Modelling Product-Service Systems: An Empirical Analysis of Requirements From a Process-oriented Perspective

Simon Hagen[1], Thorsten Schoormann[2], Sven Jannaber[1], Ralf Knackstedt[2], and Oliver Thomas[1]

1 University of Osnabrueck, Chair of Information Management and Information Systems
Katharinenstr. 3, 49074 Osnabrueck, Germany
{simon.hagen,sven.jannaber,oliver.thomas}@uni-osnabrueck.de
2 University of Hildesheim, Department of Information Systems and Enterprise Modelling,
Universitätsplatz 1, 31141 Hildesheim, Germany
{thorsten.schoormann,ralf.knackstedt}@uni-hildesheim.de

Abstract. Product-service systems (PSS) offer great opportunities for companies to alter their business model, for example, for being competitive in the digital era. One important contributor for developing PSS is the presence of suitable modelling languages. In order to obtain requirements from the industry regarding process-oriented PSS modelling languages, we conducted 18 interviews with experts from 14 different companies. Based on over eight hours of transcripts, a total of 27 requirements were identified and compared to well-established process-oriented modelling languages (e.g., BPMN, UML and EPC). The findings can act as a framework for developing or adapting modelling languages—method engineering—to fulfil current industry needs.

Keywords: Product-service system, Modelling, Process modelling language

1 Introduction

In an increasingly digitalized world, the concept of product-service systems (PSS) gains importance (e.g., [1]) and classic business models aim at shifting to PSS [2]. However, the transformation from a product to a service-centred value proposition is challenging and is requiring changes in different aspects of a company such as the infrastructure and supporting business processes [3, 4]. From a process-perspective of PSS, for example, workflows of services, products and software have to be unified, coordinated and communicated (also across different companies) in order to allow for offering a solution that best fits individual needs of customers. Moreover, due to the increased complexity, further capabilities are needed for which reason value-adding networks are created [5]. In order to tackle these challenges, developing and modelling PSS becomes a crucial aspect in research and practice (e.g., [3, 6]).

Because modelling techniques can support transformation, the demand in adequate languages is present and has been explored in recent years and decades in the context

Multikonferenz Wirtschaftsinformatik 2018,
March 06-09, 2018, Lüneburg, Germany

of PSS [6]. As a result, numerous modelling languages with various purposes are proposed. Due to the variety of such languages and their extensive set of applicability, the selection of appropriate approaches becomes a challenging task. While prior studies on PSS modelling tend to be theoretically, this study aims at empirically gathering requirements from practice that should be addressed by a process modelling technique in PSS. Thus, we formulated the following research question: *Which requirements for process-oriented PSS modelling languages do practitioners currently have?*

Our contribution is a categorized set of requirements that provides guidance for researchers and practitioners in developing process-oriented PSS modelling languages and supporting tools. As a first step, we briefly outline the background related to modelling in PSS (Section 2). Based on our research design (Section 3), we derived five categories and 27 requirements regarding process modelling languages in the context of PSS. To do so, 18 interviews with experts from 14 different companies with more than eight hours of audio recording material were conducted, transcribed and coded. In order to validate our findings and to highlight if current languages support these requirements, we compared the 27 items to six established (BPM) modelling approaches [7] (Section 4). Finally, we discuss the results, provide aspects for further research (Section 5) and conclude with our main findings (Section 6).

2 Process-modelling in Product-service Systems

The integration of (physical) products and services to one marketable bundle of product(s) and service(s), which in combination serve the demands of the user, can be defined as product-service system (e.g., [8, 9]). The share of product or service components is hereby not defined and the system can be offered by a singular firm or a network of companies [10, 11]. The primary goal of a PSS is to provide a solution to the customer that best satisfies his needs in order to generate value for him, and therefore for the offering company. This is accomplished by provisioning the solution through a certain period of time, which leads also to a continuous revenue stream, rather than the onetime sale of a (physical) product. The development of PSS is referred to as product-service systems engineering or service engineering [12].

