2.4, are a perfectly good choice for representing and expressing process flows. However, a major drawback is that the tokens cannot be extended with additional parameters and that no comments can be expressed. An even greater drawback of the mathematically supported modeling approach is, that nearly no PPR requirements are fulfilled. Only processes are depicted, as this concept is also in petri nets a core concept realized as transitions. Petri nets thus achieve only 8 out of 17 points.

5.3.5 extended EPC (eEPC)

The eEPC approach has many similarities with BPMN 2.0, but lacks the possibility to add additional parameters and has no comments integrated. As already stated, is it also a problem of eEPC diagrams that organizational units need to be annotated to every task they belong to, making the diagrams cumbersome to work within larger examples. Also like petri nets before, are eEPC diagrams not suited to express PPR knowledge, as major concepts are missing. eEPC diagrams also score 8 out of 17 total points.

5.3.6 IDEF0

The IDEF0 approach is the only one, that has no explicit concepts for complex logical operators. In the language is it not possible to express AND or XOR branches/joins, making it already hard to work with process language for complex situations. It is further hard to add any additional parameters for example tasks/activities, as the principle is not supported by the standard. IDEF0 has also more drawbacks when looking into PPR modeling. Just like petri nets and eEPC, there are no vital concepts of PPR modeling supported by this approach, achieving 8 out of 17 points.

5.3.7 Sequential Funciton Charts (SFC)

The SFC approach, targeting already the production systems domain with the capability of expressing Programmable Logic Controller (PLC) code via diagrams, performs worst of all investigated languages. Basic elements like additional parameters, comments or organizational responsibilities are missing, making it already hard to express basic PPR concepts. In regard of PPR pragmatics the approach performs worse than all other approaches, making it an impossible task to express PPR knowledge in this language. The approach of using SFC as possible foundation is not well founded, as it only achieves 7 out of 17 points.

5.4 Adaptation proposition

The previous two sections presented a) requirements and criteria that need to be met by a modeling language to fully express PPR knowledge and b) capabilities of existing languages regarding these requirements. Table 5.4 presented the results of the existing approaches. The best result is achieved with the VDI/VDE 3682, also known as FPD [79], achieving 14 out of 17 points. This means that the FPD represents a good base for possible adaptations.

Necessary adaptations that need to be made, will be presented in this section. The main focus of this section lies on presenting new elements, based on [9], that are needed, and not on giving again a full overview of the modeling language itself. For an introduction of the FPD, section 2.4 presented the basic concepts and capabilities. All of the proposed adaptations build on the design science cycle from [82]. In an iterative approach, many extensions have been proposed, evaluated implemented and then with domain experts and stakeholders been discussed. Some extensions were directly included while others have been discarded or reworked several times to fit the needs of the target group of modelers, the basic planners in a Production Systems Engineering (PSE) organization.

5.4.1 Relations

FPD has already two kinds of relations, namely for relations between products and processes and between processes and resources. Both of these relations are visually distinguishable. The product to process relation has a normal line with an arrowhead at the end of the relation. Relations between resources and processes are expressed by dotted lines and arrowheads on both the start and the end.

Even though FPD has these two kinds of relations, there are no specifications on how to deal with relations when expressing only one aspect for example only the product side of a model. This is especially of interest when handling different views of a model or the modeling is done in a stepwise approach. Such an approach could be that first the product model is created, then the processes are added and finally a full PPR model is created. To enable such an approach and have visual different elements, figure 5.1 presents relations between the same concepts.

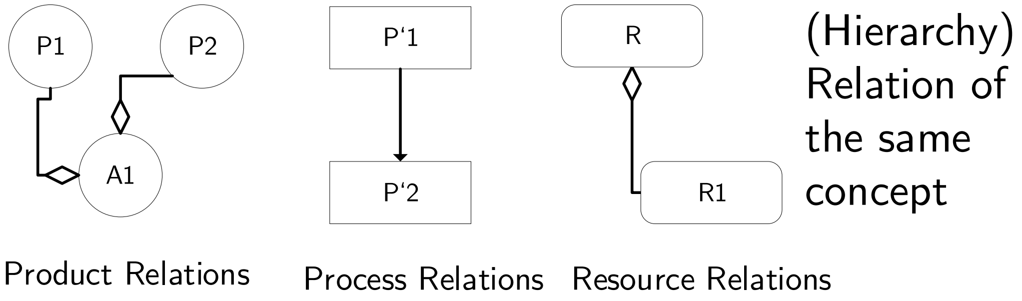


Figure 5.1: Relation extension for individual modeling of Product, Process, Resource (PPR) concepts.

On the very left-hand side are product relations depicted. The connection is a solid line and the end is a white diamond, similar to the Unified Modeling Language (UML) concept of aggregation and specialization. In the example, the two products P1 and P2 are part of the assembly group A1, which is why this shape was chosen. The example also shows already a basic hierarchy, where A1 is the parent to both P1 and P2. P1 and P2 are siblings in this example.

In the middle of figure 5.1, the already existing connection of FPD is reused to visualize process executions. The example shows two processes P’1 and P’2, where P’1 is executed before P’2.

On the very right side of the figure are resource relations depicted. It is very common in modeling production systems to have hierarchies for resources. Often the higher up a resource is the more abstract it is, for example, are welding cells or robots very abstract in contrast to an ultrasonic welding cell. To depict such hierarchies again the diamond shape has been chosen, to allow expressing hierarchies of resources but also to model complete layouts and the containing resources.

Another kind of a relation is expressing relations between sibling nodes in a hierarchy. In figure 5.2 again a very basic hierarchy is depicted. The assembly group A1 has two children P1 and P2. However, in this example is it of interest to express a relation between these two sibling nodes.

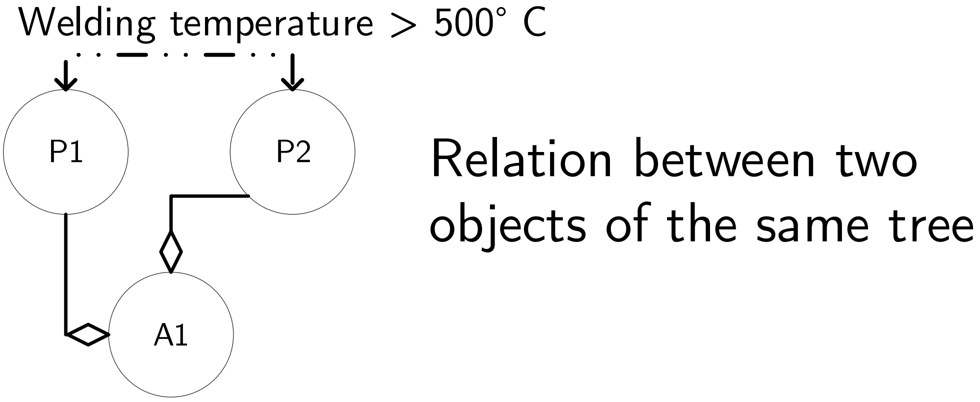


Figure 5.2: Extension for expressing relations between the same concept of Product, Process, Resource (PPR).

When the two product parts P1 and P2 are assembled together, it is of interest to express important dependencies like the welding temperature. This relation was added to the FPD because of again the stepwise modeling. When a basic planner starts with the product tree and does not think about processes or even resources, but wants to express a dependency between two product parts, it should be possible to model such knowledge. Further, does making these concepts explicit help in tracing certain design decisions. For example, if a laser welder is chosen later on with a certain power level, then the decision can be traced back to the product tree already and probably to some product specifications.

5.4.2 Consistency

By following the design science approach [82] and conducting interviews, the domain experts and stakeholders expressed a major concern for modeling PPR. In many cases, it is currently not clear why certain decisions are made. One case is that a basic planner simply has the experience and knowledge and chooses a certain assembly sequence due to this experience, without any certain well founded reasons. This is also called *degree of freedom* in engineering. Not every choice has a well-founded base and in certain situations, the experience of a domain expert is needed. Another case could be, however, that the product, or product parts, have certain properties which influence the process execution and resource selection. For example, parts which are fragile must not be accelerated too quickly in a transport process or plastic parts must not be heated over a certain threshold of degree Celsius.

The second example is the main reason, that consistency relations or dependencies are proposed as an extension for the FPD. A very basic example can be seen in figure 5.3. One product P1 serving as input for the process P'1 and a resource R1 are depicted. Between the product and process as well as the process and the resource there are dot-dashed lines, with arrowheads on the start and end, expressing a consistency relation.

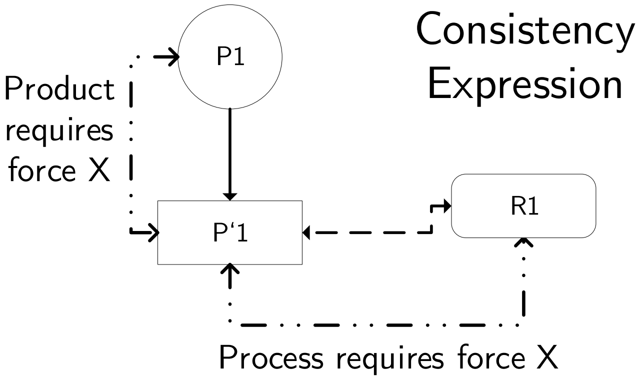


Figure 5.3: Extension for expressing consistencies across Product, Process, Resource (PPR) concepts.

The product part has a requirement that a certain force X needs to be applied. This can be a minimum amount of force or also maximum in case the product might break. But also intervals can be annotated to this consistency relation. Through requiring a certain force level, the process P'1 also requires now a resource which is capable of applying such forces, that is why the link between resource and process is made.

5.4.3 Further Extensions

Through iteratively using the already proposed extensions in the design science cycle with small use cases, further extensions were elicited. These extensions are to some degree only *nice to have*, and not absolute must extensions to be able to model PPR knowledge.

In figure 5.4 all further extensions are presented. A description of each element will follow now.

Starting on the top left corner is the concept of multiplicity. In many use cases is it common to have multiple processes or resources executing the same or at least similar tasks. The reason for this is, that a process might take up several minutes but the requirements specify that the overall process execution must be shorter than that. This leaves only the solution of creating multiple instances of the process. Examples are check processes or firmware flashes for storage chips. These processes usually take up a multiplex than all other processes and thus would increase a potential cycle time enormously. To express this situation and to not have to copy and paste the process and resources several times, resulting in a blown up model, the multiplicity element was created. Multiplicity elements are normal PPR elements from the FPD, but have in the right corner a # and an according number, indicating how often this concept is used or instantiated.

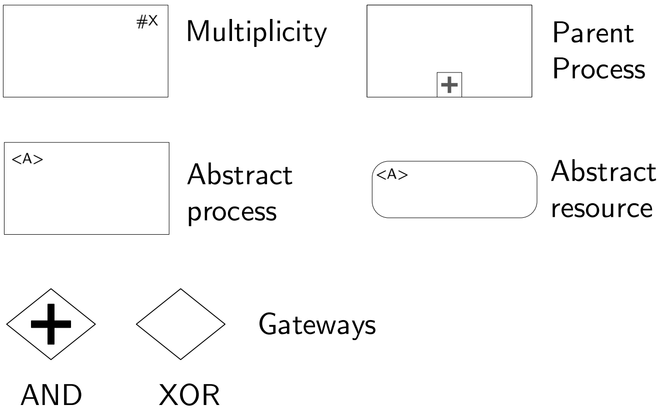


Figure 5.4: Extension for expressing multiplicity, abstract concepts, gateways and meta-processes.

On the top right corner is the concept of a parent or meta-process. This concept is already present to some degree in the FPD, however not explicitly indicated. In the standard [79], a parent process can be modeled and potential sub-processes are indicated by more detailed implementation right next to the parent process. In a complex model with several parent processes, this is, however, a cumbersome solution and makes it hard to work with. This is why a small adaptation has been proposed. The + in the middle of a process indicates that this process is a parent or meta-process and, that there is a more concrete implementation of a sub-process available. This concept allows for a cleaner and layered modeling and is also used for example in BPMN 2.0.

The middle tier of figure 5.4 presents the concept of *abstract processes* and *abstract resources*. As already mentioned is it often the case that in the planning of a production system, an iterative approach is used. This also means, that a potential basic planner starts with very abstract models and later starts to add more details. To allow to express such an iterative approach the abstract processes and resources have been added. An abstract process can be for example a joining process, with no specification if this is a welding, gluing or pressing operation to name just a few. The same concept applies to abstract resources. In the beginning, it is not of utmost interest to specify a concrete implementation of a resource but it is sufficient to say that a resource is a joining resource and has the capability of joining two product parts together. Only in later steps are the concrete implementations like a welding or screwing roboter added and inserted into the model.

The last concept is an explicit visual representation of gateways. In figure 5.4 two examples of gateways are depicted, namely AND and XOR logical operators. The first version of the FPD standard [79], did not consider parallelism or exclusive gateways. This is why in [17], two extensions have been proposed to fill this gap in the FPD. However, the proposed adaptation is visually not very explicit and many interview partners did not like the proposed extension as they found the approach to implicit and not user-friendly. Based on this and the fact that the PPR Engineering Process Analysis (EPA) has already gateways form BPMN 2.0, the extension in this work proposes explicit gateways for logical operations like AND, XOR and so on.

5.5 Evaluation

The last step to close this chapter and the second research issue is the evaluation of the proposed adaptations. This step also, just like the first research issue, builds on a case study approach [64]. The case study for this research issue is methodologically the same as in the previous chapter 4, the difference lies however in the involved stakeholders. As the focus of the PPR modeling language lies on supporting basic planners in a first attempt and in a second step to help further roles to have explicit design decisions, a basic planner was interviewed thoroughly regarding the PPR concepts. For other disciplines in detail planning like mechanical or electrical engineers, process optimizers or operators, proxies have been interviewed for evaluating the proposed extensions.

This last section is split up into two parts. The next subsection 5.5.1 presents a proof of concept implementation with all proposed extensions. First, a general example is presented and then, in an iterative and extendable manner the product, process and finally PPR trees are presented. The second subsection 5.5.2 is then concerned with evaluating the proof of concept with metrics and again a Likert scale. Most of the evaluation examples and outcomes are presented in these two subsections, further details are added to the appendix due to space limitations.

5.5.1 Proof of Concept

The evaluation builds on one example that is used for the proof of concept as well as the evaluation later on with the domain experts. The example is a magnet assembly group, where the final product is a part of the mechanism used in cars to shift the gears. The magnet assembly consists of only three individual parts. First, is a pole metal, a thin sheet of metal with two holes in the middle. Second, are two magnets, which are used to attach the magnet assembly to the gear shift mechanism. Third, is a magnetguard, a plastic frame, enclosing the two magnets. In the middle of the magnetguard construction are two thin pillars that are arranged through the two holes of the pole metal. Figure 5.5 visualizes on the left-hand side the three individual parts.

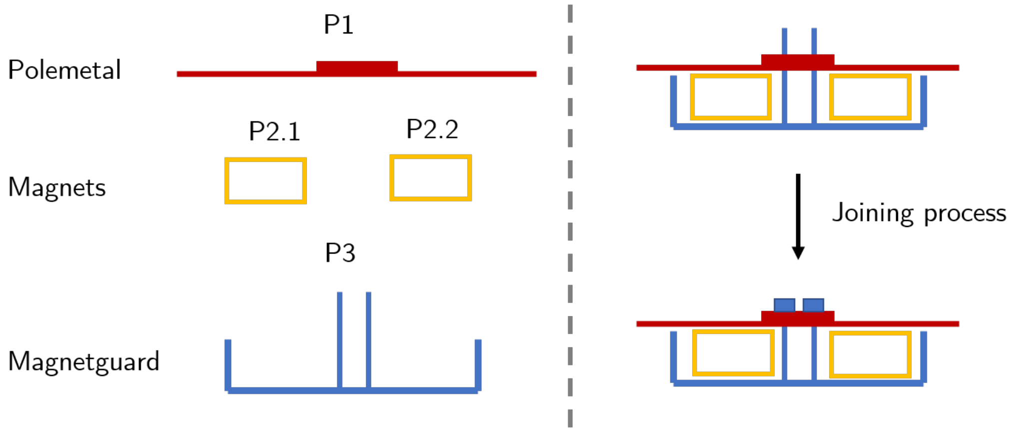


Figure 5.5: Conceptual model of magnet assembly, left the individual parts and right the assembly with a joining process depicting the final product in the bottom right corner.

The three parts are assembled together so that the magnetguard encloses the magnets and the polemetal is placed on top of the magnet guard. Due to the fact that the pillars of the magnetguard are a little bit higher than the pole metal, a joining process is executed to join the magnetguard and the polemetal together. This joining process finalizes the product and is necessary that the individual parts are not released while operations.

In figure 5.5 this joining process is visualized on the right-hand side. On the top, the individual parts are only assembled together while the bottom representation shows the final product where the pillars are joined together with the polemetal through a joining and pressing process.

This conceptualization of the magnetguard assembly group can also be expressed as a product tree with the extensions proposed in the previous section. Figure 5.6 presents the corresponding magnetguard product tree. The figure is structured that the individual parts are on top and are assembled together on their way down through the tree and making up the final product on the very bottom of the figure. The earlier presented relation visualizations with the open diamond are used to express a hierarchy and parent-child relation between assembly groups and the individual parts.

