applied sciences

MDPI

Article

SemDaServ: A Systematic Approach for Semantic Data Specification of AI-Based Smart Service Systems

Maurice Preidel [1,*] and Rainer Stark [1,2]

[1] Department of Industrial Information Technology, Institute for Machine Tools and Factory Management, Technische Universität Berlin, Pascalstr. 8-9, 10587 Berlin, Germany; rainer.stark@ipk.fraunhofer.de
[2] Division of Virtual Product Creation, Fraunhofer Institute for Production Systems and Design Technology IPK, Pascalstr. 8-9, 10587 Berlin, Germany
* Correspondence: maurice.preidel@tu-berlin.de

Abstract: To develop smart services to successfully operate as a component of smart service systems (SSS), they need qualitatively and quantitatively sufficient data. This is especially true when using statistical methods from the field of artificial intelligence (AI): training data quality directly determines the quality of resulting AI models. However, AI model quality is only known when AI training can take place. Additionally, the creation of not yet available data sources (e.g., sensors) takes time. Therefore, systematic specification is needed alongside SSS development. Today, there is a lack of systematic support for specifying data relevant to smart services. This gap can be closed by realizing the systematic approach SemDaServ presented in this article. The research approach is based on Blessing's Design Research Methodology (literature study, derivation of key factors, success criteria, solution functions, solution development, applicability evaluation). SemDaServ provides a three-step process and five accompanying artifacts. Using domain knowledge for data specification is critical and creates additional challenges. Therefore, the SemDaServ approach systematically captures and semantically formalizes domain knowledge in SysML-based models for information and data. The applicability evaluation in expert interviews and expert workshops has confirmed the suitability of SemDaServ for data specification in the context of SSS development. SemDaServ thus offers a systematic approach to specify the data requirements of smart services early on to aid development to continuous integration and continuous delivery scenarios.

Keywords: smart services; data specification; domain knowledge; information needs; data needs; knowledge needs; data quality; smart service systems engineering

Citation: Preidel, M.; Stark, R. SemDaServ: A Systematic Approach for Semantic Data Specification of AI-Based Smart Service Systems. *Appl. Sci.* **2021**, *11*, 5148. https://doi.org/10.3390/app11115148

Academic Editors: Marlene Amorim, Yuval Cohen and João Reis

Received: 1 May 2021
Accepted: 27 May 2021
Published: 1 June 2021

1. Introduction

For smart services to be developed and successfully operated as a component of smart service system (SSS), a qualitatively and quantitatively sufficient amount of data is required. This is especially true when data-driven software components are implemented by using statistical methods from the field of artificial intelligence: the quality of the training data directly determines the quality of the trained AI model. However, the quality of the training data can only be evaluated at a late stage using existing methods in the context of smart service development, since the training success is only known when the artificial intelligence (AI) training takes place. Furthermore, automated machine learning (AutoML) approaches are rising [1–9]. AutoML is aiming for the automation of machine learning (ML) model development. This means that SSS development projects using AutoML will need to rely even more on sufficient data carrying the relevant information because the ML model created with AutoML relies completely on statistics on the raw data and ignores causality only available from domain experts. Therefore, the availability of sufficient data and especially data engineering [10,11] will stay a major bottle-neck of AI applications. If the information in the training data or the data to be analyzed in operations is missing, a statistical AI model will be of poor quality. If irrelevant data is fed

into AutoML pipelines, spurious correlations could result in unpredictable and dangerous product behavior occurring during the operation of the SSS.

The purpose of the presented Semantic Data Specification for AI-Based Smart Services (SemDaServ) approach is to specify the data needs of an AI-based smart service as part of a SSS as early as possible using expert domain knowledge. This allows the identification of insufficient data quality and quantity without the need to train AI models. This is often necessary for practice if the data for a smart service in development is not yet available and therefore needs to be acquired. To acquire relevant data while omitting irrelevant data (and therefore saving time and cost) data specification describing the data needs of smart service's AI and other software components is required. Furthermore, it is crucial to identify all areas of smart service's data needs; an unidentified data need may require changing the physical product (e.g., integrating new sensors into the main bearings of an aircraft turbine)—which could result in high costs and delays. To make the domain knowledge of experts available to smart service and AI engineers, the domain knowledge is formalized in a way that preserves the semantic meaning of the data specified.

Take, for example, the service-oriented Power-by-the-Hour business model for aerospace engines, where the customers pay for hours using the engine while not owning the engine. The engine provider (SSS provider) needs to make sure that the customers always got an operational engine at the wing. If a critical component like a bearing is going to fail soon, this must be immediately known to the engine provider to trigger actions for maintenance or exchange of the engine. Therefore, the remaining life prediction for the bearing is a critical component of the SSS offering. Within this article, this kind of remaining life prediction is understood as a smart service being part of a SSS.

The structure of the article is presented in Figure 1: (A) Based on the state of the research field and resulting research gaps described in Section 1, research questions, working hypotheses, and research methods are described in Section 2. (B) Additionally, the purpose of the SemDaServ approach described in Section 1 and (C) the research questions described in Section 2 are setting the scope for Section 3. Section 3 is the main section of this article containing a detailed description of the SemDaServ approach in four sub-sections: (D) Section 3.1 describes the business roles required to conduct the guided process for semantic data specification described in Section 3.2. (E) The artifacts used in this process are described in Section 3.3. (F) The IT systems and tools described in Section 3.4 are helpful to support the realization of the process described in Section 3.2. (G) Additionally, IT Systems and tools described in Section 3.4 can be used for the creation and storage of the artifacts described in Section 3.3. The coloring of the boxes in Figure 1 that represent Sections 3.2–3.4 is used throughout the article: Blue represents processes, yellow represents artifacts, and green represents IT systems and tools. (H) Section 4 presents the evaluation of the SemDaServ approach. (I) The study design of the evaluation (Section 4) is described in Section 2. (I, J) The working hypotheses described in Section 2 are discussed in Section 5 while taking the results of the evaluation (Section 4) into account. (K) Section 5.1 describes the limitations of the presented SemDaServ approach, especially regarding the evaluation scope. These limitations will be addressed in future research, described in Section 5.2. (L) Section 6 summarizes the article. Appendix A presents details linked to the descriptive study I (cf. Section 2). As the focus of this article is on the SemDaServ approach, the descriptive study I—which mainly focuses on the state of the art and research gaps—is necessary to understand the research approach but is not part of the SemDaServ approach.

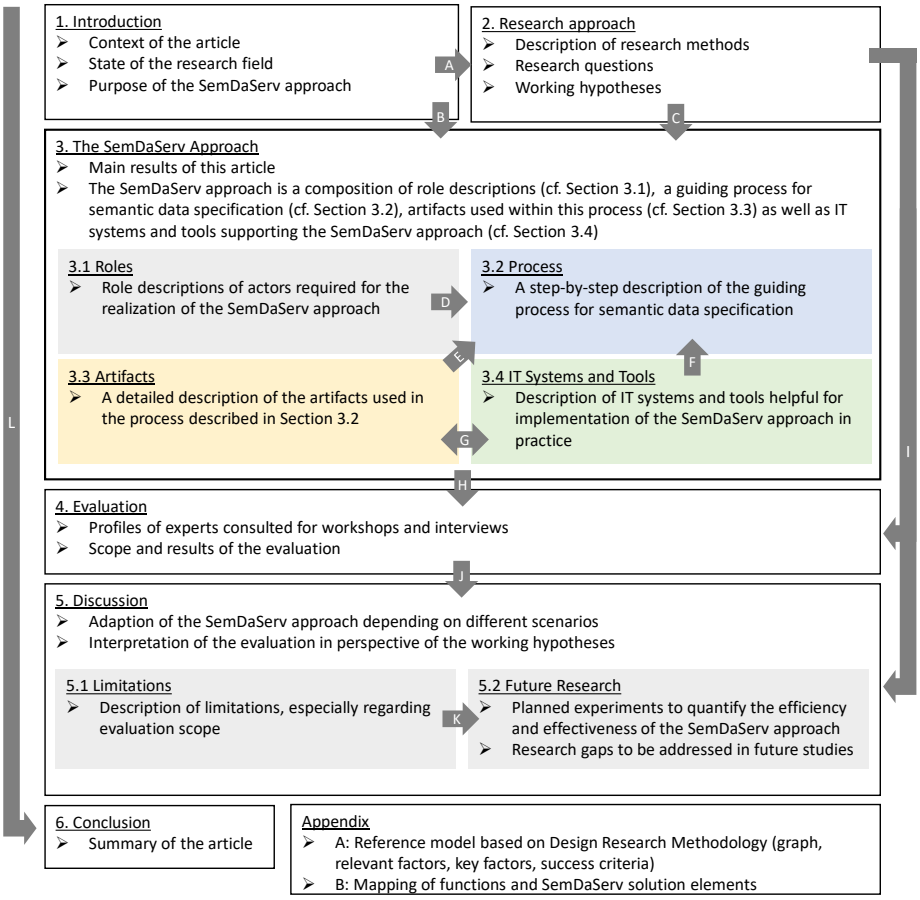

Figure 1. The structure of the article.

Reviewing the state of the art, new service engineering methods are required to systematically develop AI components of smart services. Ref. [12] While there are many methods, tools, and processes for SSS development (cf. [13] giving an overview), there is a lack of approaches that systematically make use of expert domain knowledge—which is highly relevant in industrial AI applications [14]—to specify the data needed for AI components of these services. This is especially the case if the data needed for AI training is not available at the stage of development. Anke [15] found that determining the required data and its quality is an important challenge. Rosa et al. [16] identify the major problem in the high probability of system designers ignoring the creation of relevant information in early product service system (PSS) design phases. They linked this problem to four main challenges: "[a] lack of completeness and structure on service-related information due to the intangibility and heterogeneity of services; [a] lack of integration among the PSS elements due to not considering its information requirements; dealing with a significant quantity and variety of knowledge; and ensuring completeness without limiting the flexibility of designers to select the methods and artifacts they intend to use" [16]. The SemDaServ approach addresses all four challenges within the scope of smart service data specification.

Data science perspectives have also contributed, well established, generic approaches to developing AI models (e.g., CRISP-DM [17], KDD [18], or SEMMA [19]). Azevedo and Santos [20] concluded that SEMMA and CRISP-DM were both implementations of the KDD process, though CRISP-DM is more complete than SEMMA. While these data-driven approaches perform well in situations where sufficient data is already available, their aim and scope limit their applicability in cases where relevant data needs to first be identified and acquired. This is especially true for SSS in development, where the data needs also define requirements for the physical components (e.g., the quality of sensors). As Wang et al. [21] pointed out, expert domain knowledge is crucial for clarifying data needs. In addition, expert knowledge is crucial in feature engineering (e.g., [22]).

Marx et al. [12] examined data science and smart service systems engineering (SSSE) together in a systematic literature review, finding that there was a lack of smart service engineering methods that dealt with the data perspective: Of 36 methods, only six mainly considered the data perspective.

2. Research Approach

The results were obtained using the Design Research Methodology described in [23] and presented in Figure 2.

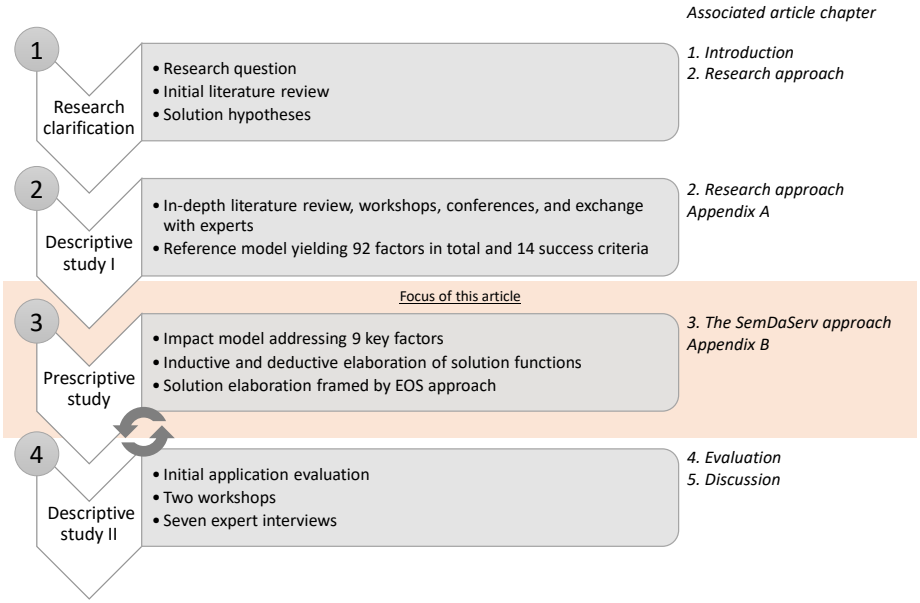

Figure 2. The research approach and focus of this article.