Since the first appearances of PSS in science and practice, many have thought about intuitive and efficient ways for planning, modelling and developing those systems. In addition, authors also stress the importance of the process dimension of a PSS, since it represents an essential part of the system [4]. Similar to information or process modelling, different approaches were proposed distinctively and in combination (e.g., [6, 13, 14]) – for example, Pezzotta et al. used Business Process Model and Notation (BPMN) combined with Service Blueprinting in their PSS design methodology [15]. However, most of them are developed in a theoretically manner and have not been extensively tested in practice [3], nor are they widely applied in industry. In order to contribute to the outlined deficits, this study focuses on empirical requirements regarding PSS modelling from a process-oriented perspective by analysing common process modelling languages such as BPMN, Unified Modelling Language (UML) and Event driven Process Chain (EPC) [7].

3 Research Design

The data collection was performed by conducting interviews with 18 (domain)experts from 14 different companies, following Meuser and Nagel [15]. In this study, a person is considered an expert due to her/his expertise in a certain domain or industry, however not because of expertise in (process) modelling. The experts selected are from different industries and departments to cover various fields such as software, production and service (Table 1). In order to achieve comparable results, an interview guideline was prepared in advance – based on recommendations by Porst [16]. The questions focus on the representation of PSS in particular, and thus, consist of the following topics: (I) documentation of processes (e.g., which notations? Use of software tools? Experiences with notations? Improvement potential for notations/tools?); (II) use of models (e.g., Where do apply such models?); (III) quality control (e.g., How do you ensure quality of the models?); (IV) responsibilities (e.g., who is in charge for managing models?); (V) data of interviewees (e.g., position in company etc.). For ensuring the applicability of the guideline, it was tested in advance in a pre-test and the interviews were carried out by three different interviewers (IS researchers).

The recorded interviews—a total length of 08:11 hours—were transformed into transcripts and analysed as proposed by Mayring [17]. After paraphrasing important statements, each statement has been generalized to a more generic abstraction level and in two additional steps reduced to the essence of the message. These steps were performed independently by three researchers and merged afterwards to guarantee a common understanding as well as to contribute to the robustness of the results. For categorization, each statement was clustered into a category by the same researchers in an independent manner. Next, in order to find a consensus, the categories were discussed. As a result, five requirement categories were determined (c.f. section 4.1).

Table 1. Overview of experts interviewed

#ID	Expert position	Industry	Length [m:s]
01	Employee	Software (requirements engineering)	10:32
02	CEO	Software (development)	09:17
03	Employee	Software (database management)	13:52
04	Employee	Service (support engineer)	10:54
05	Academic associate	Service	13:28
06	Manager IT	Service (process management, ERP)	11:23
07	Manager „process engineering“	Production (steel industry)	34:03
08	Manager „functional safety“	Production (automotive industry)	7:55
09	Manager „maintenance“	Production (automotive industry)	10:02
10	CEO	Electrical engineering	32:27
11	Manager „service“	Electrical engineering	49:54
12	Deputy CEO	Electrical engineering	23:58
13	CEO	Mobility	25:56
14	Product development	Mobility	49:53
15	Product & company development	Agricultural engineering	57:17
16	Manager „smart services“	Heating, Ventilation & Air Condition industry	40:44
17	Manager „sales“	Heating, Ventilation & Air Condition industry	42:32
18	Manager „market & service“	Agricultural engineering	47:34

4 Results

4.1 Requirements

In total, 27 items were identified as requirements that are titled and described in the following section. Five different categories, namely (1) representation, (2) navigation/ linking, (3) additional information, (4) design/handling, and (5) transformation were derived. Moreover, each of the requirements was assigned with an ID to allow for referencing them in the later sections.