In figure 5.6 there is also a consistency expression, the top yellow circle. The consistency dependency states that the magnets must not be magnetic when they are inserted later on into the physical machine. This is because the placing of the polemetal might be influenced if the magnets are already magnetic. Such a scenario shows already that it is

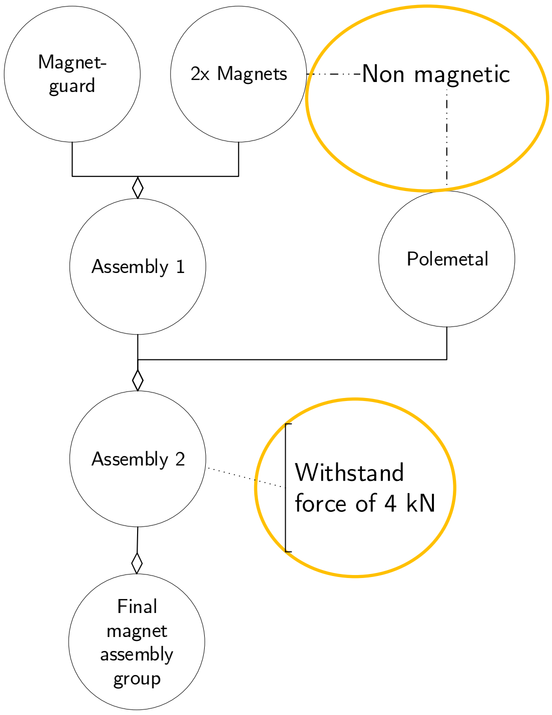


Figure 5.6: Adapted VDI 3682 [79] product tree, depicting a hierarchy of product parts, assembly groups, and the final product

important to support iterative modeling, wherein the beginning only product properties and their impacts are modeled without too much detail of processes or resources.

A second yellow circle shows a comment for assembly group number two. This comment states that the joining operation should yield a result that can withstand a force of 4-kilonewton (kN). Such annotations or comments are necessary because they are not directly derivable from product properties but might be a requirement from the customer or are based on experience of the basic planner. This example shows that not all PPR knowledge can be made explicit and some knowledge is based on degrees of freedom of the domain expert.

After creating the product tree like depicted in figure 5.6, it is the next step to add processes. Also here a conceptualization between products and processes is possible. Figure 5.7 shows the individual parts and which station in the final physical machinery is responsible for what process and later which resource. A very important point to note here is that the sequence of the product parts is not the same as in the previous product tree. This is due to some optimizations and physical space restrictions of the machinery, but does not change anything else from the magnetguard assembly group.

On the very top, station 1 starts and inserts the magnetguard (P3), which is available

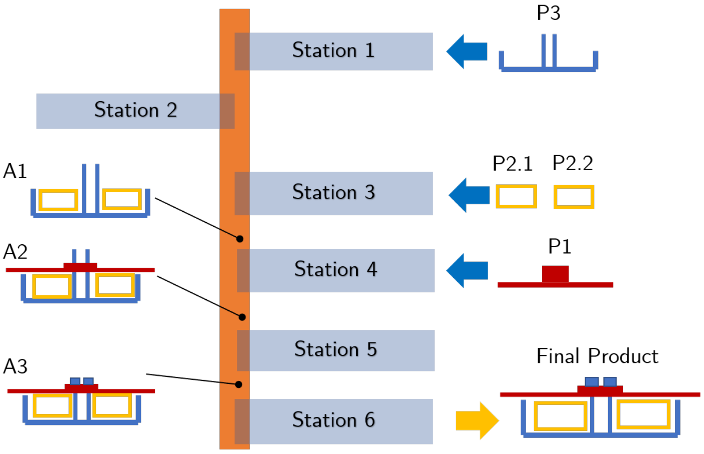


Figure 5.7: Conceptual model of a magnet assembly example.

in containers as bulk material. Due to varying quality and potential damage, station 2 checks if a) the magnetguard has been inserted correctly into the center of the transport system and b) if the product part has no obvious damages. Station 3 adds then the two magnets, which are again present in containers as bulk goods. The two magnets are inserted into exactly into the two notches of the magnetguard. Station 4 inserts then the polemetal on top of the existing assembly group A1. After the fourth process all individual parts are present and assembly group A2 is checked by station 5 again for quality and centering. Station 5 is also responsible for joining the polemetal with the magnetguard together. This joining process is done in two steps, one is heating the plastic to a certain temperature and the second one is pressing the notches on top of the polemetal. After station 5 the final product is basically finished, however, station 6 has to check if the previous processes have all been successful or if any obvious damages have been made to individual parts of the magnetguard assembly group. If all quality checks are successful the final product is dispensed, if not a second dispense container collects all parts with bad quality.

The conceptual process or physical model from figure 5.5 can also be translated into a product tree extended by processes. This is visualized in figure 5.8, but only with three processes, due to space limitations in this work. For the complete model please refer to the appendix.

Figure 5.8 shows all three product parts and the corresponding processes. The first insertion process corresponds to the third station in the conceptual model (figure 5.5). The magnet guard is at this point of time already placed on the transportation system and the two magnets are added into the notches. Also highlighted in this figure just like before with the product tree visualization (figure 5.6), is the consistency relation between the magnet and the polemetal. However, in this extension with the processes, the consistency relation is also added to the process because the process can also not handle magnetic parts, as most of the machinery is made out of metal. If the magnets

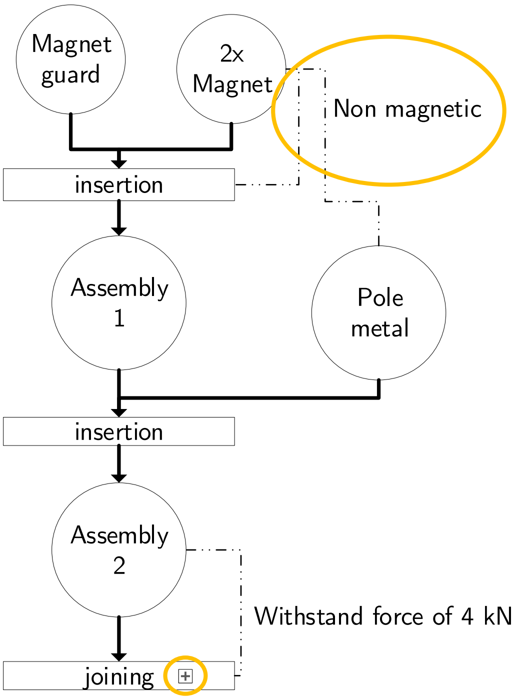


Figure 5.8: Adapted FPD model consisting of products (circles) and processes (squares).

were already in a magnetic state it would be impossible to place them and hold them in place for further processing. After the second insertion of the polemetal assembly group number two is created containing all product parts. The joining process which was already mentioned in the conceptual product model (figure 5.5) and corresponds to station number five, is a parent process. The fact that it this process is a parent process, can be seen by the indicated +, next to the label of the process rectangle. Joining in this process sequence is more complex than a simple one-step process as are the other insertion processes, thus a parent process is chosen. The concrete implementation of the process will be presented later with the full PPR model.

Extending the previous model of products and processes (figure 5.8) with resources, presents figure 5.9 the complete PPR model. The processes now are also in relation with resources that actually provide the capabilities to executed the process, this is indicated by the standard FPD relation symbol a double-headed connector with dotted lines. The example below also presents a custom extension, namely the *abstract resource* concept, highlighted again with a yellow circle. Both the joining and dispense process have an abstract resource with is not yet fully specified. Only the general type of the resource is defined but on a concrete instance is added to the model, which allows the modeler as already said to model in iterations and to postpone certain decisions regarding the resource selection. A postponement might be motivated by limited knowledge about the project which requires callbacks to the customer to later then specify the resource when the necessary blank spots are filled.

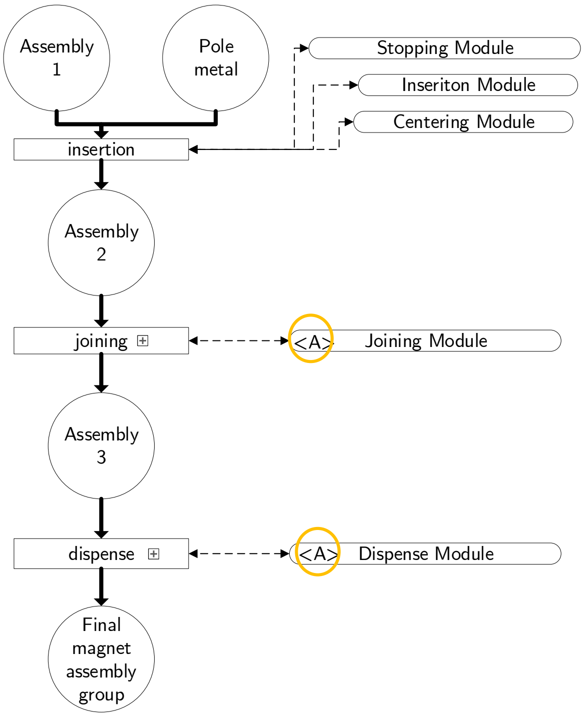


Figure 5.9: Final Product, Process, Resource (PPR) model of the magnetguard assembly group in FPD, with extensions.

The example above consists mainly of standard elements of the FPD. This shows that the proposed adaptations to model PPR knowledge, are not profound and would require high and steep learning curves for stakeholders, but that mainly the necessary adaptations have been introduced.

In figure 5.10 the sub-process and more detailed representation of the joining parent process is depicted. The model uses only already known concepts with the exception of the newly explicitly introduced gateways. In the example AND gateways are used to model the parallel execution of the welding process and the pressing process. Both processes have respective resources, additionally has the pressing process also a consistency dependency to the process as already introduced in the product tree model (figure 5.6. The AND gateway in this example needs to be understood like *sequence parallel* and not on an exact timely basis. This form of modeling does not mean that both processes or even resources must start and end at exactly the same time, or even that PLC code is timed to fulfill such requirements.

The last example for the proof of concept modeling is depicted in figure 5.11 and contains

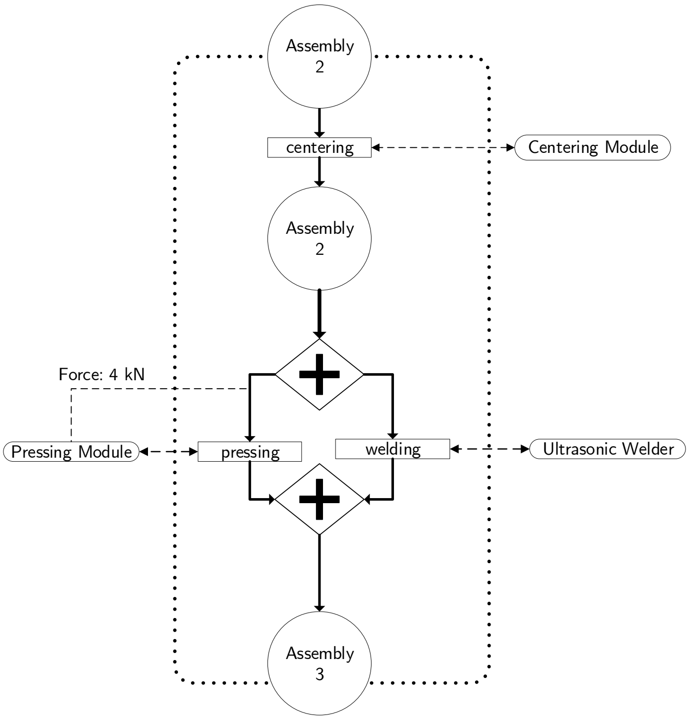


Figure 5.10: Detailed joining process including AND gateway for parallel execution of two processes and their resources.

an XOR gateway. This gateway manipulates the execution paths in such a way that only one of the following processes is executed. As in the example, this allows to model circumstances that are dependent on the outcomes of previous processes. If for example the quality check is negative, then the process corresponding to a not OK state will be executed, otherwise, if the quality check is OK then the sorter will dispense the product in the good quality container and execute the corresponding process.

The example in figure 5.11 also contains a multiplicity extension for the quality check resource. As already described in the adaptations section, multiplicity elements are often used when a process or resource takes up more time than is allowed by customer specifications. In the example, the quality module is available two times and thus reducing conceptually the time needed by half.

5.5.2 Expert Interviews

This subsection presents evaluation results that were elicited through domain expert interviews. The metrics used follow, as in chapter 4, the ISO 25010 metrics [7] and the domain experts had the choice of the same five Likert scale points [2] as before. Newly add metrics will shortly be described before the evaluation results will be presented. The

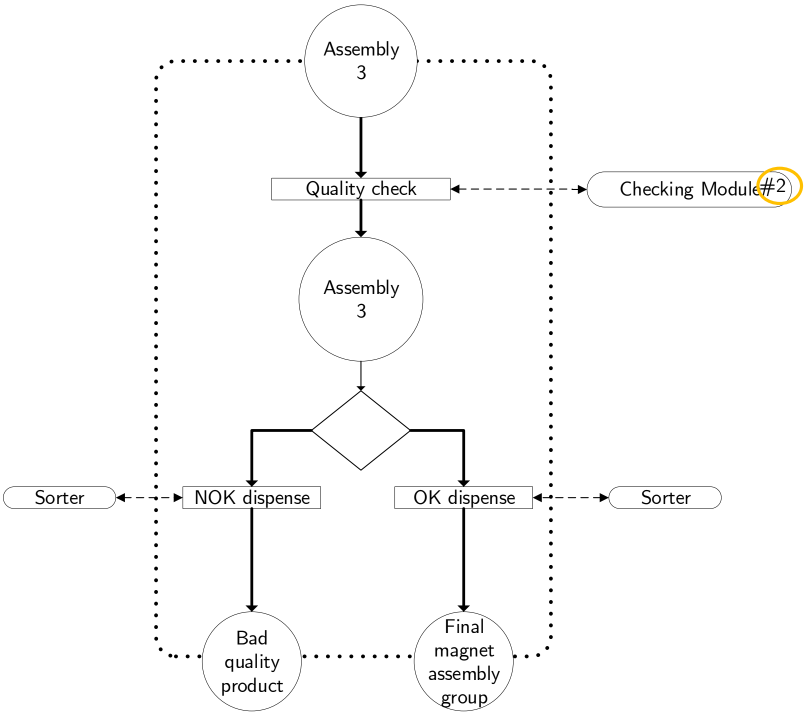


Figure 5.11: Detailed release process, of the individual parts, including XOR gateway for exclusive execution of two process depending on a previous process.

evaluation was done for each concept of PPR separately with multiple domain experts of the different PSE aspects present in a multi-disciplinary engineering (MDE) PSE environment.

Maturity represents how well the needs of a user are met in regard for reliability under normal operation. In the modeling context of a PPR solution this means, that the used concepts are appropriate and mature enough for a domain expert or stakeholder that they would integrate these concepts into their daily work-flows. Additionally, does it incorporate how expressive the potential PPR language with its concepts is.

Learnability is according to the ISO 2510 standard the degree to which a product can be used by users to achieve specified goals. This would mean for a PPR modeling language, that a user is capable of modeling and expressing his thoughts and circumstances as a model. Further are measures for learnability: effectiveness, efficiency, and satisfaction in the specified context. Learnability is an important metric for a new treatment design, as many domain experts need to pick up new concepts and familiarize themselves and switch from an existing solution to something they have not used before.

Operability states how easy it is to operate and control a system. In the context of PPR modeling, operability contains concepts of understandability and usefulness, as potential modelers need to be able to understand the concepts and also find them useful for their daily works. Operability extends the concept of learnability as it requires the domain experts to actually use the concepts and not only understand them.

Time Behaviour expresses requirements regarding the performance and throughput of a product. For modeling PPR knowledge, this means that the models and concepts a modeler wants to express can be mastered in reasonable time. This requirement is very important for the stakeholders as they are not interested in having a perfect solution but which takes up a manifold of time with respect to a current solution.

Co-Existence measures the degree to which a solution can perform its functions while sharing a common environment and also resources with other solutions without impacting the environment negatively. Currently, the basic planners have already a solution, this metric measures how well a potential new solution can co-exist with the already existing environment or if the two solutions would impact one another negatively. An important note here is, that the current solution with excel files is not seen as worthwhile keeping, neither by domain experts nor by stakeholders in the management level. This means that any new solution would inevitably replace the current solution, as long as it does improve the expressiveness and reusability of PPR knowledge. This explains why in nearly all of the following tables the worst grade of the Likert scale is given to this metric.

All of the following tables present in the first column the metric that the interview partner could rate and then follow the three different concepts of product, process, and resource. As a comparison for the grading serves the current approach, which consists of using digital spreadsheets. In the spreadsheets a lot of product information, little process information and additionally a lot more resource information. The interpretation of the Likert scale is thus as follows: "++" means that the modeling of a concept in regard of the metric is much better than the current solution. "+" means that it is better than the spreadsheet approach. "O" can be seen as equally good or bad. "-" means that the new modeling approach is not so good as the current situation. "- -" expresses that a new approach is much worse than the current situation.