Research clarification (1): First, we formulated the following research questions (RQ):

RQ1 How can the probability of AI development with quality of results being in line with smart service requirements be increased?
RQ2 How can a clear understanding of relevant data for the development and operation of a smart service be systematically generated?
RQ3 How can domain knowledge relevant to AI training be formalized?

In light of these research questions, we initially reviewed the literature on the topics of SSSE, data science, data engineering, and semantic data modeling, as well as the use of domain knowledge in AI development. We used the Web of Science (webofknowledge.com accessed on 6 April 2021) , Google Scholar (scholar.google.de accessed on 7 April 2021), and Google Search (google.com accessed on 8 April 2021) to identify sources. After the initial literature review, we formulated the following working hypotheses (WH):

WH1 The probability of AI development with quality of results in line with smart service requirements can be increased by a domain knowledge-driven data specification approach.
WH2 A clear understanding of relevant data for the development and operation of a smart service can be systematically generated by using domain knowledge to clarify information needs and derive data needs from information needs.
WH3 Domain knowledge relevant for AI training can be formalized using the Systems Modeling Language (SysML).
WH4 Domain knowledge relevant for AI training can be formalized using a guided process.

Descriptive study (2): To get a deeper understanding of the current state-of-the-art as well as existing challenges in practice, we conducted an in-depth literature review, using the same database, search engines, and research areas as in Step 1. The insights gained were modeled in a graph-based reference model according to [23] and presented in Figure A1. The following sources from the literature yielded factors and links for the reference model: Refs. [10,14,15,21,24–32]. The first author of the research team also attended relevant conferences and workshops, resulting in an exchange with experts over a period of more than five years. This yielded additional factors and links, which were added to the reference model. In the last step, the reference model was analyzed regarding missing links and nodes from a logical point of view. Overall, 92 factors were identified within the descriptive study and 43 factors were declared outside the scope of the study (cf. Table A1), too far away from the core of the research questions and working hypotheses. This left 49 factors (cf. Table A2). From these factors, we identified 14 factors as success criteria (The research goal is to improve these factors as they are the most relevant factors to define success for the contribution to practice. ([23] p. 26) (cf. Table A3)).

Prescriptive study (3): The reference model was used to formulate the desired situation of positively impacting the success criteria. For this purpose, we first identified nine key factors (the most promising factors for improving on the existing situation ([23] p. 21), cf. Table A4). Then, we used an inductive approach to define solution functions in the context of the key factors. To do so, we abstracted sub-functions to more general main functions. After that, we applied a deductive approach to close gaps in the resulting functions architecture, detailing the main functions. The resulting functions architecture, as well as a mapping connecting it to the solution elements of the SemDaServ approach, are presented in Table A5. After designing the functions architecture, we systematically designed SemDaServ by addressing these functions according to the Engineering Operating System (EOS) [33] shown in Figure 3.

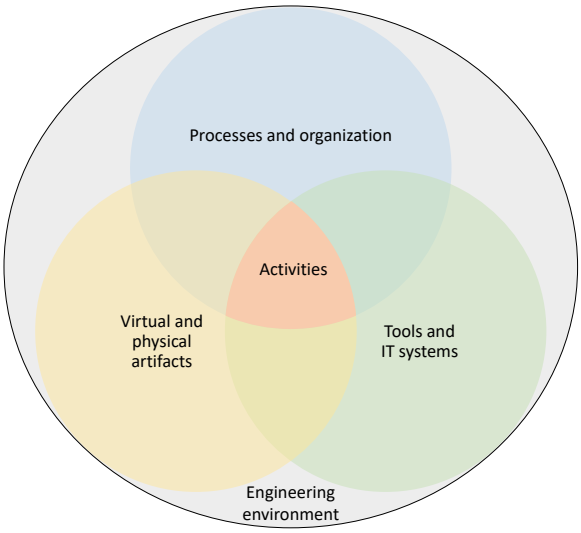

Figure 3. Engineering operating system, adapted from ([33] p. 319) with permission from IEEE.

Descriptive study II (4): This part of the research approach focuses on validation SemDaServ regarding logical correctness, applicability, and usefulness for real-world applications. The applicability of the SemDaServ approach was validated in two workshops lasting about 90 min each. Within the workshops, the SemDaServ approach was presented by the first author of this article for open discussion. The research questions RQ1, RQ2, and RQ3 guided the discussion. The second workshop was focused on the industrial point of view. In addition, the first author interviewed seven experts from academia and industry. The interviews were related to the specialization of the interviewed experts and therefore focused on specific aspects of the SemDaServ approach. The workshops and expert interviews took place over a period of five months. During this time, the SemDaServ

approach was continuously improved based on the results of the workshops and expert interviews. Therefore, there were iterations between prescriptive study and descriptive study II. The profiles of the workshop participants and the interviewed experts as well as the outcomes of descriptive study II are presented in Section 4.

3. The SemDaServ Approach

The activity of semantic data specification of smart services is at the center of the proposed SemDaServ approach presented in Figure 4. The guidance to successfully conduct this activity is primarily provided by the SemDaServ process dimension: The three-step data specification process (Clarify domain knowledge needs, Clarify information needs, Specify data needs) systematically guides the actors involved through the individual steps of data specification and ensures that the knowledge pyramid is systematically traversed starting from domain knowledge (definition: information plus context, experience and cross-linking [34]), to information (definition: data plus meaning [34]) needs, and then to data (definition: symbols plus syntax [34]) needs (cf. [21]). As the SemDaServ approach is generally based on traversing the knowledge pyramid, the SemDaServ is applicable from new development of a smart service to continuous integration (CI) and continuous delivery (CD) scenarios. Realizing SemDaServ is expected to result in (1) a clear understanding of relevant data for the development and operation of the smart service, (2) formalized domain knowledge relevant for AI training, and (3) increased probability of AI development with quality of results in line with smart service requirements.

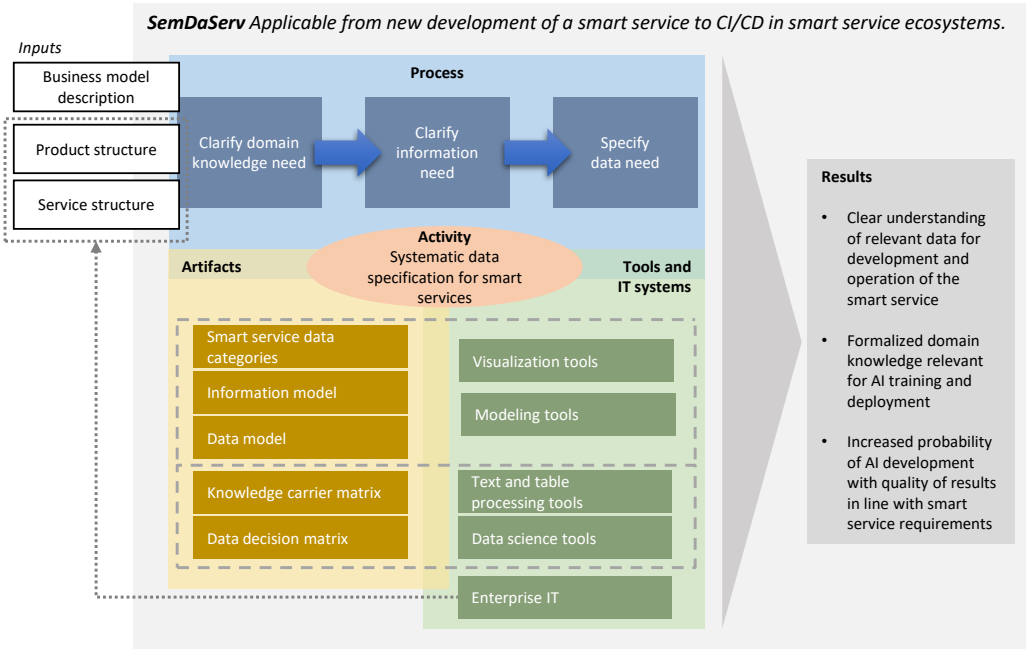

Figure 4. Overview of the SemDaServ approach.

The Clarify information needs process step is the most important step here, since the information level is where knowledge and data come together. Without the information level, it is difficult to infer concrete data needs (e.g., vibration sensor signal sampled at 200 Hz at the uppermost point of the outer ring) from general domain knowledge (e.g., damage in the bearing causes vibrations in the system). Here, the information layer serves to formalize the relevant domain knowledge in the context of the system (e.g., increasing vibration at the outer ring implies increasing wear of the ball bearings).

Various artifacts are used and created as part of the data specification process. On the one hand, these artifacts serve to facilitate individual process steps by specifying the result formats in a structured manner. On the other hand, the artifacts also document the process results. The artifacts can be divided into two groups: (1) artifacts for describing data and information and (2) artifacts for supporting neural points during process execution. Group 1 artifacts use SysML for three following reasons. Frist, computer scientists and data scientists benefit from semantic data specification due to the formalized domain knowledge, and SysML's proximity to Unified Modeling Language (UML), which is commonly used in SSSE [15], is useful here. Next, SysML is a powerful modeling language that can represent arbitrary technical systems and software components. This makes it possible to describe the connection between the smart service and the physical holistically. Third, the increasing diffusion of model-based systems engineering (MBSE) makes it likely that the diffusion of SysML will further grow as well, and will be increasingly supported by IT tools. Thus, in an MBSE environment, semantic data specification for smart services can be carried out without media discontinuity in the context of system specifications, and can be seamlessly integrated into the model-based engineering of the future.

Tools and IT systems increase the efficiency and quality of artifact creation. For example, visualization and modeling tools support the creation of SysML models. Text and spreadsheet tools enable the digital creation of documentation, such as tables and checklists. Theoretically, the creation of SysML models, checklists, and tables is also possible in a paper-based manner. However, using software tools for these activities is much more convenient, avoids errors, and increases process efficiency. Software tools from the field of data science (e.g., Python development environments) are another crucial element allowing insights from existing data to become part of the data specification process. This is necessary because domain experts may be unaware of correlations carrying causal links already present in the existing data. However, these previously unknown correlations may be of interest to the smart service but must be checked by domain experts, because these correlations may simply be spurious. The enterprise IT domain supports data specification as a data provider for important inputs (especially product and service structure) as well as Internet-of-Things (IoT) data and their context.

3.1. Roles

To design the responsibilities within the data specification process, we linked roles from (Hildebrand et al. [34] p. 240) (hereafter: "data-oriented roles") with roles of SSSE according to Anke et al. [35] (hereafter: "SSSE roles"). However, it is necessary to first analyze the intersection of the data-oriented and the SSSE roles. The analysis result and the associated role descriptions are shown in Table 1. It turns out that the primary SSSE roles largely overlap with the data-oriented roles. The data-oriented roles located in each row of Table 1 are mapped to the SSSE roles. This results in multiple assignments for the Service Operator role: The Service Operator acts simultaneously in the roles of Data Provider, Data Consumer, and Data Owner. This is caused by SSS data sources (e.g., sensors to monitor bearings) and sinks (e.g., ML model predicting the remaining life of a bearing) are equally present within the data specification process. Since the Service Provider is responsible for the technical operation of the SSS as a whole, the Data Provider and Data Consumer roles both fall to the Service Provider. Since the Service Provider is also responsible for service compliance in addition to the operational operation of the SSS, the Service Provider thus also has the role of Data Owner. The primary role of Digital Innovator is not assigned to a data-oriented role because the Digital Innovator is focused on idea generation and the business model. These aspects are upstream of the data specification process. Nevertheless, the Digital Innovator is an essential actor in the data specification process, whose role is particularly important in the first step of data specification (Clarify domain knowledge needs). The roles of Project Sponsor, System Integrator, and Service Provider are thus the main actors of the data specification process and are referred to as the Core Team below.

Table 1. Mapping of data-oriented roles to the SSSE roles that make up the Core Team for realizing the SemDaServ approach.

Data-Oriented Roles Described in ([34] p. 240)			**Core Team: Assigned SSSE Roles Described in [35]**	
Role	**Description**		**Role**	**Description**
Process Owner	Responsible for the overall process including process definition, documentation, improvement, and timelines.	$\Longrightarrow$	Project Sponsor	Responsible for SSS development project from initiation to completion including time and cost management.
Data Definition Owner	Responsible for data specification, including data quality, granularity, and format as well as storage media, if applicable. Usually shares the role of Data Consumer. Coordinates Data Consumers, should there be more than one.	$\Longrightarrow$	System Integrator	Responsible for development and implementation of technical system elements including system architecture, technical conceptualization, and integration with existing systems.
Data Consumer	Beneficiaries of the data.	$\Longrightarrow$	Service Operator	Responsible for the technical operation of the SSS, including software management, service availability, and compliance with existing policies.
Data Provider	Responsible for the timely acquisition and delivery of data as defined by the Data Definition Owner.			
Data Owner	Owner of the data, who is therefore responsible for its use, including data acquisition, and security as well as measurement ranges and methods.			

