Vehicle Information Systems in the Internet of Vehicle Era – A Pattern Language to guide Manufacturers to gather, manage, store and analyze Vehicle Status Data

Dissertation

zur Erlangung des Doktorgrades

Dr. rer. pol.

der Fakultät für Wirtschaftswissenschaften

der Universität Duisburg-Essen

vorgelegt
von

Niklas Brandt

aus

Hannover, Deutschland

Betreuer:

Prof. Dr. Frederik Ahlemann

Lehrstuhl für Wirtschaftsinformatik und Strategisches IT-Management

Essen, 07.10.2021

Gutachterinnen und Gutachter

Prof. Dr. Frederik Ahlemann

Prof. Dr. Stefan Eicker

Tag der mündlichen Prüfung:

11.07.2022

DOI: 10.17185/duepublico/77031
URN: urn:nbn:de:hbz:465-20221027-083112-7

The results, opinions, and conclusions expressed in this thesis are not necessarily those of Volkswagen Aktiengesellschaft.

Preface

For some years, the automotive industry has faced more disruptive changes than ever. The transitions toward e-mobility, autonomous driving, and ubiquitous connectivity require automakers to manage multiple significant innovations at the same time. In the best case, these innovations offer automakers a chance to gain a competitive advantage. On the other hand, it constitutes a risk for their future success. As an employee of an automotive original equipment manufacturer, I have the chance to be a part of these innovations and support the successful transformation of my employer. In 2018, I got the opportunity to start a dissertation in one of the innovation fields, i.e., ubiquitous connectivity, in parallel to my employment at the automaker. It quickly became clear that I must take this opportunity and start academic research on internet-connected vehicles. Conducting academic research and writing a dissertation alone already requires support and patience from others. This demand for support and patience is even greater when employed elsewhere in parallel. It is not possible to thank all adequately. Nevertheless, I would like to thank those who supported me the most during my thesis. This dissertation would not have been possible without these.

First, my special thanks go to Prof. Dr. Frederik Ahlemann, who allowed me to conduct my research at his chair. I am thankful for his immense work in guiding me and giving constructive feedback to all my essays. I appreciate his engagement even more, as I have rarely been present at the chair due to my parallel employment. In addition, I would like to thank Prof. Dr. Stefan Eicker for accepting the role as a second supervisor for this thesis and for providing me with constructive feedback throughout my research.

Second, I thank all my colleagues at the chair that discussed my overall research design frequently and always shared their constructive feedback for specific research projects. I appreciate these discussions as they helped me to rethink bad ideas. For their support in individual essays, I explicitly thank Stefan Reining, Kevin Rehring, and Helge Alsdorf.

Third, I want to thank all my Volkswagen colleagues for supporting me for the last three years. I want to mention my manager, who gave me maximum flexibility and helped me while conducting research in my work environment.

Last, I am deeply grateful for having a supportive family and girlfriend. Even though I had long days and needed to work in my spare time, they continuously motivated me. Moreover, they always believed in me even when there were setbacks.

Braunschweig, October 2021 Niklas Brandt

Table of Contents

1

INTRODUCTION TO “VEHICLE INFORMATION SYSTEMS IN THE INTERNET OF VEHICLE ERA – A PATTERN LANGUAGE TO GUIDE MANUFACTURERS TO GATHER, MANAGE, STORE AND ANALYZE VEHICLE STATUS DATA“

Brandt, Niklas, University of Duisburg-Essen, Essen, Germany

Abstract

This chapter introduces the dissertation and structures the following essays. It explains the overall gap in research and the problem in practice that both motivate the research of this thesis. In addition, this introduction explains relevant key concepts that are continually used in the essays of this dissertation. It further outlines the research process and the methods that are used to answer the overall research question. Based on this, the research questions of the separated essays are presented and the contribution of each essay to answer the overall research question is discussed. In the end, this chapter also summarizes the main findings of each research essay and how these contribute to solve the research problem.

Keywords: *Introduction, motivation, research domain, research design, research process*

Table of contents

List of Figures

1 Introduction

Prior to industrialization, manufacturing physical products involved just a few stakeholders. Due to the simplistic nature of most products back then, these stakeholders were able to manage all product related information along the life of the products without assistance (Terzi et al., 2010, p. 362). Nowadays, complex products, short innovation cycles and new product related services require organizations to systematically manage product related data that is spread over multiple stakeholders along the individual product lifecycles (Kiritsis, 2011, p. 481). A product centric way to do so is the approach of Product Lifecycle Management (PLM) (Stark, 2018, p. 13). PLM is not a set of tools that are used store and analyze lifecycle related data (Terzi et al., 2010, p. 363). Rather, PLM is a strategic approach (Staisch et al., 2012, p. 1) to create, manage, maintain, and provide product related information (Myllärniemi et al., 2009, p. 9). Per definition, PLM covers the whole product lifecycle. Nevertheless, the amount of available product related information traditionally drops, after the product is sold to the customer (Holler et al., 2016b, p. 7). Therefore, focus of PLM in practice is on the design and the manufacturing phases, where the manufacturer has physical access to the product and can gather all data directly (Holler et al., 2016b, p. 7). As a result, product related tasks that require product information from the use phase often rely on hypothetical data and assumptions (Zhang et al., 2017, p. 230). By integrating sensors, processors and communication technologies into physical products, products become able to provide this traditionally missing product-related information from the use phase (Bajyere et al., 2020, p. 565). By systematically analyzing these data with the PLM approach, many benefits for manufacturers, customers, and third parties arise (Bilgeri et al., 2019, p. 191). Experts expect high benefits, especially for technological products such as cars that have plenty of sensors already embedded (Stocker et al., 2017, p. 126). These high expectations in the automotive industry led to forecasts that manufacturers worldwide will equip one-third of all manufactured vehicles in 2023 with internet connectivity units (Cäsar et al., 2019, p. 4) and enable vehicles to participate in the internet of things (IoT). In this IoT, vehicles transfer detailed status data from embedded sensors to a product external information system (IS) along their individual lives.

Like the growing interest of practitioners in the IoT, researchers' interest rapidly increased in the last few years. Although there is such a high interest among practitioners and researchers in internet-connected vehicles, it remains a challenge for manufacturers to systematically gather, manage, store, and analyze these data from individual products along their use phase (Stocker et al., 2017, p. 126) (Holler et al., 2016a, p. 486–487) (Brandt and Ahlemann, 2020, p. 12). To overcome these challenges, guidelines for practitioners to integrate the vehicle status data from the use phase into PLM are needed (Zhang et al., 2017, p. 229). By then, vehicle manufacturers can design new or extend their existing IS to gather, manage, store, and analyze these new data from individual products (Stocker et al., 2017, p. 128) systematically and holistically. As the integration of IoT data from internet-connected products involves domains such as business, IT, social science, and technology, IS researchers are well prepared to make substantial contributions (Baiyere et al., 2020, p. 567). Furthermore, IS Researchers have a long history of developing conceptual models that guide others in the design of IS (Legner et al., 2020, p. 735-736). Nevertheless, IS researchers paid little attention to gathering, managing, storing, and analyzing status data from connected products during the use phase and the integration into PLM in the past (Marabelli et al., 2017, p. 352).

I split this thesis into six essays to develop an artifact that guides vehicle manufacturers and other stakeholders to design such IS. In the first essay, i.e., this one, I motivate the research, introduce key concepts, and structure the following essays. The main emphasis is on essays two to five that document the design process of the central research artifact, i.e., the pattern language. In essays two and three, I derive the demand for and the requirements for a pattern language for vehicle IS in the era of the IoT from existing research and practice. I design and evaluate the research artifact in essay four and five. In the last essay of this dissertation, I discuss the results and limitations of the pattern language. In addition, I include an appendix to this dissertation that discusses the artifact type that this thesis develops, i.e., a pattern language. The essay in the appendix is not a part of this dissertation and is just added for faster access.

This introductory essay is structured as follows. After motivating the research of this thesis and the overall research question in this chapter, key concepts are explained in chapter 2. Subsequently, the overarching Design Science Research (DSR) process is outlined, and the research questions are stated in chapter 3. Last, chapter 4 summarizes the results of the research essays.

2 Research Background

2.1 Introduction to the Vehicle Information System

Before the Vehicle Information Systems (VIS) are introduced, key terms are defined shortly, and their use in this dissertation is clarified. The term software is used for any kind of computer program. The term application is used more specifically for application software. An IS expands this specific technology focus of software and applications towards this technology in a socio-technical system within its context of use.

Sensors are devices that can transform events into measurable electrical signals. When such sensors are embedded into individual physical objects, they can transform real-world events that occur at the object into machine-readable data (Stocker et al., 2017, p. 125). By sharing these data with communication technology, the physical objects can interact with other objects and share information about their status along their whole and individual lifecycles – from manufacturing over the use phase to recycling or disposal (Baiyere et al., 2020, p. 565). As objects gain the capability to interact with other objects, they build an IoT (Xia et al., 2012, p. 1101). Besides, objects in the IoT can share information about their status with other IS that could store and analyze these data (Wortmann and Flüchter, 2015, p. 223). By capturing the data about the status of individual products throughout their lives, a digital shadow of unique products can be created (Bauernhansl et al., 2016, p. 23). This digital shadow is a virtual representation of individual physical objects. By analyzing the data and performing actions on the physical object in the real world, a digital twin arises that virtually represents the status of a physical object and interacts with it (Liu et al., 2020, p. 4). Experts project that the number of objects with embedded IoT technology will increase significantly in the following years (Shim et al., 2017, p. 2), which brings manufacturers and others more and more transparency about the objects and their usage in the real world. This also explains why the interest of researchers (Liu et al., 2020, p. 1) and practitioners (Baiyere et al., 2020, p. 565) in the IoT and the subsequent analysis of the digital shadows and twins increased in the last years.

According to Bilgeri et al. (2019, p. 191), data from IoT objects can be analyzed to improve the product creation or enhance product functionalities. Shipping containers that share information about their position for more efficient supply chain management are examples of an improved product creation process (Baiyere et al., 2020, p. 565). Wind turbines that share real-time status information to improve their maintenance efficiency are an example of enhanced product functionality (Bilgeri et al., 2019, p. 192). Previous researchers put immense effort into improving the production processes by analyzing existing data from the machines (Liu et al., 2020, p. 1). Experts expect many benefits for industrial companies through applying the IoT in an industrial context. Therefore, it is already called the fourth industrial revolution or Industry 4.0 (Gilchrist, 2014, p. 223). The automotive industry has an excellent basis to bring the concept of the IoT from manufacturing to their products, as modern vehicles already capture detailed data about their status and environment (Kaiser et al., 2018, p. 1). These data are created in vehicles to enable driving and comfort functionalities. When vehicles can share these data over the internet, organizations could gather and subsequently analyze the data captured anyway (Lee et al., 2016, p. 1). Hence, experts expect that the automotive industry will greatly benefit from the IoT for manufacturing process improvements and enhanced product functionalities (Kaiser et al., 2019, p. 7). When multiple vehicles communicate and share data with an external IS, an Internet of Vehicles (IoV) as a sub-group of the IoT is created (Yang et al., 2014, p. 2). In an IoV, manufacturers could analyze the vehicles' usage to improve the design of the products (Holler et al., 2016a, p. 486). Manufacturers could proactively detect imminent faults and react accordingly (Främling et al., 2013, p. 798). Furthermore, they could offer customers new services based on the vehicle status, like an insurance based on the driving style (Kubler et al., 2015, p. 94) or predictive maintenance services (Tao et al., 2016, p. 35). To realize these benefits, manufacturers need a suitable IS within each vehicle that gathers the in-vehicle data along the vehicle lifetime, and a suitable central IS that manages, stores, and analyzes the IoV data from the numerous vehicles. Kaiser et al. (2018, p. 3) call an IS to acquire, preprocess, analyze, store, and use data that vehicles generate along their lives a VIS. A VIS might exceed system boundaries when data is gathered and preprocessed inside a vehicle and stored and analyzed outside of the vehicle in a central IS. The capabilities of such a VIS build the basis for manufacturers to improve the product creation process and enhance the product functionalities (Bilgeri et al., 2019, p. 191).

2.2 Pattern Languages in Information Systems Research

Researchers frequently conduct research to develop an artifact that solves either an unsolved problem or an artifact that solves a problem in a new and innovative way (Hevner et al., 2004, p. 81). Researchers in the IS domain commonly refers to the process of developing such artifacts as Design Science Research (DSR) (Peffers et al., 2006, p. 86). Following the DSR process, researchers might develop a model as a design artifact. Such an artifact is a reference for further implementations of good solutions to recurring problems. In the IS domain, researchers often call their DSR artifacts "reference models" (Junginger, 2009, p. 472). Reference models contain descriptive and prescriptive knowledge that is aggregated over time by researchers and practitioners to guide others in designing good solutions to common problems (Legner et al., 2020, p. 754). Researchers in other domains, such as architecture (Alexander, 1977, p. 10) or computer science (Gamma et al., 1997, p. 394) and some information systems researchers (Sprenger and Mettler, 2016, p. 8) also call artifacts with a similar aim as a pattern language (Winter, 2009, p. 468). Although the artifacts are named differently, there is no common understanding about the differences between the differently named artifacts – if any. In a previous research project that was already rated in another study and added to the appendix of this thesis for faster access, we investigated the characteristics of existing artifacts of type models that are called a pattern language. The following results are based on the previous study. We found no characteristics that all existing pattern languages have in common and differentiate them from artifacts called reference models. Instead, we identified attributes that most artifacts called pattern languages have in common, and that can be defined as characteristics of pattern languages. Pattern languages contain the knowledge to design solutions to common problems. They consist of multiple patterns that solve individual sub-problems of the overall problem (Gamma et al., 1997, p. 10). Their main aim is to provide practical instructions on designing a solution (Alexander et al., 1977). These instructions view a problem from a single perspective and one level of abstraction (Noble, 1998, p. 4). Typically, the individual patterns of a pattern language are derived from existing solutions (Voit et al., 2020, p. 330). This means that the artifact is not actively designed based on recurring requirements. Instead, researchers identify existing and reasonable solutions in practice and abstract them by defining patterns (Moreira and Paiva, 2009, p. 3). In addition, pattern languages might define multiple complementary patterns for the same problem (DeHon et al., 2004, p. 7). These alternative patterns ease the adoption in different contexts (German and Cowan, 2000, p. 4).

2.3 Research Gap and Research Problem

The automotive industry recently transformed their traditionally hardware-oriented business model into a business model based on data-driven services (Kaiser et al., 2019, p. 7). Therefore, systematically managing and analyzing IoT data from individual vehicles is a key to future success. Nevertheless, it is a hurdle for manufacturers of physical goods to utilize their knowledge and create value with digital goods (Baiyere et al., 2020, p. 575). They do not have the capabilities to gather and store data from connected products throughout their lives (Holler et al., 2016a, p. 485). Furthermore, even analyzing existing IoT data for innovative applications is challenging for many manufacturers (Stocker et al., 2017, p. 126). Due to these missing technological capabilities of vehicle manufacturers (Holler et al., 2016b, p. 7), automotive experts expect specialized service providers will take over the promising field of revenue (Kaiser et al., 2019, p. 8). In such a scenario, manufacturers could neither digitize their hardware-oriented business model nor could they leverage the product data for product improvements. Nevertheless, with their control over the vehicle architectures, vehicle manufacturers have the potential to control over the vehicle data and could prevent others from accessing it (Kaiser et al., 2019, p. 8). Due to a lack of research and missing existing solutions in practice, Kaiser et al. (2019, p. 9) call on researchers to propose design guidelines that could enable researchers and practitioners to build robust vehicle IS. Baiyere et al. (2020, p. 564) suggest that IS researchers should focus on existing IoT research gaps, as IS researchers practice solving interdisciplinary social, business, and technical problems. Although the IS community seems well prepared to contribute to IoT research gaps, they devoted little attention to the IoT (Marabelli et al., 2017, p. 351) and related IS.

Research Problem: There is no suitable guidance for vehicle manufacturers to design a holistic information system to gather, manage, store, and analyze data from individual internet-connected vehicles along their lives.

I solve the research problem by conducting design as a science (Simon, 1996, p. 111). The design artifact that I develop is a pattern language that guides researchers and practitioners in designing a VIS for the IoV. To do so, I break down the research problem into four essays that answer individual research questions. The essays and the underlying research questions are presented in the following chapters.

3 Overall Research Paradigm

In this thesis, I develop a digital artifact, i.e., a pattern language that solves an unsolved practical problem in a new and innovative way. By creating such an artifact, I perform design as a science by solving an unsolved problem with a creative solution (Gregor and Hevner, 2013, p. 347). The fact that I develop a solution to a class of problems differentiates my research from routine design (Winter, 2008, p. 471). Moreover, routine design provides known solutions to problems solved (Gregor and Hevner, 2013, p. 347). While scientific design as a methodology has a long history in other domains like engineering, it is relatively new but increasingly popular in IS research (Peffers et al., 2007, p. 47). To systematically develop the artifact, I apply the design science research (DSR) guidelines as the overall development process. DSR is ideally suited to develop an innovative artifact to solve a human problem and, thereby, create fundamental knowledge (Hevner and Chatterjee, 2010, p. 5).