Representation (1). The first category 'representation' (Table 2) deals with aspects that should be illustrated by a modelling language in an explicit way, for example by providing graphical or textual constructs. Due to current changes (e.g., smart factories, smart cities and smart "X"), there is an increasing relevance of data and information. Thus, *information flows (1.4)* to highlight, which actor needs which information at what time (e.g., in a network with various organizations) and *data-objects (1.6)* to consider technology and supporting software should be represented. In addition, *machine/service-related information (1.2), e.g.* availability or downtimes, should be visualized. Experts stated that "machine information" or "electrical circuit diagrams" need to be provided because often an entire process is affected by this.

Another cluster of requirements is dealing with legal aspects, standards and specifications. In global cooperation's, *legal aspects (1.7)* such as employment laws and labour agreements have to be considered, for example, to coordinate which actor may support customer (e.g., 24/7 support). Experts argued that "legal regulations should be represented, for example, in collaboration with locations in the US and Europe […] to spread the service delivery." Besides legal guidelines, internal specifications are relevant such as *safety instructions (1.11*)—which resources (e.g., helmet or safety jacket) are prescribed for certain tasks?—and *time aspects (1.12)*—"visualize deadlines and different timelines" because in a collaboration the execution of certain activities often depend on prior processes and deadlines. In addition, the experts stated that many tasks are carried out because they need to follow *standards, norms or best practices (1.8)*. For example, "models should highlight that machine maintenance is executed following DIN 31051". Accordingly, a modelling language should allow visualizing which steps base on a certain standard, norm or best practice, because it supports the understanding of why a process is done in a specific manner and highlights the quality standards of a company.

Generally, actors, products and tasks are essential for every business process. Thus, a modelling language should allow representing (physical) *products and service objects (1.1)* and provide detailed models such as CAD-diagrams. Experts argued that models should "represent (sub-)products to assign activities to a specific output". Furthermore, different *types of activities (1.9)* are required to allow identifying analogue and digital ones. Experts highlighted this and argued that they "would like to analyse, which steps are executed analogue and which ones digitally." In doing so, *decisions (1.5)* are important for processes because "what happens if a common process has to be done in a divergent way?" or "We would like to see consequences of different process paths". Internal *actors (1.3)* "like users" as well as "employees and their qualifications" need to be assigned to a model as well as *communication interfaces (1.10)* because various experts argued, "it is a huge challenge to demonstrate, which steps have to be carried out by the customer". "The customer has to understand that certain tasks require collaboration." Accordingly, a model should allow for highlighting crucial tasks that need input from another actor such as customers. Especially for co-creation scenarios, this is important.

Table 2. Model representation

Requirement (#)	*Description*
1.1: Products and service objects	Representation of (physical) products and service objects, to show more detailed models such as CAD-diagrams or object lists.
1.2: Machine/service-object information	Integration of information (e.g., availability or downtimes) about resources such as production machines.
1.3: Actors	Representation of actors that are involved in a process or task as well as qualifications that are required to carry out a certain task.
1.4: Information flows	Representation of information flows to show which actor needs which information on what time.
1.5: Decisions	Highlighting decisions (and options), because they play an essential role in coordinating a collaboration
1.6: Data-objects	Representation of data-objects, because of the increasing relevance of information, technology and supporting software systems.
1.7: Legal aspects	Consideration of legal aspects such as employment laws, which especially becomes important in global cooperation's.
1.8: Standards/Norms	Visualization of steps and elements in a process model that base on a certain standard, norm or best practice.
1.9: Activity types	Representing of different activity types to allow identifying digital and analogue ones.
1.10: Communication interfaces	Highlighting crucial tasks that need input from other actors like customers. Especially for co-creation scenarios, this is important.
1.11: Safety instructions	Visualization of safety instructions related to the entire processes and single tasks to provide guidance for the employees.
1.12: Time aspects	Representation of time aspects to show which activities depend on prior processes and deadlines.