Table 5.5 presents the first evaluation results that were achieved by interviewing a basic planner, namely the production process planner. As the production process planner is a major stakeholder of the modeling approach for PPR knowledge, it is not surprising that this role sees many benefits of explicitly representing the knowledge which is currently only hard to express and reuse. As already stated by the explanation of co-existence, the stakeholder does not see the possibility to have multiple solutions present at the same time. Important to note here is, that for the stakeholder this is a positive thing, in terms of forced change so that the old inadequate solution gets replaced by a better fitting one.

All three modeling approaches are seen as very positive (overall quality is "++"). The product modeling approach is seen as positive, only maturity scores equally good as the current approach. This is because now already a lot of information is stored in the spreadsheets and the information gain is not so great. Both process and resource modeling approaches are also very positive in their overall quality. Time behavior is seen as good

Metric	Product	Product + Process	PPR
Functional Appropriateness	++	+	+
Maturity	O	++	++
Learnability	++	++	++
Operability	++	+	+
Time Behaviour	+	O	O
Analyzability	+	++	++
Co-Existence	- -	−	- -
Overall Quality	++	++	++

Table 5.5: Expert interview (production process planner) results regarding the proof of concept modeling examples.

or bad as the current approach, as it requires annotating and adding process/resource information to a product tree.

Now the individual metrics will shortly be discussed for the first domain expert. Functional appropriateness is seen as very positive for the product modeling approach and positive for the other two. This is because the new approaches allow a more explicit modeling and reuse of knowledge. Maturity is as good for the product approach as the current one, however process and resource extensions make the model much more interesting for the production process planner as new information is added. Learnability is seen in all concepts way better than the current approach, as it is argued to be cleaner and easier to understand. Operability is very positive regarding product modeling and positive for the next two approaches. The models from the proof of concept were easy to operate for the stakeholder. Analyzability is positive for the product as it does not bring much more detail. However, process and resource extensions and their relations allow a much better analyzability as is currently present. Finally, co-existence is seen as not possible but which is not a problem for the domain expert.

The results of the next domain expert and stakeholder, the production system planner are depicted in table 5.6. The focus of the production system planner lies on the mechanical construction of the production system. Thus his main interest is resource information with detailed 2/3 D drawings including exact measurements and dimensions.

From the point of view of a production system planner the product modeling approach does not bring a benefit. Currently, he receives a lot of information in CAD drawings which already have all the information he needs. Especially the maturity of modeling a product tree is seen as worse than the current approach, as this would mean that a lot of information is lost, which also explains the negative grade for analyzability. However, is it possible in this situation that both solutions co-exist.

The process approach is seen as more interesting and has an overall good quality grading from the production system planner. Only again maturity is negatively graded all other aspects are a clear improvement, even though only slightly.

Modeling resource aspects is more mixed, regarding the metrics. Functional appropriateness and operability are seen as better than the current approach, as the full PPR modeling allows an overview of the complete production system including relations between the concepts. Learnability and Co-existence are seen neutral in respect to the current approach. Analyzability is graded very positive as currently, only little resource information including relationships is present.

Metric	Product	Product + Process	PPR
Functional Appropriateness	+	+	+
Maturity	- -	- -	- -
Learnability	O	+	O
Operability	O	+	+
Time Behaviour	O	+	O
Analyzability	-	+	++
Co-Existence	O	O	O
Overall Quality	O	+	+

Table 5.6: Expert interview (production system planner) results regarding the proof of concept modeling examples.

Functional appropriateness is seen as better in all three approaches than the current approach. It can be argued that this is because of a better way of abstraction and an easier way to express simple concepts without having the need to use complex proprietary software. Maturity is graded overall negative, which is clear as the model does not contain any detailed measurements of 3D visualizations which is a key aspect for the production system planner. Learnability and operability are seen as neutral for product and PPR modeling, process modeling is positive and better than the current approach. This result is interesting and based on the fact that the mechanically driven production system planner receives more insights from process knowledge and is thus more inclined to learn and operate such models. Time behavior is neutral for the product and PPR approach, only process extensions are better than the current approach. The positive remark is explained through newly introduced knowledge, with upgrades the time to value ratio for the domain expert. Analyzability is worse for the product model because many dimensions and annotations are lost through the basic representations of circles. Product + process models are however better than the current approach and the full PPR model is seen as top. Both of the positive grads can be explained as the domain expert currently has a lot, and nearly too much information and thus analyses are harder to make. Coexistence is overall neutral and two solutions could possibly co-exist from the point of view of a production system planner.

The process optimizer also graded the proof of concept examples in regard to the presented metrics. The results of the domain expert can be seen in table 5.7.

As the name already suggests is the process optimizer very much interested in everything which is process related. Because of that, it is not surprising that the product and

Metric	Product	Product + Process	PPR
Functional Appropriateness	++	++	-
Maturity	++	++	-
Learnability	+	+	O
Operability	++	++	- -
Time Behaviour	++	++	O
Analyzability	++	++	-
Co-Existence	O	O	-
Overall Quality	++	++	-

Table 5.7: Expert interview (process optimizer) results regarding the proof of concept modeling examples.

product + process approach are seen as very positive and that the PPR approach with the extension of resources is not of interest for the domain expert.

Product and product + process modeling is overall very positive, only learnability and co-existence are not graded with the highest remark. Learnability is better than the current approach which consists of using PLC code or in some cases SFC diagrams. A possible PPR solution is overall also seen as neutral in regards to co-existence.

The PPR approach is more mixed than the two approaches before. Learnability and time behavior are seen as neutral and impose the same effort as is currently needed. The remaining metrics are all negative, this is because the process optimizer does care very little about resource information and the addition of this information is of no interest to him.

Due to the fact that the product and product + process approach are graded exactly the same and the PPR approach is of no interest to this domain expert, the discussion of the individual metrics will be omitted as it does not provide any new insights into the situation.

5.6 Summary

Focus of this chapter was to investigate how PPR knowledge can be expressed through the use of a modeling language. Requirements for a good solution were presented in section 5.2 and existing approaches benchmarked in section 5.3. These two outcomes fulfill the promise that was imposed by Research Question (RQ)2a+b.

As the proposed requirements, which have been elicited through domain expert interviews and literature, are not met by any existing approach, section 5.4 proposed adaptations. Through these adaptations, use cases from basic engineers were modeled and proofed the capabilities of the extensions made to the FPD (VDI/VDE 3682) [79]. The domain experts, who evaluated the adaptations, found the outcome of this chapter in its entirety very usable and useful for the different involved engineering disciplines.

CHAPTER 6

PPR Persistence Requirements

The previous two chapters presented a) how an engineering process can be analyzed and represented and b) how PPR knowledge can visually be expressed in a modeling approach. Both chapters represent very important research outcomes. However, it is not enough to simply be able to identify the need to persist PPR knowledge, or be able to express it, but it is also a key point of interest to make this knowledge accessible over a longer time period. This was identified through the evaluation of the PPR modeling language, and a wish of practitioners to be able to retrieve PPR knowledge and reuse it.

This chapter now focuses on use cases that are relevant in regard to PPR knowledge persistence. Found use cases, groups of data and requirements are based on [9]. Also does this chapter answer the last RQ: *RQ3: What are primary use cases that require the persistence of different categories of PPR knowledge?* Section 6.1 focuses here on the existing context, but with a little bit of a different viewpoint and presents four elicited use cases to describe the needs for PPR knowledge persistence. In the second section, 6.2, the use cases are used to derive groups of different data that are present and possibly have implications regarding the requirements for a possible persistence solution. The chapter will be closed by section 6.3, which is concerned with the requirements that are elicited from a) the use cases and b) the groups of data and their characteristics.

6.1 Use Case Elicitation

This section describes how the use cases for PPR knowledge persistence were elicited. As already presented in chapter 3, the research approach uses the design science cycle from Wieringa [82]. Through the analysis of an engineering process, different use cases were elicited and identified. Domain experts, who are concerned with the information aspect and represent the discipline of software engineering, were interviewed regarding these use cases. Here insight into the already known was given, as the basis was the engineering process described in chapter 4.

6.1.1 Use Case 1: Product/ion-Aware Engineering Tool Support

The first use case is concerned with how PPR knowledge and the persistence can be supported through tool support, and also what toolchains should fulfill to be able to persist PPR knowledge. The aspect of persistence was thus investigated from both sides, namely how tools can be supportive but how the development of tools can be supported.

In this field two different use cases can be identified. It is common that engineers need to check characteristics of production process and if they fit to the characteristics of the product that should be manufactured. For this, advanced engineering tools are required. Basic and also detail engineering have here common and different needs that should be investigated and taken into account.

Use Case 1a: Basic Engineering

As already presented in chapter 4, the basic engineers are interested in creating, from product and process knowledge, new resource knowledge. This is schematically depicted in figure 6.1.

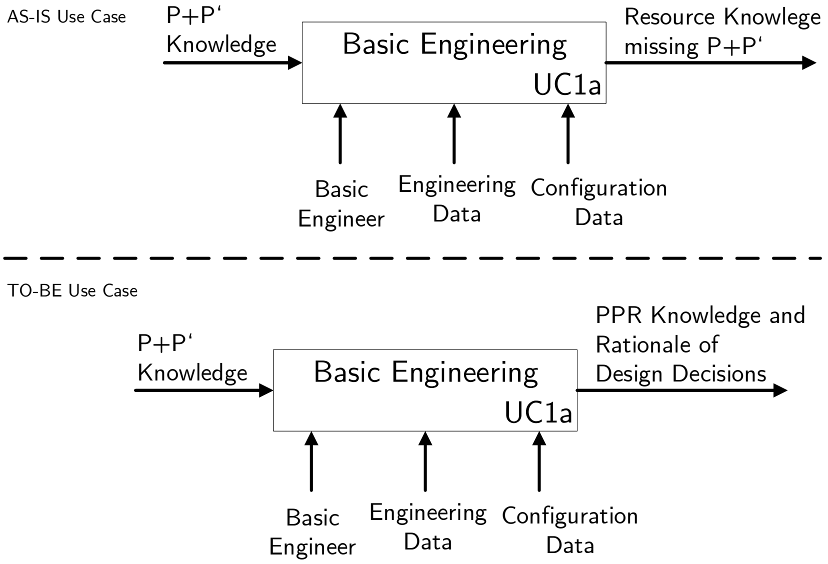


Figure 6.1: Use case 1a, basic engineering, regarding Product, Process, Resource (PPR) knowledge persistence.

The figure consists of two parts. The upper half of the figure is the current process, as already described in length in chapter 4. The lower half describes the TO-BE process, how it should be, so that the use case can successfully be executed. Both parts are again an IDEF0 diagrams, and have product and process knowledge as inputs. The use case *basic engineering*, described here, is the action and produces resource knowledge. In the current situation, only the resource knowledge is persisted. Important to note here is, that for a successful PPR persistence solution, it is required that the product and process information is not lost. The output of the use case indicates this.

The current approach has only resource knowledge where the product and process information is missing and lost. The TO-BE process has however, PPR knowledge and the rationale of design decisions. It can easily be seen that the AS-IS and TO-Be use case have a delta in form of product and process information and their relations. The basic engineer is the role accompanying this use case, visualized as a mechanism (arrow from the bottom).

It was already argued that the basic engineer needs to enrich the product and process knowledge by requirements of product, process and their relationships and all decision rationales. Engineering and configuration data are two categories that are also mechanisms that influence or characterize this use case. This was already, to some extend, presented in chapter 4 and will be discussed in more detail in the next section.

The requirements and what to store in a PPR persistence solution is done by executing the PPR EPA and identifying relevant but missing knowledge and then analyzing these shortcomings and investigating solutions that overcome these limitations.

A key part of the use case is the mapping of product parts to process steps. The current approach in PSE organizations is to store this information in spreadsheets. These engineering artifacts have, however, only poor support for executing this task successfully and efficiently. To identify resource candidates which meet the product and process characteristics/requirements, a basic engineer requires access to these mappings of product features to production resource characteristics. To also be able to access variations of production processes, is it required that a distinction between planned and actual implemented process can be made. These process variations can either be derived from product specifications or resource components and combinations of the two.

Use Case 1b: Detail Engineering

After the basic engineer finishes his work and detail engineering starts, many roles and domain experts work in parallel. This was already depicted in figure 4.3, by the in parallel running engineering tasks of detail engineers. From a persistence point of view, the use case can be depicted as in figure 6.2.

The figure follows the same concept as already presented for *UC1a*. Currently the use case of detail engineering only receives resource knowledge. Product and process knowledge are missing from the current AS-IS use case input. This is clear, as the currently deployed resource knowledge base does not allow storing product and process information. This was already described in the motivation section of this work 1.1. The detail engineer is also a mechanism (arrow from the bottom), and executes the detail engineering use case activity. In this case, engineering data and to some extend configuration data is present. Here it is worth noting, that the configuration data does not have any rationale of design decisions, as the basic engineer does not convey this knowledge. Output from the AS-IS use case is again resource knowledge but with some more details.

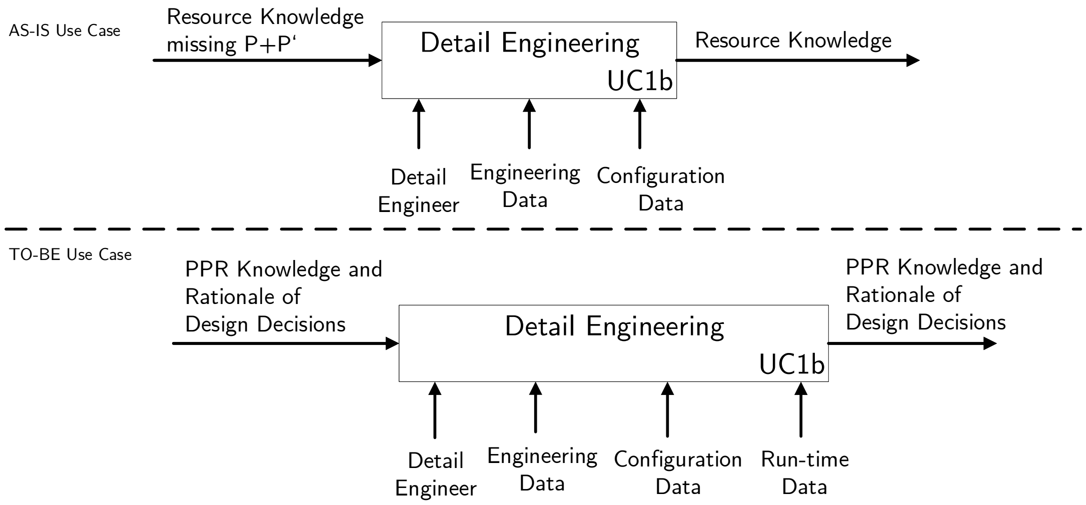


Figure 6.2: Use case 1b, detail engineering, regarding Product, Process, Resource (PPR) knowledge persistence.

The TO-BE use case differs in three aspects. First, input is PPR knowledge from basic engineering and the design decisions. Second, configuration data is available and can be combined with the design decisions for greater quality. Also is now run-time data present which is needed to gain valuable insights into the detail engineering. Third, the output is now also PPR knowledge and again the design decisions. The design decisions are in this use case from both the basic and detail engineer. Design decisions that were made by the basic planner will still be stored, except the detail planner changes them and overrules them with new design aspects.

In the use case execution, the detail engineer is concerned with detailing the received resource knowledge. Here different view-points, like mechanical or electrical are required. Often in this phase there exist design dependencies ranging across disciplines. An example here would be resource configurations, that potentially are stored in spreadsheets or, for example, a relational database. A key challenge for the detail engineer is to reuse already existing components. A differentiation between product specific requirements and solutions and the use of libraries of product families is needed. A library that the detail engineer, currently has in his head or in a nonstructural form stored is the different kinds of resources, for example, motors like electric or combustion. In the context of PPR knowledge the discipline specific view should be linked to the relevant characteristics of the production system under construction. To also be able to optimize the production system, run-time data could provide valuable insights on how the machinery operates and provide valuable insights into quality measurements and optimization potential.

6.1.2 Use Case 2: Product, Process, Resource (PPR) - based Run-time Data Analysis

The second use case is concerned how the engineering process can be improved with new insights coming from run-time data as well as analyses based on this data. From the interviews, there were again two use cases identified which are concerned with integrating new groups of data and gaining knowledge, also in form of backflows.

Use Case 2a: Run-time Process Data Analysis

In the operations phase, detail engineers, and specially the production process optimizer, are interested in how the actual implemented and manufacturing process is executed and behaves. The behaviour that is of interest here focuses on quality aspects of the produced parts, cycle times the machinery needs to produce one complete assembly, and possible maintenance schedules. In figure 6.3, two IDEF0 diagrams are present. The first one, on the upper half, is the current AS-IS use case. The second one, on the lower half,is the TO-BE use case.