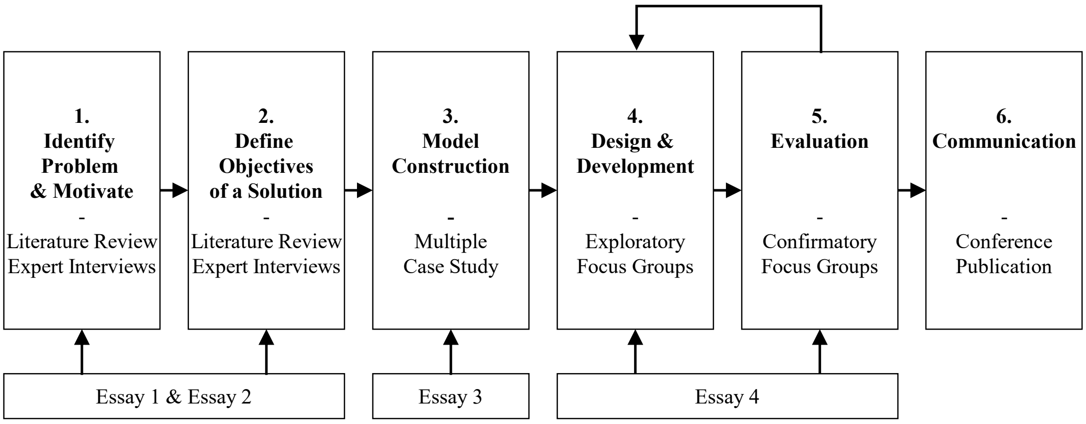


Figure 3-1 Combination of the DSR process and the process model

The process that guides the development of this thesis research artifact is shown in Figure 3-1. It is based on the well-acknowledged Design Science Research Methodology (DSRM) from Peffers et al. (2007, p. 54) and the process model for an empirically grounded reference model construction from Ahlemann and Gastl (2007, p. 12). The DSRM builds the basis (Peffers et al., 2007, p. 54) of the process as it was already successfully applied for similar projects. Nevertheless, there is a need to partially adopt the process model (Ahlemann and Gastl, 2007, p. 12). I do so by creating an initial model based on existing knowledge before the iterative design and evaluation phases start, as proposed by the process model. I do so as I see the need to systematically aggregate existing knowledge and use the initial model to have a foundation to build on in the design and evaluation phases. This need for an initial model is generally useful but required for our data collection method, which is design-oriented focus groups. The focus groups use the initial model to unveil patterns for each element of the initial model. Hence, it serves as a structure for the focus groups. Second, I adapt from the process model that there is no loop from the evaluation phase to the object definition phase. Changing the objectives would lead to the previous design and evaluation iterations based on different knowledge. This would limit the value of the earlier iterations. Hence, I define the objectives upfront with no re-definition after the initial model is constructed. In case of new goals or changes to existing objectives, a new initial model will be built, and the design and evaluation of the artifact will be initiated again.

Title	**Internet of Things in Product Lifecycle Management – A Review on Value Creation through Product Status Data**
Publication	Proceedings of the 27th American Conference on Information Systems (AMCIS 2021)
Method	Literature review
Co-authors	Frederik Ahlemann, Kevin Rehring, Stefan Reining
Own contribution	Research design, data collection, data analysis, essay development

Title	**How Do You Drive? Analyzing Vehicle Sensor Data in Product Lifecycle Management Systems**
Publication	Proceedings of the 28th European Conference on Information Systems (ECIS 2020)
Method	Expert interviews
Co-authors	Frederik Ahlemann, Stefan Reining
Own contribution	Research design, data collection and coding, data analysis, theorizing, essay development

Title	**Structure of a Pattern Language to Integrate Vehicle Sensor Data into Manufacturers' Information Systems**
Publication	-
Method	Multiple case study
Co-authors	Frederik Ahlemann
Own contribution	Research design, data collection, data analysis, design pattern language structure, essay development

Title	**Pattern Language to Gather, Manage, Store and Analyze Vehicle Sensor Data**
Publication	-
Method	Focus groups
Co-authors	-
Own contribution	Research design, data collection, data analysis, derive and validate design patterns, essay development

Figure 3-2 Summary of the essays of this dissertation

In total, this cumulative dissertation consists of four essays. Figure 3-2 summarizes the essays, their publication, the methods used, the co-authors and my contribution. Two of those essays have already been published, while the other two are currently preparing for publication.

3.1 Literature Review

I started the research project with a systematic literature review. The review identified and motivated the problem and defined the objectives of a solution (Peffers et al., 2007, p. 54). It supported the first two phases of my DSR process in Figure 3-1. A systematic literature review is ideally suited as an initial method to gather existing knowledge in a concept-centric approach (Webster and Watson, 2002, p. 16) when the research is widespread over different sources (Rowley and Slack, 2004, p. 31). The review of existing literature provides an overview of the domain of research and explains what has been solved already (Knopf, 2006, p. 127). Subsequently, it supports identifying unanswered research questions for this thesis (Rowley and Slack, 2004, p. 32). The review results are summarized in a topic map that makes propositions about barriers, benefits, and applications for IoT data and contextualizes them. Furthermore, the review identified that existing reference models lack the guidance of manufacturers to design IS to systematically gather, manage, store, and analyze IoT data from their products. Hence, the review motivates the artifact development.

3.2 Expert Interview

Besides a literature review, I conducted semi-structured interviews with pre-defined but open-ended questions (Myers and Newman, 2007, p. 3) to support the first two phases of the research process. Interviews are one of the most critical qualitative research methods for gathering data (Myers and Newman, 2007, p. 2). I, as a researcher, acted as an interviewer who gained firsthand data from the interviewed experts (Gillham, 2000, p. 1). Thereby, the results of this study are co-constructions between me as a researcher and the interviewees (Mann, 2016, p. 50). I deal with the critiques of researchers (Mayring, 2001, p. 2) on non-transparent and non-generalizable results by following the guidelines from Myers and Newman (2007, p. 16) for good qualitative research. With the qualitative interviews, I investigated the influencing factors, i.e., positive and negative factors, on the decision of automakers to realize the vehicle digital twin. Ultimately, I conceptualized the factors based on the Technology-Organization-Environment Framework (Brandt and Ahlemann, 2020, p. 10).

3.3 Multiple Case Study

For the construction of my initial model, I considered expert interviews, standards, a literature review, and the domain knowledge of the researchers for data collection (Ahlemann and Gastl, 2007, p. 24-25). With a multiple case study, I investigated numerous existing but separated cases that are somehow similar (Stake, 2006, p. 1). Case study research tends to be more qualitative than quantitative (Starman, 2013, p. 30) and can be used to learn how cases, i.e., an observed phenomenon, work (Gerring, 2007, p. 19). The cases that I observed for constructing the initial model are existing reference models, commercially offered software solutions, and insights into the automotive industry. I analyzed 22 cases to construct the initial model that builds the basis for the central research artifact, i.e., the pattern language.

3.4 Focus Groups

I conducted design-oriented focus groups to develop and evaluate the patterns of the pattern language. Traditionally, focus groups are conducted to study cross-cultural and social factors (Kitzinger, 1995, p. 299-300). Tremblay et al. (2010, p. 602) were one of the first who adapted this research methodology to design artifacts with exploratory (EFG) and confirmatory focus groups (CFG). Such design-oriented EFG and CFG are ideally suited for my interdisciplinary research, as the roots of this method for data collection deal with diverse stakeholders (Stocker et al., 2017, p. 128). Furthermore, researchers have already proven that EFG and CFG are well suited to develop and validate a pattern language (Sprenger and Mettler, 2016, p. 7). Based on this research methodology, I developed 38 patterns and assessed their suitability for internet-connected vehicles to solve the initial research problem.

4 Summary of the Research Essays

This cumulative dissertation combines four self-contained research essays with individual objectives and research methods. Although all essays are self-contained and stand-alone, they are guided by the overall research process in Figure 3-1. The main contribution of this thesis are the research essays, which are two to five. Essay one serves as an introduction, and essay six as a conclusion and summary. Essay seven is attached in the appendix to ease access to it due to the relevance of this essay for this thesis. As this essay was already graded in another academic course, it is not part of this dissertation.

	Title	Objective	Method
Paper 1	Introduction	Introduce the research questions & structure the thesis	-
Paper 2	Internet of Things in Product Lifecycle Management – A Re-view on Value Creation through Product Status Data	Aggregate existing research results	Systematic Literature Review
Paper 3	How Do You Drive? Analyzing Vehicle Sensor Data in Product Lifecycle Management Systems	Unveil influences on the manufacturers' decision to adapt the vehicle digital twin.	Qualitative Expert Interviews
Paper 4	Structure of a Pattern Language to Integrate Vehicle Sensor Data into Manufacturers' Information Systems	Design the structure of a pattern language that guides manufacturers to gather, store and manage IoT data	Multiple Case Study
Paper 5	Pattern Language to Gather, Manage, Store and Analyze Vehi-cle Sensor Data	Unveil and assess patterns to gather, store and manage IoT data	Design-oriented Focus Groups
Paper 6	Conclusion	Summarize the thesis and outline further research	-
Paper 7	Delimitation of Pattern Languages and Reference Models as Design Artifacts	Research on the differences and similarities of design artifacts that are called reference models and pattern languages	Systematic Literature Review

Figure 4-1 Essays of this dissertation

Figure 4-1 depicts the process that puts the essays in chronological and content-related order. The process summarizes the main objectives of the individual essays and gives an overview of the applied methods. Essay one serves as an introduction. It briefly introduces and motivates the research, explains the key concepts and derives the central research problem, as already stated in chapter 2.3. Furthermore, it structures the dissertation by giving an overview of the following essays. Essays two to five contain the primary research and systematically develop the pattern language. Therefore, in essay two, previous research was aggregated in a content-related approach. In addition, essay three integrates knowledge of experts. Essay four develops an initial model, i.e., a pattern language structure based on this knowledge. For each element of the pattern language structure, essay five describes one to many patterns that build the pattern language, which is the final artifact. This dissertation concludes with essay six. This last essay summarizes the main achievements of the individual essays and the overall dissertation. Furthermore, it presents the limitations of this thesis and suggests further research. The essay in the appendix is not part of this dissertation as it was already graded in another context. It is included to ease access to it.

Research Question 1: Which PLM applications could benefit from information about individual product in-stances and what hinders organizations to realize them? What is the limitation of existing PLM IS that were designed to support the lifelong gathering and analysis of product instance data?

Before a DSR artifact can be designed, existing research results need to be analyzed to identify the existing designs and unveil their limitations – if any. A systematic literature review in the first research essay aggregates existing research results in a concept-centric approach (Webster and Watson, 2002, p. 16). Therefore, peer-reviewed publications from major scientific databases were identified with a keyword search (Vom Brocke et al., 2015, p. 210). In addition, a back- and forward search was conducted to ensure that all significant publications were included (Webster and Watson, 2002, p. 16). The objective of the review was two-folded. The first objective was identifying applications, drivers, and barriers to gather IoT data and contextualizing them in a topic map. This topic map depicts the reasons for gathering IoT data and the general limitations of doing so. In addition, it contextualizes the factors by making propositions. The second objective was to create an overview of existing models that guide practitioners in designing an IS to gather IoT data from product instances and unveil their limitations. An inductive framework guided the analysis of the publications in the literature pool to ensure a systematic process (Mayring, 2010, p. 603). The results of the analysis of the body of knowledge of the body of knowledge analysis were used later in the subsequent essays to identify and motivate the overall research as welforl as to define the requirements to a solution, as depicted in Figure 3-1.

Research Question 2: Which factors influence the decision of vehicle manufacturers to adopt the concept of the vehicle digital twin on a firm level?

Based on the literature review results, an in-depth qualitative interview study was conducted. The results were analyzed to develop a theory for explaining (Gregor, 2006, p. 624-625). This theory is designed to answer, why automakers want to gather IoT data from internet-connected vehicles. In total, 30 experts from one car and one truck manufacturer participated in the interview research. The interviewees' answers to the open-ended questions (Yin, 2011, p. 135) were systematically analyzed with qualitative content analysis by Mayring (2000, p. 213) to identify the benefits and challenges to gather, store and analyze vehicle IoT data from the manufacturers' perspective. By abstracting the benefits and barriers identified in the interviews, fifteen factors that influence the decision of vehicle manufacturers to adopt the vehicle digital twin or not were identified.

Research Question 3: What are the elements of a pattern language that guides the design of an IS to gather, manage, store and analyze sensor data from individual vehicles?

After the limitations of existing models were identified by answering research question one and the requirements for a solution were identified by answering research question two, an initial model needed to be constructed according to the DSR process in Figure 3-1. This initial model aggregates existing research (Ahlemann and Gastl, 2007, p. 24-25), i.e., the previous research essays and existing models using publicly available documentation of commercially offered IS. The initial model comprises 21 cases, which are different commercially offered software, research artifacts, and commercially used software in two organizations. The cases are analyzed and aggregated with a classification scheme (Fettke and Loos, 2003, p. 43 - 49) to derive a pattern language structure (Alexander et al., 1977, p. 35). The pattern language structure contains thirteen elements. Each element describes a problem that a component of an IS needs to be solved to enable vehicle manufacturers to gather, manage, store, and analyze vehicle status data. For each element of the initial pattern language structure, one to many best practices can be identified in subsequent steps.

Research Question 4: Which are the patterns, i.e., best practices, to design an IS to gather, manage, store, and analyze sensor signal based status data from individual internet-connected vehicles?

Patterns, i.e., best practices, needed to be identified and evaluated in iterative steps based on the pattern language structure (March and Smith, 1995, p. 254). To identify such patterns in practice, design-oriented focus groups with multiple experts at a time were conducted (Tremblay et al., 2010, p. 602). With exploratory focus groups, patterns were unveiled from experts' experiences. They were conducted with experts that work in industries that already gather, manage, store, and analyze IoT data. With subsequent confirmatory focus groups, the ability to adopt the patterns to the automotive industry were investigated with experts from a vehicle manufacturer. For each element of the pattern language structure, multiple patterns were identified where each pattern might be helpful in a different context.

5 References

Ahlemann, F. & Gastl, H. (2007). “Process model for an Empirically Grounded Reference Model Construction.” In: *Reference Modeling for Business Systems Analysis*. Idea Group Publishing, Hershey, p. 77-97.

Alexander, C., Ishikawa, S. & Silverstein, M. (1977). „A Pattern Language. Town – Buildings - Construction.” Oxford University Press. New York.

Alharahsheh, H. and Pius, A. (2020). “A Review of key paradigms: positivism VS interpretivism.” In: *Global Academic Journal of Humanities and Social Sciences*, 2(3), p. 39-43.

Baiyere, A., Salmela, H. and Tapanainen, T. (2020). ”Digital Transformation and the New Logics of Business Process Management.” In: *European Journal of Information Systems*, p. 1-48.

Bauernhansl, T.,Krüger, J., Reinhart, G. and Schuh, G. (2016). „WGP-Standpunkt Industrie 4.0.“ Wissenschaftliche Gesellschaft für Produktionstechnik WGP e. v.

Becker, J. and Niehaves, B. (2007). “Epistemological perspectives on IS research: a framework for analyzing and systematizing epistemological assumptions.” In: *Info Systems Journal*, 17, p. 197-214.

Bilgeri, D., Fleisch, E., Gebauer, H. and Wortmann, F. (2019). „Driving Process Innovation with IoT Field Data. “In: *MIS Quarterly Executive*, p. 191-207.

Brandt, N. and Ahlemann, F. (2020). "How Do You Drive? Analyzing Vehicle Sensor Data in Product Lifecycle Management Systems.” In: *Proceedings of the 28th European Conference on Information Systems (ECIS).”* An Online AIS Conference, June 15-17, p. 4.

Cäsar, M., Mehl, R., Tschödrich, S., Pauli, M., Ruther, B., Wendt, R., Blöchl, D., Stotz, A., Völkoi, M., Song, W. and Dupont, W. (2019). „Connected Vehicle Trend Radar - The drive to add value: lessons from China.“ Capgemini.

DeHon, A., Adams, J., DeLorimier, M., Kapre, N., Matsuda, Y., Naeimi, H., Vanier M., and Wrighton M. (2004). “Design Patterns for Reconfigurable Computing.” In: *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, p. 13-23.

Fettke, P. and Loos P. (2003). “Classification of reference models: a methodology and its application.” In: *Information Systems and e-Business Management*. Springer-Verlag, p. 35-53.

Främling, K, Holmström, J., Loukkola, J., Nyman, J. & Kaustell, A. (2013). „Sustainable PLM through Intelligent Products.“ In: *Engineering Applications of Artificial Intelligence* Volume 26, p. 789-799.

Gamma, E., Helm, R., Ralph, E. and Vlissides, J. (1997). ”Design Patterns Elements of Reusable Object-Oriented Software.” Prentice Hall; 1st ed.

German, D., and Cowan D. (2000). “Towards a unified catalog of hypermedia design patterns.” *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 1-8.

Gerring, J. (2007). “Case Study Research: Principles and Practices.” Cambridge University Press.

Gilchrist, A. (2016). “Industry 4.0: The Industrial Internet of Thing.” Apress, Thailand.

Gillham, B. (2000). “The research interviews.” Continuum, London.

Gregor, S. & Hevner, A. (2013). “Positioning and Presenting Design Science Research for Maximum Impact.” In: *MIS Quarterly*, Volume 37, Issue 2, p. 337-355.

Gregor, S. (2006). “The Nature of Theory in Information Systems, In: *MIS Quarterly* (30:3), p. 611-642.

Guba, E. and Lincoln, Y. (1994). “Competing Paradigms in Qualitative Research.” In: *N. K. Denzin & Y. S. Lincoln (Eds.), Handbook of qualitative research*, p. 105-117.

Hevner, A. and Chatterjee, S. (2010). “Design Research in Information Systems.” Springer Science+Business Media.

Hevner, A., March, S. and Park, J. (2004). “Design Science in Information Systems Research.” In: *MIS Quarterly* Vol. 28 No. 1, p. 75-105.

Holler, M., Stoeckli, E., Uebernickel, F. & Brenner, W. (2016a). „Towards Understanding closed-loop PLM: The Role of Product Usage Data for Product Development enabled by intelligent Properties.” In: *BLED Proceedings*, p. 479-491.

Holler, M., Übernickel, F. and Walter, B. (2016b). „Challenges in product lifecycle management: evidence from the automotive supply industry.“ In: *27th Australasian Conference on Information Systems*, p. 1-12.

Junginger, S. and Moser, C. (2009). “Patterns in Business and Information Systems Engineering.” In: *Patterns in Business and Information Systems Engineering*. 471-473.

Kaiser, C., Stocker, A., Festl, A., Lechner, G. and Fellmann, M. (2018). „A Research Agenda for Vehicle Information Systems.“ In: *Proceedings of the Twenty-Sixth European Conference on Information Systems*, p. 1-12.

Kaiser, C., Stocker, A. and Fellmann, M. (2019). „Understanding Data-driven Service Ecosystems in the Automotive Domain.“ In: *Proceedings of the Twenty-fifth Americas Conference on Information Systems,* p. 1-10.

Kiritsis, D. (2011). “Closed-loop PLM for intelligent products in the era of the Internet of things.” In: *Computer Aided Design* (43), p. 479-501.

Kitzinger, J. (1995). „Introducing focus groups.“ In: *BMJ Clinical Research*, Volume 311, p. 299-302.

Knopf, J. (2006). “Doing a Literature Review.” In: *Political Science Journal*, Volume 39, p. 127-132.

Kubler, S., Främling, K. & Derigent, W. (2015). „P2P Data synchronization for product lifecycle management.” Computers in Industries (66), p. 82-98.