Navigation and linking (2). In order to reduce the complexity of creating a PSS (Table 3) as well as providing guidance in constructing a PSS, the *core phases (2.1*) of a collaborative project should be considered in a process model, for example, to position processes or single tasks to a certain phase. Experts stated that "it must be clear, which tasks belong to the development phase of a service." Regarding PSS, standardized phases (e.g., conception, implementation and dissolution) can be used such as the DIN PAS 1094. This also refers to the need of following norms (1.8). Correspondingly, a language should allow for representing phases.

Moreover, the structure should contribute to the comprehensibility, which can be supported by aspects such as *hierarchies (2.2)* to allow for integrating different levels of abstraction or *call of further workflows (2.3)* to refer to further processes and workflows (e.g., by implementing a service-oriented architecture). Experts stated that "it has to be visualized, which use cases and workflows are triggered by a specific process step." In the era of digitalization, internet of things (IoT), smart X and industry 4.0, further types of models become increasingly important for PSSs. The experts argued that particularly software-oriented diagrams (e.g., for components, code and interfaces) are crucial, for example to "represent which steps a customer has to carry out in a software" or "which components are applied to support specific tasks". A language should allow for *embedding different types of diagrams (2.4)*.

Table 3. Model navigation and linking

Requirement (#)	*Description and motivation through expert interviews*
2.1: PSS phases	Representing core phases of PSS construction (e.g., DIN PAS 1094) and allowing for assign objects and tasks to them.
2.2: Hierarchies	Creating hierarchies of processes and provide relationships between different levels of abstraction.
2.3: Further workflows	Presenting references to further processes/workflows (e.g., service-oriented architecture SOA).
2.4: Further diagrams	Embedding of different types of diagrams such as from software engineering (e.g., UML class diagram).

Additional information (3). This category (Table 4) addresses the integration of different quantitative aspects into a process model. As stated before, the integration of data gains increasing relevance for organizations. Consequently, *key performance indicator (KPI) (3.1)* need to be integrated into process models to provide information about the workload of employees, capacity of machines or tools, use of material etc. Many experts argued that indicators like time, costs, capacities of resources, quality etc. are required because they are important for coordinating the actors involved (e.g., to spread support tasks across different locations equality). In addition, data regarding a concrete processes should be *tracked (3.2)* and provided in a model. Various experts stated that information needs to be logged, for example, to "optimize processes that are too slow" or "to follow specified criteria". For the coordination of processes while collaborating with different organizations and locations, *collaboration data (3.3)* such as contact information are needed. Coordinating such networks requires "additional information of the actors involved".

Table 4. Additional model information

Requirement (#)	*Description and motivation through expert interviews*
3.1: KPI	Integration of KPIs related to products and service-objects to provide relevant information (e.g., workload, availability).
3.2: Tracking process data	Continuous tracking of process data to support versioning of process models and their performances.
3.3: Collaboration data	Consideration of collaboration data such as contact information to support the coordination between organizations and locations.

Design and handling (4). This category (Table 5) consists of aspects for the design, handling and construction of a process model and its language. Various experts stated that it is an fundamental issue to "allow the use of existing models on further devices like Smart Glasses" to better integrate them into daily practices of an organization as well as enable creating further formats (e.g., videos and games) based on a model. This requires configuration parameters, for example, related to the responsive design of a process model and the transformability to other formats and *further devices (4.1)*.

Moreover, modelling languages should be *adoptable (4.2)* to fulfil individual needs. For example, experts would like to reduce elements provided by a language to a 'core set' of constructs that are sufficient for a specific situation. Generally, a modelling language should have *limited complexity (4.4)*, for example, regarding the number of elements. Experts claimed for "simplicity", "transparency", and "slimness" of a model (see 4.2). In addition, although different interviewees stated that they "do not focus on rules", numerous of them followed standardized steps related to the construction of a model—mostly by following rules that are provided by a tool. Thus, a modelling language should allow for *standardized construction (4.3)*.