3.2. Process

The data specification process is described in the following on the first sublevel of the process using Business Process Modeling Language (BPMN). It is based on three process steps Clarify domain knowledge needs, Clarify information needs and Specify data needs (cf. Figure 4). These are based on the clarify needs section of the reference model described in [21].

3.2.1. Clarify Domain Knowledge Needs

The goal of the process step Clarify domain knowledge needs is the documented identification of the knowledge required for the data specification as well as the associated knowledge carriers from the domain of the SSS. This involves all roles of the Core Team as well as the Digital Innovator. To perform this process step, the following inputs must be available: The business model description (e.g., in the form of the Smart Service Canvas [36] or the framework for data-driven business models described in Exner et al. [37]), the product structure (derived from the product data management (PDM) system, see Section 3.4) and the service structure (e.g., created according to the MESSIAH [38] or PSS-layer method [39]). The results of this process should be an understanding of the smart service from within the application domain as well as the roles and the names of the knowledge carriers required for the data specification. The whole Clarify domain knowledge needs process step can be performed in a single kick-off meeting. Depending on the complexity of the smart service as well as the number of domain knowledge carriers in question, a short meeting (about one hour) is sufficient. If necessary, up to two full-day workshops are required, but this represents an extreme case. The sub-processes explained below are shown in Figure 5.

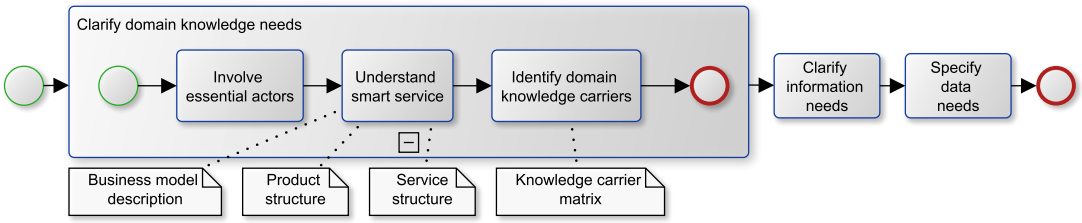

Figure 5. Process steps to clarify domain knowledge needs.

The goal of the Involve essential actors sub-process step is to onboard the Core Team to initially start the data specification process. Only the Core Team (cf. Table 1) is involved. This process step is completed when the Core Team has an understanding (tasks, areas of responsibility, own role in the data specification process) of how to execute the data specification process (as well as initiating contact among team members). For this purpose, the Project Sponsor explains the data specification process and general conditions (time, costs, quality). The Core Team discusses questions concerning the understanding of the process as well as the necessary process adjustments in light of given conditions, and they decide on the first process adjustments (e.g., the definition of the required level of detail of the results or the maximum number of domain experts to be involved).

The goal of the Understand smart service sub-process step is to give the Core Team as a whole a thorough understanding of the smart service at the outset so that they can competently guide the data specification process and identify what domain knowledge is required and who holds that in the next step. In addition to the Core Team, the Digital Innovator is also involved: This person carries knowledge regarding the business model and the innovative core of the smart service, which is fundamental to the holistic understanding of the smart service. The result of this sub-process step is therefore the Core Team's holistic understanding of the smart service. To achieve this, the Digital Innovator first explains the business model. Then the System Integrator guides the team through the service structure and explains the relevant parts of the product structure. In the process, questions of understanding are clarified so that at the end of this sub-process step all actors have an agreed level of knowledge on the smart service. Since the smart service is the focus (business model, product, and service structure) and the System Integrator, who has a close relationship to the knowledge domain of the SSS, is involved, an initial exchange of domain knowledge takes place at this point. This ensures that the Core Team has a basic understanding of the relevant domain knowledge. This is the basic prerequisite for starting the next process step.

The goal of the Identify domain knowledge carriers sub-process step is to determine the domain knowledge carriers to be involved downstream from the data specification process. At a minimum, the Core Team is involved. Optionally, other people can be involved as needed (e.g., ball bearing monitoring specialists if the smart service includes ball bearing failure prediction as an essential component). However, the group of people involved should be kept to a minimum in this step, because the data specification process is a framework allowing the integration of domain knowledge carriers as needed in any process step. At this point, however, the focus is on establishing a good starting point that takes into account the essential domain knowledge areas. The result of the process step is therefore to designate which knowledge carriers are to be involved in the data specification process. To do so, the Core Team fills in the knowledge carrier matrix described in Section 3.3.1, and, if necessary, adds additional knowledge areas. The Core Team should also keep the business model description and the product and service structure in mind: Valuable information on relevant domain knowledge areas (e.g., components or assemblies of the SSS concerned) can be found here.

3.2.2. Clarify Information Needs

The goal of the Clarify information needs process step is to use the domain knowledge of the previously identified knowledge carriers to describe what information needs the smart service will have. All sub-process steps in this section involve the Core Team and the identified knowledge carriers. The result is the description of the required information flow as an information model (including a description of the information quality) for the development and operation of the smart service under consideration. The entire process step Clarify information needs can be done synchronously (in the form of workshops) or asynchronously (by modeling using a shared SysML information model)(see Section 3.3.4). The sub-processes explained in the following are shown in Figure 6.

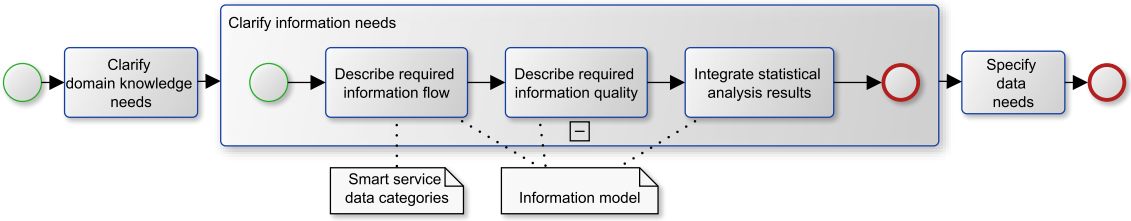

Figure 6. Process steps to clarify information needs.

The goal of the Describe required information flow sub-process step is to formalize the information flow required for the development and operation of the smart service. The result of the process step is the information flow of the information model described in Section 3.3.4. Guided by the System Integrator, the knowledge carriers answer the question: What information from which sources must flow into the components of the service structure so that the smart service can be developed and operated? The answers to this question can be described in the form of a graph: The target nodes are elements of the service structure, the source nodes are elements of the service or product structure, and possibly external data sources (e.g., weather data from an external information provider). The information content is represented by SysML Information Items in free text.

The goal of the Describe required information quality sub-process step is to detail the information flow concerning information quality. The result of this process step should be an information model that describes information flow as well as information quality. Guided by the System Integrator, this time, the knowledge carriers answer the question: What quality level must the information described within the information model have for the smart service to be developed and operated? The answers to this question are documented in the information model of the previous process step Describe required information flow by textual annotation of the Information Items. Consideration of the quality dimension is fundamental, because not only can costs increase exponentially with quality requirements, but also the probability of success in developing the desired quality of service (e.g., measurement every day versus every second for thousands of bearings) depends critically on the quality of information available in the data.

The goal of the Integrate statistical analysis results sub-process step is to account for any correlations that may occur in existing data that could be useful for the operation and development of the smart service. This is necessary because the data specification process has so far been designed to be purely knowledge-driven. However, data relevant for the smart service is often already available in the company. Targeted statistical analyses of the available data can unearth previously undiscovered or even largely unknown relationships in a data-driven manner. The members involved include the Core Team, domain experts brought in as needed, and Data Analytics Specialists (Data analytics specialist and ML expert. Responsible for development and implementation of big data solutions [35]). The data analysis conducted by the Data Analytics Specialists is presented to the Core Team and relevant domain experts so that they can check whether there are spurious correlations or trustworthy causal relationships. This review is necessary because otherwise there

is a risk of training AI models with spurious correlations, which in turn would lead to unpredictable misbehavior of the SSS in the operation of the smart service once the spurious correlation no longer holds. The knowledge gained from the statistical analyses is added to the information model. This describes the information requirements. In the next step, the information level (e.g., vibrations along the outer ring of the Ball Bearing 2 must be measured hourly) is broken down to the data level (e.g., vibrations at the Ball Bearing 2 are measured hourly with a sampling rate of 200 Hz and stored under the variable name vibration_mainBearing2_200).

3.2.3. Specify Data Needs

The goal of the Specify data needs process step is the final specification of the data requirements of the smart service. This involves all roles of the Core Team as well as the Data Analytics Specialist. If necessary, domain experts can be involved upon request of the Data Analytics Specialist. The sub-processes explained below and shown in Figure 7 are carried out.

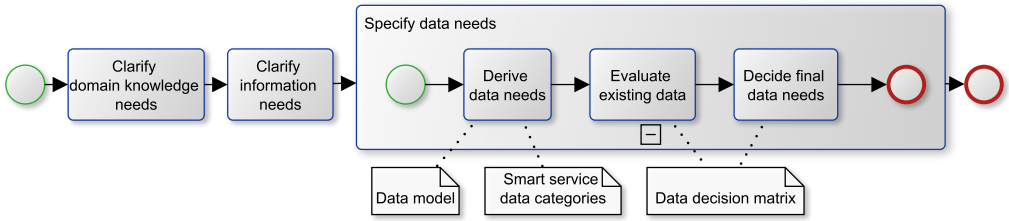

Figure 7. Process steps to specify data needs.

The goal of the Derive data needs sub-process step is to convert the information model into a data model. The data model describes which data are needed to meet the information requirements described in the information model. For this purpose, variables are defined (or documented in the case of existing data) and the information quality described is supplemented by figures, data, and facts from the field of data quality management. The data types are also defined in the process.

The goal of the Evaluate existing data sub-process step is to complete a technical fit-gap analysis comparing the data requirements described in the data model and the existing data. This involves assessing the extent to which the existing data meets the data needs. In addition to searching for relevant data, this also requires a technical assessment of the data quality. Here, the Data Analytics Specialist can be supported by the Information Service Provider (Provides supplementary data from external sources [35]), the Data Center Operator (Operates the IT infrastructure [35]), the Cloud Platform Provider (Operator of application-independent (external) cloud components [35]), and the Connectivity Provider (Responsible for technical interface (e.g., mobile network) between (smart) product and IT infrastructure [35]) are supported. The result of this process step is a qualitative assessment of the gap between data requirements from the domain expert's perspective and the existing data. Nevertheless, the economic perspective is still not yet included in this final definition of the data requirements but is taken into account in the next step of the data specification process.

The goal of the Decide final data needs sub-process step is the final specification of data requirements. However, after the domain experts finish specifying the data and information needs, maybe the costs for data acquisition exceed the expected revenue of the smart service. In this case, the smart service would be a loss-making business. Therefore, at this point, the profitability of the smart service is reviewed and optimized–if necessary by reducing the data requirements or removing variables. This process step is guided by the Project Sponsor. It involves the Core Team, the Data Analytics Specialist, and domain experts, as needed. This process step relies heavily on the data decision matrix (see Section 3.3.3). The final determination of data requirements concludes the data specification process. The artifacts described below add up to the semantic data specification.

3.3. Artifacts

The artifacts used within the data specification process are described below.

3.3.1. Knowledge Carrier Matrix

For the systematic documentation of relevant knowledge areas and the associated knowledge carriers, we developed a Knowledge carrier matrix, presented in Table 2. To do so, we took the matrix-like representation of knowledge requirements described in [40] and adapted it for data specification.

The Knowledge carrier matrix organizes knowledge into four knowledge area: service (e.g., predicting the remaining lifetime of the main ball bearing), physical product (e.g., main ball bearing wear behavior), data science and data engineering (e.g., the training of ML models), and other (e.g., legal requirements of aircraft maintenance).

Table 2. Knowledge carrier matrix.

	Knowledge Carrier		
Field of Knowledge	**Role (Name/Organization)**	**Role (Name/Organization)**	**...**
Service			
Sub-field 1	X		
...			
Physical product			
Sub-field 1	X		
...			
Data science and data engineering			
Sub-field 1		X	
...			
Other			
Sub-field 1		X	X
...			

The knowledge areas are noted in the first column and—depending on the project requirements (level of detail, relevant areas)—supplemented by additional knowledge areas in the rows. To fill this matrix, the following guiding question needs to be answered: Which areas from the categories of "service", "physical product", and "data science and data engineering" are directly or indirectly affected by this smart service development project? In answering this question, there will also be areas that do not fit into any of these three categories. Such knowledge areas can be noted in the Other category.