Lee, E., Gerla, M., Pau, G., Lee, U., and Lim, J. (2016). „Internet of Vehicles: From intelligent grid to autonomous cars and vehicular fogs.“ In: *International Journal of Distributed Sensor Networks*, Vol. 12(9), p. 1-14.

Legner, C., Pentek, T. and Orro, B. (2020). “Accumulating Design Knowledge with Reference Models: Insights from 12 Years’ Research into Data Management.” *Journal of the Association for Information Systems*, p. 735-770.

Liu, Y., Zhang, Y., Ren, S., Yang, M., Wang, Y. and Huisingh, D. (2020). “How can smart technologies contribute to sustainable product lifecycle management?” In: *Journal of Cleaner Production*, Vol. 249, p. 1-5.

Mann, S. (2016). “The Research Interview - Reflective Practice and Reflexivity in Research Processes.” Palgrave Macmillan, London.

Marabelli, M., Hansen, S., Newell, S. and Frigerio, C. (2017). „The Light and Dark Side of the Black Box: Sensor-based Technology in the Automotive Industry.“ In: *Communications of the Association for Information Systems*, Volume 40, p. 351-374.

March, S. & Smith, G. (1995). „Design and natural science research on information technology.“ In: *Decision Support Systems,* 15, p. 251-266.

Mayring, P. (2000). “Qualitative Content Analysis.” In: *Forum: Qualitative Social Research Sozialforschung*, Volume 1, No. 2, Art. 20, p. 1-10.

Mayring, P. (2001). “Kombination und Integration qualitativer und quantitativer Analyse.” In: *Forum Qualitative Sozialforschung / Forum Qualitative Social Research*, Volume 2, Issue 1, p. 1-14.

Mayring, P. (2010). “Qualitative Inhaltsanalyse.” In: *Handbuch Qualitative Forschung in der Psychologie*, p. 601-613.

Moreira, R. and Paiva, A. (2009). “Towards a Pattern Language for Model-Based GUI Testing.” In: *Proceedings of the 19th European Conference on Pattern Languages of Programs,* p. 1-8.

Myers, M. and Newman, M. (2007). “The qualitative interview in IS research: Examining the craft.” In: *Information and Organization*, Volume 7, p. 2-26.

Myllärniemi, J., Okkonen, J. and Kärkkäinen, H. (2009). ”Utilizing Business Intelligence Framework for Leveraging Product Lifecycle Management.” In: *The 9th International Conference on Electronic Business*, p. 9-16.

Noble, J. (1998). “Towards a Pattern Language for Object Oriented Design.” In: *Proceedings Technology of Object-Oriented Languages*, p. 2-13.

Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V. and Bragge, J. (2006). ”The Design Science Research Process: A Model For Producing And Presenting Information Systems Research.” In: *Proceedings Design Research Information Systems and Technology*, p. 83-106.

Peffers, K., Tuunanen, T., Rothenberger, M. and Chatterjee, S. (2007). “A design science research methodology for information systems research.” In: *Journal of Management Information Systems*, 24(3), p. 45-77.

Ponterotto, J. (2005). “Qualitative Research in Counseling Psychology: A Primer on Research Paradigms and Philosophy of Science.” In: *Journal of Counseling Psychology*, Vol. 52, No. 2, p. 126-136.

Rowley, J. and Slack, F. (2004). “Conducting a literature review.” In: *MCN The American Journal of Maternal/Child Nursing*, 27(6), p. 31-39.

Saunders, M., Lewis, P. and Thornhill, A. (2019). „Research methods for business students.” Pearson Education Limited, United Kingdom.

Shim, J., Avital, M., Dennis, A., Sheng, O. and Rossi, M. (2017). “Internet of Things: Opportunities and Challenges to Business, Society, and IS Research.” In: *Proceedings of the Thirty Eighth International Conference on Information Systems*, p. 1-6.

Simon, H. (1996). „The Sciences of the Artificial”. *MIT Press*, Cambridge, p. 1-231.

Sprenger, M. and Mettler, T. (2016). „On the utility of e-health business model design patterns.” *Twenty-Fourth European Conference on Information Systems*, p. 1-16.

Staisch, A., Peters, G., Stueckl, T. and Seruga, J. (2012). „Current Trends in Product Lifecycle Management.“ In: *ACIS 2012 Proceedings*, Volume 37, p. 1-10.

Stake, R. (2006). “Multiple case study analysis.” The Guilford Press, New York.

Stark, J. (2018). “Product Lifecycle Management (Volume 3): The Executive Summary.” Springer International Publishing.

Starman, A. (2013). “The case study as a type of qualitative research.” In: *Journal of Contemporary Educational Studies*, p. 28-43.

Stocker, A., Kaiser, C. and Fellmann, M. (2017). „Quantified Vehicles – Novel Services for Vehicle Lifecycle Data.“ In: *Business and Information Systems Engineering*, Volume 59, p. 125-130.

Tao, F., Wang, Y., Zuo, Y., Yang, H. & Zhang, M. (2016). „Internet of Things in product life-cycle energy management.“ In: Journal of Industrial Information Integration 1, p. 26-39.

Tekin, A. and Kotaman, H. (2013). „The Epistemological Perspectives on Action Research.” In: *Journal of Educational and Social Research*, Vol. 3 (1), p. 81-91.

Terzi, S., Bouras, A., Garetti, M. and Kiritsis, D. (2010). “Product lifecycle management - From its history to its new role.” In: *International Journal of Product Lifecycle Management*, Volume 4, p. 360-389.

Tremblay, M., Hevner, A., Berndt, D. and Chatterjee, S. (2010). „The Use of Focus Groups in Design Science Research.“ In: *Communications of the Association for Information Systems*, Volume 26, p. 599-618.

Tuli, F. (2010). “The Basis of Distinction Between Qualitative and Quantitative Research in Social Science: Reflection on Ontological, Epistemological and Methodological Perspectives.” In: *The Basis of Distinction Between Qualitative and Quantitative Research*, Vol. 6 No 1, p. 97-108.

Voit, T., Schneider, A. and Kriegbaum, M. (2020). „Towards an Empirically Based Gamification Pattern Language Using Machine Learning Techniques.” In: *IEEE 32nd Conference on Software Engineering Education and Training*, p. 1-4.

Vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R. (2015). “Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information.” In: *Communications of the Association for Information Systems* (37): p. 205-224.

Weber, R. (2004). “The Rhetoric of Positivism Versus Interpretivism: A Personal View.” In: *MIS Quarterly*, Vol. 28, No. 1, p. 3-12.

Webster, J. and Watson, R. (2002). “Analyzing the Past to Prepare for the Future: Writing a Literature Review.” In: *MIS Quarterly*, 26(2), p. 13-23.

Winter, R., vom Brocke, J., Fettke, P., Loos, P. Junginger, S., Moser, C., Keller, W., Matthes, F. & Ernst, A. (2009). “Patterns in Business and Information Systems Engineering.” In: *Business & Information Systems Engineering,* p. 468-474.

Wortmann, F. and Flüchter, K. (2015). „Internet of Things - Technology and Value Added.“ In: *Business & Information Systems Engineering* 57(3), p. 221-224.

Xia, F., Yang, L., Wang, L. and Vinel, A. (2012). „Internet of Things.“ In: *Proceedings of the International journal of communication systems*, Vol 25, p. 1101-1102.

Yang, F., Shangguang, W., Li, J., Liu, Z and Sun, Q. (2014). “An Overview of Internet of Vehicles.” In: *China Communications*, p. 1-15.

Yin, R. K. (2011). “Qualitative Research from Start to Finish.” The Guilford Press, New York.

Zhang, Y., Ren, S., Liu, Y., Sakao, T. and Huisingh, D. (2017). „A framework for Big Data driven product lifecycle management.“ In: *Journal of Cleaner Production*, Volume 159, p. 229-240.

2

INTERNET OF THINGS IN PRODUCT LIFECYCLE MANAGEMENT – A REVIEW ON VALUE CREATION THROUGH PRODUCT STATUS DATA

Brandt, Niklas, University of Duisburg-Essen, Essen, Germany

Ahlemann, Frederik, University of Duisburg-Essen, Essen, Germany

Reining, Stefan, University of Duisburg-Essen, Essen, Germany

Rehring, Kevin, University of Duisburg-Essen, Essen, Germany

Abstract

The increasing number of sensors and communication technologies built into customer products make information about the product status accessible over the internet. Nonetheless, manufacturers' Product Lifecycle Management (PLM) Information Systems (IS) do not integrate these product instance data. Furthermore, existing PLM IS do not support product-related tasks after the manufacturing phase. We conducted a systematic literature review to understand the impact of data from connected products as new sources of information on PLM. Our results led to a topic map that identifies drivers, applications, and barriers and contextualizes them. We further analyzed PLM reference models that support the analysis of product instance data. We found that the existing PLM IS have several limitations that hinder a successful realization in practice. The results of this review are the first steps toward understanding how the Internet of Things (IoT) influences PLM on a firm level.

Keywords: *Product Lifecycle Management, Internet of Things, close-loop, Literature Review.*

A previous version of this essay was published in the Proceedings of the American Conference on Information Systems 2021.

Brandt, N.; Ahlemann, F.; Rehring, K.; Reining, S., " INTERNET OF THINGS IN PRODUCT LIFECYCLE MANAGEMENT – A REVIEW ON VALUE CREATION THROUGH PRODUCT STATUS DATA" (2021). In Proceedings of the 27th American Conference on Information Systems (AMCIS), An Online AIS Conference, August 9-13, 2021.

Table of contents

List of Figures

1 Introduction

Industrial companies use Information Systems (IS) to create, store, analyze and exchange product-related information. The strategic approach of Product Lifecycle Management (PLM) is the product-centric way to do so (Ameri and Dutta, 2005, p. 577) (Hribernik et al., 2017, p. 1408). With PLM IS, products are designed and simulated. Furthermore, firms generate manufacturing instructions and simulate production processes. These data management and analysis tools for design and manufacturing are the core elements of PLM IS today (Ameri and Dutta, 2005, p. 578-579). On the other hand, data about the usage of individual product instances are often not considered in such IS (Wellsandt et al., 2015, p. 1). Nonetheless, analyzing status information from individual products might interest manufacturers, as each product instance is used differently. One reason for the missing integration of these data into PLM IS is that information about individual products is mostly unavailable after a product is sold to a customer (Yoo et al., 2016, p. 2). This drop in available data after manufacturing is still valid for products today (Lucero, 2016, p. 5). The Internet of Things (IoT), where products share data from embedded sensors over the internet, make these traditionally missing pieces of information about product instances available (Wagner et al., 2018, p. 1004) (Haag and Anderl, 2018, p. 65). Examples are smart home devices (Kubler et al., 2015, p. 83), connected cars (Han et al., 2018, p. 365), and healthcare products (Sodhro et al., 2018, p. 380).

Holler et al. (2016a, p. 488) found in their empirical study that product usage data could improve the product design process as the customer demand could be measured. Nevertheless, due to the young nature of the IoT and the rare integration of sensors and communication technology even in technology products (Kiritsis, 2011, p. 481), data acquisition in the use phase remains a challenge today (Ren et al., 2019, p. 1356). As a result, organizational tasks, which could benefit from detailed product usage information, are based on hypothetical data and various forms of customer feedback and market analysis (Goto et al., 2016, p. 1021). Previous research mainly investigated specific applications and does not provide a holistic overview of organizations' overall potentials. Furthermore, researchers have already proposed several IT artifacts to gather, store and analyze product status data along the products' lifecycles. Often those models only fit specific applications and are not harmonized with previous works (Kadiri and Kiritsis, 2015, p. 5661). Conducting a literature review about this research topic is especially useful, as actual research achievements are widespread in the engineering, information systems, and management domain. Our research aims to holistically depict and contextualize the benefits and challenges of using data about product instances in PLM (1) and to analyze existing PLM IS that support the gathering, storage, and analysis of product lifecycle data (2). Precisely, this paper aims to answer two research questions:

1. Which PLM applications could benefit from information about individual product instances, and what hinders organizations from realizing them?

2. What are the limitations of existing PLM IS designed to support the lifelong gathering and analysis of product instance data?

We chose a systematic literature review to unveil previous achievements by collecting and aggregating related research (Seuring and Mueller, 2008, p. 1700) in a concept-centric approach (Webster and Watson, 2002, p. 16). Through systematic analysis and aggregation, our review contributes to the body of knowledge and sets the direction for future research possibilities. Research question one is answered by a topic map that contextualizes barriers, drivers, and use cases. Afterward, we used this topic map to analyze existing PLM IS.

The paper is organized as follows. We explain the main concepts in section 2 and the research method in section 3, followed by a descriptive and qualitative presentation of the findings in section 4. Subsequently, we discuss the results in section 5 and conclude in section 6.

2 Conceptual Foundations

2.1 Product Lifecycle Management

The lifecycle of physical products is divided into the phases Begin of Life (BOL), Mid of Life (MOL), and End of Life (EOL) (Kiritsis, 2011, p. 481). During BOL, a product is designed and manufactured (Cao and Folan, 2012, p. 25). After that, during MOL, a product is individually used and maintained until it is disposed or recycled at its EOL. PLM IS support the storage and analysis of product-related information along these phases. Since its first usage, the scope of PLM IS has grown. Today, PLM ISs include software applications for storing product-related data (e.g., Product Data Management) and creating/analyzing design models (e.g., Computer Aided Design) (Cao and Folan, 2012, p. 25) (Holligan et al., 2017, p. 909). Although the concept of PLM includes all lifecycle phases, the focus of existing applications is on static data from BOL. Furthermore, existing PLM IS do not incorporate data about individual product instances but product classes (Wellsandt et al., 2015, p. 245). Data about classes are valid for a group of individual products, like design drawings or manufacturing instructions. Nonetheless, when a physical product is manufactured and used afterward, the information differs between the products as no product is manufactured and used the same way. The missing integration of product instance information into PLM applications, as measured manufacturing conditions or usage information, can be explained by the unavailability of instance information from manufacturing, MOL, and EOL and a lack of available IS (Zhang et al., 2017, p. 229).

2.2 Internet of Things

After physical products are sold to the customers, manufacturers only receive status information in case of maintenance, repair, and overhaul. With product embedded IoT technology instead, individual products share actual information about their status over the internet. The basis for the IoT are Cyber-Physical-Systems (CPS). CPS interface with the global internet via mostly wireless communication technology and the real world via actuators and sensors (Gubbi et al., 2013, p. 1647). The increase from 12.5 billion connected devices in 2010 (Evans, 2011, p. 3) to 26.66 billion in 2019 (Lucero, 2016, p. 5) demonstrates that more and more products can provide valuable information about their status and behavior. Thus, detailed status information from product instances during manufacturing, MOL, and EOL become available to organizations. Due to this, organizations might extend their traditional PLM IS and replace hypothetical data that they use to perform lifecycle-related tasks with real data. In the past, researchers introduced the term closed-loop PLM for this extension of PLM IS, which suggests closing information loops between all lifecycle stages (Kiritsis, 2011, p. 481). Researchers have already designed IS that allow the integration of IoT information into PLM IS and realize several benefits for organizations (Hribernik et al., 2017, p. 1410) (Yoo et al., 2016, p. 3) (Cai et al., 2014, p. 1563).

3 Research Methodology

3.1 Literature Selection Process

To increase the reliability of this review, we follow the guidelines on performing a systematic literature review from vom Brocke et al. (2015, p. 2016) and Webster and Watson (2002, p. 20). We queried databases from different disciplines to satisfy both aspects of IoT and PLMIoT and PLM aspects, selected an appropriate time frame, continually revised our keywords, and documented the review process in detail (Vom Brocke et al., 2015, p. 210). Our research questions are related to engineering, information systems, and management. Therefore, we selected the Elsevier database, the Xplore digital library, the EBSCO Business Source Ultimate and the Applied Science and Technology databases and the AIS eLibrary. We only considered peer-reviewed papers that are written in English language. We chose the time frame from 2010 to 2020 for two reasons. Xin and Ojanen (2017, p. 1099), who systematically analyzes the impact of digitalization on PLM, found that most results with a focus on IoT were published after 2010. Before 2010, the leading IoT sources were RFID and PDA (Kiritsis, 2011, p. 481), which clarifies that the early definition of the IoT differs from our view. Second, significant results from previous years would be referenced from sources within our time frame to recognize these results through our backward search. For the keywords, we searched in titles, abstracts, and keywords for combining words from two categories. We used the keywords "Internet of Things" and "IoT" in the first category. Furthermore, we used the keyword "closed-loop," as this term is used in previous related research to depict the information flows between lifecycle stages. We chose the keywords "PLM" and "Product Lifecycle Management" for the second category. We linked the keywords within each category with a logical OR and the two categories with a logical AND. In addition to the keyword search, we applied back- and forward search (Webster and Watson, 2002, p. 16). In total, we ended up with 173 papers after removing exact duplicates. In the first step, we excluded 122 papers due to a lack of relevance after reading the titles, abstracts, and conclusions. After back- and forward search, we added a number of six often cited papers to the literature pool (Yoo et al., 2016) (Cerri et al., 2015) (Li et al., 2015) (Guo et al., 2014) (Hribernik et al., 2013) (Zhang et al., 2017) so that this review is based on 57 papers.

3.2 Review Framework

We defined a review framework for systematically analyzing the resulting literature pool. According to Mayring (2010, p. 603), inductive and deductive frameworks can be differentiated. An inductive framework is developed during the review, whereas a deductive one is proposed upfront and continuously refined. We derived the categories from answering the first research question by adapting the triggers from Seuring's well-acknowledged literature review on sustainable Supply Chain Management (Seuring and Mueller, 2008, p. 1703-1705). The categories "barriers," "drivers" and "use cases" are well suited, as Seuring used them to answer a research question that is like ours. Furthermore, the categories are validated with a high number of publications. "Barriers" summarize a lack of companies' IS to integrate IoT data in PLM. With the category "drivers," we aggregate the organization-external circumstances that explain why the topic came up now. The category "use cases" creates an understanding of the upcoming applications. As we aim to identify the barriers, drivers, and use cases that are currently unknown, we chose to develop an inductive framework. Thus, we defined the categories up front and continually redefined the factors for each category during the review using the open coding technique.