In contrast to the documentation and easy interpretation of a model, experts require the instantiation of a process model. Accordingly, a modelling language should allow for *being executable (4.6)*, for example, in a workflow management system. Furthermore, experts claim for "opportunities that allow for commenting certain model constructs", for example, to point out potential for improvement of a process. Hence, *remarks (4.5)* and comments need to be integrated.

Table 5. Model design and handling

Requirement (#)	*Description and motivation through expert interviews*
4.1: Further devices	Usage and transformability of a model to further devices such as Smart Glasses in order to better integrate them into daily processes.
4.2: Adoption of model constructs	Adaptation of model constructs to fulfil individual needs like reducing an entire set of modelling elements to a 'core set'.
4.3: Construction procedure	Standardized steps and rules for constructing of a certain model in order to provide guidance for the modeller and unity of a model.
4.4: Limited complexity	Limited complexity regarding the application of a modelling language, for example, regarding the number of elements (see 4.2).
4.5: Remarks	Annotation of remarks and comments to a process model in order to visualize, for example, potential or needs for improvement.
4.6: Execution	Instantiation and execution of a process model, for example, to enable the application in 'real scenarios'.

Transformation (5). Finally, the fifth category (Table 6) deals with requirements concerning the transformation of a process model and ways for further use. First, there is a need for *automatic generation (5.1)* of artefacts, for example, "software code should be transformable into models", "a diagram into a SQL-script", or "a model needs to be transformable into a database". Typically, this is discussed with terms such as model driven architecture” (e.g., [18]). Therefore, *exchange formats* (5.2) of process models are important to allow for exchanging certain data (e.g., [19]).

Table 6. Model transformation

Requirement (#)	*Description and motivation through expert interviews*
5.1: Automatic generate artefacts	Support the transformation and automatic generation of artefacts such as code to model or model to code.
5.2: Exchange models	Support the specification of export and exchange formats in order to allow for transforming data between process models.

4.2 Evaluation

Following the identification of requirements, selected business process modelling notations are assessed regarding their applicability for process modelling in the context of PSS development. The selection of modelling notations adheres to previous work by [20, 21], and thus, encompasses the most prominent and wide-spread notations for process modelling. The notations under consideration are depicted in Table 7. Each notation is evaluated against the defined requirements presented in Section 4.1. The evaluation has been conducted by a team of three researchers with high expertise in the field of business process modelling. Following a two-step procedure, each participant evaluated the notations distinctively first. Afterwards, a workshop took place, in which the initial results have been aligned and finalized. The evaluation is based on available language specifications published by the respective standard development organization. In the EPC case, we have referred to the specification efforts conducted by [22, www.epc-standard.org] and previous consolidation work by for example [23]. The requirements presented in Section 4 are assessed regarding suitable modelling constructs offered by each respective notation that are capable to sufficiently represent each requirement within a model. Accordingly, workaround solutions such as text annotations to overcome specific requirements have not been considered. Fulfilled requirements, marked by an “X”, reflect that a particular notation offers modelling constructs that fully support the requirement. A partial fulfilment is marked by an “(X)” and requirements not fulfilled are marked by an “-”.

Reviewing the results in Table 7, it becomes apparent that the three dominant notations for business process modelling considered within this assessment, BPMN, EPC and UML Activity Diagram, yield the largest degree of coverage of the requirements extracted from the expert interviews. Especially in terms of process representation and transformation, the aforementioned notations provide a respective set of modelling constructs that suit the needs from practice. Exemplarily, most notations are able to depict a basic information flow as well as provide modelling constructs for resources (e.g., data object and actors). However, the analysis reveals that there are severe shortcomings across all evaluated notations that specifically concern their application within a PSS context. This becomes visible via PSS-related requirements such as the integration of domain-specific procedure models for PSS development within a modelling notation, or the necessity to express legal issues in a modelled business process, which is of utmost importance for enterprises in the service domain. Even more striking are the shortcomings of traditional notations in the context of data-centric requirements, which have been frequently referred to over the course of the expert interviews, underlining a growing importance of data and data processing across all industry sectors. Here, the assessed notations are not able to handle various forms of data modelling and data integration, exemplarily by incorporating machine and sensor data into modelled processes. Furthermore, requirements summarized in Table 4 indicate that no notation fulfils practical demands concerning additional model information such as quantitative data, which facilitates model analysis or log data that supports model versioning.