The column headers are filled in with the name of the person who has the most knowledge in the knowledge area of the corresponding row. If the role is carried out outside the company, the corresponding organizational name is entered. For filling the column headers, the following question needs to be answered: Which role carries the broadest knowledge in area A? The A is replaced by the knowledge area of the respective line. An X in a cell means that the person represented by this column has the broadest knowledge in the knowledge domain of the corresponding row. The formulation of this question ensures that the most relevant roles are documented as knowledge carriers. In this context, it is also possible that a single knowledge carrier will carry knowledge in many knowledge areas. This is justifiable and even helpful: a reduced number of knowledge carriers means a leaner data specification process (see Section 5). By focusing this question on the knowledge carriers that have the broadest knowledge, the process minimizes the number of people involved, while still maintaining comprehensive coverage of the

knowledge domain. The wording of the question means that mainly technical leadership roles are documented in the first iteration of the knowledge carrier matrix. This is legitimate since the data specification process first takes place at the system-architect level and the technical leadership roles can easily involve additional subject matter experts for details if needed.

3.3.2. Data Categories

SemDaServ offers a hierarchical model of smart service data categories to create the most complete possible specification of the data required for a given smart service. The data category model presented in Figure 8 helps actors during the data specification to check if all data categories have been considered. The data category model is described hierarchically to tailor the abstraction level of the data specification appropriately to the smart service under consideration as well as to the particular requirements of the project (Particularly concerning the level of detail of the data specification (see Section 5)). We derived the data category model from the literature by combining data categorization systems from multiple domains. These domains included engineering, statistics, and computer science. The "data science" domain itself uses categorizations from statistics and computer science, so this domain was not considered separately.

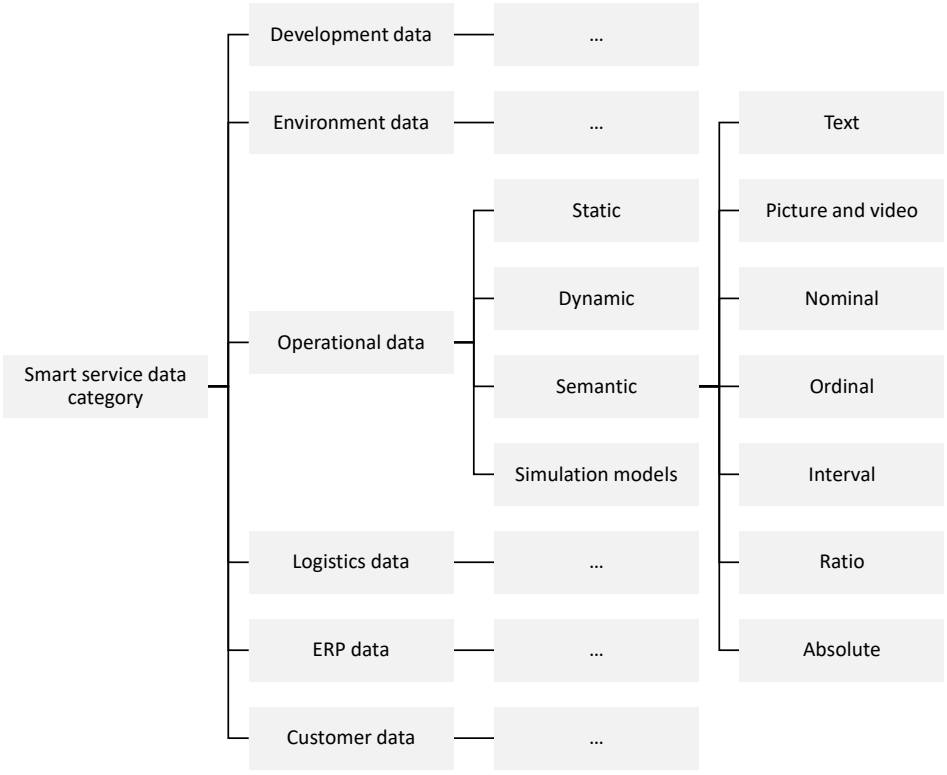

Figure 8. Data categories for the specification of smart service data.

After analyzing the presented data categorization systems, it turned out that the categorization system presented in ([41] p. 79) was the closest to the application domain of smart services and the most intuitive for actors with heterogeneous professional backgrounds to understand. For this reason, we adopted it as the top level of the data category model. There were two exceptions made: One exception was Exner et al. [41]'s category of expert knowledge, which SemDaServ does not conceptualize as a data category but as a type of knowledge, and therefore is not considered in the data category model. The second exception is in Exner et al. [41]'s data category machine data which we think is too narrow. Therefore, we reframed this data category as operational data (e.g., to cover data acquired from humans). A detailing of the data categories according to ([41] p. 79) is achieved by the data categories from ([42] pp. 40–41): This categorization classifies data according to

various properties and can therefore be applied downstream to the domain-oriented data categories from ([41] p. 79). One level downstream, the categories from statistics can be found, as they are domain-independent and applicable to all other data categories from engineering, thus generating a deeper level of detail. Furthermore, the scale system of statistics is close to the methods from data science, which is why a description of the data in the categories of statistics is helpful for AI training in the context of smart service development bridging the gap between the application domain and statistical AI models. Text, image and video data categories as defined by (Runkler [43] pp. 1–2) are also included at the most detailed level because these categories cover important data domains that are not represented in the statistics scale system. The remaining data categories in Runkler's [43] text are were already included in the categories outlined in ([41] p. 79) and ([42] pp. 40–41), and do not need to be repeated.

3.3.3. Data Decision Matrix

The data decision matrix supports and documents the decision as to which data from which sources should ultimately be consumed by the smart service. To do so, the system takes technical (data quality) and economic (cost-benefit) aspects of the data collection requirements into account. Filling out the data decision matrix requires an upstream assessment of the gap between the defined data needs and the quality of existing data, as well as an estimation of what costs will be incurred to sufficiently cover the data needs.

For the smart service development project to be successful, the resulting smart service must be profitable. For this reason, the data decision matrix includes information about the cost–benefit ratio of data collection. Methodologically, this is done through the cost-benefit analysis presented in Figure 9. The goal is to maximize the cost-benefit ratio. The economic benefit of the smart service is usually already determined and thus known before the start of the data specification process in the context of business model development and analysis through appropriate SSS requirements. The costs of developing and operating the smart service result from data requirements, IT infrastructure, the required quality of the AI models, and data quality. Data requirements result in costs for adapting or redeveloping physical product components for product development, manufacturing and operation. Furthermore, the IT infrastructure must be adapted due to additional data streams: Again, this includes development, initial deployment, and operating costs. The data requirements of the smart service can be partly covered by synthetic data from simulation models. This reduces costs in the area of physical product components but generates modeling costs if the existing simulation models have to be adapted, created, or expanded.

Once the technical and economic assessment of the data requirements is available, the data decision matrix shown in Table 3 can be filled in by the actors involved in the decision-making process based on the results of earlier steps in the data specification process. To do this, the variables specified in the data model are entered in the first column. This is followed by the decision of the Project Sponsor whether the variable located in the respective row should be measured in real terms or generated synthetically by simulation models. Data requirements and the existing data quality are entered in the following columns. This is done by first assessing the relevance of the information content (RI) on a scale from 0 (information content of the variable is irrelevant) to 6 (the smart service cannot be developed according to the requirements without this variable). The Data Quality Score (DQS) is determined by assessing the existing data quality using the data quality dimensions based on [34,44] and shown in Figure 10. To do this, each major criterion is qualitatively estimated on a scale from 0 to 2. The meaning of the scale is defined as follows:

0 Existing data does not satisfy the data requirement of this data quality factor.

1 Existing data probably satisfy the data needs of this data quality factor.

2 Existing data satisfy the data needs of this data quality factor.

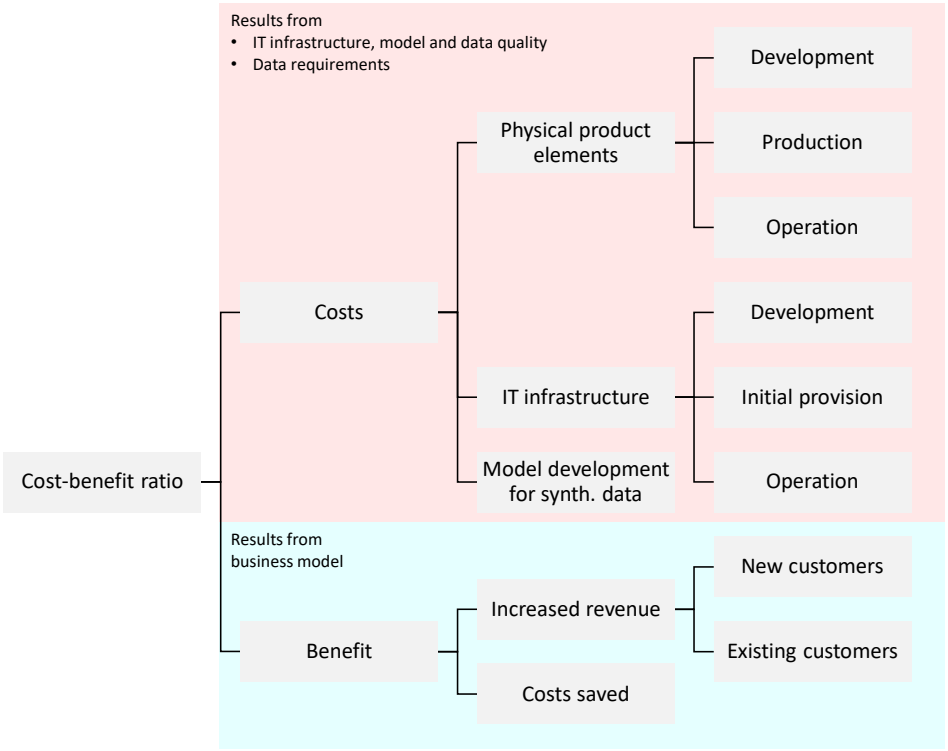

Figure 9. Cost–benefit analysis.

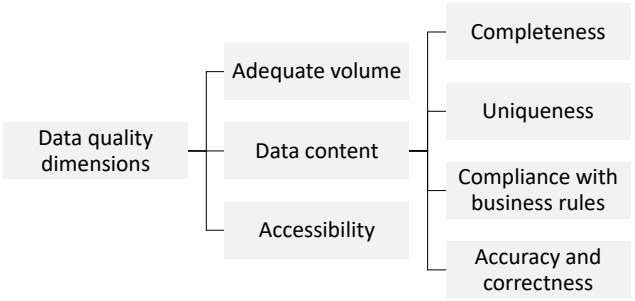

Figure 10. Data quality dimensions based on [34,44].

The rating for the main criterion of content data quality is the arithmetic average of the four sub-criteria but can be determined directly for efficiency reasons. The data quality value results from the sum of the ratings of the three main criteria and is thus at most six (existing data quality fulfills the data needs in all factors) and, at least zero (existing data quality fulfills the data needs in no factors). Thus, the Relevance to Quality Score (RQS) can be calculated according to Formula (1):

$$RQS = \frac{RI}{1 + \Delta DQS} = \frac{RI}{1 + 6 - DQS} = \frac{RI}{7 - DQS}. \quad (1)$$

The RQS is thus a means of focusing on the most relevant variables, which are already present in the highest quality. The maximum value of the RQS is 6 ($RI = DQS = 6$), meaning that the variable has the highest possible relevance for the smart service while already having sufficient data quality available. The cost of collecting this variable, in this case, is zero because the variable is already provided with existing simulation models, IT systems, and physical product components. If this were not the case, the DQS would be less than six, since at least one data quality factor would not fully satisfy the data requirement, resulting in costs for data collection. An RQS value of 0, on the other hand, would mean that a variable has no relevance to the smart service ($RI = 0; 0 \leq DQS \leq 6$). If the data decision matrix is now sorted in descending order by the RQS, the most relevant

variables with the highest pre-existing data quality will be at the top. This allows the Core Team to prioritize which final data to collect.

To prioritize data needs, it is also important to consider the costs of data collection: For this purpose, the costs required to adequately collect a variable are filled in the rows physical product components, IT infrastructure, as well as modeling for synthetic data and totaled.

After all these elements have been entered, it is possible to make a final decision for each variable, whether it shall be collected or not. This decision can now be documented in the last column for each variable. The entry of yes or no in the last column indicates whether the data on the variable located in the row should be collected. The sum of the variables with a yes entry in this column represents the final set of data to be collected.

Table 3. Data decision matrix.