We defined the categories "research result" and "object of analysis" to answer the second research question. In those categories, we systematically cluster the models of the literature. Finally, "data gathering" as a well-established category in systematic reviews helps to understand the empirical foundation of each proposed research result (Hesselmann and Kunal, 2014, p. 7). While we developed the factors for the first research question in an inductive approach, we used a deductive approach to analyze the literature pool for the second research question, as the kind of research results and objects of analysis are predefined by the second research question to be data models and architectures. Furthermore, the factors for the data-gathering category are well known from previous literature reviews.

4 Paper Analysis

4.1 Topic Map to Integrate Data from CPS into PLM IS

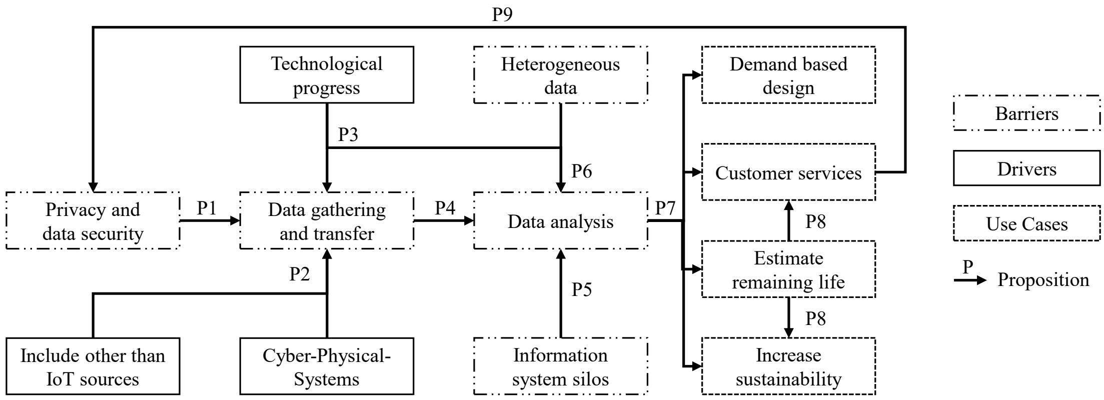


Figure 4-1 Topic map with barriers, drivers and use cases

The topic map in Figure 4-1 gives an overview of the integration of IoT data from product instances into PLM IS. Thereby, it addresses the first research question. By contextualizing the factors from the three independent categories of the classification framework, it contextualizes the requirements to realize the identified use cases. With the aggregation of existing knowledge, the topic map fulfills the requirements of a theory for analyzing (class 1) (Gregor, 2006, p. 622-623).

Proposition 1 (9 papers): Strict governmental privacy regulations and high data security requirements negatively influence data gathering and transfer.

In the use phase, products are operated by or on behalf of customers. As IoT data unveil information about the product's usage, it indirectly unveils the customers' behavior as well (Kubler et al., 2015, p. 83). Due to the criticality of customer behavioral information, governmental privacy regulations in several countries (Hribernik et al., 2013, p. 7) and data security requirements (Kunz et al., 2010, p. 1311) complicate the gathering of product instance IoT data along the use phase. The strength of the negative data privacy impact on data gathering and transfer can be reduced when the product's owner and/or user agrees that data from his product is gathered and transferred (Kubler et al., 2015, p. 91). Anonymizing IoT data to keep the individual customer's identity and behavior secret (Xin and Ojanen, 2017, p. 1101), along with taking measures to increase data security, might support companies to overcome this issue.

Proposition 2 (26 papers): Data gathering is supported by product-embedded sensors and connectivity units and other internet sources.

Many technological products today are equipped with embedded sensors and communication technology to extent products functionalities (Hribernik et al., 2017, p. 1407-1408) (Xin and Ojanen, 2017, p. 1098) (Robert et al., 2016, p. 4908). This strengthens data gathering as the existing product embedded hardware can be used to gather and transfer IoT data to the manufacturers' IS (Främling et al., 2013, p. 799). From a technological perspective, IoT data could then be gathered along the products' lives. In addition, researchers found that the social media entries, customer ratings, and complaints can be systematically analyzed to get feedback about customers' product instances (Li et al., 2015, p. 667) (Wellsandt et al., 2015, p. 9). Thereby, data gathered with product-embedded IoT technologies can be complemented to gain a holistic view.

Proposition 3 (27 papers): Continuous technological progress supports the gathering and analysis of data along the lifecycle.

Much research and development is conducted in the fields of sensors and communication technologies that enable the gathering and transfer of IoT data using computing power (Yoo et al., 2016, p. 2), communication standards (Kiritsis, 2011, p. 479), miniaturization (Kaed et al., 2018, p. 4182) and costs (Goto et al., 2016, p. 1022). These trends allow companies to embed IoT technology into more and more products and gather data along the products' lives (Lucero, 2016, p. 5). Without product-embedded technology for data gathering and transfer, information is often only gathered manually (Tao et al., 2014, p. 1253) during maintenance, repair, and overhaul (Yoo et al., 2016, p. 2) (Hribernik et al., 2017, p. 1407). Besides hardware optimizations, new and innovative tools for data analysis strengthen the value of IoT data in PLM applications (Goto et al., 2016, p. 1022) (Lowenstein and Slater, 2018, p. 1) (Li et al., 2015, p. 13).

Proposition 4 (27 papers): To analyze product-related IoT data along the product lifecycle, product information needs to be accessible to the manufacturer.

The analysis of IoT data in PLM applications requires that the product-embedded sensors create data on an instance level. Furthermore, these data must be accessible to the manufacturer (Kiritsis, 2011, p. 479). Without access to these data, organizations are restricted from performing tasks that traditional PLM IS offers (Holligan et al., 2017, p. 911-912) (Wellsandt et al., 2015, p. 479-480). On the other hand, new or improved traditional applications identified in this review would not be possible (Goto et al., 2016, p. 1023). Thus, it becomes clear that product data gathering and transfer on an instance level is a prerequisite for a successful analysis in PLM applications.

Proposition 5 (31 papers): Information system silos hinder IoT data analysis, as the product lifecycle and the product-related data might relate to multiple separated IS.

As a product's lifecycle relates to many organizational units of an industrial company, multiple separate IS exist that perform lifecycle-related tasks (Tantik and Anderl, 2017, p. 91) (Kaed et al., 2018, p. 4183). To holistically analyze the IoT data that the product creates along their lives, data from multiple IS might need to be accessed (Goto et al., 2016, p. 1023). Thus, the heterogeneity of the legacy IS further complicates the analysis, as these IS might not be designed for this (Franke et al., 2014, p. 226). Furthermore, product instance IoT data cannot be analyzed with existing IS, as these IS analyze data only about product classes (Robert et al., 2016, p. 4909) (Kunz et al., 2010, p. 1308). Flexible PLM IS are required to meet the new requirements (Främling et al., 2013, p. 319). A common platform to analyze product lifecycle data is often proposed as a solution (Kubler et al., 2015, p. 83) (Cai et al., 2014, p. 1565).

Proposition 6 (28 papers): The multiple sources of product (instance) information along the lifecycle lead to heterogeneous data, complicating its analysis.

The product-related IoT data type might vary from one lifecycle phase to another and within a phase. This is due to different sensor types or versions and communication technologies that provide varying signals (Sodhro et al., 2018, p. 380). In addition, multiple information from traditional PLM IS needs to be integrated. The challenge can be visualized with a mixture of structured and unstructured (Kunz et al., 2010, p. 1310) as well as static and dynamic data (Kiritsis, 2011, p. 481). Furthermore, the high volume (Holligan et al., 2017, p. 910) and variety of data (Hribernik et al., 2017, p. 1411) further complicate the analysis. Hence, the heterogeneity of data gathered makes the analysis more difficult.

Proposition 7 (18 papers): Analyzing IoT data within PLM applications brings value to manufacturers, customers, and other stakeholders.

Without data analysis, valuable insights are wasted, and lifecycle-related tasks are not supported (Matsokis and Kiritsis, 2010, p. 787). The most prominent application of product data analysis is the improvement of future product designs. This application's prominence can be explained by the fact that the design process traditionally relies on hypothesizing how the product will be used during its MOL (Goto et al., 2016, p. 1021) (Emmanouilidis et al., 2018, p. 436). The real customer demand can be measured precisely by analyzing how individual products are used instead (Goto et al., 2016, p. 1021). Furthermore, products can be customized to the individual customers' actual demand (Holligan et al., 2017, p. 910), and over-dimensioning and early failures of components can be avoided in future designs (Shin et al., 2014, p. 552-553). Evaluation of a part's remaining lifetime is the second most prominent application for IoT data in PLM. The remaining life can be estimated by analyzing the entire product's lifecycle data (Shin et al., 2014, p. 553). When the remaining life is known, the value of EOL parts (Kiritsis, 2011, p. 482) can be calculated (Xin and Ojanen, 2017, p. 1100) without the need for manual inspections (Ondemir et al., 2012, p. 722). Additionally, lifetime estimation can be used for predictive maintenance applications through detecting abnormalities (Tao et al., 2016, p. 35) to reduce unplanned failures and replace functional parts in fixed intervals (Xin and Ojanen, 2017, p. 1100). When data about product instances become available, new services based on product status (Han et al., 2018, p. 368) become possible. For instance, data about the product's usage can offer customers dynamic insurance based on their behavior (Kubler et al., 2015, p. 94) (Han et al., 2018, p. 374). Those new services and business models are the third most often mentioned application. The last application is to increase the overall product sustainability to reduce the environmental impact. These sustainability applications include energy efficiency and CO2 emissions (Wang et al., 2016, p. 1999) (Främling et al., 2013, p. 795). On the other hand, it also includes improving the recycling process by estimating EOL products remaining life (Xin and Ojanen, 2017, p. 1100).

Proposition 8 (32 papers): Evaluation of individual products' remaining service life enables predictive maintenance services and recycling decision support.

The actual usage of products, which varies from customer to customer, significantly influences the products' remaining life. As mentioned in proposition 7, knowledge about the remaining product life or its parts might be used as input for new services, i.e., predictive maintenance (Tao et al., 2016, p. 35), and to increase sustainability, i.e., recycling of individual parts based on their conditions (Xin and Ojanen, 2017, p. 1100). Hence, calculating the remaining life is not only an application by itself but also the enabler for the other applications.

Proposition 9 (23 papers): Privacy restrictions are lowered, as customers want the manufacturer to gather product status information to extend product functionalities.

Customers require new services that analyze sensor-based product information (Tao et al., 2016, p. 35). Thus, they might be willing to accept that the manufacturer subsequently gathers and stores IoT data about their product and behavior. Furthermore, customers have particular requirements to a product, which an individualized service that has access to the user's behavior might be able to address (Holligan et al., 2017, p. 910). Nevertheless, in the case of a new application, the customer can opt in or –out depending on the purpose of the new services and the requested data. Thus, he controls the data that unveils his behavior (Hribernik et al., 2013, p. 10). With the explicit agreement of customers to gather and analyze information about the product's usage, they authorize the manufacturers to gather the data (Kunz et al., 2010, p. 1311) and to overcome the strict privacy regulations.

4.2 Analysis of existing Research Results

According to our second research question, we analyze the PLM IS, i.e., reference and data models that propose methods to store and analyze IoT data for PLM applications. For the analysis, we derived the evaluation criteria from our topic map in Figure 4-1. The barriers are factors that need to be overcome with the research results, the drivers are factors that need to be considered as increasing sources of information and the use cases are factors that need to be enabled by the IS. In our analysis, we did not further consider generic and abstract reference and data models (21 out of 61), as those generic concepts do not provide detailed information on managing and using IoT data in PLM IS. Furthermore, we excluded results that only consider specific lifecycle phases (27 out of 51), as the main objective of our analysis is the lifelong analysis of product-related information. By filtering out generic concepts and lifecycle-specific results, we ended up with 14 reference models and 11 data models to analyze. As four of the architectures and three of the data models are based on the PROMISE project, we grouped them. In addition, four data models are based on the Open Data Format (O-DF) / Open Messaging Interface, and thereby, we grouped them. Thus, we analyze eleven reference models and six data models.

While nine of the thirteen reference models are independent of each other, the QLM cloud, the DIALOG, and the QLM standard are based on the PROMISE project (Främling et al., 2014, p. 321) (Kubler et al., 2015, p. 83) (Kiritsis, 2011, p. 483-484) (Främling et al., 2013, p. 793). Due to the early stage of the IoT technology at the time of the PROMISE project, the model considers structured events instead of heterogeneous sensor data. The IoT-based Configurable Information Service Platform (ICISP) proposes technology-independent information management by storing the product instance information into sub-models for each lifecycle stage and component (Cai et al., 2014, p. 1563). In addition, the ICISP has standard interfaces to all sub-models for data analysis over lifecycle stages. The Big data-driven PLM proposes specific tools for data analysis to extract knowledge from lifecycle data (Zhang et al., 2017, p. 233). Focus varies from abstract processing of data to precise methods. Nevertheless, information system silos of organizations and traditional PLM information from BOL are not integrated. In the FALCON architecture, data streams and events from IoT instances are stored in a Triple Store, which serves as a central database (Hribernik et al., 2017, p. 1411). This model introduces specific PLM applications and focuses on MOL data, while interfaces to data from other lifecycle phases are important. Besides the abovementioned holistic reference models, others are restricted to specific industries and/or use cases. The PLM Service Architecture integrates boat data and internet sources to manage a boat's lifecycle (Hribernik et al., 2013, p. 8). With the division into four layers, the Big Bill of Material (BOM) precise data-gathering technologies to measure and analyze energy consumption (Tao et al., 2014, p. 1254). IoT in Product Lifecycle Energy Management (PLEM) is also applicable to evaluate energy efficiency and focuses on technologies to gather, transmit and analyze data (Tao et al., 2016, p. 34). The Product Lifecycle Management House deals with data gathering technologies, where a core data lake all data (Shen et al., 2016, p. 2). The Linked Engineering mAnufacturing Platform (LEAP) uses the PROMISE message interface to reduce lifecycle costs and environmental impact (Cerri et al., 2015, p. 129). Through the division of the information flow into application-specific information loops, the complex product-closed loop PLM system proposes storing product-related data according to the use cases (Guo et al., 2014, p. 134). Last, the Big

Data driven Hierarchical Digital Twin Predictive Remanufacturing (BDHDTPREMfg) enables the gathering of product instance energy consumption information (Wang et al., 2020, p. 1).

Besides the reference models, we identified 14 data models that propose methods to store product instances information along the product lifecycles. Based on the PROMISE reference model, Främling et al. (2013, p. 321), Kubler et al. (2015, p. 86), Cerri et al. (2015, p. 129) and Robert et al. (2016, p. 4909) introduced the O-DF to manage product instance data in objects, information items, and values. As a completely different approach, Kubler et al. (2015, p. 194) propose storing product-related information locally on a product-embedded database. The Asset Administration Shell offers a similar approach to store asset information locally in the asset and offers to store general and static product properties in a header and dynamic lifecycle information in the body of a product instance data model (Tantik and Anderl, 2017, p. 88-89). The resource model for IoT from Cai et al. (2014, p. 1564) introduces a two-dimensional data structure, where the data model of a product instance grows along the lifecycle. The Wrapper for Web Services from Franke et al. (2014, p. 225-226) focuses on PLM applications that are traditionally interoperable but need to exchange information during closed-loop PLM. They propose an exchange of information between applications that support the tasks along the product lifecycle via self-explaining wrappers

5 Discussion

PLM IS supports industrial companies in storing and analyzing product-related information in a product-centric approach. Although a product's life consists of BOL, MOL ,and EOL, existing IS mainly concentrate on static BOL information (Bruno et al., 2015, p. 45) (Kunz et al., 2010, p. 1308). In addition, information gathered by product instance sensors are not considered in those IS. Instead, many product-related tasks within industrial companies that rely on product instance data with hypothetical data (Goto et al., 2016, p. 1021).

We derived a topic map from our systematic literature review that depicts and contextualizes the interdependencies between the barriers, drivers, and benefits. First, data from product-embedded sensors might unveil customer behavior; thus, gathering this data is critical. Second, data about the status of individual products need to be collected along individual products' lives and transferred to a database using sensors, data preprocessing, and communication technology. This requires that manufacturers redesign their products. Third, analyzing the high amount of data requires manufacturers to rethink existing processes and how product-lifecycle-related tasks are performed. Fourth, the heterogeneity of the information from product instances and the combination of static data about product classes and changing/increasing data about product instances, complicate the analysis. Fifth, lifelong data analysis requires the cooperation of different organizations' departments and the introduction of interfaces between applications, which generally do not exchange information. A systematic way to exchange information between existing legacy IS and new applications is required. Nevertheless, when information about product instances become available and is systematically analyzed, we found that products can be designed to meet the measured customer demand, new services for higher revenues and customer satisfaction can be established, and the remaining life of individual products can be estimated, and the overall product sustainability can be improved. Due to the dependencies of the identified barriers, drivers, and use cases in our topic map, it becomes clear that all the diverse factors need to be addressed by a PLM IS together. Thereby, the identified factors work as requirements to design closed-loop PLM IS.

Researchers of different domains have already proposed PLM IS that aim to enable the use cases and overcome the barriers identified in this research paper. In total, we identified 16 independent research results using reference and data models. The focus of the specified reference models range from descriptions of precise technologies to exchange information (Shen et al., 2016, p. 2) to application-specific (Hribernik et al., 2013, p. 8) and data-analysis-focused (Zhang et al., 2017, p. 233) models as well as abstract definitions of information exchange loops (Guo et al., 2014, p. 134) to more detailed reference models which describe the data aggregation and analysis process more holistically over the whole lifecycle in a use cases-independent way (Hribernik et al., 2017, p. 1410). For a new generation of PLM IS, use case-and technology-independent architectures help to face the challenges of technological progress and the increasing number of objects that participate in the IoT. Furthermore, flexible architecture configurations support the inclusion of heterogeneous information as the sources of information vary not only between companies but also between the (future) products of a single manufacturer (Främling et al., 2014, p. 319). One aspect not mentioned in the paper is the local preprocessing in the products to efficiently transfer the relevant data only. Last, new sources of information might be added, and existing data from PLM IS about product classes need to be combined with information about product instances.

In our review, we identified data models that focus on data transfer (Cai et al., 2014, p. 1563). Other models dealt with specific data types, such as events (PROMISE Semantic Object Model), which limits the storage of information to a subset of all available information. Furthermore, data models were found to store information locally in product-embedded storage devices (Kubler et al. 2015, p. 194). Nonetheless, such systems restrict information availability and hinder the analysis. Another approach (Resource model for IoT) is to store the gathered data in lifecycle-specific data warehouses and establish interfaces for data analysis between them.