Table 7. Validation of requirements
Caption: “X” requirement fulfilled, “(X)” partly fulfilled, “-“ not fulfilled

#	Requirement title	BPMN 2.0	UML Activity	EPC	PetriNet	IDEF3	Ansi Flowchart
1.1	Products and service objects	(X)	(X)	(X)	-	X	-
1.2	Machine/service-object information	-	-	-	-	-	-
1.3	Actors	X	X	X	-	-	-
1.4	Information flows	X	X	X	X	X	X
1.5	Decisions	X	X	X	X	X	X
1.6	Data-objects	X	X	X	-	-	X
1.7	Legal aspects	-	-	-	-	-	X
1.8	Standards/Norms	(X)	-	-	-	-	-
1.9	Activity types	X	X	-	-	-	-
1.10	Communication interfaces	X	X	-	-	-	-
1.11	Safety instructions	-	-	-	-	-	-
1.12	Time aspects	X	(X)	-	-	-	(X)
2.1	PSS phases	-	-	-	-	-	-
2.2	Hierarchies	X	X	X	-	-	-
2.3	Further workflows	(X)	(X)	(X)	-	-	(X)
2.4	Further diagrams	-	(X)	(X)	-	-	-
3.1	Key performance indicator (KPI)	-	-	-	-	-	-
3.2	Tracking process data	-	-	-	-	-	-
3.3	Collaboration data	-	-	-	-	-	-
4.1	Further devices	-	-	-	-	-	-
4.2	Adoption of model constructs	-	-	-	-	-	-
4.3	Construction procedure	X	X	(X)	-	-	-
4.4	Limited complexity	(X)		X	-	-	X
4.5	Remarks	X	X	X	-	-	X
4.6	Execution	X	(X)	(X)	X	-	-
5.1	Automatic generate artefacts	X	X	(X)	X	-	-
5.2	Exchange models	X	X	X	X	-	-

5 Discussion and Further Research

Based on the results derived, we—in the following section—discuss different clusters and challenges that should be addressed by further research.

Being compliant with legal aspects and standards. Various aspects from the experts deal with following standards, norms and best practice, considering legal aspects as well as representing them in process models. This cluster can be addressed by research streams that (a) investigate the applicability of methods from IS research such as process modelling languages to represent legal and contract issues as well as (b) investigate extensions for such languages like BPMN (e.g., [24]). However, currently no well-accepted set of constructs for supporting law and legal aspects exists, thus, further research can deal with (1) method engineering to support the representation of relevant constructs, (2) evaluation to analyse if the suggested extensions are applicable and (3) standardization of the evaluated constructs to allow for developing new software tools that support the construction of them.

Handling the increased relevance of data. Many experts highlighted the utmost importance of data, data structure and data-objects as well as applied software components. These criteria are particularly related to current topics such digitalization of processes, application of new devices or transparency of activities. As a result, data and evolvable information of entire processes, single objects and actors needs to be provided. Due to the importance, different experts ask for the integration of further detail diagrams that allow for presenting these aspects.

Using new devices. One major aspect in the transformation of models is the consideration of new devices for representing process models. Common, real-world scenarios such as maintaining production machines or using big tools to carry out a task hinder the use of process models in a traditional manner (e.g., a digital version on a computer or a printed version). Experts argued that the use of process models on new devices such as Smart Glasses or Smart Watches need to be improved [25]. In doing so, employees are enable to watch models while continuing work. Hence, further research on modelling languages and model formats needs to deal with design issues such as being responsive and being readable on different types of devices.