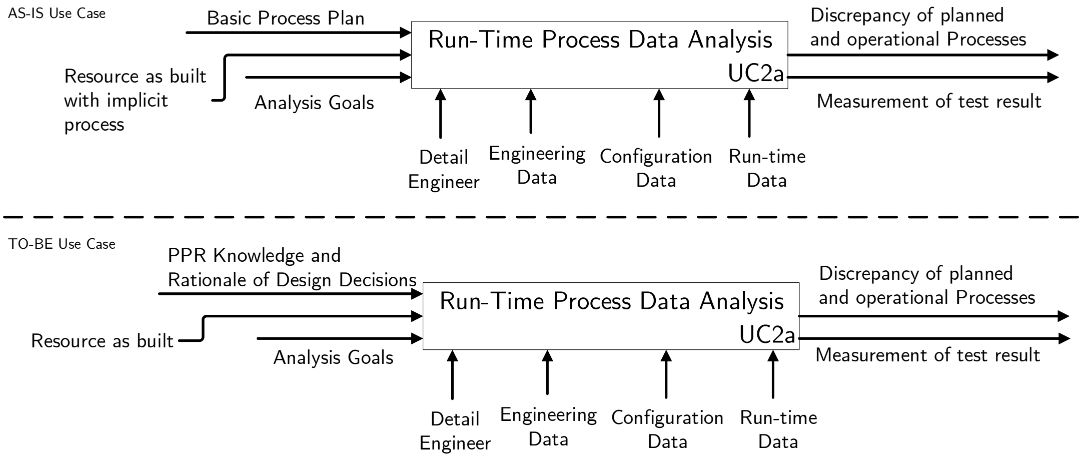


Figure 6.3: Use case 2a, process analysis, regarding Product, Process, Resource (PPR) knowledge persistence.

Current input for this use case are basic process plans coming from basic and detail engineering, resources in physical form with the implicit process (running on the machinery), and analysis goals that should be met. The basic process plan is an artifact, that results from the first use case *UC1*. At this point in time, when this use case is executed, the resource is already present in a physical form. The process is implicit, because the resources execute already their capabilities in form of a process. However, there is no explicit description of the process, which is why there might be a discrepancy between the planned and actual process. The analysis goals represent test-cases and parameter settings for the resource that need to be evaluated. Goal of the use case is to find out if the planned and implemented process deviate. If the planned process is the same as the implemented one, there are no further actions required. If the two processes deviate the analysis should yield if the planned process is better than he actual process, which would suggest some quality issue in the implementation. If the actual process is better or changed, then an improvement was created which should in turn be tracked and fed back to basic engineering for the next projects. The here described output, is only possible if PPR knowledge and rationale about the design decisions is present. This is the case, for example, in the TO-BE use case. In the AS-IS case, the missing knowledge about design decisions, does not allow for fully harvesting optimization potential, making the job of the production process optimizer harder than it should be. Also an output of the use case are measurements of the test results.

Input for the use case execution should be PPR knowledge, including parameter settings for the resource, and also design decisions. The design decisions are here very crucial, as the detail engineer needs to discern between the planned and then the actual process execution. As basic planners do not know or concern themselves too much with the fine tuning of the machinery, the rationales behind chosen settings are enormously important. Only by knowing why certain decisions were made is it possible for the detail engineer to really optimize the production system and not worsen an already fragile situation.

The use case execution of a detail engineer was already described in chapter 4 with the data processing map (see figure 4.3 and to some extend in the previous section with *UC1b*). With the focus of this section and possible PPR persistence approaches, an additional aspect is taken into account. With the knowledge about run-time behavior and data analysis, is the detail engineer in the position to analyze the production process with actual and real operational data logs. These logs including resource information and process execution relevant information give new insights into the overall production process and its system.

Output, as already stated, are distinctions between the planned and the implemented process. This output is achieved by looking into the three different kinds of data that are also input mechanisms (arrows from the bottom) for the use case in figure 6.3. In the next section a more detailed description of these groups of data will be presented.

Use Case 2b: Run-time Data Mining

The fourth overall use case goes one step further than *UC2a* and is also concerned with impacts of design decisions that are made in early engineering phases. Detail engineers, but also possibly basic engineers can gain valuable insights from data mining and analysis, as here potential new knowledge about the production system can be gained. Through analyzing the behaviour of the system and also investigating relationships of design decisions and operations, a link between design and product quality and operation can be made. In figure 6.4, the use case, AS-IS and TO-BE, with its inputs (arrow from the left), outputs (arrow leaving to the right), and mechanisms (arrows coming from bottom) is depicted.

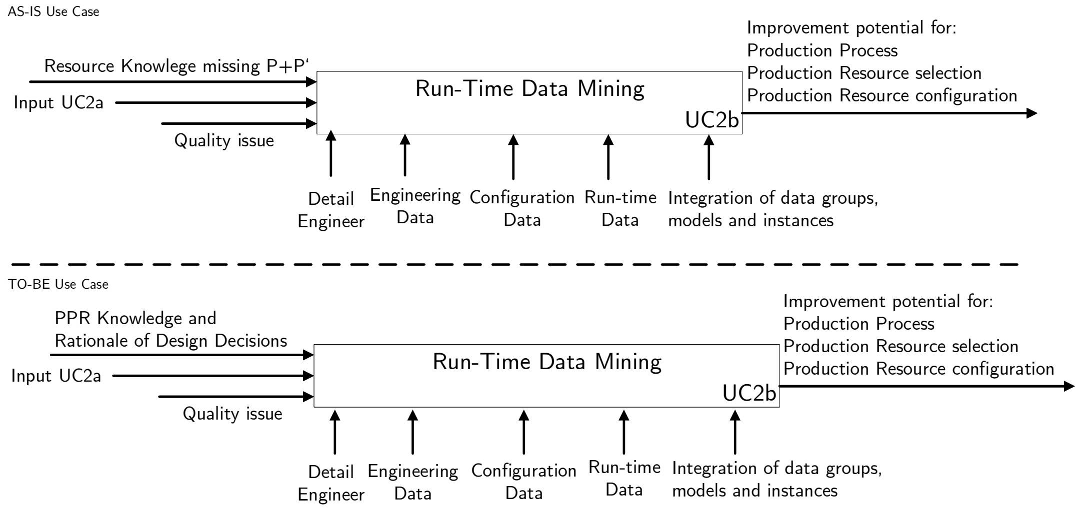


Figure 6.4: Use case 2b, data mining, regarding Product, Process, Resource (PPR) knowledge persistence.

In the AS-IS use case, only product and process information is present, leading to issues in the use case execution which will be taken up at the end of this description. For the execution of the TO-BE use case, PPR knowledge and also the rationale behind design decisions are present. Just like in *UC2a*, the rationale is a very important, and key part in the successful execution of the use case. Only by knowing why certain design decisions were made, is it possible for the detail engineer to actually trace back parameter settings or selections of resources. The behaviour of these resources can then be analyzed by the means of data analysis and data mining. Further, is the outcome of the *UC2a* used for further analysis. Possible quality issues that are the whole reason for investigating and using a data mining approach are also needed.

To be able to execute the use case, a data integration approach needs to be executed and implemented, that allows the integration and aggregation of different groups of data. This is indicated by the mechanism (arrow from the bottom) on the very right of the IDEF0 activity. The use case has engineering, configuration, and run-time data as additional mechanisms. Each category yields valuable insights already by its own, however, the combination and linking between these three data categories allows for even more knowledge harvesting. Detail engineers in this use case are interested in improvements for the production process with the selected resources and also to know what the capabilities, and possible limitations, of the production system family are. Basic planners, even though they are not directly involved in the use case, might get insights into how the production system behaves. This in turn shows them new ways and possibilities for the next project and might influence their choice of a production resource.

Output of the use case are improvement potentials for PPR selections in terms of process and resources combinations, but also possible insights and influential factors regarding resource selection and configuration.

The difference between the current AS-IS process and the TO-BE use case lies in the time, effort and risk it takes to execute the use case. If there is no PPR knowledge present and important design decisions and their rationale are missing, then the whole execution takes up much more time and resources. Further, is it not possible, or only very limited, to investigate issues that affect all three aspects of PPR. For example, design decisions based on the product, that affect the resource, are only hard to retrace as there is no knowledge present. If, however, this knowledge is present, the execution and integration of such a use case with data model and instances becomes much easier and nearly risk free.

6.2 Identification of Groups of Data

The previous section described four main use cases, that are relevant when talking about PPR persistence solutions. For each use case, one ore more groups of data were already presented as mechanisms for the use case execution. These groups of data are already now present in the engineering process. Each category has its own characteristics and different engineering artifacts as examples. In this section the three major categories, that were derived and identified through the use case elicitation will be presented.

6.2.1 Engineering Data

With engineering data, all data that is created during the engineering process, as presented in chapter 4 and figure 4.3, is concerned. Most of the engineering artifacts that are classified as engineering data, contain for example knowledge about the design of a resource station/cell that is manufactured for producing some part of the assembly line. CAD drawings, spreadsheet documents, product hierarchies and assemblies, as well as mappings between individual product parts and the respective resource or process part are examples for this category.

Engineering artifacts and data may vary in their structure from project to project and often also depend on the input that is received from the customer. In some cases customer specifications or requirements have great and deep impact on not only the project in terms of the production system but also how the engineering process is executed and which artifacts are created. Both, basic and detail planners produce engineering data in their work. However, the basic planners and their engineering data represents the starting point for the later phases and is thus very crucial for a successful project execution, because unclear design decisions may impact later reuse and task executions of other roles enormously.

6.2.2 Configuration Data

Second category, configuration data, consists of data that describes the resource under construction with all its relationships between production components and their parameter settings. This data can be seen as configuration of the production system and is very similar to other configuration data and their characteristics, as it can also be characterized in very simple terms. Most of this data consists namely of primitive values like integers, floating point numbers or string representations. This characteristic of nearly a key-value approach can be stored in a classical table structure as there are little requirements for complex object representations or mappings. The schemas that underlie this category of data are rather stable and only the value settings change, but not the whole structure. Challenges arise however, if the structure needs to be tracked and versioned. A special aggravating case is that the semantics of these parameter settings should be explicit and also be tracked. In the interviews however, it was quickly identified that this is not the case and often different parameters for the same configuration or one parameter for different configurations is chosen. In some very important cases, where the settings affected nearly all involved disciplines, one parameter and its name were chosen and named with a typical children's name, so that all involved personnel knew that this is the setting that should not be changed. Even though this was only an idea to bridge a temporal challenge, the parameter and its name made their way through all projects and is now a standard.

Of special interest in this group of data is the relationship between the parameter settings and the other two groups of data. The relationship to engineering data is of interest, as the configuration data is often derived from the engineering data, this link is in later stages or in reuse use cases of interest. The other interesting relation is, from configuration to run-time data, as the parameter settings in a resource are responsible for the produced parts and their quality. Measurements of products and quality metrics provide only then valuable insights, if it is clear how these values came to be, namely through the parameter settings.

6.2.3 Run-Time Data

The last identified data group are run-time data. This set of data consists of all data that is accumulated during the operation of the manufacturing system. In the previous subsection this was already mentioned, the relationship to configuration data is enormously important, as this allows to backtrack the rationale behind design decisions and investigate possible problem areas of a production system. Examples here are analyses, logs, quality measurements from different sensors and other operational data. This group of data can contain very fine grained and granular data objects, with detailed descriptions of the run-time behaviour of a system. Characterising for this data is, that nearly all of it is in some form a time series. As the production system is only turned off for maintenance or set-ups, the machinery produces parts non-stop and thus also produces non-stop data. For this, the different quality metrics have all their individual time series and depict the changes of values over time.

An identified, current, problem is, that this data is stored in relational databases. This is a problem because the characteristics of this set of data and the characteristics and strengths of relational databases do not align. The decision to store this group of data in a relational database lead to a great accumulation of technical debt over the years in the PSE organization. It is now very hard to understand the complete data model, the performance lacks behind modern approaches and the maintainability and flexibility of the data model are nearly non-existing. Another problem is, that this data often only contains resource aspect and does not represent fully PPR knowledge. This is because the data model is already so inflexible that no easy changes can be made and early design decisions can not be reversed anymore.

6.3 Requirements Elicitation

This section now uses the previous two sections to present requirements that can be elicited in regard to PPR knowledge persistence. The here presented set, is only a sub-set and should provide some insights into the challenges that arise when focusing on persisting PPR knowledge. However, the here presented requirements represent the most important ones and are a starting point for further analyses.

6.3.1 Representation of different PPR knowledge groups

The previous two sections already introduced the three main groups of data, namely: engineering, configuration, and run-time data. All of these groups of data have different characteristics and requirements regarding persistence. The rationale behind this requirement can also be backtracked to the use case description in this chapter. *UC1a* focuses on the basic engineering phase, here is it important to create knowledge and have some mechanism to store it. In *UC1b*, detail engineering, is it more of interest to enrich existing knowledge, understand design decisions and if need be create optimizations. On the data analysis side, *UC2a* is interested in investigating potential deviations of processes. There the planned and implemented process might be different and it is the task of a domain expert to find out the consequences and probable causes. *UC2b* is then interested in optimizing or fixing problems that are probably related to all three concepts of PPR knowledge and their relationships. It is the case, that a possible PPR persistence solution needs to be able to fulfill this requirement coming from a) the use cases and b) groups of data. Also should it be possible to represent these different groups of data in an adequate persistence solutions, where the different characteristics of the groups of data are regarded. In chapter 2, the concept of polyglot data storage was introduced [67]. In [54] it was motivated that there is no exclusive relationship between single persistence solutions and multiple, best fitting, approaches. This is exactly what this requirement represents, based on the different use cases and groups of data.

6.3.2 Data Integration and Aggregation

Through the use case descriptions is it evident, that there exist many different data groups. All of the identified groups of data have been described in the previous section. For a PPR persistence solution the requirement to fulfill now, lies in integrating and aggregating these different categories. As already motivated, is it required that the relationships between these different groups of data need to be stored as they represent vital building blocks for back-tracking design decisions. This can be seen throughout all presented use cases. The link between product, process and resource leads always to greater potential in analysis or allows for better quality of the use case execution. The requirement here is also, that possibly a shared but distributed data model needs to be created and then managed. Often is the part of aggregating and cleaning data the most time consuming one. If now a PPR knowledge persistence solution stores already the design decisions with rationale and relationships, is it possible to minimize the time needed for finding complex correlations or relationships, as they are, on a fundamental level, already present. However, it has to be pointed out, that the engineer still needs to model the relationships, otherwise the data integration is still hard to manage.

6.3.3 PPR Variability

As already described, the planned and actual process are often to some extend different. This means, that especially parameter settings or even execution orders differ through a project. *UC2a* is thus solely focused on investigating possible deviations between plan and implementation. Possible causes for a deviation are in any case, good or bad, required to be persisted. In the case the process is not able to provide the needed quality, process optimizations and investigations need to started. This knowledge then needs to be stored and flagged for other projects, as not usable. In case the implementation is better than expected, a backflow and reuse should be possible. A PPR persistence solution thus should also allow for creating and providing access to product, project or library specific knowledge. Access to this knowledge could also be reglemented through one common programmable interface. Regarding parameter settings should a PPR solution be able to discern between process variabilities and differentiated between the planned settings and the current operational settings. Such a distinction then leads to traceable design decisions and backflows for new projects.

The modeling and further persisting of variabilities regarding product, process, or even combination of both should be possible in a PPR persistence solution. Already in basic engineering is it often the case, that different variations of products are present. The creation and manufacturing of whole product families on one production system, minimize the costs per part for the PSE organization and are thus a key point of interest. Through small variations in the production system, are often whole new lines of products possible to be manufactured. Mappings between product parts and resource parts, including possible variability points, should thus also be investigated for a PPR persistence solution.

6.3.4 Reusability of PPR Knowledge

By now it has already be mentioned several times, that the reuse of PPR knowledge is very important. The current described case of the basic engineer (see chapter 4), where the knowledge is created once and can then not be reused, leads to high project costs. However, it is often the case that projects are similar but not exactly the same. The detail engineers thus tend to reuse existing solutions and modify them to new requirements. Reuse in this case is very limited to tools and project specifications. It is not possible to reuse complete process segments or production systems.

In chapter 4 the flow of an engineering process was described. This flow is executed for each new project, resulting in the challenges from *UC1a+b*. To overcome this challenge and minimize project planning times, a possible solution for persisting PPR knowledge should be able to provide reusability. Reusability should, however, not be as is now, informal and expert dependent, but more formal and discipline universal. For this product, process and also resource libraries could be created which can then be used for searching and selecting parts of existing solutions.

6.4 Summary

This last result chapter was focused on investigating use cases from engineering in terms of PPR knowledge persistence. In section 6.1 two use case groups, each containing two use cases, were presented. There the current AS-IS situation was described as well as the elicited TO-BE use cases depicted. The TO-BE models are built on domain expert interviews.

From the use cases was it possible to derive different groups of data (see section 6.2). Three main groups were identified and presented, as well as described with characteristics and examples. Section 6.3 then elicited requirements, fully closing the research of the third and last research issue.

CHAPTER 7

Discussion and Limitations

Now that all results with their evaluations have been presented throughout the last three chapters, this chapter discusses the Research Issues (RI's) and limitations of the solution approach.

7.1 Research Issues

In the following sections, the individual RI's are discussed considering their approach, results and evaluations.