6 Conclusion

We conducted a systematic literature review to existing aggregate knowledge about the issues to utilize product instance data from CPS in PLM IS. We found that product-embedded connectivity units are not only beneficial for customers in terms of new product functionalities. Moreover, we identified that the product design process could be improved based on measured customer demand, the customer service can be individualized based on the actual product status, the remaining lifetime of products can be calculated based on the product usage, and the lifecycle sustainability can be increased with detailed knowledge about the usage of the product. Thus, our topic map shows that connected products become a valuable source to improve product-related tasks.

In total, we identified twelve influencing factors and contextualized them by developing nine propositions about their interaction with and dependencies on each other. The resulting topic map shows that closed-loop PLM requires enterprise IS and the design and other internal processes of a manufacturer that are traditionally based on hypothetical customer data to be adapted. By demonstrating the diversity of influencing factors and the stakeholders along the product lifecycle, we can highlight the need to break organizational silos. We used the topic map to analyze the existing reference and data models that propose methods to realize closed-loop PLM. We contribute to the body of knowledge by presenting issues, and dependencies for future PLM IS.

Nonetheless, our research is not free of limitations. There is a need to extend and validate the barriers, drivers, and use cases, and the topic map's interdependencies with further empirical research. This research might also unveil barriers, drivers, and use cases specific to any product domain. With other evidence, our topic map can be developed toward a theoretical model that is valid independently of time and use cases (Gregor, 2006, p. 622-623). Such a theoretical model would contain the requirements for developing future PLM IS. Based on such requirements, the limitations of existing models we unveiled in the paper analysis subchapter "analysis of existing research results" can be addressed.

7 References

Ameri, F. & Dutta, D. (2005). “Product Lifecycle Management: Closing the Knowledge Loops.” In: *Computer-Aided Design & Applications (2:5)*, p. 577-590.

Bruno, G., Antonelli, S. & Villa, A. (2015). “A reference ontology to support product lifecycle management.” In: *Procedia CIRP* (33), p. 41-46.

Cai, H, Xu, L., Xu, B., Xie, C., Qin, S. & Jiang, L. (2014). “IoT-Based Configurable Information Service Platform for Product Lifecycle Management.” In: *IEEE Transactions on Industrial Informatics*, p. 1558-1567.

Cao, H. & Folan, P. (2012). “Product Life Cycle: the evolution of a paradigm and literature review from 1950-2009.” In: *Journal of Production Planning & Control* (23:8), p. 641-662.

Cerri, D., Taisch, M., Terzi, S., Buda, A., Framling, K., Kaddiri, S., Milicic, A., Kiritsis, D., Parrotta, S. & Peukert, E. (2015). “Proposal of a Closed Loop Framework for the Improvement of Industrial Systems’ Life Cycle Performances.” In: *Procedia CIRP* (29), p. 126-131.

Emmanouilidis, C., Bertoncelj, L., Bevilacqua, M., Tedeschi, S. & Ruiz-Carcel, C. (2018). “Internet of Things - Enabled Visual Analytics for Linked Maintenance and Product Lifecycle Management.” In: *IFAC-Papers Online* (51:11), p. 435-440.

Evans, D. (2011). “The Internet of Things: How the Next Evolution of the Internet Is Changing Everything.” In: *Cisco Internet Business Solutions Group*, p. 1-11.

Främling, K., Holmström, J., Loukkola, J., Nyman, J. & Kaustell, A. (2013). ”Sustainable PLM through Intelligent Products.” In: *Engineering Applications of Artificial Intelligence* (26:2), p. 789-799.

Främling, K., Kubler, S. & Buda, A. (2014). “Universal Messaging Standards for the IoT From a Lifecycle Management Perspective.” In: *IEEE Internet of Things Journal* (1:4), p. 319-327.

Franke, M., Klein, K., Hribernik, K., Lappe, D., Veigt, M. & Thoben, K. (2014). „Semantic Web Service Wrappers as a Foundation for Interoperability in Closed-loop Product Lifecycle Management.” In: *Procedia CIRP* (22), p. 225-230.

Goto, S., Yoshie, O. & Fujimura, S. (2016). “Internet of things value for mechanical engineers and evolving commercial product lifecycle management system.” In: *IEEE International Conference on Industrial Engineering and Engineering Management*, p. 1021-1024.

Gregor, S. (2006). “The Nature of Theory in Information Systems.” In: *MIS Quarterly* (30:3), p. 611-642.

Gubbi, J. Buyya, R., Marusic, S. & Palaniswami, M. (2013). “Internet of Things (IoT): A vision, architectural elements, and future directions.” In: *Future Generation Computer Systems* (29), p. 1645-1660.

Guo, W., Zheng, Q., Zuo, B. & Shao, H. (2014). “Complex Product triple Closed-loop PLM Model.” In: *Advances in Transdisciplinary Engineering* (1), p. 132-139.

Haag, S. & Anderl, R. (2018). “Digital twin – Proof of concept.” In: *Manufacturing Letters* (15), p. 64-66.

Han, J., Kim, H., Heo, S., Lee, N., Kang, D., Oh, B., Kim, K., Yoo, W., Byun, J. & Kim, D. (2018). “GS1 Connected Car: An Integrated Vehicle Information Platform for Connected Car Services.” In: *IEEE Intelligent Vehicles Symposium (IV)*, p. 367-374.

Hesselmann,F & Kunal, M. (2014). “Where are we headed with benefits management research? Current shortcomings and avenues for future research.” In: *European Conference on Information Systems* (22), p. 1-17.

Holler, M., Stoeckli, E., Uebernickel, F. & Brenner, W. (2016a). “Towards Understanding closed-loop PLM: The Role of Product Usage Data for Product Development enabled by intelligent Properties.” In: *BLED Proceedings*, p. 479-491.

Holligan, C., Hargaden, V. & Papakostas, N. (2017). ”Product lifecycle management and digital manufacturing technologies in the era of cloud computing.” In: *International Conference on Engineering, Technology and Innovation* (ICE/ITMC), p. 909-918.

Hribernik, K., Franke, M., Klein, P., Thoben, K. & Coscia, E. (2017). „Towards a platform for integrating product usage information into innovative product-service design.” In: *International Conference on Engineering, Technology and Innovation* (ICE/ITMC), p. 1407-1413.

Hribernik, K., Wuest, T. & Thoben, K. (2013). „A Product Avatar for Leisure Boats Owners.” In: *IFIP Advances in Information and Communication Technology* (409), p. 560-569.

Kadiri, S. & Kiritsis, D. (2015). “Ontologies in the context of product lifecycle management.” In: *International Journal of Production Research* (53:18), p. 5657-5668.

Kaed, C., Danilchenko, V., Delpech, F., Brodeur, J. & Radisson, A. (2018). “Linking an Asset and a Domain Specific Ontology for a Simple Asset Time Series Application.” In: *IEEE Conference on Big Data*, p. 4182-4188.

Kiritsis, D. (2011). “Closed-loop PLM for intelligent products in the era of the Internet of things.” In: *Computer Aided Design* (43), p. 479-501.

Kubler, S., Främling, K. & Derigent, W. (2015). ”P2P Data synchronization for product lifecycle management.” In: *Computers in Industries* (66), p. 82-98.

Kunz, S., Brecht, F., Fabian, B., Aleksy, M. & Wauer, M. (2010). „Aletheia - Improving Industrial Service Lifecycle Management by Semantic Data Federations.” In: *24th IEEE International Conference on Advanced Information Networking and Applications*, p. 1308-1314.

Li, J., Cheng, Y. & Zhao, L. (2015). “Big data in product lifecycle management.” In: *The International Journal of Advanced Manufacturing Technology* (81:1-4), p. 667-684.

Lowenstein, D. & Slater, C. (2018). “Management of Test Utilization, Optimization, and Health through Real-Time Data.” In: *IEEE AUTOTESTCON*, p. 1-6.

Lucero, S. (2016). “IoT platforms: enabling the Internet of Things – Whitepaper.” HIS Technology, p. 1-21.

Matsokis, M. & Kiritsis, D. (2010). “An ontology-based approach for Product Lifecycle Management.” In: *Computers in Industry* (61:8), p. 787-797.

Ondemir, O., Ilgin, M. & Gupta, S. (2012). “Optimal End-of-Life Management in Closed-Loop Supply Chains Using RFID and Sensors.” In: *IEEE Transactions on Industrial Informatics* (3:8), p. 719-728.

Ren, S., Zhang, Y., Liu, Y., Sakao, T., Huisingh, D. & Almeida, C. (2019). “A comprehensive review of big data analytics throughout product lifecycle to support sustainable smart manufacturing.” In: *Journal of Cleaner Production* (210), p. 1343-1365.

Robert, J., Kubler, S., Traon, Y. & Främling, K. (2016). ”O-MI/O-DF standards as interoperability enablers for Industrial Internet.” In: *42nd Annual Conference of the IEEE Industrial Electronics Society*, p. 4908-4915.

Seuring, S. & Mueller, M. (2008). “From a literature review to a conceptual framework for sustainable supply chain management.” In: *Journal of Cleaner Production* (16), p. 1699-1710.

Shen, X., Cao, M., Lu, Y. & Zhang, L. (2016). “Life cycle management system of power transmission and transformation equipment based on Internet of Things.” In: *China International Conference on Electricity Distribution*, p. 1-5.

Shin, J., Kiritsis, D. & Xirouchakis, P. (2014). ”Design modification supporting method based on product usage data in closed-loop PLM.” In: *International Journal of Computer Integrated Man.* (28:6), p. 551-568.

Sodhro, A., Pirbhulal, S. & Sangaiah, A. (2018). “Convergence of IoT and PLM in medical health care.” In: *Future Generation Computer Systems* (86), p. 380-391.

Tantik, E. & Anderl, R. (2017). “Integrated Data Model and Structure for the Asset Administration Shell in Industry 4.0.” In: *Procedia CIRP* (60), p. 86-91.

Tao, F., Wang, Y., Zuo, Y., Yang, H. & Zhang, M. (2016). „Internet of Things in product life-cycle energy management.” In: *Journal of Industrial Information Integration* (1), p. 26-39.

Tao, F., Zuo, Y., Xu, L., Lv, L. & Zhang, L. (2014). „Internet of Things and BOM-Based Life Cycle Assessment of Energy-Saving and Emission-Reduction of Products.” In: *IEEE Transactions on Industrial Informatics* (10:2), p. 1252-1261.

Vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R. (2015). „Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information.” In: *Communications of the Association for Information Systems* (37), p. 205-224.

Wagner, T., Herrmann, C. & Thiede, S. (2018). „Identifying target oriented Industrie 4.0 potentials in lean automotive electronics value streams.” In: *CIRP Conference on Manufacturing Systems* (72), p. 1003-1008.

Wang, Y., Zahng, M. & Zuo, Y. (2016). „Potential applications of IoT-based product lifecycle energy management.” In: *IEEE 11th Conference on Industrial Electronics and Applications*, p. 1999-2002.

Wang, Y., Wang, S., Yang, B., Zhu, L. & Liu, F. (2020). „Big data driven Hierarchical Digital Twin Predictive Remanufacturing paradigm.” In: *Journal of Cleaner Production* (248), p. 1-15.

Webster, J. & Watson, R. (2002). “Analyzing the Past to prepare for the Future: Writing a Literature Review.” In: *MIS Quarterly* (26:2), p. 13-23.

Wellsandt, S., Hribernik, K. & Thoben, K. (2015). „Content analysis of product usage information from embedded sensors and web 2.0 sources.” In: *IEEE ICE/ITMC conference*, p. 1-9.

Xin, Y. & Ojanen, V. (2017). “The impact of digitalization on product lifecycle management: How to deal with it? .” In: *International Conference on Industrial Engineering and Engineering Management*, p. 1098-1102.

Yoo, M., Grozel, C. & Kiritsis, D. (2016). „Closed-loop lifecycle management of service and product in the internet of things.” In: *Sensors* (16:7), p. 1-26.

Zhang, Y., Ren, S., Liu, Y., Sakao, T. Huisingh, D. (2017). “A framework for Big Data driven product lifecycle management.” In: *Journal of Cleaner Production* (159), p. 229-240.

3

HOW DO YOU DRIVE? ANALYZING VEHICLE SENSOR DATA IN PRODUCT LIFECYCLE MANAGEMENT SYSTEMS

Brandt, Niklas, University of Duisburg-Essen, Essen, Germany

Ahlemann, Frederik, University of Duisburg-Essen, Essen, Germany

Abstract

For most car and truck manufacturers access to information about the status of produced vehicles end with their sale. With embedded connectivity units, this changes so that manufacturers can gather data from vehicle in-build sensors and thus, gain insights into each products usage. Our study investigates why manufacturers aim to integrate connectivity units into their products to gather sensor-based usage information and what hinders them to do so. Thereby, we aim to identify the factors that have an influence on the decision of vehicle manufacturers to adopt the concept of the vehicle digital twin. To unveil those factors, we conducted 30 expert interviews at two vehicle manufacturers in which we found benefits and barriers regarding the realization of a product digital twin. By conceptualizing the benefits and barriers, which have a positive or negative influence on the adoption decision from the perspective of the manufacturers, we were able to derive general influences on the adoption decision. In our adapted technology-organization-environment framework we found influences in the technological (i.e., product data and product heterogeneity), the organizational (i.e., business model orientation, readiness of internal processes, and digital culture), and the environmental (i.e., competitors' behavior, legal regulations, and customer demand) dimension.

Keywords: *Innovation Decision, Digital Twin, Product Lifecycle Management, Vehicle Industry.*

A previous version of this essay was published in the Proceedings of the European Conference on Information Systems 2020.

Brandt, N. and Ahlemann, F., "HOW DO YOU DRIVE? ANALYZING VEHICLE SENSORDATA IN PRODUCT LIFECYCLE MANAGEMENT SYSTEMS" (2020). In Proceedings of the 28th European Conference on Information Systems (ECIS), An Online AIS Conference, June 15-17, 2020.

Table of contents

List of Figures

List of Tables

1 Introduction

Recently, customer demand for digitized products with embedded sensors and communication technology increased. Connected devices like mobiles, wearables, and smart home gadgets continue to play an essential role in the daily life of many private consumers. In addition, global trends like the industrial internet of things underline the high expectations of organizations to improve their businesses with the support of connected assets (Zehnder and Riemer, 2018, p. 4224). Vehicles, i.e., passenger cars and commercially used trucks, are important transport mediums for private and commercial users. Thus, the vehicle industry is not spared by the market demand for connectivity, and 340 million cars are expected to be connected to the internet of things (IoT) by 2023 (Statista, 2018). With product embedded sensors and connectivity units, detailed information about the product´s usage and behavior becomes available over the global internet (Kiritsis, 2011, p. 482). Thereby, product status information which manufacturers could typically only access in case of physical contact, for example, during maintenance, become constantly available (Hribernik et al., 2006, p. 368). When detailed information about individual vehicle usage are stored in a suitable data model, the vehicle digital twin - a virtual model of each product - can be realized (Glassgen, 2012, p. 7). By analyzing these twins in the manufacturers, information systems (IS), knowledge-based decisions can be made to design physical parts based on measured loads at customers' products (Holler et al., 2016, p. 485). Examples from the software industry show that automatic feedback from defective products about the cause of the failure in the context where they occur is a valuable source for organizations to improve product recall processes and adapt the design of future vehicles to prevent such failures (Teixeira et al., 2018, p. 1). Furthermore, analyzing usage data in the vehicle industry may help manufacturers innovate their processes and differentiate from competitors with new services to achieve a sustainable advantage (Saarikko et al. 2017, p. 668). Zolnowski et al. (2016, p. 13) identified that both customers and manufacturers might be able to improve their business processes and increase productivity by analyzing information from connected devices. To realize the benefits that arise from connected product sensor information, manufacturers need to adapt the concept of the vehicle digital twin. Subsequently, they need to introduce a new IS that enables a systematic analysis of such product lifecycle information from product instances. However, as discussed by Tao et al. (2018, p. 3563-3564), current research lacks such integration of product usage data to individual digital product twins, which hinders the analysis of these data for design, manufacturing, and service activities. Our study aims to understand the impact of product usage information from individual product instances on vehicle manufacturers. Thereby we define our research questions as follows:

Which factors influence the decision of vehicle manufacturers to adopt the concept of the vehicle digital twin on a firm level?

We conducted semi-structured interviews in the passenger car and truck industry to answer this question. Conducting expert interviews in industrial companies is promising, as our research question deals with an adoption decision on a firm level. Thus, we gathered the empirical data for our research on where the adoption might occur. During the interviews, we first asked the experts how a digital twin of each vehicle could enhance or does already support the execution of their or any other lifecycle tasks since these benefits positively influence organizational adoption decisions. Furthermore, we asked the experts about their experience with challenges gathering and analyzing data about individual products as those hinder organizations from collecting and/or analyzing usage data and thus negatively influence the adoption decision. After that, we used the technology-organization-environment framework (TOE) from Tornatzky and Fleischer (1990, p. 154) to conceptualize the benefits and barriers of the interviews. The remainder of this paper is structured as follows: In section 2, we define the conceptual foundations on which the paper builds o. After that, our data collection and analysis process is described in section 3. The content analysis in section 4 summarizes our findings and is followed by a discussion and conceptualization in section 5. The paper is concluded with an outlook in section 6.

2 Conceptual Background

2.1 The digital Twin

The idea of the digital twin was introduced in 2010 by the National Aeronautics and Space Administration (NASA) for their concept to increase transparency about space shuttle components' degree of wear (Shafto et al., 2010, p. 11). According to their often-cited definition, the digital twin is a virtual representative of an as-build physical product (Glassgen, 2012, p. 7). Its behavior mirrors the physical products’ status through updates from data sources like sensors or maintenance reports (Shafto et al., 2010, p.11). The quality of such a twin is limited by the amount and accuracy of available product information about the objects (Zehnder and Riemer, 2018, p. 4224). Sensors within many technology products already gather detailed information about the product status to enable product functionalities (Schütze et al., 2018, p. 360). When these data can be made available from outside the product, a detailed digital twin of those physical objects can be realized (Haag and Anderl, 2018, p. 64). Thereby, manufacturers can measure how each product is used and behaves after it is delivered to the customer (Haque et al., 2014, p. 8).