Modelling language vs. tool-support. Although our interviews aim at identifying requirements for process modelling languages, the implementation of some criteria depends on supporting software-tools. For example, KPIs, the tracking of process data, and the execution of a process model as well as the connection to further diagrams can be improved by software in particular. However, modelling languages can contribute to this by providing specific constructs that allow for visualizing these aspects (e.g., an icon that represents a stored detail diagram). Thus, especially the requirements, which are not fulfilled (c.f. section 4.2), might be supported by tools. In addition, tools enable the transformation of process models to further devices, which can barely enabled by a modelling language itself.

Model complexity vs. execution. Some of the requirements derived are in contrast. Especially criteria related to the reduction of model complexity and criteria related to the transformation and execution of a model. On one hand, experts argued for simplicity and slimness by reducing the set of modelling constructs provided or creating own rules for the application of a modelling language. On the other hand, experts require, such as, an automatic generation of models to software code or automatic transformation into an executable process model (e.g., in form of a workflow). Correspondingly, a modelling language needs to address the challenge between both streams, for example, by providing different level of abstractions or by providing features that operate in the background of a model (e.g., semi-formal, graphical representation layer and formal transformation layer).

6 Conclusion

Due to the importance of business processes in the development and implementation of PSS, the goal of our study was to identify requirements and challenges related to modelling PSS from a process-oriented perspective in an empirical manner. Our contribution is a set of 27 requirements that are based on 18 expert interviews from 14 companies and over eight hours of interview transcripts. Although we derived helpful insights for modelling PSS, our study is not free of limitations. First, our investigation is limited to industry partners and interviewees selected. The participants are from different fields to ensure a broad overview on the integration of service, products, software etc. Nevertheless, further interviewees (e.g., with domain and modelling experienced experts) may provide new aspects. We used an inductive approach that conceptualizes data from empiricism – as a next step, we plan to compare our set of requirements to the literature (deductive approach) to evaluate and revise our initial findings. Second, although our coding follows established guidelines [15, 17], the process is based on own interpretations and decisions. However, to contribute to the reliability, three different researchers coded independently and consolidated the results in a follow-up workshop. Third, in order to initial validate our results, we compared them to common modelling languages. Additionally, the results can be evaluated, for example, with experts to prove the coding and the generic usefulness (e.g., in a Delphi Study or in focus groups). As some of the 'general purpose languages' (e.g., BPMN) already address many aspects, future research can deal with the investigation of domain-specific variants for PSS.

Overall, our work gives orientation as to which requirements and challenges need to be addressed by modelling languages and can be used to redesign existing or develop new languages that support PSS modelling from a process-perspective.

7 Acknowledgments

This research is partly funded by the European Regional Development Fund and the State of Lower Saxony (NBank) in the scope of the research project SmartHybrid—Service Engineering (ID: 6-85003236) and Process Engineering (ID: ZW 6-85003451). We would like to thank them for their support.

References

1. Thomas, O., Nüttgens, M., Fellmann, M.: Smart Service Engineering: Konzepte und Anwendungsszenarien für die digitale Transformation. Springer Fachmedien Wiesbaden, Wiesbaden (2017)
2. Frank, U., Strecker, S., Fettke, P., vom Brocke, J., Becker, J., Sinz, E.: Das Forschungsfeld „Modellierung betrieblicher Informationssysteme“. Wirtschaftsinformatik. 56, 49–54 (2014)
3. Vasantha, G., Roy, R., Lelah, A., Brissaud, D.: A review of product–service systems design methodologies. J. Eng. Des. 23, 635–659 (2012)
4. Becker, J., Beverungen, D.F., Knackstedt, R.: The challenge of conceptual modeling for