7.1.1 RI-1: Engineering Process Analysis Representation

The aim of the first RI was to investigate a possible engineering process analysis and representing the outcome of the elicited knowledge. The first research question *RQ1a: What adaptions or combinations of business/engineering process analysis methods allow overcoming the limitations of these communities regarding product/ion-aware engineering processes?*, was interested in existing approaches from related work as well as possible adaptations for a multi-disciplinary engineering (MDE) environment. This research issue was mainly motivated by the fact, that existing engineering process analysis methods from both Business Informatics (BI)/Information Systems (IS) and Production Systems Engineering (PSE) communities do not focus on, and subsequently not allow, representing Product, Process, Resource (PPR) knowledge. Based on this, Research Question (RQ)1a investigated a combination of Business Process Analysis (BPA) and Engineering Process Analysis (EPA) methods to overcome their respective limitations which are: BPA methods allow representing the big-picture of an engineering process, are however often too general [62][68]. EPA methods often only focus on intra process improvements, neglecting common interfaces between workgroups and the possible improvements for collaboration and coordination [47][46][36]. The proposed solution focused on providing a repeatable process so that the outcome can be recreated later on. Especially process improvement and engineering stakeholders are interested in this fact, as it is important to have measurable and comparable representations of individual engineering disciplines and their execution paths.

Through following the design science approach [82] and a case study [64], was it possible to create a new treatment for investigating engineering processes and also evaluate the treatment accordingly. However, it was never motivated to provide a full set of interview questions that could be used as a template for future investigation or applications in other PSE organizations. Even though in appendix A there are questions from the conducted interviews, these can not be seen as complete. Many questions were indirectly answered by the interview partners, as they had a sense of foreboding what a follow-up question might be, which made it possible to conduct the interviews in the given time.

In chapter 4 a small sample of engineering task descriptions was presented. The presented table was only a representation and foundation for discussion. Appendix B presents several more task descriptions of the PPR Data Processing Map (DPM). For the chosen tasks input and output artifacts were classified just like in the result table and thus depict better how the results of the PPR EPA look like.

A second key point of interest for the first RI was, to investigate the representation of engineering knowledge, the flow of engineering artifacts and involved stakeholders with their engineering tasks and processes. The according RQ was formulated as follows: *RQ1b: What adaptions or combinations of business/engineering process notations allow overcoming the limitations of these communities for representing stakeholders, processes, and documents that may represent PPR knowledge?* The research in this area was also motivated by the limitations of existing visual representations of knowledge from both BI/IS and PSE communities. Existing approaches like extended EPC (eEPC)[70], are cumbersome to work with and are not suitable for expressing PPR knowledge in an engineering process. This was seen in the evaluation results of chatper 4. Thus, an adaptation for Business Process Model and Notation 2.0 (BPMN 2.0)[3] was chosen and proposed to be able to represent the elicited knowledge appropriately. The adaptation was especially focused on indicating the flow of PPR knowledge through the engineering process, but also which tasks and stakeholders are involved in the engineering process. This work proposed an extension that is capable of indicating where PPR knowledge is created, present, absent or required but not available. Requirements were additionally divided into two levels of importance: *important* and *crucial* for task execution. Possible extensions which are capable of representing general processes and go beyond PSE organizations were not in the scope of the research or the proposed adaptations. This means, that it was never a point of interest to investigate annotations to represent for example batch processing knowledge and possible dependencies in such an engineering process. Even though the proposed visual representation is very expressive, due to the foundation of BPMN 2.0 as standard, are further investigations needed where possible limitations lie. The research did not address very large examples, because it was also the aim of the work to have representative and analyzable representations. Especially the combination of the proposed annotation with process improvements or analysis needs to be considered, as these two major fields of research were not part of this RI.

The outcomes of this first RI are capable of investigating PPR knowledge, which flows through an engineering process, and further to visually represent the outcomes of an engineering process investigation. However, a point of discussion that still remains is the fact, that future research will show how well suited the proposed approach is for other application areas and organizations.

7.1.2 RI-2: Modeling of Product, Process, Resource (PPR) concepts

RI two, focused on modeling and expressing of PPR knowledge. The main application area for this was defined by the requirements and tasks of basic planning in a PSE organization. One such stakeholder is the production system planner, as he starts of each project and his initial created knowledge is later on reused and if possible refined. Motivated by the lack of reuse possibilities, did this RI focus on introducing an adaptation for a modeling language to express PPR knowledge. A first step in this research was to investigate and elicit adequate requirements. These requirements were the main focus of *RQ2a: What are the requirements for modeling PPR concepts in a MDE context for PSE?*

Related work and interviews with domain experts, were the main methodological forces driving the results of this RQ. A detailed description of the requirements can be found in Appendix C. Section 5.2 only gave an overview and a general motivation. All presented requirements are needed for either representing an aspect of PPR, structural or behavioral information. Basic modeling requirements were presented as they are the key building block for nearly all modeling languages and, thus also for representing PPR knowledge. More advanced and specific requirements, mainly focusing on expressing PPR knowledge, were presented as research outcome. The list presented, however, was never motivated to be complete and with high probability is not. In other domains or PSE organizations this list could be used as a basis but should be adapted to specific needs. The second bibliography at the end of this work presents all relevant literature for this outcome. The existing works were the main building block and legitimize many choices of the relevant criteria.

As second key point of investigation was the capabilities analysis of existing approaches introduced. *RQ2b: What are the capabilities and limitations of different existing modeling language approaches considering the different concepts of PPR modeling?* The Formal Process Description (FPD) [79] achieved the highest result based on the elicited requirements. This fact is not surprising as the approach already allows modeling PPR knowledge out of the box, and is easy to adapt for missing concepts like comments or further relations. Other approaches like BPMN 2.0 or Systems Modeling Language (SysML)-Activity Diagram (AD)[11][28] did also perform very well, however, have no concepts to express PPR knowledge which is their major drawback. This major limitation rendered both approaches useless, in the context of this work. The investigated set of modeling approaches consisted of seven different approaches, all focused on process modeling. Investigating approaches for variant or feature modeling were not motivated for this work. However, in the next chapter 8, these two concepts will again be visited.

The *RQ2c: How can representative PPR use cases from domain experts be supported by the identified modeling languages?*, was answered by the elicited use cases and the proof of concept implementation with evaluation through domain experts. For choosing the use cases was it a key requirement, that the use cases were easy to understand for non-experts, as informatics researchers are not educated in complex mechatronical approaches and standards. By choosing these simple, yet representative use cases was it possible to create a proof of concept using all proposed adaptations and evaluating them. However, more complex and larger examples need to be considered, as they are commonly present in an engineering process, and it is vital to know if the proposed approach is also capable of expressing larger examples.

	PPP			PSP			PPO		
Metric	P	P + P’	PPR	P	P+P’	PPR	P	P+P’	PPR
Appropriateness	++	+	+	+	+	+	++	++	-
Maturity	O	++	++	- -	- -	- -	++	++	-
Learnability	++	++	++	O	+	O	+	+	O
Operability	++	+	+	O	+	+	++	++	- -
Time Behaviour	+	O	O	O	+	+	++	++	O
Analyzability	+	++	++	-	+	++	++	++	-
Co-Existence	- -	−	- -	O	O	O	O	O	-
Overall Quality	++	++	++	O	+	+	++	++	-

Table 7.1: Combined evaluation results from the domain expert interviews regarding the Product, Process, Resource (PPR) proof of concept modeling approach.

In table 7.1, the combined results of the evaluation for the found use cases and the proof of concept modeling are presented. The abbreviations are as follows: PPP stands for the first domain expert the production process planner. PSP stands for the second domain expert the production system planner. PPO is the abbreviation for the production process optimizer, who is representative for both detail planning and the operations phase. In the second row, the modeling approaches are also abbreviated. P stands for modeling product assemblies. P + P’ is the combination of product trees with process information. PPR represents the evaluation results for the complete models containing products, processes and resources.

Section 5.5.2 presented already the individual results, that are now simply combined into one table and depicted in table 7.1. For that reason, the individual results will not be discussed here again. However, an overall discussion based on all three outcomes will be made. As can be seen in the overall quality rating, the proposed approach for modeling product assemblies, products and processes and finally PPR is accepted and seen as useful. In five out of nine times, the best grade was achieved, three positive grades, one neutral and also one slightly negative grade was given.

The negative grade is based on the focus that this domain expert has in his field of work. For the production process optimizer, the resource model does not add very much value and thus makes the PPR model not so useful. It has to be noted here, however, that both other approaches are graded very good from the same domain expert.

The only neutral grade can similarly be discussed. The production system planner is not very much interested in the product, and for his line of work he already receives very detailed information about the product. This fact makes it clear why in the total grading, the product assembly modeling did only achieve a neutral grade.

All other seven quality ratings are positive, which indicates that the domain experts expect and look forward to applying these concepts. The ratings also show, that the proposed solution is not only well suited but also needed and useful for their work. This is based on the fact that the proposed approach was compared with the existing solution, which is an unstructured collection of spreadsheets, including many reworks and additional manual tasks. So it can be argued, that in comparison to the existing approach the here proposed solution achieves much better results and probably will have a high acceptance in the PSE organization.

It can be summarized that the research conducted to answer this RI, presented major adaptations for expressing PPR knowledge driven by requirements elicited out of the context of PSE and respective stakeholders.

7.1.3 RI-3: Product, Process, Resource (PPR) Knowledge Persistence

The third research issue focused on how PPR knowledge and persistence solutions come together. Through use case elicitation with domain experts, were two groups and four use cases (two in each group) described. *UC1* investigated again basic and detail engineering phases, but under the light of a persistence solution. The requirements of previous chapters were again seen, as it is also a need for storing PPR knowledge, that the rationale and relationships are stored. This was already seen in the first research issue.

The use cases defined already to some extend different groups of data, which were then in chapter 6 further described. Three main groups were addressed by this work namely engineering, configuration and run-time data. Each use case focuses a little bit more on one of the categories, or is interested in all of them with the relationships between the different aspects. Each category has its different characteristics, leading to requirements for PPR persistence.

The research issue with *RQ3: What are primary use cases that require the persistence of different categories of PPR knowledge?* was motivated by the need of data curators and technical architects to choose an appropriate solution for PPR knowledge persistence. Methodologically did the research issue follow to some extend the design science cycle [82], by interviewing domain experts, eliciting use cases, proposing possible treatments and validating the solution approach through interviews with practitioners. The presented use cases, groups of data and requirements, are thus a good foundation for investigations of possible PPR persistence solutions.

The research issue also picked up an already existing topic, *polyglot data storage*. In chapter 2, the concept introduced by [67], focuses on using multiple storage solutions. Each solution or approach has strengths and limitations and is chosen because a part of the (shared) data model or a group of data fits best to the solution approach. By combining different aspects of persistence technologies, there is no longer the need to decide between relational or NoSQL databases. This was also already discussed, and presented in [54].

Even though the research issue addressed the RQ, there are still open issues that need to be addressed, as they were not motivated for this work. An example is, that the found use cases and groups of data, even though the are representative, need to be investigated further. Especially the *UC1* needs to be investigated regarding differentiation's to the elicited engineering process presented in chapter 4. Another example worth noting is, that *UC2* needs also further investigation. *UC2a* is, as described here in this work a subset of *UC2b*, as the use case focuses only on a process perspective, whereas *UC2b* is a more wholesome approach. It was not motivated how the different approaches and use cases differ or are defined in detail and it was thus never the aim of this research issue to propose a detailed solution. However, further investigations and also prototypical implementations with different persistence solutions could enrich the research of the proposed solution.

It can be summarized, that the presented research for the proposed RQ's and also RI's, represents important building blocks for future research. Through this first approach and investigation of the domain, its context and use cases with groups of data in regard to PPR knowledge persistence, is it now more clear which next steps should be taken and how to address issues, that were identified during the work here.

7.2 Limitations

The conducted research investigated three major research issues, however, several limitations were found during the course of the presented research. Major limitations, that were not possible to overcome in the scope of this work are now presented in the following subsections.

7.2.1 Limitations of the Evaluation Process

As mentioned in section 4.4, the proposed EPA approach was evaluated in an engineering company which is specialized in manufacturing highly automated production systems. The company and its respective engineering process can be seen as representative for the field of PSE organizations, as they follow known engineering phases that were also presented in chapter 2 like [46][78]. Even though the engineering company can be seen as representative, it still is only one company which limits the evaluation concept. Further, was the number of investigated projects and stakeholders, involved in the overall engineering process, limited to one or two representative ones per discipline. Also, was the interview unstructured and did not follow a thorough design phase. The proposed approach for investigating an engineering process might not be fully capable of representing other, more general or specific engineering processes as the evaluation only investigated one PSE organization where no errors were found.

7.2.2 Variability of Engineering Processes

In the subsection above it was already mentioned, that the evaluation conducted only used one engineering company as its basis. However, the engineering process itself also underlies variabilities depending on the project or company. The achieved results of the EPA, thus need to be considered with care, as in the current example and result only one variant of an engineering process was investigated. A second execution of the proposed EPA would benefit the credibility of the results and provide probably new insights into a) the proposed approach and b) the found results and their generalization.

7.2.3 Limited Proof of Concept Evaluation Example

The proof of concept and evaluation for the PPR modeling language was also done in a limited evaluation scenario. A simple use case for the proof of concept was chosen, where only two levels of detail are present. This is a deep enough level for a conceptual model, however, the daily work and projects of engineering domain experts often involve much more detailed examples. The evaluation only focused on the proposed extensions and adaptations but not how well this approach might fit for daily work. Further, did the domain experts not model themselves but only evaluated already predefined and discussed examples. The evaluation in section 5.5.2 was also conducted with a limited number of domain experts for the proof of concepts. Other domain experts might have other results and find limitations that are now not represented due to the small number of available interview partners. However, all interview partners and domain experts were representative for their field and have a huge amount of knowledge and build upon many decades of experience in their fields.

7.2.4 Limited Expertise with Persistence Technologies

The third research issue was focused on eliciting use cases, data groups and requirements for a possible solution to persist PPR knowledge. For this currently, responsible stakeholders were interviewed. These interview partners are responsible for developing engineering tools and supporting the overall software and system infrastructure. However, the interviewed domain experts were all not educated informatics personnel, specialized in developing persistence solutions. Most of them developed their skills on the job, which means the focus lied on developing solutions for current problems. Also had the overall development situation of the PSE organization an impact on the viewpoints of the domain experts responsible for knowledge storage and retrieval. From this point of view, it has to be considered, that the domain experts led to the development of the current system which is cumbersome and ridged. This fact has then also be seen as a limitation because the people who were responsible for the development of the first version of the system are now the interview partners and visionaries responsible for a new approach.

CHAPTER 8

Conclusion and Future Work

As this chapter is the last one of this work, it is now time to draw a conclusion and present future work. Presented future work is based on results, limitations and general open research topics identified in this work.

8.1 Conclusion

The engineering of large scale, highly automated, production systems requires a variety of different disciplines, domains, workgroups and domain experts. This multi-disciplinary engineering (MDE) environment is characterized by many interdependent engineering processes and interfaces between work groups. These interfaces make it often hard for the domain experts to know what impact their decisions have and how they can improve the overall engineering process. The lack of knowledge about the impact of design decisions combined with the focus of domain experts on intra process improvements often lead to information silos [61] Further, even if domain experts want to express all their knowledge regarding Product, Process, Resource (PPR), they currently often do not have the means to convey this knowledge, as little to no means are available.

This work set out to answer the question of *How can the Production Systems Engineering (PSE) process be improved with better PPR knowledge representation?* To answer this question, several steps were undertaken.

First an adapted Engineering Process Analysis (EPA) method was proposed based on existing works from both Business Informatics (BI)/Information Systems (IS) and the PSE community. Both communities provided vital building blocks for a good solution and allowed, in combination, to overcome their respective limitations. With the knowledge of how the engineering process looks like and what possible impact the use and expression of PPR knowledge has, is it possible for domain experts to trace design decisions and start to think about what knowledge they want or need to convey through the engineering process.

Second, requirements were elicited to benchmark existing solution candidates that could be used for expressing PPR knowledge. Also here an adaptation was proposed using the Formal Process Description (FPD) [79]. The focus of this step was to make it possible for domain experts to easily express their design decisions and provide future reuse opportunities through the use of qualitative models, replacing old spreadsheet approaches. Third, use cases were elicited to provide a set of requirements for possible future PPR persistence solutions. In chapter, 1 the introduced *resource knowledge base* was identified as one main bottleneck which is to a large part responsible for technical debt and inflexible engineering situation currently present.

Main results were evaluated by following the design science approach from Wieringa [82] and the application of a case study [64]. The involved domain experts all build upon multiple decades of experience and thus were able to provide invaluable insights into their daily workflows, possible improvements and current process sequences or knowledge that should be adapted or expressed.

In regard to the challenges presented in section 1.2, the research successfully addressed them. Challenge one was concerned with the *insufficient and explicit PP+R knowledge representation.* Here the outcomes presented in chapter 5, address exactly this issue, by providing simple means in terms of a graphical modeling language to express PPR knowledge. For researchers in the area of models and verification, this adapted modeling language presents new insights into how PSE organizations convey knowledge. Thus, is it a good first step to look at these models and apply them in other, maybe more software driven, research areas.