2.2 Extending Product Lifecycle Management Information Systems

To systematically store and analyze product-related information, industrial companies follow the strategic approach of Product Lifecycle Management (PLM) (Ameri and Dutta, 2005, p. 577). Nonetheless, its lifecycle definition, PLM IS, mainly focuses on the phase Begin of Life (BOL), in which the product is developed and manufactured (Bruno et al., 2015, p. 41). Product-related information from the phases Mid Of Life (MOL), in which the product is used and maintained, and End Of Life (EOL), in which the product is disposed or recycled (Kiritsis et al., 2003, p. 190), are thereby not considered in PLM IS. Thereby a lack of IS to manage MOL and EOL data, including but not limited to data from the digital product twin, can be observed (Hribernik et al., 2013, p. 86). Companies face challenges in integrating IoT data from the digital product twins into their PLM IS (Capgemini, 2019, p. 17). With the concept of closed-loop information flows in PLM, Kiritsis (2011, p. 482) suggests that organizations improve their processes by utilizing information from connected products to enrich traditional PLM IS with MOL and EOL product information. For this concept, it was already identified in literature that opportunities to improve product design, increase sustainability, offer new services, and evaluate products remaining lifetime emerge. Nonetheless, either previous work focused on connected products to improve tasks in specific lifecycle phases like design and production (Holler et al., 2016, p. 485) (Boyes et al., 2018, p.2) or broadly deals with product-related data from all lifecycle phases (Kiritsis, 2011, p. 482). On the other hand, our research unveils specifically how the tasks along the lifecycle are performed or could be enhanced with information about the products' usage and the challenges. As defined by our research question, we do not limit our research to the manufacturers but also include external stakeholders that perform tasks along a product's life like customers, service providers, and governments.

2.3 Analyzing Benefits and Challenges with the Technology-Organization-Environment Framework

Our study investigates the adoption of a vehicle, the digital twin. Professionals expect this will discontinuously change the vehicle industry (Baker, 2012, p. 233). A frequently cited framework to research such a technology innovation adoption on a firm level is the TOE framework (Tornatzky and Fleischer, 1990, p.154) (Oliveira and Martins, 2011, p.110). According to the TOE, factors in the dimension of technology, organization, and environment influence whether an organization adopts an innovative technology (Oliveira and Martins, 2011, p.110-111). The technology dimension addresses the internal and external technologies relevant to the organization (Tornatzky and Fleischer, 1990, p. 154). The organizational size defines a firm's inner properties and culture that influence the innovation decision (Tornatzky and Fleischer, 1990, p. 152). This includes the infrastructure for data exchange between departments and lifecycle phases (Baker, 2012, p. 233). The environment dimension specifies the environmental factors influencing the organizations' decisions (Tornatzky and Fleischer, 1990, p. 154). Ortbach, Brockmann, and Stieglitz (2014, p. 11), who analyzed fourteen TOE-based studies, identified that the framework dimensions had been successfully utilized to understand the adoption of different IT systems. Furthermore, Baker (2012, p. 238-239) found that researchers used the TOE in prior studies for a wide range of innovative technologies to identify the positive and negative influences on organizational decisions to adopt specific technologies and IS. Pumplun, Tauchert, and Heidt (2019, p. 12) also proved that the framework is suitable for adopting concepts like artificial intelligence on a firm level. According to Stieglitz et al. (2018, p. 8-10), who researched social media applications in crisis management, the TOE can qualitatively categorize challenges in adopting innovative technologies. During their study on adopting e-commerce, Zhu, Kraemer, and Xu (2002, p. 339) also identified positive influences using the TOE. Most TOE frameworks were developed for software innovations used inside a company. For the vehicle, digital twin physical and communicating hardware is used outside the company and changes in external processes. Therefore, we think the TOE for the vehicle digital twin differs significantly from previous TOE frameworks. Through the conceptualization of our findings, researchers and practitioners can apply the extracted knowledge to find out why the vehicle digital twin is or should be adopted by a specific vehicle manufacturer or not.

3 Research Method

We chose to conduct research in a practitioner's environment, as our research question asks for the adoption decision on a firm level and practitioners are most involved in this issue. More specifically, we focused on the vehicle industry because vehicles already gather information with embedded sensors, and there is a trend for vehicle connectivity (Statista, 2018). Furthermore, qualitative research was more promising than quantitative as we aim to explore new and unknown aspects. Mayring (2001, p. 2) criticizes that a qualitative approach generally risks lacking transparency and being systematic. This will be countered by following Myers and Newman (2007, p. 16) seven guidelines for qualitative research. The interviews took place at two companies, CAR, a manufacturer of passenger cars, and TRUCK, a manufacturer of trucks and buses. According to two experts, CAR and TRUCK are ideally suited to conduct expert interviews regarding the digital twin, as more than 90% of their vehicles already have an embedded connectivity unit. In addition, diverse stakeholders in these companies already require information about individual product usage (Terzi et al., 2010, p. 365). Traditionally CAR and TRUCK use separate tools or gather necessary information at service points, which the vehicle digital twin can deliver more accurately. Our interviews focused on determining which tasks along the vehicle lifecycle can potentially or already benefit from a detailed and accurate digital twin continuously updated by vehicle sensors data and what the barriers are doing so. Furthermore, we were interested if there are already measures to lower the barriers.

From an organization's perspective, the tasks along the product lifecycle differ between the design, production, and use phase and within a single phase. Hence, we decided to interview different experts from CAR and TRUCK from all lifecycle phases. Nevertheless, most of the experts work in use phase-related departments. That is because, generally, experts from a MOL role are responsible for gathering and analyzing MOL data even if the tasks have implications on design and manufacturing. Before the candidates for the interview were asked to participate in the study, the organization's approval was obtained, and the works councils were informed. After that, potential experts were contacted by phone based on their job descriptions and recommendations from CAR and TRUCK employees. In this first call, it was clarified whether the potential participants were experts in the field of this study and if they voluntarily wanted to participate. In case of interest in participating, a research summary was sent to the experts by mail. The interviews were conducted at the interviewees' workplace or via phone in a secure environment. Additionally, the interviewer works for the same employer without prior work-related contacts. This helps to ensure that the researcher's interests don´t influence the responses. In addition, this research topic is not critical regarding personal interests as we aim to document actual organizational activities and explicitly encourage the encourage participants to mention positive and negative factors (Potter and Hepburn, 2012, p. 565). Table 3-1 gives an overview of the profiles of the interviewees. In total, our study is based on 26 semi-structured interviews with 30 experts conducted between July and October 2019, with one or two interviewees at a time, either in person or via video conference. Seventeen interviewees have expertise in development, 11 in production, 25 in MOL, and 7 in EOL. Each interview lasted between 29 and 85 minutes, with a median duration of 45 minutes. After permission was granted, each interview was recorded. We ended our interviewing process after no more new insights were gained. All interviews were conducted in German, which is the first language of most experts, by using an interview guideline. Four knowledge-specific questionnaires (Design, Production, MOL, and EOL) were derived from the main guideline concerning our experts' diverse backgrounds of exact quotes in our analysis and discussion, and the interviewees' statements were directly translated into English. The questionnaires are structured into four paragraphs as follows. After a short introduction of the interviewer and the scope of the interview, the focus was on understanding the interviewee's tasks and their (potential) interaction with usage data. Afterward, the benefits of a vehicle digital twin were unveiled, and questions were asked to understand the challenges that hinder the creation and use of a digital twin for their tasks and the measures taken to overcome these. We concluded the interviews with open questions about missing and unasked aspects. To explore new and unknown elements, we used open-ended questions encouraging experts to give new insights into benefits, challenges, and measures from their perspectives (Yin, 2011, p. 135).

Company	Role	ID	Job description	Des.	Prod.	MOL	EOL
CAR	Des	I-01	Engineer Design Optimization	X		X	
		I-02	R&D Vehicle Concepts	X			
	Prod.	I-03	Production Strategy		X	X	X
		I-04	Quality Control		X		
		I-05	Digital Production		X	X	X
	Use	I-06	Service Process	X		X	X
		I-07	Quality Management	X	X	X	
		I-08	Connected Car Operations			X	
		I-09	Connected Car Operations			X	
		I-10	Service Process Diagnosis			X	
		I-11	Predictive Maintenance			X	
		I-12	Predictive Maintenance	X		X	
		I-13	Connected Car Retrofit	X		X	
		I-14	User Experience Analysis	X		X	
		I-15	Connected Car Projects	X		X	
		I-16	Swarm Data Aggregation	X		X	
		I-17	Product Care	X	X	X	X
		I-18	Product Care	X	X	X	X
		I-19	Connected Car Services			X	
		I-20	IT Architecture	X		X	
		I-21	Digital Business Models			X	
Company	**Role**	**ID**	**Job description**	**Des.**	**Prod.**	**MOL**	**EOL**
TRUCK	Des.	I-22	Self-Driving Car Engineer	X		X	
		I-23	PLM Manager	X	X		X
	Prod.	I-24	Product Structure	X	X		
	Use	I-25	Service Manager			X	
		I-26	Driving Analysis	X	X	X	
		I-27	Digital Services			X	
		I-28	Product Reliability	X	X	X	X
		I-29	Digital Transformation	X	X	X	
		I-30	IT Architecture			X	

Table 3-1 Interviewees and their professional background

After transcription, we pseudonymized the interviews to ensure the anonymity of the experts while simultaneously being able to correlate given statements with the function and employer of the interviewee. After that, we coded each interview using MAXQDA to increase the consistency of our terminology (Yin, 2011, p. 185). We followed the qualitative content analysis by Mayring (2000, p. 213) to extract benefits and challenges influencing the vehicles manufacturer's decision to digitize their products and realize the vehicle digital twin. In the first step, we derived Level 1 or open codes in an inductive approach (Yin, 2011, p. 187). The code system was considered complete after eight interviews when no new first-level codes appeared. The final design was developed by grouping these codes into category codes (Yin, 2011, p. 188). In the last step, all interviews were coded deductively by applying the category codes (Mayring, 2000, p. 213). In the application phase, the interviews that were already analyzed to derive the first-level codes were coded again to keep the analysis consistent.

4 Empirical Findings

4.1 Benefits

Vehicle-embedded connectivity units and the vehicle digital twin enable new applications to create value along the lifecycle. Through this characteristic, applications have a positive influence on the adoption decision of manufacturers. In Table 4-1, the identified applications are grouped into benefits.

TOE dimension	Benefit	Application
Organization	Design based on measured demand	Mechanical design based on vehicle usage Development of functions based on user interaction
	Improve production process	Transparency for vehicle logistics Precise and fast fault process
	Reduce repairs during warranty	Decrease warranty costs Greater customer loyalty
	Optimize the failure process	Prediction of product faults Understand faults context Lifelong failure measurement
	Business model innovation	Sales / Subscription of additional services Extent business model to other companies
Environment	Fulfilment of legal requirements	Ensure products' safety Documentation to avoid liability
	Customer satisfaction	Vehicle independent services Steering of vehicle functions Services based on vehicle information
	Improved service	Predictive maintenance Fast and transparent maintenance, repair, and overhaul
	Third party business models	Additional customer services from external providers Vehicles as environmental sensors

Table 4-1 Benefits from connectivity units within vehicles to improve lifecycle tasks

Benefits through connected vehicles that enable the analysis of usage data were found in the two TOE dimensions, organization, and environment. First, future vehicles can be designed based on the measured usage information, i.e., actual customer demand. I-01 stated that requirements "defined at some time by somebody based on random samples in the field" could be replaced. This is important as several customer groups with different mobility requirements use the same type of vehicle. Thus, I-01, I-07, I-11, I-12, I-18, I-25, and I-26 named it a benefit to gain insights into individual customers driving patterns to improve mechanical design for higher durability. According to I-9, I-16, I-23, and I-27, manufacturers need an "understanding how the vehicle functions work. How often they are used" (I-14) is seen as important information for prioritization and further development.

I-04, I-05, I-15, and I-18 from CAR also stated that a vehicle digital twin could benefit the production process. I-15 mentioned that the digital twin could be used after final assembly "tell logistics where individual cars are (and prevent thefts, as) vehicles get stolen from production." Furthermore, I-04 and I-18 expect that defects, which occur during product usage, can be directly correlated with sensor data from the fault-causing process in production to fasten the quality process.

According to I-11, I-20, and I-30, a vehicle digital twin can also reduce customer visits to service points during the warranty period. With wireless data transmissions, service costs can be eliminated in case of software updates or at least minimize service time and thereby costs if failure data are wirelessly transmitted to the workshop before the visit. I-01 and I-07 from CAR explained that fault reductions could significantly reduce warranty costs and risks. Furthermore, I-09, I-10, I-13, I-16, and I-20 stated that customer loyalty to the manufacturer owned service stations could be increased through individualized offers through direct interaction with the customer without the need to wait for the customer to initiate the contact. Without these active offers, I-28 mentioned that customers often “choose independent service providers after the warranty period ends.”

Furthermore, handling product failures were mentioned to benefit from connected vehicles. I-07, I-18, and I-30 stated that product usage data could be analyzed to predict product failures. More precisely, I-07, I-09, I-20, and I-28 see a potential to identify warranty risks in the market before the vehicles arrive at service points. Based on several expected failures, I-07 stated that manufacturers could decide “whether to put 200.000 or 3 million spare parts to the inventory”. In addition, when manufacturers gain access to vehicle information, details about a failure context are expected to support the avoidance of faults in future vehicle designs (I-04, I-14, and I-18). I-11, I-20, and I-28 added that connected vehicles are valuable data sources, enabling manufacturers to learn from failures treated at third-party service points after the warranty period has ended.

As stated by I-03, embedded connectivity units are also “driven by business models” to increase revenues. I-09, I-13, I-19, I-29, and I-30 describe lifelong revenues through digital services as essential applications. In addition to services for own vehicles, it was stated by I-25 and I-27 from TRUCK that services are already offered to customers of foreign manufacturers by using a retrofit connectivity unit on standardized vehicle interfaces. Thereby, digital services become hardware-independent competitive factors.

In the environment dimension, the government, customers, and third parties were mentioned to benefit from connected vehicles and the available information about product usage. When gathering fault information from products, I-07, I-15, I-18, and I-20 from CAR expect to fulfill better legal requirements of ensuring product safety. I-01 mentioned, “it is a legal requirement, and thus, it is obligatory to have a connectivity unit in the vehicle” to improve medical support in case of accidents. This is reported to be a significant reason for connectivity units within nearly all vehicles. Furthermore, I-03, I-11, I-20, I-23, and I-28 said that vehicle connectivity units could be used to update product configurations automatically after parts are replaced for lifelong documentation. I-23 named the "documentation which software is installed on the control units of actual vehicles" for strict homologation reasons as an example.

I-20 mentioned that his employer sees the vehicle digital twin as a measure to increase customer satisfaction by "adding value to the customer with digital services besides the vehicle." I-09, I-10, I-13, and I-20 from CAR stated that customers require general online functions such as park services and online news. I-09, I-16, and I-21 additionally identified a demand from customers to steer vehicle functions like the lock, the auxiliary heating, and pre-setting of navigation directions online. I-08-13, I-15, I-16, I-19, I-20, I-25, I-26, and I-30 from CAR and TRUCK, on the other hand, mentioned that services, which are based on vehicle information such as driving analysis or displaying the vehicle health status, are relevant to customers. Those services were explained to guide efficient driving or simplify the owners/drivers processes, thus applicable to private and business customers.

Besides the previously mentioned organizational benefits regarding the service process, the customers are described as benefiting from an improved service process. I-01, I-07, I-11, I-13, I-20, I-25, and I-27 expect faults to be predictable if manufacturers can understand the context of other vehicles. Additionally, I-10, I-12, I-13, I-20, and I-30 described that the transparency about the vehicle status increases transparency for customers and the workshop to prepare and fasten the repair process. Transparency is also of interest for commercially used vehicles, as the drivers do not report vehicle damages. I-01 stated that those drivers often "park the vehicles on the yard as long as they drive, put the key to the office and just leave."

I-15 and I-16 from CAR also pointed out that data from connected vehicles can be beneficial to external stakeholders as sensor information enable third-party business models. Besides services offered to vehicle owners by external partners, e.g., insurance based on the driving pattern, the experts reported that vehicles could be used as environmental sensors to measure the weather and determine road quality.

4.2 Challenges

The identification of applications and benefits showed that individual vehicle sensor information offers manufacturers new opportunities to improve their value-creation process. Nonetheless, in Table 4-2, we aggregated the barriers that hinder the holistic realization of the vehicle digital twin and its benefits. Furthermore, we summarized measures of how CAR and TRUCK aim to overcome the issues.

TOE dimension	Challenge	Measures
Technology	Data only locally available	Bring transceiver module to each product Temporary installation of transceiver hardware Retrofit of existing vehicles
	Traditional PLM IS	Interfaces between data lakes Development of a specific digital twin IS
	Complex big data analysis	Local Preprocessing Local storage Simple analysis algorithms
	High product variance	Normalization of models, sensors, and values Central communication gateway Modular operating system
	Long lifetime of products	Update of components and services
Organization	Limited warranty period	Extent warranty period Lifelong wireless aggregation of fault information
	No holistic strategy	Increase management attention Holistic identification of applications
	New competencies required	External partnerships Internal startups / labs Open innovation
Environment	Low customer acceptance	Monetary incentives Non-monetary incentives Professional drivers agree to gather data in their workers contract
	Data Protection Regulation	Anonymization and pseudonymization

Table 4-2 Challenges with a negative influence on the innovation-decision and measures taken in practice to overcome these

Challenges in developing connected vehicles and utilizing insights into product usage are found in all TOE dimensions. We found that practitioners see ten barriers that hinder the manufacturers from holistically gathering information about the vehicles' usage and analyzing them during BOL and MOL. In the technology dimension, I-04, I-8-16, I-18, I-25-27, and I-30 stated that many sensor data are only locally available within the vehicles, and due to the traditional vehicle architectures, these data cannot be accessed centrally. According to I-20 and I-25, connectivity units in every newly produced vehicle already access with a limited set of usage data. Due to technological limitations, powerful data loggers are occasionally applied to customers' vehicles to gain additional insights (I-01, I-15, I-18, I-26). Furthermore, I-01, I-07, I-10-13, and I-30 stated that retrofit solutions for standard vehicle interfaces are suitable for gathering data from – primarily older – vehicles that do not have an in-built connectivity unit.