		Data Need and Existing Data Quality			**Costs**				
Variable Name	**Data Source**	**Relevance of Information *RI***	**Data Quality Score *DQS***	***Relevance to Quality Score RQS***	**Physical Product**	**IT Infrastructure**	**Modeling for Synthetic Data**	***Sum***	**Collect Data?**
Variable 1	real/synthetic	0–6	0–6	*0–6*	X $	X $	X $	*X $*	yes/no
…	…	…	…	…	…	…	…	…	…
Variable n	real/synthetic	0–6	0–6	*0–6*	X $	X $	X $	*X $*	yes/no

3.3.4. Data and Information Models

SemDaServ uses diagrams and modeling elements of the SysML language to model the information and data requirements. The data model is based on the information model and therefore uses the same SysML diagrams. The difference between information models and data models is in the object of study. However, the representation and modeling approach is identical—except for minor differences outlined below.

The central element of the data specification is the block definition diagram: This is used to build both the information model and data model. The basic elements of the information model are shown in Figure 11: The SysML element entitled Information Item is at the center of the information flow of the Producing element block (e.g., a sensor) and the Consuming element block (usually a software component of the smart service). These two blocks are linked to the Information Item by the SysML element of Information flow. The information contained in the information flow is modeled by a textual description of the Information Item. The quality requirements for the information are stored as free text by the SysML note element Description. This results in an information flow from the Producing element (source) to the Consuming element (target). This can be a one-to-n-relationship, which can be represented by additional information flows of an Information Item additional Consuming elements.

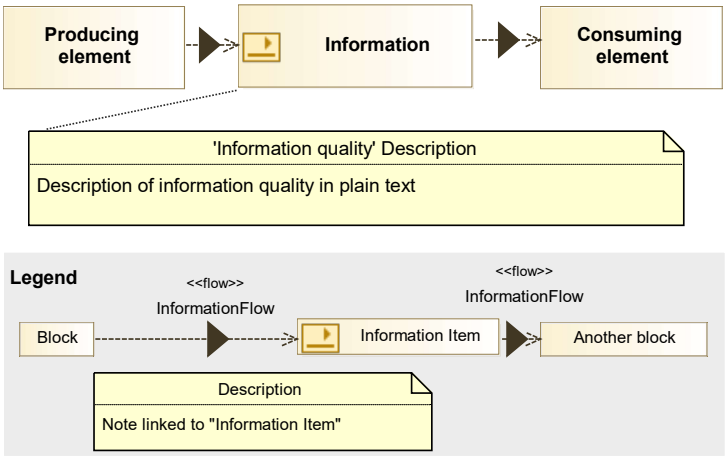

Figure 11. Information model.

The data model results from resolving the Information Items of the information model. The Information Item is replaced by a direct information flow linked to the data variable of the producing and the consuming elements. The variables are modeled as attributes of the blocks. If the information described in the Information Item can be completely covered by one variable, the information flow is modeled at the variable level (by direct linking of the variable using the SysML element Information flow). If several variables are required to realize the information flow, an Information Flow is modeled between the Producing element and Consuming element blocks. The basic elements of the data model are shown in Figure 12. The placeholder <no type> represents arbitrary data types. To resolve the Information Item, the information quality must also be broken down to the data quality. For this purpose, SemDaServ uses the SysML note element Description similarly as in the information model, but in this case, this element is directly linked with the variables. Thus, the data quality is described for each variable. The description of the data quality follows the scheme described in Figure 10. To increase the clarity of all information models and data models, the Note elements can be modeled outside the diagram by using a model-oriented tool.

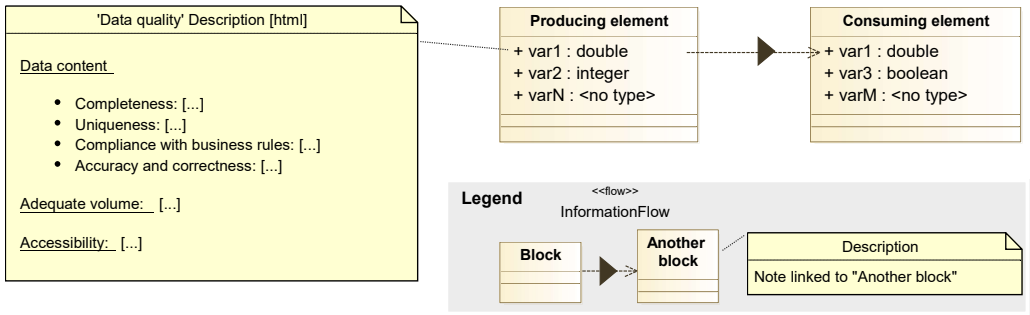

Figure 12. Data model.

If these initial modeling possibilities are not sufficient in a particular use case, the information model and the data model can be supplemented by additional UML and SysML diagrams: sequence diagrams, for example, can be used to model time-dependent relationships. The use of use case diagrams can help in understanding the use case when the business model description, as well as the product and service structure, are too complex, too complicated, or too superficial from the perspective of the actors involved. The use case diagram can then be used to focus on the essentials—especially from the user perspective. If a large number of detailed quality descriptions arise and/or exhibit a large number of mutual dependencies, the information and data quality descriptions can be combined in a requirement diagram, linked to the associated blocks, and modeled to take these dependencies into account. If it turns out during the data specification process that

a block (e.g., an assembly of the SSS, represented as a block in the data model) needs to be considered in more detail, the corresponding block can be linked to an internal block diagram and described in detail. If the description of the generation of synthetic data by simulation models is particularly important or specific, the relevant correlations and calculation rules can be described in the parametric diagram and subsequently linked to the corresponding variables.

3.4. IT Systems and Tools

The SemDaServ approach benefits from the application of software tools and IT systems used in product creation. The following section gives an overview of how SemDaServ does so, based on typical tools and IT systems from three categories: System description tools, data science tools, and enterprise IT systems.

Model-based and document-oriented tools for system description support the creation of information and data models. A distinction must be made between model-based and document-based tools. Both model-oriented and document-oriented tools can be usefully employed for data specification. Therefore, we recommend users to rely on the approaches already established in their respective companies, to optimize the training effort, and thus the cost-benefit ratio of the data specification is optimized. Document-oriented tools include text and table processing tools (e.g., Microsoft Office, Libre Office, etc.)—programs generally used for office work. The advantage of this is that many people are familiar with them and they are widely distributed, meaning companies can easily access them. One disadvantage in the context of data specification is the poorly developed support for modeling with UML and SysML. Specialized visualization tools such as Microsoft Visio, yED, or DIA provide better support. The advantage here is again the comparatively high distribution and easy accessibility of visualization tools in contrast to specialized modeling tools. However, since these visualization tools take a document-based approach, there is a significant disadvantage that changes to the information or data models (e.g., renaming a block) may not be propagated throughout the model. In contrast, the use of model-based tools enables the creation of an information and data model that can be validated, is machine-readable, and maintains links among its elements. This provides higher compliance with modeling language specifications (e.g., model-based tools can alert users to model errors and can prevent elements from being used out-of-specification) and higher model quality (e.g., name changes of variables or blocks propagate themselves throughout the model).

Data science tools primarily support the role of the Data Analytics Specialist by providing ways to evaluate existing data (e.g., from older generations of products already in use). The tools presented in this section exemplify the range of data science tools that can be used to analyze and evaluate existing data as part of the data specification process. It is also possible to use these tools to train the AI models required for the smart service with AutoML methods on existing data to obtain a sound evaluation of the quality of this data.

A large number of IT systems are used in companies, and advancing digitalization means that the amount of data and information stored in these systems will only increase further (cf. [45]). This makes the enterprise IT of a company an important supplier of data and information in the context of data specification. In practice, the characteristics and operational use of a company's IT landscape are very heterogeneous. In principle, data specification can be performed with any type of enterprise IT landscape. However, a well-developed enterprise IT infrastructure in the areas of PDM and IoT is especially helpful: PDM systems provide a good source of structured, contextualized data (e.g., product structure), while IoT systems can manage the operational data of SSS being in the use phase of the product lifecycle.

4. Evaluation

The evaluation is based on two workshops and seven expert interviews described below. The profiles of experts consulted for evaluation are outlined in Table 4. The experts for interviews and workshops were chosen in a way that a broad range of professional backgrounds of actors relevant for SSSE as well as different company sizes are represented. The resulting group of chosen experts, therefore, ranges from academic to small and big companies. The sectors are focused on mobility, but the experts from academia have a history of researching a broad range of engineering-focused sectors. Overall, the majority of experts come from industry (7 out of 12). The following experts share the same employer: Experts 1, 6, and 10; experts 2 and 8; experts 3, 4, and 5; experts 7 and 9. The specialization of the experts covers all professional backgrounds relevant to the SemDaServ approach, ranging from the business perspective (expert IDs 5, 7, 11), to the SSSE perspective (expert IDs 1, 2, 8), the MBSE perspective (expert ID 6) a highly specialized knowledge carrier perspective (expert ID 12), the data science and data engineering perspective (expert IDs 3, 4, 7, 9), and the knowledge management in engineering perspective (expert ID 10). The professional experience of the experts also spans a wide range from more than two years up to more than 40 years of professional experience.

Table 4. Profiles of experts consulted for evaluation of the SemDaServ approach.

ID	Job Title	Specialization	Professional Experience	Company Type	Sector
1	Research assistant	Smart service platforms, Internet-of-Things, cloud computing	>2 years	Technical university (>7.500 employees)	Academia
2	Research engineer	Smart service systems engineering, Internet-of-Things	>3 years	Research institute for applied science (>25.000 employees)	Academia
3	Data scientist	Smart services, natural language processing, condition monitoring	>3 years	Big company (>30.000 employees)	Rail
4	Data scientist	Artificial intelligence, operations research, data engineering	>6 years	Big company (>30.000 employees)	Rail
5	Principal key expert	Project management, business strategy	>20 years	Big company (>30.000 employees)	Rail
6	Research assistant	Model-based systems engineering	>3 years	Technical university (>7.500 employees)	Academia
7	IT project manager	Artificial intelligence, production planning, six sigma	>4 years	Big company (>150.000 employees)	Automotive
8	Researcher and managing director	Smart service systems engineering, product lifecycle management	>7 years	Research institute for applied science (>25.000 employees)	Academia
9	Doctoral candidate	Artificial intelligence and digital twins	>6 years	Big company (>150.000 employees)	Automotive
10	Research assistant	Knowledge management in engineering	>8 years	Technical university (>7.500 employees)	Academia
11	CEO	Innovation, management and technology consulting	>30 years	Small company (<20 employees)	Consulting
12	System architect	Predictive maintenance, engine health monitoring, system architecture	>40 years	Big company (>50.000 employees)	Aerospace

The first workshop (participating experts: IDs 1 and 2; cf. Table 4) confirmed, that the SemDaServ approach is logically correct, consistent, and fills a research gap in the field of SSSE. Regarding the success criteria (cf. Table A3), it was pointed out that more experiments are needed to measure the impact of the SemDaServ regarding success criteria 20 (efficiency of smart service development methods) and 22 (efficiency of the application of domain knowledge). This is caused by the fact that the SemDaServ approach itself generates efforts that need to be compared to approaches not using the SemDaServ approach (e.g., trial and error mixed with explorative data analysis using established data science tools and methods). For all other success criteria, it could be validated from a logical perspective that they are well addressed by the SemDaServ approach.

The second workshop (participating experts: IDs 3, 4, and 5; cf. Table 4) confirmed, that the SemDaServ approach is logically correct, applicable, and useful for real-world applications. The comprehensibility and low access barrier of the artifacts used and the process steps were rated as very good. It was pointed out that scaling SemDaServ according to real-world project scenarios (budget, time, quality as well as the availability of experts) is an important aspect. Therefore, guidelines on tailoring SemDaServ to different project scenarios will be developed in future research. It was confirmed that the relevant stakeholders in the workshop participants' company can understand the artifacts resulting from SemDaServ and therefore a benefit is generated for the collaboration of product and service development. From a data scientist perspective, it was confirmed that the resulting semantic data specification provides great added value, as SemDaServ systematically describes relevant data and domain knowledge relevant for data understanding. Thus, iterations during smart service development can be prevented by developing a suitable data basis at an early stage.

The interview with expert 6 (cf. Table 4) was focused on the topic of MBSE and the related use of SysML. In this interview, it was confirmed that the SemDaServ approach is compatible with the MBSE procedures and that the chosen representation type in the block definition diagram, as well as the use of the diagrams optionally mentioned in Section 3.3.4, is reasonable and logically correct.