Challenge two, *no backflows*, is addressed more indirectly. Through the use of a modeling language, that is capable of expressing PPR knowledge, the models can be adapted in detail engineering and later be reused. With the proposed concepts, is it now possible to use the PPR models as a single point of truth and create different versions, based on the engineering phase or discipline that is currently working on them. By using one model, which gets adapted and changed but is in its core still the project-specific model, allows seeing the changes that the model underwent and were needed to overcome challenges specific to that project.

The *limited quality assurance* challenge, was addressed in two ways. Also, here the modeling language helps for a more transparent expression of knowledge. But more importantly, the investigated use cases and groups of data, leading to persistence requirements allow now further investigations on concrete implementations and solution approaches. Researchers in the fields of knowledge representation or database design, might be interested in these requirements and use them for their own research agendas.

Even though the research represents a first building block and vital outcomes for research in this area, due to the limited scope of the work in regards of time, several identified topics remain as future work. The next section presents the identified topics for future work and how research in these areas could be beneficial for both BI/IS and PSE communities.

8.2 Future Work

This section investigates possible future work, motivated by the discussion and limitations form the previous chapters.

8.2.1 Process improvement

Chapter 4, presented an approach to investigate the flow of PPR knowledge through an engineering process and an adapted Business Process Model and Notation 2.0 (BPMN 2.0) notation to represent this knowledge. Both, the approach plus the notation were evaluated in an engineering company which is representative for the PSE domain, however further investigations in other PSE organizations should be done. One point of research could be to look into PSE organizations that are not directly driven by PPR knowledge and thus evaluate how well applicable this special approach is in a more general setting. Another key point of research should be the investigation of PPR knowledge and process improvement approaches. As described in chapter 2, there are already many approaches for process investigation and improvement available. Thus a point of interest for future research could be how well the proposed PPR EPA method behaves in a process improvement setting.

8.2.2 Product, Process, Resource (PPR) Language extension

In chapter 5, the process of choosing and adapting a modeling language for expressing PPR knowledge was presented. The language is capable of expressing the most vital concepts like individual modeling of product assemblies or the interconnection of product, process and resource trees. With the proposed solution and presented proof of work is it easily possible to express how a production system is conceptualized and how certain decisions were made. However, the language does not fully allow at this point to express the path how a model came to be. This means especially that there is no possibility for modeling for example variants of products, the creation of product families which also would influence the creation of system families. Further is the language currently limited to expressing one system and its parts, it is not easily possible to reuse or even merge parts of other projects into a new design. The currently proposed language should in next steps be formalized even further and described in a technical setting like Unified Modeling Language (UML). Also, a machine processable form of the modeling language should be investigated so that possible tools and data exchanges can be done.

The presented and evaluated approach is also limited currently to small models and thus it is a key interest for future work to extend the examples used to model and express PPR knowledge. Simple cases for modeling could depict dependencies on the same level of different PPR trees, similar to the currently presented approach. More advanced examples could focus on dependency modeling through different levels between different trees. This means that requirements or consistencies should be expressed for example from a top-level layer of a product to a nested sub-process or even a resource. These examples could then be evaluated in larger scale studies and also under the point of view how concepts of other languages could be adapted for a better fit and easier modeling. In section 5.5 existing languages were benchmarked for the modeling of PPR knowledge, this table could be the basis for future investigations and adaptations with other, already presented solutions.

8.2.3 Variant Modeling

The above presented future field of language extensions in future work, could in one aspect also focus on expressing variants of different kinds like product variants, product families, product and process variants, or production system variants and possible families. Product family and variation assembly/manufacturing were also identified in some interviews with domain experts as a challenges, but never was a focus of this work. Future work could also investigate possibilities of feature modeling focusing on variant management as well as library design for basic modeling concepts and reuse.

8.2.4 Product, Process, Resource (PPR) Data Model

The presented PPR language, allows to express knowledge in a visual way and so depict dependencies and complex situations concerning a production system. However, the sole representation of PPR knowledge is not enough, as this knowledge needs also to be represented in a structured way, requiring a data model. There are already data models, which are focused on the representation of knowledge between a Manufacturing Execution System (MES) and Enterprise Planning System (ERP) system. Such an approach is, for example, ISA 95 [21], which already presents the most common data elements and their interconnections and relations. For example is it possible to depict material, material lots and the processes that are executed for a production system. However, the ISA 95 standard is mostly concerned with batch processing of products, for example, the mixing of chemical substances in tonnes or liters.

A point for future investigation is how a PPR data model could look like and if an existing solution like ISA 95 could be adapted in such a way that it is possible to express all relevant knowledge, without rigorously adhering to a fixed standard but to take the best and fitting ideas and concepts. Further, should a possible data model be investigated with a strong focus on PSE organizations and their need. Research in this field should be driven by the needs of domain experts and stakeholders for utilizing PPR models in their daily work as well as the capabilities of a PPR language. Toolchains spanning over multiple disciplines and phases of engineering could be a second area of future work with a PPR data model. Different tools along a toolchain and their respective users often focus on varying foci. One common data model could provide valuable insights on how to address and overcome challenges in a MDE environment concerning toolchains and their interoperability.

The presented solution in chapter 5, could thus be investigated on how a mapping form this approach to for example ISA95 and AutomationML (AML) could look like. Research in this area could allow presenting useful and usable high-level modeling approaches which are then later mapped for persistence and reuse.

8.2.5 Product, Process, Resource (PPR) Knowledge Persistence

In chapter 6, use cases, different data groups and requirements for a possible PPR knowledge persistence solution were presented and elicited. Only representing PPR knowledge and making it more explicit are not enough, as many engineering projects build on similar but slightly different environmental circumstances. A key challenge for future research is thus how to store PPR knowledge. Existing and well-established solutions in the field of database management and storage need to be investigated under the focus of how well they are suited for storing and querying PPR knowledge. Research in this area could provide more flexibility in the development and maintenance of engineering tools which are created by PSE organizations.

Possible research in the field of PPR knowledge persistence should focus on the application of existing storage solutions and not on creating completely new solutions as there are already good and fitting approaches in existence. The storage should also be combined with approaches regarding functionality and application of knowledge. This would mean that on a very high-level, further use cases for knowledge persistence should be elicited, functionalities like searching, querying and indexing should be investigated and finally the application of persistence solutions investigated. All of these areas present their individual set of challenges and motivate future research for a better and more complete application of PPR knowledge throughout an engineering process.

Future researchers can use this work as a building block for their own research. This might be that the proposed notations or process descriptions seem useful for their application areas, like, for example, process improvements. On another side, can the PPR modeling language be investigated in terms of variability modeling. Identified use cases and groups of data for PPR knowledge persistence, could also be used for a foundation of future discussion with other researchers in disciplines like database design or knowledge representation.

On a more practical note, the research also proposed many applicable concepts, that are useful for practitioners in the fields of BI/IS and PSE. Especially the proposed PPR modeling language can be used by them and further investigated for daily work applicability. Possible fields of application do represent an interesting point for future work and thus bring practitioners and researchers together. Both groups could benefit from future work to further expand the concepts of PPR knowledge modeling and thus improve engineering processes, just like it was the aim of this work.

List of Figures

List of Tables

Acronyms

AD Activity Diagram. 21, 27, 76, 109

AML AutomationML. 31, 32, 119

BI Business Informatics. 2, 8, 9, 20, 28, 33, 34, 40, 49, 56, 107, 108, 115, 117, 119

BPA Business Process Analysis. 3, 16–18, 20, 22, 28, 34–36, 49, 52, 107

BPMN 2.0 Business Process Model and Notation 2.0. 21, 22, 24, 28, 29, 35, 38, 40, 52, 55–57, 59–61, 64–68, 76, 77, 81, 82, 108, 109, 117, 121, 153

DPM Data Processing Map. 35, 49, 51, 52, 54, 55, 57–61, 65, 66, 68, 108

eEPC extended EPC. 22, 28, 34, 35, 40, 52, 56, 77, 108

EO Engineering Organization. 11, 12, 19, 20

EPA Engineering Process Analysis. 3, 8, 16, 19, 20, 22, 24–26, 28, 33–36, 38, 39, 49–59, 61, 65–68, 82, 97, 107, 108, 112, 113, 115, 117

EPC Eventdriven Process Chains. 28, 65–68

ERP Enterprise Planning System. 30, 63, 118

FPD Formal Process Description. 21, 23, 24, 40, 41, 76–82, 86, 87, 93, 109, 116, 121, 122

IS Information Systems. 2, 8, 20, 28, 33, 34, 40, 49, 56, 107, 108, 115, 117, 119

MDE multi-disciplinary engineering. 1–3, 7–10, 33, 34, 39–41, 43, 45, 51, 52, 71, 89, 107, 109, 115, 119

MES Manufacturing Execution System. 30, 118

MU Mechatronical Unit. 12, 13

PLC Programmable Logic Controller. 19, 26, 77, 87, 93

PPR Product, Process, Resource. 1–9, 11, 14–17, 19–22, 24–35, 38–45, 49–52, 54–61, 64–71, 73, 74, 76–82, 84, 86, 87, 89, 90, 92, 93, 95–98, 100–102, 104–113, 115–119, 137, 138, 141, 147–157

PSE Production Systems Engineering. 1–11, 19, 20, 26, 28, 29, 33, 34, 37, 39–41, 44, 45, 49, 52, 56, 58, 61, 67, 71, 78, 89, 97, 104, 105, 107–109, 111–119, 141

RI Research Issue. 8, 34, 44, 45, 107–109, 111, 112

RQ Research Question. 8, 9, 34–38, 40–42, 44, 47, 49, 52, 56, 61, 68, 69, 93, 95, 107–109, 112

RTE Round-Trip Engineering. 1, 5

SE Software Engineering. 2, 3, 9, 10, 37, 44

SFC Sequential Funciton Charts. 21, 26, 40, 77, 93, 121

SysML Systems Modeling Language. 21, 27, 76, 109, 121

UML Unified Modeling Language. 27, 28, 56, 78, 117, 153

XML Extensible Markup Language. 32

Bibliography

[1] JabRef. `http://www.jabref.org/`. Accessed: 2019-01-19 10:48.

[2] I. Elaine Allen and Christopher A. Seaman. Likert scales and data analyses. *Quality progress*, 40(7):64–65, 2007.

[3] Thomas Allweyer. *BPMN 2.0: introduction to the standard for business process modeling.* BoD–Books on Demand, 2016.

[4] Armin Balalaie, Abbas Heydarnoori, and Pooyan Jamshidi. Microservices migration patterns. Technical report, Automated Software Engineering Group., 2015.

[5] Kitchenham Barbara. Procedures for performing systematic reviews. *Joint Technical Report*, 2004.

[6] Kent Beck. *Test-driven development: by example.* Addison-Wesley Professional, 2003.

[7] Nigel Bevan. International standards for usability should be more widely used. *Journal of Usability Studies*, 4(3):106–113, 2009.

[8] Stefan Biffl, Detlef Gerhard, and Arndt Lüder. Introduction to the multi-disciplinary engineering for cyber-physical production systems. In *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, pages 1–24. Springer, 2017.

[9] Stefan Biffl, Lukas Kathrein, Arndt Lüder, Kristof Meixner, and Dietmar Winkler. Data interface for coil car simulation (case study) part ii - detailed data and process models. Technical report, CDL-SQI, TU Wien, 2018.

[10] Stefan Biffl, Dietmar Winkler, Richard Mordinyi, Stefan Scheiber, and Gerald Holl. Efficient monitoring of multi-disciplinary engineering constraints with semantic data integration in the multi-model dashboard process. *IEEE Emerging Technology and Factory Automation (ETFA)*, January 2015.

[11] Conrad Bock. Sysml and uml 2 support for activity modeling. *Systems Engineering*, 9(2):160–186, 2006.

[12] Sébastien Bornot, Ralf Huuck, Ben Lukoschus, and Yassine Lakhnech. Verification of sequential function charts using smv. In *In PDPTA 2000: International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas*. Citeseer, 2000.

[13] Marco Brambilla, Manuel Wimmer, and Jordi Cabot. *Model-driven software engineering in practice*. Synthesis lectures on software engineering. Morgan & Claypool,, second edition. edition, 2012.

[14] David S. Bushnell. Input, process, output: A model for evaluating training. *Training & Development Journal*, 44(3):41–44, 1990.

[15] Cristina Cabanillas, David Knuplesch, Manuel Resinas, Manfred Reichert, Jan Mendling, and Antonio Ruiz-Cortés. Ralph: a graphical notation for resource assignments in business processes. In *International Conference on Advanced Information Systems Engineering*, pages 53–68. Springer, 2015.

[16] Haibo Cheng, Lingling Xue, Peng Wang, Peng Zeng, and Haibin Yu. Discrete manufacturing ontology development. In *Industrial Technology (ICIT), 2017 IEEE International Conference on*, pages 1393–1396. IEEE, 2017.

[17] Lars Christiansen, Tobias Jäger, Frank Schumacher, and Alexander Fay. Modellierungsvorschlag zur grafischen beschreibung alternativer und paralleler prozessabläufe auf basis eines vergleichs bestehender beschreibungsmittel.

[18] Wallace Clark, Walter Nicholas Polakov, and Frank W. Trabold. *The Gantt chart: A working tool of management*. Ronald Press Company, 1922.

[19] David Cohn and Richard Hull. Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull.*, 32(3):3–9, 2009.

[20] International Electrotechnical Commission et al. Technical committee no. 65. *Programmable Controllers–Programming Languages, IEC*, pages 61131–3, 1998.

[21] International Electrotechnical Commission et al. Iec 62264-1 enterprise-control system integration–part 1: Models and terminology. *IEC, Genf*, 2003.

[22] Desmond L. Cook. PROGRAM EVALUATION AND REVIEW TECHNIQUE–APPLICATIONS IN EDUCATION., 1966.

[23] Chris J. Date and Hugh Darwen. A guide to the sql standard, vol. 3, 1987.

[24] Hoda A. ElMaraghy. Changing and evolving products and systems–models and enablers. In *Changeable and reconfigurable manufacturing systems*, pages 25–45. Springer, 2009.

[25] K. Feldmann, T. Schmuck, M. Brossog, and J. Dreyer. Beschreibungsmodell zur planung von produktionssystemen. entwicklung eines beschreibungsmodells für produkte, prozesse und ressourcen zur rechnergestützten planung produktionstechnischer systeme. *wt Werkstattstechnik*, 98(3), 2008.

[26] US Air Force. Integrated computer aided manufacturing (icam) architecture part ii. *Volume IV-Functional Modeling Manual (IDEF0), Air Force Materials Laboratory, Wright-Patterson AFB, Ohio*, 45433, 1981.

[27] Martin Fowler, Cris Kobryn, and Kendall Scott. *UML distilled: a brief guide to the standard object modeling language.* Addison-Wesley Professional, 2004.

[28] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A practical guide to SysML: the systems modeling language.* Morgan Kaufmann, 2014.

[29] Cristina Venera Geambaşu. Bpmn vs uml activity diagram for business process modeling. *Accounting and Management Information Systems*, 11(4):637–651, 2012.

[30] Varun Grover and William J Kettinger. *Process think: winning perspectives for business change in the information age.* IGI Global, 2000.

[31] Xiao He, Zhiyi Ma, Weizhong Shao, and Ge Li. A metamodel for the notation of graphical modeling languages. In *Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International*, volume 1, pages 219–224. IEEE, 2007.

[32] Anders Hellgren, Martin Fabian, and Bengt Lennartson. On the execution of sequential function charts. *Control Engineering Practice*, 13(10):1283–1293, 2005.

[33] Yuze Huang, Jiwei Huang, Budan Wu, and Junliang Chen. Modeling and analysis of data dependencies in business process for data-intensive services. *China Communications*, 14(10):151–163, 2017.

[34] Richard Hull. Artifact-centric business process models: Brief survey of research results and challenges. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1152–1163. Springer, 2008.

[35] Jez Humble and David Farley. *Continuous delivery: reliable software releases through build, test, and deployment automation.* Pearson Education, 2010.

[36] L. Hundt and A. Lüder. Development of a method for the implementation of interoperable tool chains applying mechatronical thinking — use case engineering of logic control. In *Proc. IEEE 17th Int. Conf. Emerging Technologies Factory Automation (ETFA 2012)*, pages 1–8, September 2012.

[37] Tobias Jäger, Alexander Fay, Thomas Wagner, and Ulrich Lowen. Mining technical dependencies throughout engineering process knowledge, 2011.

[38] Nasser Jazdi, Camelia Maga, Peter Göhner, Thomas Ehben, Thilo Tetzner, and Ulrich Löwen. Mehr systematik für den anlagenbau und das industrielle lösungsgeschäft—gesteigerte effizienz durch domain engineering. *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik*, 58(9):524–532, 2010.

[39] Biolchini Jorge, Mian Paula Gomes, Natali Ana Candida Cruz, and Travassos Guilherme Horta. Systematic review in software engineering. *Technical Report RT – ES 679/05, COPPE / UFRJ*, 2005.

[40] Mahmood Reza Khabbazi, Mohammad Khatim Hasan, Riza Sulaiman, and Azrulhizam Shapi'i. Business process modeling in production logistics: Complementary use of bpmn and uml. *Middle East Journal of Scientific Research*, 15(4):516–529, 2013.

[41] Soung-Hie Kim and Ki-Jin Jang. Designing performance analysis and idef0 for enterprise modelling in bpr. *International Journal of production economics*, 76(2):121–133, 2002.