I-07, I-13, I-23, I-24, and I-28 described a lack of PLM IS to handle usage data and enable the identified benefits along the lifecycle and across lifecycle phases. In addition, the diverse systems require technical development of individual interfaces between the data lakes and organizational efforts, according to I-07, I-17, I-26, and I-29. Moreover, I-23 from TRUCK stated that manufacturers must develop systems to store and analyze usage information alone due to the lack of available market solutions.

A group of experts also stated that it is challenging to systematically transmit and analyze the high velocity, variety, and volume of data that is created by the vehicle sensors (I-01, I-04, I-07, I-10, I-11, I-13, I-15, I-18, I-20, I-23, I-25-27, and I-30). As a measure, I-16, I-20, I-25-27, and I-30 saw a need to pre-process the vehicle's sensor data, thereby reducing the amount of transmitted and centrally stored data. I-12, I-13, I-18, I-26, and I-27 also stated that the actual focus lies on algorithms that are easy to compute, like summations to measure usage frequency due to the lack of experience and low pre-processing capabilities of existing vehicles.

I-07, I-11, I-13, I-16, I-20, and I-24 also reported that the numerous vehicle variants with different sensors that depend on the vehicle equipment and the high number of connectivity unit versions impede frequent communication. As a measure, I-13, I-16, I-20, I-24, and I-26 saw the need for a normalization layer between the vehicles and their virtual twins. Furthermore, the need for a central communication gateway inside the vehicle that is compatible with prior versions and one operating system that runs on all vehicles was stated as measures from I-11, I-21, and I-30.

According to I-01, I-14, I-15, I-21, and I-25, the long lifespan of a vehicle is a challenge that requires services and software that can be updated while maintaining the homologation criteria.

From the perspective of an organization, missing interaction with the customers in service shops after the warranty period ended was reported as a challenge by I-05, I-12, I-18, and I-28 as the most relevant faults to improve the reliability of future products are stated to occur after a couple of years. Besides equipping the vehicles with connectivity units to gather failure information directly, extending the warranty period was identified as a measure by I-18 and I-28.

As another challenge, I-01, I-12, I-14, I-15, I-18, I-20, I-23, I-29, and I-30 pointed out that the vehicle industry traditionally has no focus on vehicle usage data, and a holistic data strategy is still missing. As an action, experts identified the need to aggregate the benefits and derive the requirements for a data model and a vehicle architecture. This was mentioned to be a necessary basis to justify the implementation effort (I-14, I-16, I-18, I-23, I-30) and to increase the internal prioritization of the vehicle digital twin through management attention (I-12, I-14, I-20, I-23).

In addition, missing expertise was mentioned as challenging in the organization dimension by I-07, I-18, I-2,3, and I-27 as traditional competencies are related to vehicle manufacturing rather than big data analytics. Measures range from external partnerships in the automotive company (I-13, I-16, I-20) to internal start-ups (I-30) and open innovation hackathons (I-29) in the truck company.

In the environment dimension, I-01, I-07-16, I-18, I-20, I-21, I-25, I-26, and I-28 mentioned the low customers' acceptance for the manufacturer to analyze the vehicle usage were often named as a challenge since the data about the vehicles' use reveals detailed information about the vehicles' users. Some interviewees reported that incentives are often chosen to get the agreement of the customers to analyze data about their vehicles' usage. I-09, I-13, I-15, and I-30 mentioned price reductions of existing services as incentives, while I-07, I-14, I-16, I-27, and I-29 mentioned completely new services unavailable without sharing usage information. I-25 and I-27 from TRUCK also explained that their customers, i.e., fleet managers, already address the driver's privacy concerns in the work contracts, as they want to analyze vehicle usage information to optimize their processes.

Due to the data protection regulations, experts saw the need to minimize the storage of customer information. According to I-01, I-08, I-10-13, I-15-17, I-23, I-26, and I-30, anonymization algorithms are applied in cases where no reference to a specific vehicle is needed. In contrast, pseudonymization is used when gathering information need to be consistently mapped to a particular vehicle.

5 Discussion

In prior research, the TOE was found to be a generic theory that explains the influences on the adoption of new technologies on a firm level (Baker, 2012, p. 237) (Pumplun, Tauchert and Heidt, 2019, p. 12). According to the TOE definition, innovative technology might enhance an organization's competency (Baker, 2012, p. 233). This is the case for manufacturers who realize the digital product twin and holistically analyze the usage information for lifecycle optimizations (Baker, 2012, p. 233). Figure 5-1 depicts our TOE framework that explains the adoption of the vehicle digital twin from the perspective of an organization. We propose eight factors influencing the adoption of the TOE dimensions by conceptualizing the benefits, barriers, and measures unveiled during our interview study. We argue that benefits, barriers, and measures can be generalized to have either a positive, no, a neutral, or negative influence on the organization's decision to adopt the vehicle digital twin. The direction of the influence, i.e., positive, neutral, or negative, depends on the specification of the criteria in the organization.

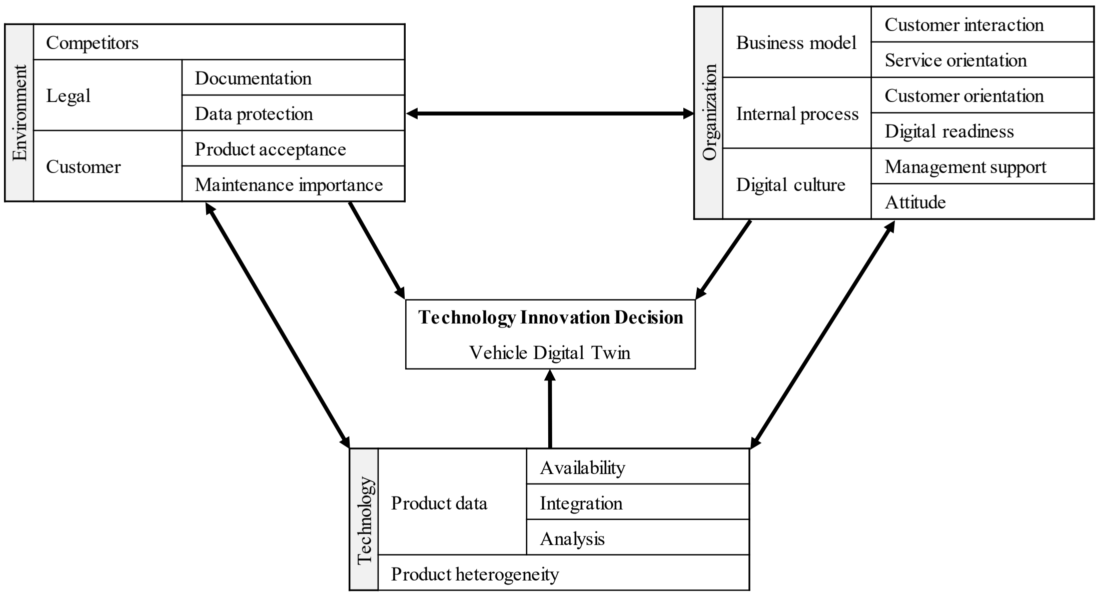


Figure 5-1 **The adapted TOE for the vehicle digital twin on a firm level**

Proposition 1: Vehicle sensor data availability and the IS capability to integrate and analyze these data are technological prerequisites to realize the vehicle digital twin.

In our study, we unveiled that in CAR and TRUCK, not all vehicle information could be accessed wirelessly, and the PLM IS are not designed to gather and analyze data from individual vehicles. Furthermore, the organization's lack of competency and expertise in interpreting available information was challenging. I-20 illustrated the traditional focus on the vehicle hardware and driving functions rather than sensor data analytics by reporting that most automotive companies "build cars (…) rather than smart devices with a car as a case". Thus, in the two case companies, the criteria product data negatively influence the adoption decision as enabling vehicle connectivity and changing the company's IT infrastructure requires high efforts. Nonetheless, other manufacturers might have continuous and wireless access to sensor data of their vehicles, integrate them into their ISs, and have experience in analyzing these data. For these manufacturers, the product data criteria have a neutral influence on the concept of the vehicle digital twin depending on the degree of realization of these prerequisites.

Proposition 2: The more heterogeneous the manufacturer´s range of vehicle architectures is the more complex and likely is a holistic implementation.

We found that the wide variety of models and vehicle features that can be individually configured by the customers and the long life of products with new models regularly might lead to different vehicle architectures. I-20 concretized this by stating that some vehicles seem similar, but "none of them works technologically identical to another." Such a variety of physical architectures was mentioned to hinder wireless access to vehicle sensor data. Therefore, data gathering must be individually configured for each vehicle architecture. Thereby, high heterogeneity was found to have a negative impact on the innovation-decision. On the other hand, a low product variety, or rather a low variety of architectures and connectivity interfaces, have a positive effect.

Proposition 3: The need for customer interaction to maintain the traditional business model and the organizations orientation on digital services constitute an influence.

During our study, it was mentioned that information about product usage is essential for the two manufacturers to reduce failures of future vehicle generations and consequently guarantee costs. Thus, vehicle sensor data enable CAR and TRUCK to maintain and improve their traditional business model. Hence, we conceptualize that an organization's need for customer interaction positively influences the decision to adopt the vehicle digital twin. On the other hand, there is no influence when the manufacturer has no demand for customer input. In addition, experts from CAR and TRUCK see a trend in their employers to extend the hardware-oriented business model with digital services. I-27 reported on digital services that it "becomes more important to deliver an on top value" to the traditional vehicle. The increasing interest of manufacturers to offer digital services to vehicle customers positively influences the decision to adopt the digital twin concept. This is because service-driven business models require more information about the product usage than a business model that is orientated on one-time sales of physical hardware. Accordingly, a focus of the business model on hardware has a neutral influence on the adoption decision.

Proposition 4: Whether the internal processes are oriented towards customer interaction and digital ready has an impact from the organization's perspective.

The customer orientation criteria specify if the organization's internal processes along the product lifecycle are already designed to integrate any kind of usage information from individual vehicles. In our study, the measurement of market demand for a customer-oriented product design was mentioned as important information for the company. I-14 summarized that connected vehicles enable product design to measure "real customers that are driving on the roads. That are obviously the most relevant data". Nonetheless, the additional benefits connected vehicles offer to improve design, manufacturers already have processes established to measure customer demand with test vehicles and customer questionnaires, for instance. Another example of established processes of manufacturers to analyze MOL information are vehicle failure data that are measured at service points and unveil errors in the manufacturing process. I-04 explained that he "checks the failure reports and groups them together with codes" to find more significant issues that are caused by manufacturing defects. Having the general internal processes and organizational structures already established for applications like customer-oriented design or manufacturing failure detection, the benefits from a vehicle digital twin can be realized with less effort. In addition, realizing the benefits requires the organizational process to be digital mature so that vehicle sensor data can be analyzed across lifecycle stages. A high digital readiness makes the organization's adoption more likely, while a lower readiness hinders the adoption. Internal process criteria have a neutral impact when the processes are customer orientated and digital mature. On the other hand, this criterion has a negative impact for an organization without customer oriented or immature digital internal processes.

Proposition 5: The digital culture in the sense of management support and overall employee's attitude influences the adoption of the vehicle digital twin.

I-10 reported, "it is more a topic of prioritization of resources" to realize the vehicle digital twin rather than missing or immature technologies on the market. For a positive decision to priorities, management needs to understand the overall benefits and chances to support the concept. Thereby I-23 stated that it is essential to develop a holistic strategy and analyze "which data are generated, which do we need and how do I use them over the lifecycle". In addition, we found that the digital attitude of employees influences the strength of an organization to use the vehicle digital twin. When an organization has a digital culture, their employees are likelier to make decisions based on vehicle usage information, that is essentially customer feedback. All in all, management support and a digital employee culture positively impact the innovation-decision. Missing management support, on the other hand, has a negative impact due to lower prioritization and fewer resources. In addition, a lack of digital culture has a negative influence as it leads to fewer employees' intent to use and further develop applications and IS.

Proposition 6: Competitors' offerings of digital services that are based on the digital twin impact the manufacturers' adoption decision.

It was mentioned by I-08 that the customers "enthusiasm for digital services is high". If these digital services that are based on vehicle sensor data fulfil customer demand to a higher degree than comparable services from competing manufacturers, product innovations are a suitable measure to gain a competitive advantage. If competitors on the other hand already fulfil customer demand to a higher degree with these digital products, following competitors' innovations enables a manufacturer to stay competitive. Both making a product innovation and following competitors has a positive impact on the adoption decision. The strength of this impact then depends on the expected influence of these digital services and functions on customer's perceived benefit. A high effect has a positive influence whereas a low or no effect has a neutral.

Proposition 7: Governmental obligations for product documentation and customer data protection regulations influence the implementation decision.

According to I-12 manufacturers can be obligated to "continually improve the product" and exchange information about the product usage with the government. Due to the mandatory nature of these laws, legal requirements for documentation have a positive influence on the adoption of the concepts. The more information is required by law, the stronger is the positive effect. On the other hand, I-20 highlighted that governments worldwide limit the amount usage information that companies can gather and analyze, as they do "not allow to perform data hoarding". Depending on the rigor, data protection regulations might have no or a negative influence.

Proposition 8: Customers influence the adoption decision by their acceptance of digital services and functionalities and their perceived benefit through maintenance reductions.

As stated by I-10, "creating value for the customer" with digital services and functions is one of the main applications of vehicle usage data. Thus, experts implicitly assumed a will from customers to use digital products. Such a customer willingness to use additional digital products affects the manufacturer decision to develop those and subsequently realize the vehicle digital twin as a data source. In addition, a faster or eliminated service process in the workshops through a vehicle digital twin was unveiled in the interviews as a customer benefit. Nonetheless, the impact of these benefits on the manufacturer's decision depends on the general importance of the maintenance process and their problems with the actual process. The more important maintenance seems to be from a customer perspective the more do the customers appreciate the improved maintenance process. Depending on the importance that the customers attribute to the new services, function and a better maintenance support, the customer criteria have a neutral to positive influence.

6 Conclusion

Connectivity units bring digital functions and services to vehicles that private and commercial customers of technological products expect. In our study, we identified that the influence of digitized vehicles goes beyond fulfilment of this customer demand. Product embedded connectivity units make local sensor data about the products status and usage available worldwide, which enables a detailed vehicle digital twin of each individual product. Based on 30 expert interviews our qualitative study creates a deep understanding about the influences on the manufacturers' decision to adopt the concept of the vehicle digital twin.

By replacing guesses and inaccurate information about the usage with detailed data from individual products, IoT embedded products promise to be competence enhancing when information is systematically gathered and analyzed. With connected vehicles, customers can influence and improve the value creation process while using the product. In total, we identified nine benefits that can be realized with a vehicle digital twin for the manufacturers, the government, and the customers. Thereby, our study encourages (vehicle) manufacturers to analyze information from product-embedded sensors for lifecycle innovations and improvements. Moreover, ten challenges were identified which hinder the holistic concept to be realized. For each challenge, the manufacturer's measures to overcome these (provisionally) were aggregated. Based on the results we derived eight groups of influences that explain the decision of manufacturers to adopt the concept of the vehicle digital twin. In addition to the explanation, they can guide manufacturers what need to be considered when they aim to enrich their PLM IS with IoT information. The results were conceptualized by using the TOE framework. We unveiled that a need for customer interaction to maintain the traditional business model and an orientation towards digital services can have a neutral to positive influence on the decision. Like that the fulfilment of market demand to stay competitive or to gain a competitive advantage as well as the regulatory requirement to analyze or transfer vehicle usage information might have a positive impact. Last the customer acceptance of new services and importance that customer attribute to a better maintenance process might positively or neutrally influence the adoption decision. On the other hand, missing product data availability and capability to integrate and analyze these data as well as the heterogeneity of vehicles architectures were found to have a neutral to negative influence. Furthermore, a neutral to negative impact on the adoption of the vehicle digital twin is linked with the property if the internal processes are oriented towards customer feedback and their digital readiness. In addition, data protection regulations might also have a neutral to negative impact. Last, the criteria of digital culture regarding management support and employees' digital attitude might have a positive or negative influence on the organizations decision to adopt the concept of the vehicle digital twin depending on the direction of the culture.

Although we rigorously followed our research agenda and academic well-established guidelines for qualitative research (Yin, 2011) (Myers and Newman, 2007), the paper at hand is not free of limitations. First, we conducted our interviews only at two companies to understand the implications of a vehicle digital twin along the whole product lifecycle. Nonetheless, other manufacturers might find other benefits, challenges, or measures. Second both manufacturers and our interview partners were situated in Germany what might have also limited our findings. Last, our qualitative approach does neither allow us to prioritize the benefits based on their potential impact nor does it allow us to evaluate the importance of a certain challenge or the suitability of any measure.

Following Yin's (2011, p. 221) guidelines to conduct qualitative research we conclude our research by pointing out further research opportunities. First, we propose to validate and extent the identified influencing factors at other vehicle manufacturers. Second, we expect relations between the influences themselves that should be investigated in further studies to find correlations. Those factors and correlations might then be used to explain the innovativeness of an organization. Third, the influences on the innovation decision of vehicle manufacturers to use IoT technology within their products and analyze usage information might be transferable to other industries. We expect that manufacturers of other complex technological products like planes, trains, mobiles, or machines have similar influences due to the similar size of most of those organizations, the already embedded sensor technology in their products and similar external stakeholders. Since we chose a qualitative and exploratory method, further research might quantitatively enrich our TOE framework with mixed methods, which is a theory for explaining, towards a higher-class theory for prediction (Gregor, 2006, p. 624-625). Based on this an artefact to gather and analyze product usage information in a systematic approach, which would be relevant to researchers and practitioners, can be developed.

7 References

Ameri, F. and Dutta, D. (2005). Product Lifecycle Management: Closing the Knowledge Loops. In: Computer-Aided Design and Applications. Volume 2, p. 577-590.

Baker, J. (2012). The Technology–Organization–Environment Framework. In: Information Systems Theory: Explaining and Predicting Our Digital Society. Volume 1, p. 231-245.

Boyes, H., Hallaq, B., Cunningham, J. and Watson, T. (2018). The industrial internet of things (IIoT): An analysis framework. In: Computers in Industry 101, p. 1-12.

Bruno, G., Antonelli, D. and Villa, A. (2015). A reference ontology to support product lifecycle management. In: 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering. Volume 33, p. 41-46.

Capgemini (2019). State- of-the-art IoT study - A generic maturity model of the "Internet of Things" across the manufacturing, automotive and healthcare industries. Capgemini invent.