The interview with experts 7 and 9 (cf. Table 4) focused on current best practices established in the industry regarding the collaboration of product and service development related to data specification for AI applications. It turned out that in practice relevant data is mainly searched for according to a data-driven trial and error approach. For this purpose, the data scientists ask domain experts known to them from the past in an unstructured way which data is relevant for the application. Since the data scientists have little knowledge of the application domain, the questioning about relevant data usually remains at too general a level, as a result of which important details are lost. This process is time-consuming and causes many smart service projects to fail due to insufficient data. The systematic approach SemDaServ was evaluated as a suitable solution for reducing try and error iteration loops and thus increasing the efficiency in smart service development resulting from an improvement of the cooperation between product and service development.

The interview with expert 8 (cf. Table 4) focused on the connection between product and service structure, the compatibility of the SemDaServ approach with other methods of SSSE, and the trade-off between document-oriented and model-oriented approaches. As a result, it was determined that there are methods such as MESSIAH [38] that should be used to develop an initial service structure. The resulting elements of the service structure can then be related to the product structure via the information items and information flows described in SemDaServ. It was confirmed that SemDaServ is equally suitable for document-based and model-based approaches and thus fits well into the current state of the art (mostly document-based approach) and at the same time is fit for the future (model-based approach).

The interview with expert 10 (cf. Table 4) focused on the formalization of domain knowledge. The discussion mainly focused on the right degree of formal specifications regarding the modeling language for explicating domain knowledge. A higher degree of detail in the specifications and language constructs of the modeling language leads to more difficult access to the modeling language and thus, in the expert's experience, to less use of the modeling language in practice. The advantage is the unambiguousness in the interpretation and thus the reusability of the explicated knowledge. A lower level of detail in the specifications (e.g., allowing free text without specific formal specifications) leads to less unambiguity, but the access barrier to the use of the modeling language is lower, which is why the circle of users increases. From the workshops as well as other expert interviews, it became clear that the SSSE requires a multidisciplinary team with a variety of different actors. Therefore, the SemDaServ approach uses as few formal specifications for information and data models as possible. Nevertheless, further SysML diagrams can be used, which accordingly also bring the advantages of the multitude of specified specifications of SysML. In the interview, the level of detail of the specifications in the information and data model was confirmed as sufficient. In addition, the conclusiveness of the sequence of the SemDaServ process steps for running through the knowledge pyramid was confirmed.

The interview with expert 11 (cf. Table 4) focused on the broad applicability of the SemDaServ approach independent of the use case, role profiles of the actors, conclusiveness, and usefulness for practice. Expert 11 confirmed that the SemDaServ approach is generic enough to be used in an industry-independent manner. The role profiles described are appropriate and can already be found in practice in several companies. The version of the knowledge carrier matrix presented to expert 11 was not yet based on roles, but only noted the names of the knowledge carriers. This was changed at the suggestion of expert 11: In the SemDaServ version, in addition to the name, the knowledge carrier matrix primarily records the role and also the organization of the knowledge carrier. The conclusiveness of the approach and its usefulness for practice was confirmed.

The interview with expert 12 was focused on the suitability of the SemDaServ approach for the possibility of achieving a complete specification of all data. The question was whether knowledge-driven domain experts can identify all relevant data for a smart service in early development phases without having to fall back on previous data. The interview revealed that domain experts can identify a large part of the relevant data by naming the required information content and information flows. Nevertheless, it is required to have domain experts check statistical analyses of already existing data-especially correlation analyses-in the context of SemDaServ. This enables the discovery of relevant correlations that were previously unknown to the domain experts or that were simply forgotten. By having the revealed correlations checked by the domain experts, spurious correlations can be discovered and excluded from the data specification. Based on the interview with expert 12, the sub-process step Integrate statistical analysis results (see Section 3.2.2) was therefore added to the data specification process Clarify information needs.

5. Discussion

The SemDaServ approach should be adapted in practice depending on the specific requirements of the SSSE project as well as available resources (time, budget, staff, external experts). SemDaServ is theoretically applicable to all kinds of SSSE projects. For practical use, adapting SemDaServ to the particular SSSE project requirements is necessary to justify the expenses linked to the realization of SemDaServ. Indeed, conceptualizing SemDaServ in practice as a guiding framework rather than a rigid system makes its application more efficient. The four most important factors for adapting SemDaServ in practice are as follows:

1. Number of product and service components
2. Number and heterogeneity of actors to be involved
3. Requirements for the modeling depth
4. Method of operation (model-based or document-based)

For Factors 1 and 2, the guiding principle should be "as much as necessary, as little as possible", because these factors exponentially scale the effort related to SemDaServ's application. For example, assume that predictive maintenance of the aircraft engine's main ball bearings would be the most critical element of a power-by-the-hour smart service. Then it would make sense to focus the data specification on the physical component main ball bearings and the smart service component predictive maintenance. This makes the relevant part of the product structure and the service structure very small. The actors required to carry out the data specification process would be, in addition to the Core Team, a few domain experts for ball bearings, and data scientists with knowledge in the area of service life prediction. This reduction to the core elements makes the SemDaServ approach feasible in a lean way in this case. Similarly, the principle of "as simple as possible, as detailed as necessary" applies to Factor 3. For example, it is not necessary to formalize every detail of possible signal waveforms from sensors on the main bearings as part of the creation of the information and data model. It is much more important to specify that suitable sensors must be installed in the engine to monitor the condition of the main bearings. Factor 4 should be aligned to the usual approach within the company. While using a model-based approach has benefits such as traceability or automatic updates of linked elements, the document-based approach is more accessible to a larger set of stakeholders, as it usually does not require specialized knowledge regarding software tools. For example, for small projects, it may make more sense to perform the SemDaServ approach in a document-oriented manner using information and data models created in Microsoft Visio than to completely abandon the use of the SemDaServ approach if neither software licenses nor the know-how to use a model-oriented approach is available.

Regarding the working hypotheses the following conclusions are drawn from the evaluation presented in Section 4: WH1 (The probability of AI development with quality of results being in line with smart service requirements can be increased by a domain knowledge-driven data specification approach.) was confirmed in the interviews with the experts 7, 8, 9 (cf. Table 4 for looking up the IDs) as well as in both workshops. WH1 was not discussed in the remaining expert interviews because these expert interviews had a different thematic focus. The interviews and workshops revealed that a data-driven approach is currently used in practice for the development of AI components. This means that all available data is checked for its suitability regarding the realization of smart service requirements. Domain knowledge is of elementary importance here, since domain experts can use their knowledge to identify relevant data and distinguish causality from spurious correlation. If the available data is too small in scope or too low in quality, physical components of the SSS must be adapted, the quality requirements for the AI components must be reduced or, in the worst case, the SSS development project must be aborted. Data specification in early product development phases can reduce the probability of occurrence of the aforementioned scenarios. For illustration purposes, imagine the following synthetic example: Requirement 1 'The remaining lifetime of the main bearings must be known with an accuracy of more than 95%' and Requirement 2 'The number of sensors must be reduced as much as possible to save costs.' To meet Requirement 1, an ML model is trained based on data from engines already in use. The engine data used for training is selected so that the ball bearings used are comparable to the new engine model. Based on the data from already in-service engines, the resulting ML model meets Requirement 1. When testing the ML model on the data from the prototype of the engine to be developed, it is found that the 95% accuracy requirement is not met, as the new engine has fewer sensors in order to meet Requirement 2. In this situation, the quality of AI results (the accuracy of the ML model) is not in line with the smart service requirements. This is caused by an unfulfilled data need (missing sensor data). In the mentioned example, extensive product changes (adding

the needed sensor as well as a redesign of affected components, electrical layout, etc.) and the need to redo the testing of all components (software and hardware) affected by the changes caused by the new sensor. Experts from the field of bearing technology would have known which data is required to predict the remaining life of a bearing. By using a domain knowledge-driven data specification approach, data needs can be discovered and cross-checked with SSS requirements. This can increase the probability of AI development with the quality of results being in line with smart service requirements. The extent to which this probability is increased is not part of WH1. Thus, WH1 can be confirmed.

WH2 (A clear understanding of relevant data for the development and operation of a smart service can be systematically generated by using domain knowledge to clarify information needs and derive data needs from information needs.) was discussed in all expert interviews and both workshops. As a result, WH2 can be partially confirmed. There was agreement that WH2 can be confirmed for small SSSE projects (e.g., predictive maintenance of a bearing) involving just the Core Team (cf. Table 1) and a handful of domain knowledge carriers. However, further research needs to be conducted to evaluate scalability for larger projects. It remains open up to which count of actors, as well as product and service components, the presented SemDaServ approach scalably leads to a clear understanding of the data relevant for the smart service. For this purpose, experiments are required to investigate the impact on individual process steps when scaling up the number of relevant actors, product, and service components. In addition, case studies on complex, industrial application examples are required to evaluate whether the presented SemDaServ approach may need to be given additional process steps or artifacts for larger SSSE projects.

WH3 (Domain knowledge relevant for AI training can be formalized using the SysML.) was confirmed in the expert interviews 6, 7, 8, 9, 11, and in both workshops. WH3 was not discussed in the remaining expert interviews because these expert interviews had a different thematic focus. Both workshops confirmed, that the SysML information model and data model of the SemDaServ approach are feasible to formalize the domain knowledge relevant for AI training. Expert 6 confirmed that the SysML elements used for the information model and data model are a valid choice to formalize domain knowledge relevant for AI training. Furthermore, 'SysML is designed to provide simple but powerful constructs for modeling a wide range of systems engineering problems. It is particularly effective in specifying requirements, structure, behavior, allocations, and constraints on system properties to support engineering analysis.' ([46] p. 1) Workshop two as well as interviews with experts 7, 8, 9, and 11 revealed, that these kinds of knowledge from the engineering domain (requirements, structure, behavior, allocations, constraints) are relevant for AI training. In workshop two, it was positively emphasized that the beneficiaries of the data specification—the data scientists, who were using their AI models on qualitatively and quantitatively sufficient data as well as a good semantic description of the data–can easily understand SysML due to its similarity to UML.

WH4 (Domain knowledge relevant for AI training can be formalized using a guided process.) was confirmed in the expert interviews 7, 8, 9, 10, 11, and both workshops. WH4 was not discussed in the remaining expert interviews because these expert interviews had a different thematic focus. According to the experts involved in the evaluation of WH4, the formalization of the domain knowledge relevant for AI training has not yet been described in practice as part of a standardized process. Different procedures established in practice for formalizing the domain knowledge relevant for AI training were therefore mentioned. As a result, the procedure differs depending on the industry, the size of the company, and the people involved. However, in both workshops and the expert interviews quoted for WH4, the following pattern emerged: the actors tasked with AI training (hereafter: AI engineers) first try to conduct data-based AI training through experiments. If this yields insufficient results, domain experts from the proximate company network are contacted. This is usually followed by an open interview of the domain expert to ask about relevant domain knowledge. If the AI engineer conducting the interview has

little to no domain knowledge, it is difficult for the AI engineer to ask relevant questions. The open interview of domain experts is then usually repeated until the AI engineer has the impression that the domain knowledge relevant for the AI training is known. Finally, the formalization of the domain knowledge is mostly done in plain text or the form of diagrams (e.g., UML or entity-relationship diagrams). The described procedure of AI engineers can be described in a guided process by putting the mentioned steps into a logical order and consequently systematizing them. Thus, WH4 can be confirmed. Participants of workshop two further noted that the semantic data specification resulting from the application of the SemDaServ approach is very helpful for the data understanding and data preparation (especially feature engineering) steps of the widely used CRISP-DM process.

5.1. Limitations

The evaluation was conducted by discussing the SemDaServ approach with experts from industry as well as academia (workshops and interviews). Therefore, SemDaServ was qualitatively evaluated using expert experience and logic. This research approach is feasible to evaluate the applicability in practice, usefulness, and logical correctness of the SemDaServ approach. To evaluate the effectiveness and efficiency in real-world SSSE projects, experiments (e.g., A to B comparisons with professionals specifying data needs without any guidance, data-driven AI training without any data specification, etc.) are needed to quantify the efficiency and effectiveness of SemDaServ. Especially the impact of SemDaServ regarding cost, quality, and time on real-world SSSE projects cannot be quantified yet. Additionally, the uncertainty factor of the data specification resulting from the SemDaServ approach cannot be quantified yet. To overcome these limitations, a significant number of case studies are needed to measure the quality of the smart services developed using the specified data needs resulting from the application of the SemDaServ approach.

5.2. Future Research

The evaluation of the SemDaServ approach revealed the needs for detailing the following aspects: (1) An exemplary agenda for workshops necessary for the Clarify domain knowledge process step. (2) Guidelines on how to tailor the SemDaServ approach to different project types (new development vs CI/CD scenarios) (3) Recommendations on building successful Core Teams (cf. Table 1) and combinations of knowledge carriers (cf. Table 2), especially regarding feasible competence mixes. Future research is planned to address these aspects. Additionally, it is planned to research a model approach to size the data types, frequencies, and reliabilities to support the creation of the data model (process step Derive data needs). Furthermore, experiments measuring how the number of actors, product, and service components impact the overall resources (time, budget, scope of required domain knowledge) needed to realize SemDaServ are planed.