[42] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – a systematic literature review. *Information and Software Technology*, 51(1):7 – 15, 2009. Special Section - Most Cited Articles in 2002 and Regular Research Papers.

[43] Barbara A. Kitchenham, Tore Dyba, and Magne Jorgensen. Evidence-based software engineering. In *Proceedings of the 26th International Conference on Software Engineering*, ICSE '04, pages 273–281, 2004.

[44] Lei Li, Jing-juan Zhu, and Hong-min Li. Idef-based construction supply chain management. In *2009 First International Conference on Information Science and Engineering*, pages 4899–4902. IEEE, 2009.

[45] Rong Liu, Kamal Bhattacharya, and Frederick Y Wu. Modeling business contexture and behavior using business artifacts. In *International Conference on Advanced Information Systems Engineering*, pages 324–339. Springer, 2007.

[46] A. Lüder, M. Foehr, L. Hundt, M. Hoffmann, Y. Langer, and S. Frank. Aggregation of engineering processes regarding the mechatronic approach. In *ETFA2011*, pages 1–8, Sep. 2011.

[47] A. Lüder, M. Foehr, A. Kohlein, and B. Bohm. Application of engineering processes analysis to evaluate benefits of mechatronic engineering, 2012.

[48] A. Lüder, L. Hundt, M. Foehr, T. Holm, T. Wagner, and J. J. Zaddach. Manufacturing system engineering with mechatronical units. In *Proc. IEEE 15th Conf. Emerging Technologies Factory Automation (ETFA 2010)*, pages 1–8, September 2010.

[49] Arndt Lüder, Nicole Schmidt, Kristofer Hell, Hannes Röpke, and Jacek Zawisza. Identification of artifacts in life cycle phases of cpps. In *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, pages 139–167. Springer, 2017.

[50] A. Lüder, J. Peschke, and D. Reinelt. Possibilities and limitations of the application of agent systems in control. In *2006 IEEE International Technology Management Conference (ICE)*, pages 1–8, June 2006.

[51] Vojtěch Merunka. Symmetries of modelling concepts and relationships in uml-advances and opportunities. In *Workshop on Enterprise and Organizational Modeling and Simulation*, pages 100–110. Springer, 2017.

[52] Seyedarya MirRashed, Mohammad Rostami Mehr, Magdalena Mißler-Behr, and Arndt Lüder. Analyzing the causes and effects of complexity on different levels of automobile manufacturing systems. In *Emerging Technologies and Factory Automation (ETFA), 2016 IEEE 21st International Conference on*, pages 1–4. IEEE, 2016.

[53] Thomas Moser, Stefan Biffl, Wikan Danar Sunindyo, and Dietmar Winkler. Integrating production automation expert knowledge across engineering stakeholder domains. In *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*, pages 352–359. IEEE, 2010.

[54] Cory Nance, Travis Losser, Reenu Iype, and Gary Harmon. Nosql vs rdbms-why there is room for both, 2013.

[55] US Navy. Program evaluation research task, summary report phase 1. *AD-735*, 902, 1958.

[56] Kristin Paetzold. Product and systems engineering/ca* tool chains. In Stefan Biffl, Arndt Lüder, and Detlef Gerhard, editors, *Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects*, pages 27–62. Springer International Publishing, Cham, 2017.

[57] Kijung Park and Gül E. Okudan Kremer. Assessment of static complexity in design and manufacturing of a product family and its impact on manufacturing performance. *International Journal of Production Economics*, 169:215 – 232, 2015.

[58] Julius Pfrommer, Denis Stogl, Kiril Aleksandrov, Stefan Escaida Navarro, Björn Hein, and Jürgen Beyerer. Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. *at-Automatisierungstechnik*, 63(10):790–800, 2015.

[59] Louchka Popova-Zeugmann. Time petri nets. In *Time and Petri nets*, pages 31–137. Springer, 2013.

[60] Adrien Presley and Donald H. Liles. The use of idef0 for the design and specification of methodologies. In *Proceedings of the 4th industrial engineering research conference*, 1995.

[61] Juergen Rilling, René Witte, Philipp Schuegerl, and Philippe Charland. Beyond information silos—an omnipresent approach to software evolution. *International Journal of Semantic Computing*, 2(04):431–468, 2008.

[62] Patrick Rosenberger, Detlef Gerhard, and Philipp Rosenberger. Context-aware system analysis: Introduction of a process model for industrial applications. In *ICEIS (2)*, pages 368–375, 2018.

[63] Jennifer Rowley. The wisdom hierarchy: representations of the dikw hierarchy. *Journal of information science*, 33(2):163–180, 2007.

[64] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131, Dec 2008.

[65] Nick Russell, Wil MP van der Aalst, Arthur HM Ter Hofstede, and Petia Wohed. On the suitability of uml 2.0 activity diagrams for business process modelling. In *Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling-Volume 53*, pages 95–104. Australian Computer Society, Inc., 2006.

[66] MirRashed S., Mehr M. R., Mißler-Behr M., and Lüder A. Analyzing the causes and effects of complexity on different levels of automobile manufacturing systems. In *Proc. IEEE 21st Int. Conf. Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, September 2016.

[67] Pramod J. Sadalage and Martin Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence.* Pearson Education, 2013.

[68] Higor Santos and Carina Alves. Exploring the ambidextrous analysis of business processes: A design science research. In *International Conference on Enterprise Information Systems*, pages 543–566. Springer, 2017.

[69] W. Schafer and H. Wehrheim. The challenges of building advanced mechatronic systems. In *Future of Software Engineering (FOSE '07)*, pages 72–84, May 2007.

[70] August-Wilhelm Scheer. Aris-vom geschäftsprozeß zum anwendungssystem, dritte, völlig neubearbeitete und erweiterte auflage. *Berlin et al*, 1998.

[71] Miriam Schleipen, Arndt Lüder, Olaf Sauer, Holger Flatt, and Jürgen Jasperneite. Requirements and concept for plug-and-work. *at-Automatisierungstechnik*, 63(10):801–820, 2015.

[72] Ken Schwaber and Mike Beedle. *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River, 2002.

[73] T. Schäffler, M. Foehr, A. Lüder, and K. Supke. Engineering process evaluation: Evaluation of the impact of internationalisation decisions on the efficiency and quality of engineering processes. In *2013 IEEE International Symposium on Industrial Electronics*, pages 1–6, 2013.

[74] Aisha Siddiqa, Ahmad Karim, and Abdullah Gani. Big data storage technologies: a survey. *Frontiers of Information Technology & Electronic Engineering*, 18(8):1040–1070, 2017.

[75] John Stark. Product lifecycle management. In *Product Lifecycle Management (Volume 1)*, pages 1–29. Springer, 2015.

[76] Margaret-Anne Storey, Emelie Engström, Martin Höst, Per Runeson, and Elizabeth Bjarnason. Using a visual abstract as a lens for communicating and promoting design science research in software engineering. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 181–186. IEEE Press, 2017.

[77] WMP Van der Aalst and KM Van Hee. Business process redesign: a petri-net-based approach. *Computers in industry*, 29(1):15–26, 1996.

[78] VDI Richtlinie 3695: Engineering von Anlagen – Evaluieren und optimieren des Engineerings, 2009.

[79] VDI/VDE 3682: Formalised process descriptions. Beuth Verlag, 2005.

[80] Kostas Vergidis, Ashutosh Tiwari, and Basim Majeed. Business process analysis and optimization: Beyond reengineering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(1):69–82, 2008.

[81] Chad Vicknair, Michael Macias, Zhendong Zhao, Xiaofei Nan, Yixin Chen, and Dawn Wilkins. A comparison of a graph database and a relational database: a data provenance perspective. In *Proceedings of the 48th annual Southeast regional conference*, page 42. ACM, 2010.

[82] Roel Wieringa. *Design science methodology for information systems and software engineering.* Springer, Berlin [u.a.], 2014.

[83] Stefan Wiesner and Klaus-Dieter Thoben. Cyber-physical product-service systems. In *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*, pages 63–88. Springer, 2017.

[84] Petia Wohed, Wil MP van der Aalst, Marlon Dumas, Arthur HM ter Hofstede, and Nick Russell. On the suitability of bpmn for business process modelling. In *International conference on business process management*, pages 161–176. Springer, 2006.

[85] Jacek Zavisa, Arndt Lüder, and Ambra Calà. Designing cooperating multi-agent systems—an extended design methodology. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 252–257. IEEE, 2018.

[86] Liming Zhu, Len Bass, and George Champlin-Scharff. Devops and its practices. *IEEE Software*, 33(3):32–34, 2016.

Criteria Bibliography

[Cri1] Vincenzo Ambriola, Reidar Conradi, and Alfonso Fuggetta. Assessing process-centered software engineering environments. *ACM Trans. Softw. Eng. Methodol.*, 6(3):283–328, July 1997.

[Cri2] M. W. Aziz and M. Rashid. Domain specific modeling language for cyber physical systems. In *2016 International Conference on Information Systems Engineering (ICISE)*, pages 29–33, April 2016.

[Cri3] J. Becker, P. Bergener, D. Breuker, and M. Rackers. Evaluating the expressiveness of domain specific modeling languages using the bunge-wand-weber ontology. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10, Jan 2010.

[Cri4] R. Bendraou, J. Jezequel, M. Gervais, and X. Blanc. A comparison of six uml-based languages for software process modeling. *IEEE Transactions on Software Engineering*, 36(5):662–675, Sept 2010.

[Cri5] Sébastien Bornot, Ralf Huuck, Ben Lukoschus, and Yassine Lakhnech. Verification of sequential function charts using smv. In *In PDPTA 2000: International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas.* Citeseer, 2000.

[Cri6] Jan Hendrik Hausmann, Reiko Heckel, and Stefan Sauer. Extended model relations with graphical consistency conditions. In *UML 2002 Workshop on Consistency Problems in UML-based Software Development*, pages 61–74, 2002.

[Cri7] Christian Cöllen Hendra. Prozessmodellierung für industrie 4.0 komponenten. Master's thesis, Otto-von-Guericke-Universität Magdeburg, 2018.

[Cri8] Birgit Korherr. *Business process modelling - languages, goals, and variabilities.* PhD thesis, Technische Universität Wien, 2008.

[Cri9] Hugues Malgouyres and Gilles Motet. A uml model consistency verification approach based on meta-modeling formalization. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1804–1809. ACM, 2006.

[Cri10] IR McChesney. Toward a classification scheme for software process modelling approaches. *Information and Software Technology*, 37(7):363 – 374, 1995.

[Cri11] Jan Mendling and Markus Nüttgens. Xml interchange formats for business process management. *Information Systems and e-Business Management*, 4(3):217–220, Jul 2006.

[Cri12] O. Schönherr, J. H. Moss, M. Rehm, and O. Rose. A free simulator for modeling production systems with sysml. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–12, Dec 2012.

[Cri13] Oliver Schönherr and Oliver Rose. Important components for modeling production systems with sysml. *IIE Annual Conference.Proceedings*, pages 1–6, 2010.

[Cri14] K. Z. Zamli and P. A. Lee. Taxonomy of process modeling languages. In *Proceedings ACS/IEEE International Conference on Computer Systems and Applications*, pages 435–437, June 2001.

Appendix A: PPR EPA Questionnaire

Before each interview was started, a general introduction into PPR and the concept was given. Each domain expert was picked up from ground zero, knowledge wise and it was tried to address any existing questions before the interview started. This allowed a) a more fluent interview and b) minimized the amount of questions form the interviewees how certain questions are meant.

General Questions for all Interview Partners

Question 1: What is representative PPR knowledge for your domain, your role?

Question 2: Can you split this knowledge up into it's individual product, process and resource parts, providing examples where possible?

Question 3: Is each attribute or property part of a tool, a document on paper or your documentation? Where do you currently receive your information from? Do you need to know for example the pahts of a file to access it and receive so your knowledge?

Question 4: What are current problems in regard of expressing and conveying PPR knowledge?

Question 5: How do you circumvent these problems? An example was given that in some tools attributions are misused to express knowledge for example general child elements are used for process information labeled with a PI and an informal standard established this as process information, how do you deal with things like that?

Question 6: What is your ideal case regarding present knowledge so that you can start your work?

Question 7: What is the normal case you work with? What knowledge is normally present and what is not?

Question 8: How do you express product specific attributions which also have consequences on process and resource design decisions?

Question 9: How important is it for you to express time in regard of PPR knowledge modeling?

Question 10: What files or more general engineering artifacts do you produce in your work/as role?

Question 11: What data, information and knowledge is present in the individual output-artifacts?

Question 12: What files or more general engineering artifacts do you consume in your work/as role?

Question 13: What data, information and knowledge is present in the individual input-artifacts?

Question 14: Can you briefly describe how you exchange your engineering artifacts? How does the data-logistics process look like?

Question 15: It is often the case that roles in later engineering phases require knowledge from your role but this knowledge is often not present. Do you know that or not? Do you have means to express this knowledge but you don't or is there currently no means to express this knowledge?

Question 16: How do you handle versioning of your engineering artifacts?

Specific questions for the production process planner

Question 17: Can you depict your 90 % use case. What are the process tasks you execute with the minimal set of engineering artifacts?

Question 18: Where do the remaining 10 % fit into the overall execution path?

Question 19: Why do you not always execute all possible engineering tasks?

Specific questions for the production system planner

Question 20: What are possible reworks that you can/must perform?

Question 21: After reworking existing artifacts, which process steps need to be re-executed or which engineering artifacts need also be reworked due to dependencies?

Question 22: How is it possible to depict the currently present knowledge and differentiate between a normal use case, a 90 % execution path and the full set of engineering processes?

Specific questions for the production process optimizer and automation engineer

Question 23: How easy is it for you to receive knowledge, which is not already conveyed through the engineering process?

Question 24: What are representative input artifacts?

Question 25: Where do you get your input artifacts from?

Question 26: What happens when you do not receive all necessary input artifacts?

Question 27: What consequences do missing artifacts have for you and your engineering process execution?

Appendix B: PPR DPM Discussion

Representative Process Task of the Production Process Planner

Process Description

The production process planner receives all product life cycle management documents from the customer. This process task is the first task in every single engineering project and depicts the overall start of a possible production system. Crucial products are signed by the customer and the PSE organizations domain experts in charge.

Input Artifact Description

Input artifact name	Customer Requirements
Description	Different requirements from the customer regarding the execution of the project. This artifact is usually a scan of a spreadsheet containing different PPR knowledge aspects.
Product relevant knowledge	Individually used product parts for the assembly. How the product parts are present, in buckets or as individual parts.
Process relevant knowledge	Requirements regarding the cycle time the production system should comply to.
Resource relevant knowledge	Knowledge on setting-up or change the resource. Maintenance times and requirements.

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
Process relevant knowledge	None
Resource relevant knowledge	The mapping between which part is created, or processed on which resource part.

Input artifact name	Product drawings
Description	A PDF file containing figures of the product. Often explosion drawings depicting individual parts are present. Annotations for properties like length, width are often annotated to cross section visualizations.
Product relevant knowledge	Dimensions of the product, individual parts and a rough assembly concept
Process relevant knowledge	None
Resource relevant knowledge	None.

Input artifact name	Timetable for the project
Description	A detailed time table including plannings for major mile stones corresponding to week numbers.
Product relevant knowledge	None.
Process relevant knowledge	None.
Resource relevant knowledge	None.

Output Artifact Description

In this process task no new knowledge is created. All artifacts are simply forwarded to the next process task for further inspection and creation of knowledge.

Representative Process Task of the Production System Planner

Process Description

At this point the production system planner, has already held an initial kick-off and the teams are working under his supervision setting up their individual project specific environments. This task is a major path-breaking point, as it represents probably one of the most vital steps for basic planning and the start for detail planning and operations.

Input Artifact Description

All artifacts, that the production process planner creates are used as input for this task. As the number of artifacts may vary they are not described in detail.

Output Artifact Description

Output artifact name	Internal Time Table
Description	An internal time table is created for synchronizing the most vital steps of the engineering process. This time table is crucial for all involved domains as they need to adhere to it.
Product relevant knowledge	None.
Process relevant knowledge	None.
Resource relevant knowledge	None.

Output artifact name	Detailed Concepts
Description	A first set of more detailed plans is created which goes beyond the knowledge represented by the production process planner.
Product relevant knowledge	None.
Process relevant knowledge	Concepts for production cells and the required processes.
Resource relevant knowledge	Layout drawings in various forms like 2/3D CAD drawings. Concepts for production cells and the required resources.

Output artifact name	Criteria Lists
Description	Success-critical requirements which need to be fulfilled for a project success. These requirements go beyond the requirements of the customer, as internal quality metrics and standards also need to be taken into consideration.
Product relevant knowledge	None.
Process relevant knowledge	None.
Resource relevant knowledge	Detailed requirements regarding the resources and their behaviour.

Representative Process Task of the Automation Engineer and Production Process Optimizer

Process Description

Both the automation engineer and production process optimizer have similar processes which is why one of them is depicted here. The domain expert receives general resource information which needs to be further detailed in regards of cycle times and positioning and wiring programmable modules etc. This process is one of the most vital and detailed processes in the complete engineering process.