product–service systems: status-quo and perspectives for reference models and modeling languages. Inf. Syst. E-bus. Manag. 8, 33–66 (2010)
5. Becker, J., Bernhold, T., Knackstedt, R., Matzner, M.: Planung koordinierter Wertschöpfungspartnerschaften. Gabler Verlag (2017)
6. Morelli, N.: Developing new product service systems (PSS): methodologies and operational tools. J. Clean. Prod. 14, 1495–1501 (2006)
7. Harmon, P., Wolf, C.: The State of Business Process Management. A BPTtrends Rep. 1–52 (2016)
8. Mont, O.: Clarifying the concept of product – service system. J. Clean. Prod. 10, 237–245 (2002)
9. Thomas, O., Walter, P., Loos, P.: Design and usage of an engineering methodology for product-service systems. J. Des. Res. 7, 177 (2008)
10. Aurich, J.C., Schweitzer, E., Mannweiler, C.: Integrated design of industrial Product-Service Systems. Manuf. Syst. Technol. New Front. 543–546 (2008).
11. Goedkoop, M.J., Van Halen, C.J.G., Te Riele, H.R.M., Rommens, P.J.M.: Product Service systems , Ecological and Economic Basics. (1999)
12. Cavalieri, S., Pezzotta, G.: Product–Service Systems Engineering: State of the art and research challenges. Comput. Ind. 63, 278–288 (2012)
13. Boucher, X., Medini, K., Fill, H.: Product-Service-System Modeling Method. In: Domain-Specific Conceptual Modeling. pp. 455–482. Springer Int. Pub. (2016)
14. Pezzotta, G., Pirola, F., Pinto, R., Akasaka, F., Shimomura, Y.: A Service Engineering framework to design and assess an integrated product-service. Mechatronics. 31, 169–179 (2015)
15. Meuser, M., Nagel, U.: Das Experteninterview - konzeptionelle Grundlagen und methodische Anlage. In: Pickel, S., Pickel, G., Lauth, H.-J., and Jahn, D. (eds.) Meth. der vgld. Politik- und Sozialwissenschaft. pp. 465–479. VS, Wiesbaden (2009)
16. Porst, R.: Frageformulierung. In: Baur, N. and Blasius, J. (eds.) Handbuch Methoden der empirischen Sozialforschung. pp. 687–699. Springer Fachmedien, Wiesbaden (2014)
17. Mayring, P.: Qualitative Inhaltsanalyse. In: Handbuch Qualitative Forschung in der Psychologie. pp. 601–613. VS Verlag für Sozialwissenschaften, Wiesbaden (2010)
18. Czarnecki, K., Helsen, S.: Feature-based survey of model transformation approaches. IBM Syst. J. 45, 621–645 (2006)
19. Riehle, D.M., Jannaber, S., Karhof, A., Delfmann, P., Thomas, O., Becker, J.: Towards an EPC Standardization – A Literature Review on Exchange Formats for EPC Models. MKWI 2016 Band III. 4, 1167–1178 (2016)
20. Harmon, P., Wolf, C.: Business Process Modeling Survey. A BPTrends Rep. 36 (2011)
21. Recker, J.C., Rosemann, M., Indulska, M., Green, P.: Business process modeling: a comparative analysis. J. Assoc. Inf. Syst. 10, 333–363 (2009)
22. Jannaber, S., Karhof, A., Riehle, D.M., Thomas, O., Delfmann, P.: Invigorating Event-driven Process Chains – Towards an integrated meta model for EPC standardization. AQEMO (2016)
23. Fellmann, M., Bittmann, S., Karhof, A., Stolze, C., Thomas, O.: Do We Need a Standard for EPC Modelling? The State of Syntactic, Semantic and Pragmatic Quality. EMISA (2013)
24. Schoormann, T., Knackstedt, R., Haapio, H.: Modeling and Visualization in Law: Past, Present and Future. In: Internationales Rechtsinformatik Symposium IRIS (2017)
25. Metzger, D., Niemöller, C., Berkemeier, L., Brenning, L., Thomas, O.: Vom Techniker zum Modellierer – Konzeption und Entwicklung eines Smart Glasses Systems zur Laufzeitmodellierung von Dienstleistungsprozessen. In: Smart Service Engineering. pp. 193–213. Springer Fachmedien, Wiesbaden (2017)