6. Conclusions

This article has presented the SemDaServ approach. SemDaServ is a systematic approach for semantic data specification in the context of AI-based SSS. In comparison to data-driven approaches, SemDaServ is driven by the knowledge of domain experts, who can define the data needs of a smart service in early development phases—even if no operational data of the embryonic SSS is available yet. The availability of operational data requires (virtual) prototypes of the SSS. Therefore, operational data of the SSS becomes available in late product development phases. If unfulfilled data needs are discovered late, costly iteration loops (e.g., returning to the requirements definition phase) and product changes (e.g., adding sensors with the resulting need for adaptation of data processing and data analysis) may be required. Hence, a goal of specifying the data needs of a smart service during early SSS development phases is the reduction of iteration loops in SSS development projects, which correlates with reduced costs and a faster time to market. The SemDaServ approach contributes to achieving this goal by providing a three-step process, five artifacts, as well as guidance on tools and IT systems supporting the realization of

SemDaServ in practice. The SemDaServ approach intends to improve the understanding of relevant data for the development and operation of smart services, guide the systematic formalization of domain knowledge relevant for AI training as part of smart service development, and increase the probability of AI development with quality of results being in line with smart service requirements. SemDaServ was validated by expert interviews and workshops. This qualitative evaluation of SemDaServ confirmed the applicability in practice, usefulness, and logical correctness of the SemDaServ approach. Based on the findings so far, we assume that SemDaServ contributes to reducing iteration loops during SSS development, resulting in fewer costs and development time. However, case studies and experiments are required and planned to quantify the effectiveness of SemDaServ.

Author Contributions: Conceptualization, M.P.; methodology, M.P.; validation, M.P.; formal analysis, M.P.; investigation, M.P.; resources, M.P. and R.S.; data curation, M.P.; writing—original draft preparation, M.P.; writing—review and editing, M.P. and R.S.; visualization, M.P. and R.S.; supervision, R.S.; project administration, M.P.; funding acquisition, M.P. and R.S. Both authors have read and agreed to the published version of the manuscript.

Funding: The research has not received external funding. The article processing charge was funded by the Open Access Publication Fund of TU Berlin.

Acknowledgments: We acknowledge support by the German Research Foundation and the Open Access Publication Fund of TU Berlin.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
AutoML	Automated machine learning
BPMN	Business Process Modeling Language
CD	Continuous delivery
CI	Continuous integration
EOS	Engineering Operating System
IoT	Internet-of-Things
MBSE	Model-based systems engineering
ML	Machine learning
PDM	Product data management
PSS	Product service system
SemDaServ	Semantic Data Specification for AI-Based Smart Services
SSS	Smart service system
SSSE	Smart service systems engineering
SysML	Systems Modeling Language
UML	Unified Modeling Language

Appendix A

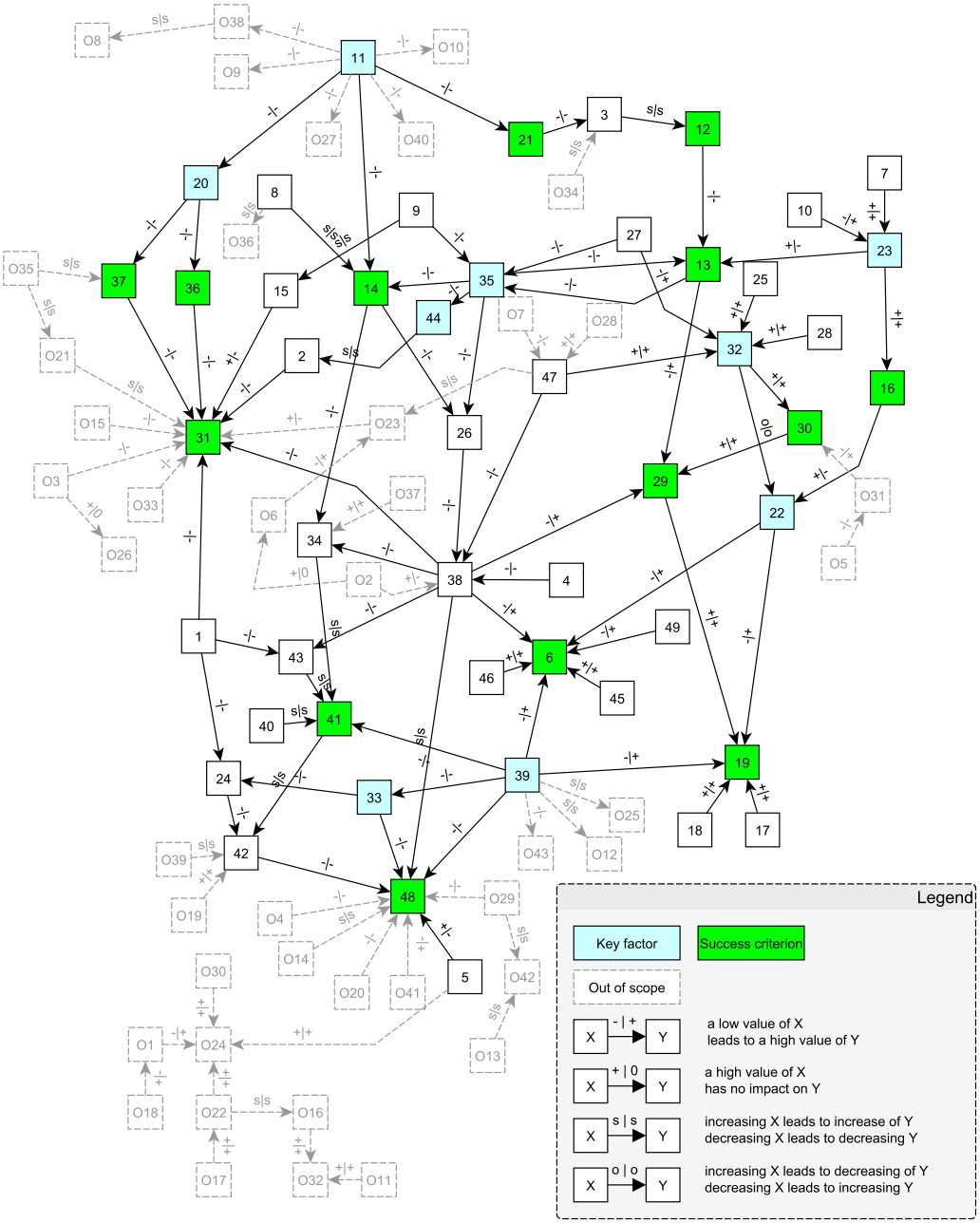

Figure A1. Reference model based on Design Research Methodology described in [23]. How to read the model: Take for example the connection 11 to O9. 11 stands for the degree of reusability of documented domain knowledge (looked up in Table A2). O9 stands for comprehensibility of the trained AI model (looked up in Table A1). 11 and O9 are linked with the label -|-. This means: Currently there is a low degree of reusability of documented domain knowledge and this leads to low comprehensibility of the trained AI model. As O9 is out of scope, the SemDaServ solution will not take this connection into account.

Table A1. Identified out-of-scope factors.

ID	Factor
O1	Agility of smart service development
O2	Assumed capabilities of data science tools to deal with low-quality data
O3	Availability of data scientists
O4	Availability/awareness of appropriate methods for service development
O5	Awareness of the added value of data analysis
O6	Capabilities of data science tools to deal with low-quality information
O7	Clarity regarding the ownership of the data
O8	Competitiveness of the company under consideration
O9	Comprehensibility of the trained AI model
O10	Contribution of domain knowledge to company productivity
O11	Degree of competitive advantage through differentiation by means of hybrid service bundles
O12	Degree of customer acceptance of the SSS
O13	Degree of customer integration in the development process
O14	Degree of customer satisfaction
O15	Degree of data literacy of actors involved
O16	Degree of the fulfillment of customer needs
O17	Degree of innovation of the smart service
O18	Degree of integration depth of smart service development methods into existing development processes, artifacts, and tools
O19	Degree of interdisciplinary collaboration
O20	Degree of management support
O21	Degree of transparency regarding the informative value of a data set
O22	Degree of uncertainty regarding customer needs, willingness to pay, market acceptance of the smart service
O23	Difficulty of data engineering
O24	Difficulty of smart service conception
O25	Duration of customer retention
O26	Duration of setting up a data engineering team
O27	Efficiency of domain knowledge generation
O28	Extent of manual data preparation
O29	Extent of testing to identify potential vulnerabilities of the SSS
O30	Individuality of the life cycles of the components of a SSS
O31	Investment in systematic the acquisition, processing, and preparation of data
O32	Level of margin
O33	Maturity of the AI-relevant IT infrastructure
O34	Professional experience of a data scientist in the target domain
O35	Quality of data science methods
O36	Quality of labels
O37	Quality of the analysis model
O38	Quality of the company's internal domain knowledge
O39	Quality of the product
O40	Reproducibility of ML algorithms from (scientific) publications
O41	Scope of the service spectrum
O42	Success rate of innovations
O43	Usefulness of the smart service

Table A2. Identified relevant factors.

ID	Factor	Key Factor?	Success Criterion?
1	Accuracy of fit of data selection	no	no
2	Availability of relevant data	no	no
3	Availability of domain knowledge required	no	no
4	Availability of synthetic data from (simulation) models	no	no
5	Complexity of the SSS	no	no
6	Cost of service development	no	yes
7	Degree of data-driven approach in the application of AI processes	no	no
8	Degree of human experience and skills	no	no
9	Degree of problem understanding with regard to the application domain	no	no
10	Degree of process-driven approach in the application of AI methods	no	no
11	Degree of reusability of documented domain knowledge	yes	no
12	Degree of systematicness in identifying relevant domain experts	no	yes
13	Degree of systematicness in linking domain knowledge and AI development	no	yes
14	Degree of utilization of relevant domain knowledge	no	yes
15	Difficulty of data analysis	no	no
16	Duration for identification of insufficient data situation	no	yes
17	Duration of adaptation of organizational processes	no	no
18	Duration of implementing changes in physical components	no	no
19	Duration until market maturity of the SSS	no	yes
20	Efficiency of domain knowledge application	yes	no
21	Efficiency of domain knowledge transfer	no	yes
22	Efficiency of smart service development methods	yes	no
23	Extent of trial and error approach to identifying relevant data	yes	no
24	Fit of requirements of existing system and new smart service	no	no
25	Heterogeneity of the data used	no	no
26	Information content of the data	no	no
27	Level of detail of problem specification	no	no
28	Number of data sources considered	no	no
29	Number of iteration loops in smart service development	no	yes
30	Number of redundancies of manual steps in data analysis	no	yes
31	Probability of success of the AI learning process	no	yes
32	Proportion of exploratory approach to data analysis	yes	no
33	Quality of collaboration between product and service development	yes	no
34	Quality of data analysis	no	no
35	Quality of data specification	yes	no
36	Quality of knowledge about data sources	no	yes
37	Quality of knowledge about relevant data	no	yes
38	Quality of relevant data	no	no
39	Quality of smart service development methods	yes	no
40	Quality of smart service development tools	no	no
41	Quality of the smart service	no	yes
42	Quality of the SSS	no	no
43	Quality of the trained AI model	no	no
44	Relevance of sensors in the product	yes	no
45	Risk of time-conditional data deviation	no	no
46	Risk of time-related model deviation	no	no
47	Scope of eligible data	no	no
48	Success of newly developed services on the market	no	yes
49	Transferability of trained AI models	no	no

Table A3. Identified success criteria for smart service data specification.

ID	Factor	Source
6	Cost of service development	[14,26,30]
12	Degree of systematicness in identifying relevant domain experts	[21]
13	Degree of systematicness in linking domain knowledge and AI development	Expert interview
14	Degree of utilization of relevant domain knowledge	Logical conclusion
16	Duration for identification of insufficient data situation	Expert interview
19	Duration until market maturity of the SSS	[26,29,30]
21	Efficiency of domain knowledge transfer	[14]
29	Number of iteration loops in smart service development	Logical conclusion
30	Number of redundancies of manual steps in data analysis	[27]
31	Probability of success of the AI learning process	[10,24,31]
36	Quality of knowledge about data sources	[31]
37	Quality of knowledge about relevant data	[24,31]
41	Quality of the smart service	[26,29,30]
48	Success of newly developed services on the market	[30]

Table A4. Identified key factors for smart service data specification.