Input Artifact Description

Input artifact name	Rough Cycle Time Analysis
Description	An analysis regarding the initial cycle time of the complete production system.
Product relevant knowledge	None.
Process relevant knowledge	Detailed listings of the relevant processes that are executed by the resources and how long each process takes.
Resource relevant knowledge	None.

Output Artifact Description

Output artifact name	Detailed Cycle Time Analysis
Description	A very detailed analysis regarding the cycle time of the complete production system.
Product relevant knowledge	None.
Process relevant knowledge	Possible optimizations of process executions and also possibly rearrangements of process sequences. Extended error state descriptions regarding the cycle times.
Resource relevant knowledge	None.

Appendix C: PPR Language Criteria Discussion

Result/State

The modeling language is capable of representing results of activities or states which are present at a given point in the model. This could be a certain state of a product f. e. representing the state after a process execution where the two input states become one output state – represented as an assembly group.

Justification
This is a very basic and fundamental element that PPR modeling needs. After a process execution a priori state “ante” is transformed into a “post” state [79]. This also can be true for resources or products, or some other parts like tools which do get transported alongside the product. Further should it be possible to model certain results of processes, that they might have on the product or themselves.

Example
Transfers from input to output states and state changes in product-process-resource
Modelling of a state the product is in after a quality check process

Priority: A
Priority Justification
It is important to have a simple representation of states to allow an in-depth modelling of the states which are f. e. present in the resource or products. To better represent the requirements of a real-world application states/results need to be present and express what has happened because of a task execution.

Mapping Explanation
Result/State modeling is required to be able to fully express PPR concepts and also to be able to represent structural information from the real world including behavioural aspects.

Literature References
[Cri10] [Cri12] [Cri8] [Cri4][Cri2] [Cri14][Cri7][Cri3]

Additional Parameters

VDI/VDE 3682 [79] describes the progress of a procedural modelling approach like this:
1. Graphical representation of the process
2. Information model of the process objects and their connection
3. Attributes of the process objects

This shows that there is a need to be able to attribute elements in a modelling language. For the needs in PPR modelling, this should however not be limited to processes only but should span over all elements including products and processes.

Justification
The roles who are working with the model should be able to represent their knowledge and decisions in some form in the model.

Example
Integer values like acceleration for a process step
(structured) text to make important notes on a resource attribute/setting
External reference to a description of resource behavior
Mathematical functions on how to calculate certain values
Pre- and postconditions to model the execution of a process
If the precondition is not met the process gets not executed and the transport is made to the next module
If the postcondition is not met – a retrial like a second approach of screwing could be possible

Priority: A
Priority Justification
This criterion represents the link between the context and the modelling language. For example a simulation tool wants that each process step has an attribute duration, this functionality thus has to be provided by the modelling language and is essential. But also for criteria which are language specific like the consistency expression between P+P and P+R, they also need parameters which can be checked.

Mapping Explanation
Additional parameters are needed to extend the common representations of PPR elements with additional information. This additional information might also have impacts on structural or behavioural aspects of a model or the underlying context.

Literature References
[Cri10] [Cri7]

Relation/Flow (as epitome of time)

Flow can also be interpreted as execution order. Relation of objects like before/after (similar to the sequence in time) but also the connections/relations of the PPR elements, which product gets processed by which production process and which production resource executes this process.

In the modelling language, the expression A –> B should indicate that A gets executed before B and that B is executed after A has finished. This is the very basic concept of time. Because there are no other time relevant aspects to consider like GANTT [18] or PERT charts [55][22] at least this notion of time or sequence is needed. This also follows the IPA – input, processing, output principle [14].

Justification
All the above presented relations should be expressible in the language. It is needed so that the relationship between objects can be inferred and can be created in the model.

The graphical representation of a model should already allow for a notion of time which is not explicitly expressed. With this criterion the basic notion of time is represented. It is easily possible to find start and end of a process and read an ordering of the process steps and thus extract non explicit knowledge from the model.

Example

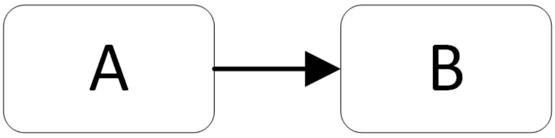


Figure 1: Relation/Flow example where task A is executed before task B is executed.

In the example above (figure 1), there are two tasks: A and B depicted as rectangles with rounded corners. Both tasks are connected via an arrow indicating the relation/flow, meaning that task A is executed *before* task b.

Priority: A
Priority Justification
This represents a very basic concept which needs to be present to be able to model any kind of relationships, due to the fact that the main goal is to model PPR and their relations it is absolutely needed. It is very important to be able to model a flow of process and also assemblies of products etc. This is only possible when a relation/flow element is in place. It is absolutely crucial to have this very basic element of time and sequencing available for the human modeler who creates the model. With this it is possible to express time in a non-explicit way. But also for the interpreter of the model is it of importance to get logical consequences out of a first reading of the model.

Mapping Explanation
The relation/flow criterion is required to be able to express all three of the PPR concepts and additionally structural information of the individual concepts.

Literature References
[Cri8] [Cri4][Cri7][Cri11]

Logical Operators/Connectors

This criterion corresponds to conditions on edges/gateways which allow an alternative routing of the flow and which tasks do get executed. Common operators are parallelism (AND), alternatives (XOR), exception handling triggers and so on. It is a specialization of the requirement “sequential series as epitome of time”, with gateways more complex flows can be expressed.

Justification
I want not only to be able to have sequential flows but also express parallelism and alternatives, thus logical operators like AND/XOR Splits and AND/XOR joins are needed.

Priority: A
Priority Justification
It needs to be possible not only to model sequential flows but more complex flows where operators allow a rerouting or simultaneous execution of tasks.

Mapping Explanation
Logical operators are able to express and manipulate the process flow with resources. But also structural information about the production system and its behaviour can be expressed through this criterion.

Literature References
[Cri10] [Cri13][Cri8] [Cri4][Cri14][Cri7]

Convergence and Divergence

In PPR is should be possible to express the interactions and relationships between the different concepts. Not all of these relations are 1:1, meaning that it is possible that a process has more than one input N:1 or produces more than one output 1:N. The same goes for resources, it is almost all the time the case that more than one resource is needed to completely execute the underlying process.

Justification
The language should not be limited in its expressiveness and allow to model as close to the reality as needed and not impose limitations on the modeler. This is also similar to the “scalability/granularity” criterion there it is more in the direction of aggregation/generalization and here this is continued.

Example
A process requires two input products and welds them together. The process thus has a two to one input to output ratio, also known as convergence.

A process splits one product assembly group up into its individual parts. This is often done if a process is not executed without errors and a (manual) rework disassembles the parts. This is then the divergence concept from one process multiple product parts are outputs.

Priority: A
Priority Justification
Without this requirement the language looses vital expressive features. Only on a very high and abstract level would it be possible to model everything in a 1:1 fashion. But due to the fact, that a PPR modeling language should be able to express more detailed facts it is of high importance.

Mapping Explanation
Convergence and divergence are mainly used to model processes and products, as these two concepts are mostly concerned with this approach. Processes can have multiple inputs or outputs which can either be products or resources. Products are assembled from multiple parts, making up a convergence hierarchy. This kind of information also yields structural knowledge.

Literature References
[Cri7][Cri5]

Function/Activity

Functions/activities represent an executable form of a task. In the concept of PPR this is equivalent to processes which get executed. Tasks do get executed in some order, where here the criterion for “relation/flow” comes into play. This criterion presents a basic building block for many modelling languages and stands in many relations to other elements and criteria to fully enable a language to represent PPR.

Justification
It should be possible to express processes and tasks which get executed either by the resource automatically or by some human worker manually.

Example
Example for tasks are: screwing process, packaging task, but also manual rework tasks or quality checks.

Priority: A
Priority Justification
As stated in the description this criterion makes up a very fundamental part of a language which should represent some form of execution or process modelling.

Mapping Explanation
Processes and their behaviour is mainly expressed by this criterion. All other information is is affected but not directly in contact with this criterion.

Literature References
[Cri1][Cri10] [Cri8] [Cri4][Cri2] [Cri14][Cri7][Cri3][Cri5]

scalability/Granularity

It should be possible to have a higher-level concept of PPR but to be able to go into more depth with each concept. This means, that a process can be a parent process with one product input but in more detail, there are several subprocesses where several input parts of the product assembly are present. This criterion allows for a quick rough layout and first calculation of the most important indicators, and after time allows to detail more fundamental concepts and thus document/model design decisions.

Justification
Different levels of abstraction should be possible to be modeled in a PPR language. This ranges from complete product assemblies with needed processes and resources to lower level concepts of assembly groups.

Example
Model a higher-level process, which can intern have subprocesses. Or model an assembly group for a quicker modeling, and in later engineering phases model the assembly with all individual parts. This can also be visualized like UML Class diagrams with aggregation and generalization [79]

Priority: B
Priority Justification
It is important to have this, but the production process planner often starts with only a rough layout and calculation. This in term means that there won't be much detail anyways in the beginning. Never the less, as time progresses in the project and more details unfold it should be possible to come back to a model and detail it more.

Mapping Explanation
All elements of PPR should be able to contain the same concept as they represent. This allows then to scale models up or down in the level of detail needed. The information about which element contains which other is also structural knowledge.

Literature References
[Cri1][Cri10] [Cri8] [Cri4][Cri14][Cri7][Cri3]

Comments

It should be possible to draw attention to some part of the model which is significant for the context, especially humans in the loop. By adding comments, the model should allow for an obvious way to point to critical knowledge and maybe trace design decisions. The comments should not be used as a form of adding parameters or circumventing the context, like limitations of a tool.

Justification
To make knowledge or certain important notes more explicit and directly visible, there should be an element which allows to model this information.

Priority: B
Priority Justification
It is possible to make a comment in the additional parameter section and not have an extra element for this. But it is a goal of this work to make PPR knowledge explicit and transfer it to later stages in a structured way. If someone else is getting the model and should be ware of special circumstances it is better to display them in the open than to hide them behind parameter settings of objects.

Mapping Explanation
Comments are mainly needed to explain structural or behavioural information. This was a key insight from domain expert interviews and following the design science cycle, that PPR knowledge should be explicit enough to not need any further comments.

Literature References
[Cri10]

Organizational responsibilities

In BPMN 2.0 this is the concept of lanes which also occurs in UML diagrams. This is practical if different roles with responsibilities collaborate to make it explicit which part gets executed under which organizational unit.

Justification
It should be possible to model the responsibilities of modules as an organizational unit. Also in combination with scaling models to higher levels organizational responsibilities gain importance. An example could be the planning of a complete factory.

Priority: C
Priority Justification
In the PPR setting, even if modelling higher level concepts, the organizational responsibility can be identified with the resource or the process or even the product. So, there is already a basic assignment of responsibilities created. It is not of interest to model the organizational responsibilities of different roles such as production process planner. Also is as already discussed the question if for higher level modelling still the same language should be used? Even if so, this criterion should not dominate the basic requirements of the language and the goals.

Mapping Explanation
Modeling organizational units in the context of PPR modeling is only of interest for the responsibilities of resources, as products or processes do not really have a responsibility. However, organizational units are also information conveyors of structural elements.

Literature References
[Cri1][Cri10] [Cri8] [Cri4][Cri14][Cri7][Cri11]

Product Assembly Modeling

Explicitly model products and their assembly via product trees. Here also comes the criterion for hierarchical structuring of PPR into play. It is a specialization of the result/state criterion. Further should this criterion be named product assembly modelling because in PPR it is actually only the case that the assembly is modelled and not the product with its product specific properties like a 3D geometry. Be able to model a product tree all with only product representative information.

Justification It is essential to be able to model the product concept of PPR.

Priority: A Priority Justification
It is essential to be able to model the product concept of PPR.

Mapping Explanation
The modeling of product information is clearly relevant for the product requirement of PPR and also for structural information.

Literature References
[Cri1][Cri12] [Cri8] [Cri14][Cri7][Cri3]

Production Process Modeling

Explicitly model the production processes and their execution. This brings into play the production resource which executes the process and the product which is transformed by the process. It is not the aim to fully model production processes – this means that PPR modelling wants to stay on a higher level than f. e. the engineering process modelling with specific mechanical or electrical process specific models. Be able to build upon the product tree and insert production processes for the assembly groups. Use the products as input for the processes and create outputs from the process execution.

Priority: A
Priority Justification
To be able to model PPR, the process concept is vital.

Mapping Explanation
Process information is needed to model both structural and behavioural aspects of a production system.

Literature References
[Cri8] [Cri2] [Cri3]

Production Resource Modeling

Modeling of the resources needed for specific production processes. Here is it also the aim to model the existence of resources and their part in PPR and not to model physical or geometrical properties. Link production resources to the respective production processes as an individual element in the modelling language.

Justification
To be able to model PPR the resource concept is needed.

Priority: A
Priority Justification
To be able to model PPR the resource concept is needed.

Mapping Explanation
Resource information also contains structural and behavioural aspects of a production system.

Literature References
[Cri8] [Cri2] [Cri7][Cri3]

Expressing Consistencies between PPR Elements

It should be possible to mark links between P+P and P+R as important because there are consistencies which need to be kept in mind.

Justification
Want to make sure that the requirements from a product are met by the process and then also for the resource. Currently there is the problem, that requirements are not fully met or found in the process. It is often the case that the parameters on a resource for a process differ from the actual product requirements.

Example
An example is the screwing process. The product has a requirement, that a force of 30 KN should be applied. The process should know this, that the parameter “force” is 30 KN. If the consistency check is violated there should be an explanation why. The same goes for the selection of a resource. If the process needs 30 KN then all resources are possible where 30 KN is possible. This could then lead to a precondition for resources – needs to be able to screw with > 30 KN

Priority: A
Priority Justification
Currently there is the problem, that often resources are configured differently than the product requirement. In the example above the product needs 30 KN but the robot executes with a force of 42 KN an nobody knows why this decision and derivation was made. This also goes into the direction of traceable design decisions.

Mapping Explanation
This criterion is needed to be able to express the consistencies on a structural level, because the consistencies also form a sort of dependency between two elements. But also are consistencies important to model behavioural aspects, namely what concrete parameters are set for an operational machinery.

Literature References
[Cri9][Cri6]

Parent-child Relations

Expressing the relations between the elements in more detail. Links between the same concept f. e. a higher level process with a nested process is a parent child relation. The link between a process and the production resource would be another relation. Because of these two links between P+P and P+R the relations should be differently modelled. This then allows for a semantic expression of the relations which is not possible with only the relation/flow criterion.

Justification
This criterion needs to be included as it is a part of scaling and nesting different concepts of PPR concepts, as they were already described. For example is it needed to know the relation of parent to subprocess, for a possible execution path.

Priority: A
Justification
PPR does not stand alone all elements in P P R are in a relation like parent child – final product and sub assemblies – so they need relations but also there are relations which will evolve over time and are not categorizable yet.

Mapping Explanation
Parent-child information only contains structural information about the model.

Literature References
[Cri1][Cri10] [Cri13][Cri8] [Cri4][Cri14][Cri7][Cri3]

Relations between PPR concepts

Not only are relations between parent and child objects of interest. Relations between the different concepts of PPR are crucial, as they make up the fundamental parts of the interlinked trees as described in section 2.2.1.

Justification
The criterion needs to be included so that the different concepts can be interlinked. This criterion is to some extend covered and related to *convergence and divergence* but is vital for the PPR modeling, as without it no *convergence and divergence* could be modeled.

Priority: B
Justification
The different concepts need to linked, however it is also possible to express this criterion implicitly, which is why it is only B, priority.

Mapping Explanation
The structuring of PPR concepts is vital for product, process and resource modeling and also contains information about the structure of the model.

Literature References
[Cri13][Cri7]

Relations between the same concepts

As it is a requirement to express consistencies between certain PPR concepts, is it also a criterion to be able to express relations between the same concepts. This ranges from parent-child relations of product assembly groups, to before/after relations of tasks/processes.

Justification
As described above, is this criterion also expressable through other criteria, however it is more important to PPR modeling than to the basic concepts and pragmatics of normal process modeling.

Priority: B
Justification
The relations between the same concepts should explicitly be representable. However, there are already more fundamental criteria which could be used to express this, which is why this is only a B requirement.

Mapping Explanation
The structuring of PPR concepts is vital for product, process and resource modeling and also contains information about the structure of the model.

Literature References
[Cri8] [Cri4][Cri7][Cri11]

Hierarchical Structuring of PPR

When modelling only the product tree a hierarchy is built up naturally, the individual parts are the leaves or children, assembly groups become parents and are also children. The final end product is the root or parent. The same goes for processes, when a process is nested, the highest level represents the parent and all levels below are children to this process.

Justification
Assemblies need to be modeled and structured in a way that the hierarchy they form, from individual part to assembly group to final product can be expressed.

Priority: B
Priority Justification
It is important to be able to express the parent-child relations also in hierarchical form. However this criterion is to some extend already covered by the parent-child concept which is why it is only B.

Mapping Explanation
The structuring of PPR concepts is vital for product, process and resource modeling and also contains information about the structure of the model.

Literature References
[Cri1][Cri10] [Cri13][Cri8] [Cri4][Cri14][Cri7][Cri3]