ID	Key Factor	Source
11	Degree of reusability of documented domain knowledge	[28]
20	Efficiency of smart service development methods	[30]
22	Efficiency of the application of domain knowledge	[14]
23	Extent of trial and error approach to identify relevant data	Expert interview
32	Proportion of the explorative approach in data analysis	[27]
33	Quality of collaboration between product and service development	[15,29,30]
35	Quality of data specification	Logical conclusion
39	Quality of smart service development methods	[15,26,29,30]
44	Relevance of sensors in the product	Logical conclusion

Appendix B

Table A5. Mapping of functions and SemDaServ solution elements.

		Solution Elements												
		Process			Artifacts					Tools and IT Systems				
ID	Function	Clarify Domain Knowledge Needs	Clarify Information Needs	Specify Data Needs	Smart Service Data Categories	Information Model	Data Model	Knowledge Carrier Matrix	Data Decision Matrix	Visualisation Tools	Modeling Tools	Text and Table Processing Tools	Data Science Tools	Enterprise IT
100	Systematically specify data relevant to the smart service and their sources in the SSS context	X	X	X	X	X	X	X	X	X	X	X	X	X
200	Describe data relevant to smart services		X	X	X	X	X	X	X	X	X	X	X	X
210	Systematically describe relevant data and their sources		X	X		X	X		X	X	X	X	X	X
211	Describe relevant data		X	X		X	X		X	X	X	X	X	
212	Systematically ensure quality of data specification			X							X	X		X
213	Describe sources of relevant data		X	X		X	X			X	X	X	X	X
220	Describe data categories of smart services		X	X	X	X		X		X	X		X	
212	Collect data categories of smart services		X		X	X		X		X	X		X	
222	Document data categories of smart services in an extensible way		X	X	X					X	X			
223	Provide a system for describing data categories for smart services		X		X	X				X	X			
300	Support collaboration between product and service development	X	X	X	X	X	X	X	X	X	X	X	X	X
310	Extend existing smart service development methods	X	X	X	X	X	X	X	X	X	X	X		
311	Describe interface of existing smart service development methods to data specification	X			X	X	X	X	X			X		
312	Describe smart service development methods that can be used for data specification	X	X	X		X	X		X			X		
313	Tailor data specification process to meet needs					X	X			X	X	X		
320	Apply domain knowledge efficiently	X	X			X		X		X	X	X	X	X
321	Describe domain knowledge required for data interpretation	X	X			X				X	X	X		
322	Identify relevant domain knowledge and its sources	X						X		X	X	X	X	X
323	Document relevant domain knowledge in a reusable way	X	X			X				X	X			

References

1. Waring, J.; Lindvall, C.; Umeton, R. Automated Machine Learning: Review of the State-of-the-Art and Opportunities for Healthcare. *Artif. Intell. Med.* **2020**, *104*, 101822. [CrossRef] [PubMed]
2. Rapidminer. *50 Ways to Impact Your Business with AI*; Technical Report; RapidMiner, Inc.: Boston, MA, USA, 2020.
3. Industrial, M.D.; Software, A. Machine Learning Einfach Gemacht. *SPS-MAGAZIN* **2020**, *7*, 32–35.

4. Chauhan, K.; Jani, S.; Thakkar, D.; Dave, R.; Bhatia, J.; Tanwar, S.; Obaidat, M.S. Automated Machine Learning: The New Wave of Machine Learning. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; pp. 205–212. [CrossRef]
5. Zöller, M.A.; Huber, M.F. Benchmark and Survey of Automated Machine Learning Frameworks. *arXiv* **2019**, arXiv:1904.12054.
6. Tuggener, L.; Amirian, M.; Rombach, K.; Lorwald, S.; Varlet, A.; Westermann, C.; Stadelmann, T. Automated Machine Learning in Practice: State of the Art and Recent Results. In Proceedings of the 2019 6th Swiss Conference on Data Science (SDS), Bern, Switzerland, 14 June 2019; pp. 31–36. [CrossRef]
7. Perrault, R.; Shoham, Y.; Brynjolfsson, E.; Clark, J.; Etchemendy, J.; Grosz Harvard, B.; Lyons, T.; Manyika, J.; Carlos Niebles, J.; Mishra, S.; et al. *Artificial Intelligence Index 2019 Annual Report*; Technical Report; Stanford University's Human-Centered Artificial Intelligence Institute (HAI): Stanford, CA, USA, 2019.
8. Elshawi, R.; Maher, M.; Sakr, S. Automated Machine Learning: State-of-The-Art and Open Challenges. *arXiv* **2019**, arXiv:1906.02287.
9. Chin, J. The Death of Data Scientists Will AutoML Replace Data Scientists? 2019. Available online: https://towardsdatascience.com/the-death-of-data-scientists-c243ae167701 (accessed on 8 April 2021).
10. Kozyrkov, C. Data Science… without Any Data?! 2020. Available online: https://towardsdatascience.com/data-science-without-any-data-6c1ae9509d92 (accessed on 8 April 2021).
11. Louis, P.; Russ, R. How to Develop Digital Products for Industrial Environments-The Data Science and Engineering Process in PLM. In Proceedings of the 9th International Conference on Data Science, Technology and Applications (DATA), Online, 7–9 July 2020. Available online: https://www.insticc.org/node/TechnicalProgram/data/2020/presentationDetails/99728 (accessed on 7 April 2021).
12. Marx, E.; Pauli, T.; Matzner, M.; Fielt, E. From Services to Smart Services: Can Service Engineering Methods Get Smarter as Well? In *WI2020 Zentrale Tracks*; GITO Verlag: Berlin, Germany, 2020; pp. 1067–1083. [CrossRef]
13. Tokarz, B.; Tokarz, B.; Fagundes, A.B.; Pereira, D.; Beuren, F.H. Product-Service Systems: A Literature Review on Assisting Development. *Int. J. Adv. Eng. Res. Sci.* **2020**, *7*, 41–51. [CrossRef]
14. Lee, J. *Industrial AI: Applications with Sustainable Performance*; Springer: Singapore, 2020. [CrossRef]
15. Anke, J. How to Tame the Tiger—Exploring the Means, Ends and Challenges in Smart Service Systems Engineering. In Proceedings of the Twenty-Eigtht European Conference on Information Systems (ECIS2020), Marrakech, Morocco, 15–17 June 2020; pp. 1–16.
16. Rosa, M.; Wang, W.M.; Stark, R.; Rozenfeld, H. A Concept Map to Support the Planning and Evaluation of Artifacts in the Initial Phases of PSS Design. *Res. Eng. Des.* **2021**. [CrossRef]
17. Shearer, C. The CRISP-DM Model: The New Blueprint for Data Mining. *J. Data Warehous.* **2000**, *5*, 13–22.
18. Feyyad, U. Data Mining and Knowledge Discovery: Making Sense out of Data. *IEEE Expert* **1996**, *11*, 20–25. [CrossRef]
19. SAS Institute Inc. SAS Help Center: Introduction to SEMMA. 2017. Available online: https://documentation.sas.com/doc/en/emref/14.3/n061bzurmej4j3n1jnj8bbjjm1a2.htm (accessed on 8 April 2021).
20. Azevedo, A.; Santos, M. KDD, SEMMA and CRISP-DM: A Parallel Overview. In Proceedings of the IADIS European Conference Data Mining, Amsterdam, The Netherlands, 24–26 July 2008.
21. Wang, W.M.; Preidel, M.; Fachbach, B.; Stark, R. Towards a Reference Model for Knowledge Driven Data Provision Processes. In *IFIP Advances in Information and Communication Technology*; Springer International Publishing: Basel, Switzerland, 2020; Volume 598, pp. 123–132. [CrossRef]
22. Roe, K.D.; Jawa, V.; Zhang, X.; Chute, C.G.; Epstein, J.A.; Matelsky, J.; Shpitser, I.; Taylor, C.O. Feature Engineering with Clinical Expert Knowledge: A Case Study Assessment of Machine Learning Model Complexity and Performance. *PLoS ONE* **2020**, *15*, e0231300. [CrossRef] [PubMed]
23. Blessing, L.T.; Chakrabarti, A. *DRM, a Design Research Methodology*; Springer: London, UK, 2009. [CrossRef]
24. Blei, D.M.; Smyth, P. Science and Data Science. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 8689–8692. [CrossRef]
25. Bundesministerium für Bildung und Forschung. Bekanntmachung der Richtlinie zur Förderung von Projekten zum Thema "Erzeugung von Synthetischen Daten für Künstliche Intelligenz". 2020. Available online: https://www.bmbf.de/foerderungen/bekanntmachung-3068.html (accessed on 8 April 2021).
26. Bullinger, H.J.; Scheer, A.W. Service Engineering—Entwicklung Und Gestaltung Innovativer Dienstleistungen. In *Service Engineering*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 3–18. [CrossRef]
27. Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI). Projekt Exdra–Idee, Übersicht, Ziele Und Ergebnisse. 2020. Available online: https://www.exdra.de/projekt/ (accessed on 8 April 2021).
28. Laroche, F.; Dhuieb, M.A.; Belkadi, F.; Bernard, A. Accessing Enterprise Knowledge: A Context-Based Approach. *CIRP Ann. Manuf. Technol.* **2016**, *65*, 189–192. [CrossRef]
29. Meyer, K.; Klingner, S.; Zinke, C. (Eds.) *Service Engineering*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2018. [CrossRef]
30. Richter, H.M.; Tschandl, M. Service Engineering–Neue Services Erfolgreich Gestalten Und Umsetzen. In *Dienstleistungen 4.0*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2017; pp. 157–184. [CrossRef]
31. Rogati, M. The AI Hierarchy of Needs | . 2017. Available online: https://hackernoon.com/the-ai-hierarchy-of-needs-18f111fcc007 (accessed on 8 April 2021).

32. Wellsandt, S.; Anke, J.; Thoben, K.D. Modellierung Der Lebenszyklen von Smart Services. In *Smart Service Engineering*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2017; pp. 233–256. [CrossRef]
33. Lünnemann, P.; Stark, R.; Wang, W.M.; Stark, R.; Manteca, P.I. Engineering Activities—Considering Value Creation from a Holistic Perspective. In Proceedings of the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira, Portugal, 27–29 June 2017; pp. 315–323. [CrossRef]
34. Hildebrand, K.; Gebauer, M.; Hinrichs, H.; Mielke, M. (Eds.) *Daten-Und Informationsqualität*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2018. [CrossRef]
35. Anke, J.; Poeppelbuss, J.; Alt, R. It Takes More than Two to Tango: Identifying Roles and Patterns in Multi-Actor Smart Service Innovation. *Schmalenbach Bus. Rev.* **2020**, *72*, 599–634. [CrossRef]
36. Poeppelbuss, J.; Durst, C. Smart Service Canvas–A Tool for Analyzing and Designing Smart Product-Service Systems. *Procedia CIRP* **2019**, *83*, 324–329. [CrossRef]
37. Exner, K.; Stark, R.; Kim, J.Y.; Stark, R. Data-Driven Business Model a Methodology to Develop Smart Services. In Proceedings of the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira, Portugal, 27–29 June 2017; pp. 146–154. [CrossRef]
38. Halstenberg, F.A.; Lindow, K.; Stark, R. Leveraging Circular Economy through a Methodology for Smart Service Systems Engineering. *Sustainability* **2019**, *11*, 3517. [CrossRef]
39. Müller, P.; Kebir, N.; Stark, R.; Blessing, L. PSS Layer Method–Application to Microenergy Systems. In *Introduction to Product/Service-System Design*; Springer: London, UK, 2009; pp. 3–30. [CrossRef]
40. VDI-Fachbereich Informationstechnik. Wissensmanagement im Ingenieurwesen–Grundlagen, Konzepte, Vorgehen. *VDI-Richtlinie* **2009**, *VDI 5610 Pt 1*, 1–28.
41. Exner, K.; Smolka, E.; Blüher, T.; Stark, R.; Blueher, T.; Stark, R. A Method to Design Smart Services Based on Information Categorization of Industrial Use Cases. *Procedia CIRP* **2019**, *83*, 77–82. [CrossRef]
42. Exner, K.; Preidel, M.; Gogineni, S.; Nickel, J.; Stark, R. Digitaler Zwilling Für Smart Services. *ProduktDaten J.* **2019**, *2*, 39–42.
43. Runkler, T.A. *Data Mining*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2015; Volume 53. [CrossRef]
44. English, L.P. *Improving Data Warehouse and Business Information Quality: Methods for Reducing Costs and Increasing Profits*; John Wiley & Sons, Inc.: New York, NY, USA, 1999.
45. Tavana, M.; Hajipour, V.; Oveisi, S. IoT-Based Enterprise Resource Planning: Challenges, Open Issues, Applications, Architecture, and Future Research Directions. *Internet Things* **2020**, *11*, 100262. [CrossRef]
46. Object Management Group. OMG Systems Modeling Language (OMG SysML™), Version 1.6. 2019. Available online: https://www.omg.org/spec/SysML/1.6/ (accessed on 10 April 2021).