Glassgen, E. H. and Stargel, D. S. (2012). The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In: 53rd Structures, Structural Dynamics, and Materials Conference, p. 1-14.

Gregor, S. (2006). The Nature of Theory in Information Systems. In: MIS Quarterly. Volume 30, Number 30, p. 611-642.

Haag, S., Anderl, R. (2018). Digital twin – Proof of concept. In: Manufacturing Letters, Volume 15, p. 64-66.

Haque, S., Aziz, S. and Rahman, M. (2014). Review of Cyber-Physical System in Healthcare. In: International Journal of Distributed Sensor Networks. Volume 10 Issue 4, p. 1-20.

Holler, M, Stoeckli, E., Uebernickel, F. and Brenner, W. (2016). Towards Understanding closed-loop PLM: The Role of Product Usage Data for Product Development enabled by intelligent Properties. In: BLED 2016 Proceedings. Slovenia: Bled, p. 479-491.

Hribernik, K., Rabe, L., Thoben, K. and Schumacher, J. (2006). The product avatar as a product-instance-centric information management concept. In: International Journal of Product Lifecycle Management. Volume 3, Issue 1, p. 366-371.

Hribernik, K., Wuest, T. and Thoben, K. (2013). Towards Product Avatars Representing Middle-of-Life Information for Improving Design, Development and Manufacturing Processes. In: IFIP International Conference on Digital Product and Process Development Systems. Berlin, p. 85-96.

Kiritsis, D., Bufardi, A. and Xirouchakis, P. (2003). Research issues on product lifecycle management and information tracking using smart embedded systems. In: Advanced Engineering Informatics. Volume 17, Issues 3-4, p. 189-202.

Kiritsis, D. (2011). Closed-loop PLM for intelligent products in the era of the Internet of things. In: Computer-Aided Design 43, p. 479-501.

Mayring, P. (2000). Qualitative Inhaltsanalyse. In: Handbuch qualitative Forschung: Grundlagen, Konzepte, Methoden und Anwendungen. Volume 1, Issue 2, p. 209-213.

Mayring, P. (2001). Kombination und Integration qualitativer und quantitativer Analyse. In: Forum Qualitative Sozialforschung / Forum Qualitative Social Research. Volume 2, Issue 1, p. 1-14.

Myers, M. and Newman, M. (2007). The qualitative interview in IS research: Examining the craft. In: Information and Organization. Volume 7, p. 2-26.

Oliveira, T. and Martins, M. (2011). Literature Review of Information Technology Adoption Models at Firm Level. In: The Electronic Journal Information Systems Evaluation. Volume 14 Issue 1, p. 110-121.

Ortbach, K., Brockmann, T. and Stieglitz, S. (2014). Drivers for the adoption of mobile device management in organizations. In: Proceedings of the European Conference on Information Systems (ECIS) 2014. Israel: Tel Aviv, p. 1-18.

Potter, J; Hepburn, A, (2012): Eight challenges for interview researchers. In: Handbook of Interview Research. Edition 2, London, England, p. 555-570.

Pumplun, L., Tauchert, C. and Heidt, M. (2019). A new organizational chassis for artificial intelligence - exploring organizational readiness factors. In: Proceedings of the 27th European Conference on Information Systems (ECIS), Stockholm & Uppsala, Sweden.

Saarikko, T., Westergren, U. H. and Blomquist, T. (2017) The Internet of Things: Are you ready for what's coming? Elsevier, vol. 60(5), p. 667-676.

Schütze, A. Helwig, N. and Schneider, T. (2018). Sensors 4.0 – smart sensors and measurement technology enable Industry 4.0. In: Journal of Sensors and Sensor Systems. Issue 7, p. 359–371.

Shafto, M., Conroy, M., Doyle, R., Glassgen, E., Kemp, C., LeMoigne, J. and Wang, L. (2010). DRAFT Modeling, Simulation, Information Technology & Processing Roadmap. USA: Washington, p. 1-32.

Statista. (2018). Prognostizierte Anzahl an vernetzten Fahrzeugen im Connected Car Markt weltweit* von 2017 bis 2023 nach Subsegmenten (in Millionen). Statista GmbH. Zugriff: 22. November 2019. https://de-statista-com.ub-proxy.fernuni-hagen.de/statistik/daten/studie/893915/umfrage/connected-car-anzahl-der-fahrzeuge-weltweit/

Stieglitz, S., Mirbabaie, M., Fromm, J. and Melzer, S. (2018). The adoption of social media analytics for crisis management — challenges and opportunities. In: Twenty-Sixth European Conference on Information Systems (ECIS2018). UK: Portsmouth, p. 1-19.

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H. and Sui, F. (2018). Digital twin-driven product design, manufacturing, and service with big data. In: The International Journal of Advanced Manufacturing Technology. Volume 94, Issue 9–12, p. 3563-3576.

Teixeira, C., Braga de Vasconcelos, J. and Pestana, G. (2019). A Knowledge Management System for Analysis of Organisational Log Files. In: 13th Iberian Conference on Information Systems and Technologies (CISTI). Spain: Casceres, p. 1-4.

Terzi, S., Bouras, A., Garetti, M. and Kiritsis, D. (2010). Product lifecycle management - From its history to its new role. In: International Journal of Product Lifecycle Management. Volume 4, p. 360-389.

Tornatzky, L.G. and Fleischer, M. (1990). The Processes of Technological Innovation. Lexington Books, Lexington.

Yin, R. K. (2011). Qualitative Research from Start to Finish. The Guilford Press, New York.

Zehnder, P. and Riemer, D. (2018). Representing Industrial Data Streams in Digital Twins using Semantic Labeling. In: IEEE International Conference on Big Data, p. 4223-4226.

Zhu, K., Kraemer, K. and Xu, S. (2002). A Cross-Country Study of Electronic Business Adoption Using the Technology-Organization-Environment Framework. In: Twenty-Third International Conference on Information Systems. Spain: Barcelona, p. 337-348.

Zolnowski, A., Christiansen, T. and Gudat, J. (2016) Business model transformation patterns of data-driven innovations. In: Twenty-Fourth European Conference on Information Systems Proceedings. Turkey: Istanbul, p. 1-25.

4

STRUCTURE OF A PATTERN LANGUAGE TO INTEGRATE VEHICLE SENSOR DATA INTO MANUFACTURERS' INFORMATION SYSTEMS

Brandt, Niklas, University of Duisburg-Essen, Essen, Germany

Ahlemann, Frederik, University of Duisburg-Essen, Essen, Germany

Abstract

Traditionally, vehicle sensor data are only accessible inside the vehicles for driving and functions purposes. By embedding connectivity units into individual vehicles, manufacturers gain access to these sensor data throughout a vehicle's entire life. Previous research has shown that it brings high value to manufacturers, customers, and other stakeholders when analyzing detailed vehicle usage data. However, missing guidelines hinder manufacturers from systematically gathering, managing, storing, and analyzing data from vehicle instances. In this paper, we introduce the structure of a pattern language to guide the design of an IS that enables the handling of vehicle usage data. Therefore, we split the overall IS design problem into thirteen sub-problems. By identifying and specifying the sub-problems, we define and contextualize the elements of a pattern language. We base our research on academic reference models, documentation of commercially offered platforms, and interviews in the vehicle industry.

Keywords: *Pattern Language, Reference Model, Internet of Things, Sensors.*

Table of contents

List of Figures

List of Tables

1 Introduction

Industrial companies traditionally lose transparency about the status of manufactured products after being sold to customers (Kiritsis, 2011, p. 482). Nonetheless, transparency about the status of individual products during their' usage would be of high value for manufacturers. In the design phase, manufacturers could benefit from insights into how the unique products are used. Based on this, they could derive, which existing functions are used, and which other functions would be of help for the customers. In addition, they could measure the lifetime loads on physical components and design the products in such a way that they resist the real loads throughout their lives and simultaneously avoid those parts that are oversized (Holler et al., 2016, p. 485). Furthermore, new services that rely on (real-time) usage data, such as predictive maintenance, could be offered to customers (Brandt and Ahlemann, 2020, p. 8). Such services would reduce product lifecycle costs and increase safety and convenience (Balasubramanian et al., 2016, p. 7). A technological approach supporting the concept of using real-time usage data from product-embedded sensors and connectivity units is the internet of things (IoT). In the IoT, knowledge about the status of individual products becomes available outside of the products. As vehicles have embedded various sensors for product functionalities (Gerla et al., 2014, p. 242), it seems that the hurdles to gather usage data from individual vehicles are low. Nevertheless, vehicle manufacturers' existing information systems (IS) are not designed to gather, manage, store, and analyze usage data from individual products over the IoT (Xin and Ojanen, 2017, p. 1099). Rather, IS such as Product Lifecycle Management (PLM) IS are designed to handle data about groups of products that remain relatively static along a product's life. Thus, new IS are required that handle usage data from individual vehicles (Yoo et al., 2016, p. 1053). To guide the development of such IS, guidelines for vehicle manufacturers to gather, manage, store, and analyze sensor data from individual vehicles are required. Researchers (Kiritsis, 2011, p. 484) (Hribernik et al., 2017, p. 1410), software vendors (Huawei Technologies Co., 2019) (IBM Corporation, 2017,) and the vehicle manufacturers themselves (WirelessCar, 2020) have already designed IS with this goal. Due to the topic's novelty, existing models lack sufficient maturity, are incomplete, or cannot be adapted due to missing adaptability into existing IS (Kaiser et al. 2018, p. 8).

As a result, guidelines are required to overcome the weaknesses of existing models. Contrary to models that already exist, i.e., reference models and software, we design a pattern language structure to solve the practitioner's problem uniquely and innovatively (Johannesson and Perjons, 2014, p. 7). A pattern language is well suited to cope with the weaknesses of existing models. The advantage of a pattern language over other artifacts is that it enables vehicle manufacturers to adapt certain parts of the overall speech and thereby complement their existing IS (Schönteich et al., 2018, p. 6). Manufacturers' can replace or adopt specific elements of the pattern language independently. Existing models, on the other hand, should be adopted entirely and do not treat the elements as exchangeable units. Therefore, we define our research question as follows:

What elements of a pattern language guide the design of an IS to gather, manage, store, and analyze sensor data from individual vehicles?

We answer the research question by conducting design science research (DSR). We analyze academic publications, documentation from software vendors, and expert interviews in the vehicle industry. In this paper, we introduce a pattern language structure that identifies and specifies the elements that an IS needs to gather, manage, store, and analyze sensor data from individual vehicles on four logical layers. This builds the basis for researchers and practitioners to develop one to many patterns for each element of the structure and creates a systematic guidance to do so.

We structure our paper as follows. First, an introduction of the technological key concepts and the notion of a pattern language as design artifact takes place in chapter 2. After that, the research methodology is presented in chapter 3. Chapter 4 presents the pattern types and the pattern language development process. Furthermore, it identifies and specifies the elements of the pattern language. The paper concludes with a discussion and an outlook in chapter 6.

2 Conceptual Background

2.1 The vehicle's digital twin

Sensors are devices that can transform physical phenomena into electrical signals (Marabelli et al., 2017, p. 352). In modern vehicles, various embedded sensors measure the status of built-in technical parts to enable driving and feature functionalities (Gerla et al., 2014, p. 242). By adding internet connectivity units (Statista, 2019, p. 5), manufacturers gain access to detailed and real-time status data of each manufactured vehicle throughout their whole individual lives (Westerbeek, 2016, p. 2). With the exchange of status data over the internet, the vehicles participate in an Internet of Vehicles (IoV) (Gerla et al., 2014, p. 1). Thereby, vehicle manufacturers can potentially gather these data from product instances and create a digital twin from each vehicle (Dietz and Pernul, 2019, p. 179). Academic definitions of such a digital twin only have in common that it is a virtual representative of an individual physical product (Enders and Hoßbach, 2019, p. 2-3). For us, a digital twin is a one-to-one virtual representative of a manufactured product that documents the status of each product along its life (Erikstad, 2018, p. 354). The virtual representative describes the status of the product in a measurable form in a product internal or external database. In addition to sensor data, the digital twin contains all information or a reference to the information that exist for the product instance. Examples include technical drawings, manufacturing instructions, customer feedback, and guarantee claim. This digital twin mirrors all product-related data over the whole lifecycle from design over manufacturing and usage to its disposal and updates information whenever available (Dietz and Pernul, 2019, p. 179).

2.2 The Digital Twin in Product Lifecycle Management

PLM is the strategic approach to systematically store, manage, and analyze product-related data along a product's life (Holler et al., 2016, p. 3). During the design phase, manufacturers create documents such as technical drawings, test results, and bill of materials for product classes (Terzi et al., 2010, p. 373). Production instructions are generated for product classes during the manufacturing phase while machine measurements from embedded sensors are gathered on an instance level (Huber, 2016, p. 38). The availability of detailed product-related data traditionally ends with the beginning of the product's usage. Only basic data from customer complaints and service point reports are available by then (Terzi et al., 2010, p. 364). These basic product usage data barely enable manufacturers to understand how each vehicle is used and behaves along the use phase (Yoo et al., 2016, p. 1053). Although detailed data about product usage and behavior are rarely available to vehicle manufacturers, researchers have already shown that these data are valuable for improving lifecycle related tasks. Moreover, Brandt and Ahlemann (2020, p. 6) found that other stakeholders, i.e., customers, government bodies and other organizations benefit from these data. Benefits include product usage-based services (Westerbeek, 2016, p. 4), product design optimizations (Holler et al., 2016, p. 485) and to fault recognitions (Främling et al., 2013, p. 798). Furthermore, other industries in which detailed product usage data are already available, such as the software and mobile phone industry, already show that this kind of data is beneficial.

Despite the high expectations, PLM applications that vehicle manufacturers use restrict data storage and analysis on design and manufacturing data from product classes. As a result, manufacturers need new IS to gather, manage, store, and analyze product instance data along the whole product lifecycles. With the concept of closed-loop PLM, Kiritsis (2011, p. 484) introduced the idea to close data loops between all phases of the product lifecycle. After the initial work, researchers developed different models to guide the design of such closed-loop PLM IS (Främling et al., 2013, p. 790) (Hribernik et al., 2017, p. 1410). Nevertheless, the missing modularity and maturity hinders a systematic realization. Thereby, researchers showed that there are unsolved problems in the vehicle industry that remain unsolved with the existing models (Brandt and Ahlemann, 2020, p. 8).

2.3 Design patterns in Information Systems research

Researchers of different domains continually develop models that guide practitioners to solve common problems. These models often summarize best practices how to solve multiple specific instances of one problem (Johannesson and Perjons, 2014, p. 29) (Alexander et al., 1977, p. 35). Therefore, the models are defined in an abstract manner so that they can be adapted and reused to solve many instances of the common problem (Aldin et al., 2009, p. 1340) (Schlieter et al., 2010, p. 413). Alexander et al. (1977, p. 11) were the first that called their model a pattern language when they designed a model to guide the creation of cities in the architecture domain. They documented 253 patterns, i.e., best practices, that city planners should follow to design a good city (Alexander et al., 1977, p. 9). Based on this initial proposal of pattern languages as a research artifact in architecture, Gamma et al. (1994, p. 99) picked up this approach in software development. Their often-cited pattern language led to the development of this model type for diverse problems in computer science (Feldhusen and Bungert, 2007, p. 125). Each pattern of such a pattern language describes a best practice on how to solve the specified part of the overall problem. Design patterns are derived from multiple cases in practice to proof their validity (Aldin et al., 2009, p. 1342). Pattern languages might also define anti-patterns (Winter et al., 2009, p. 469), which summarize solutions that are often observed in practice, but have a lack in solving the stated problem. Due to their practical relevance and their origin in research, Maaß and Storey (2015, p. 135) call pattern languages a mediator between engineering and design theory. IS researchers on the other hand often call their models reference models. As reference models have a similar aim and the term of a reference model is not standardized, researchers call reference models as the IS domain equivalent to pattern languages (Fettke and Loos, 2009, p.474). Due to missing standardization of the concepts of reference models and pattern languages, the model types cannot be separated sharply from each other. Nevertheless, the individual parts of a pattern language (patterns) tend to be modular and are explicitly defined self-contained so that they can be applied individually (Schönteich et al., 2018, p. 6). Due to this modularity, each pattern is formulated in a customizable way and allows the creation of various alternatives (Paschke, 2008, p. 1). On the other hand, reference models are most often intended to be adapted completely (Schermann et al., 2007, p. 188). Thereby, the documentation of reference models and their aim to provide complete solutions often leads to missing ability to adapt specific increments and complement existing systems (Keller, 2009, p. 472). Although this difference cannot be generally observed in pattern languages and reference models, we decide to call our artifact a pattern language. Thereby we acknowledge the observed differences in previous developments. Nevertheless, we keep the close relation of both models in mind in our further process and call for further research.

3 Research Methodology

Contrary to routine design, Design Science Research (DSR) deals with the design of artifacts with a low solution and / or application domain maturity (Gregor, 2013, p. 345). According to Gregor (2013, p. 345), the structure of a pattern language constitutes an invention, as we address an unsolved problem of the vehicle industry in a unique and innovative way. The design of an artifact in the DSR approach generally consists of the two activities build and evaluate (March and Smith, 1995, p. 254). Based on these basic activities, Ahlemann and Gastl (2007, p. 12) developed the process model for an empirically grounded reference model construction that explicitly guides the design process of reference models. In chapter 2.3 we already unveiled that there are no sharp differences between reference models and pattern languages. Thus, the process is generally applicable to our research. The process model proposes five consecutive phases. After a planning phase that includes the problem identification and the goal definition, an initial model is constructed in the second phase based on existing knowledge. During the third phase, the initial model is refined and validated, and its' operation is tested in the fourth phase. In the fifth phase, the documentation and communication take place. The phases of the process model are passed one after another with not iterations to previous phases. Another guideline for the design of DSR artifacts, which is often followed by IS researchers, is the Design Science Research Methodology (DSRM) (Peffers et al., 2014, p. 43-44). Unlike the process model, the DSRM guides the design of DSR artifacts independent from the artifact type. Thereby, it is generally applicable to our development of a pattern language structure. The DSRM starts with an identification and motivation of the problem in phase one. In the second phase, the objectives are defined and in the third phase, the artifact is developed. Thereafter, the artifact is tested in phase four, subsequently evaluated in phase five and communicated in phase six. Different from the process model, the DSRM proposes iterations from phase five and six, which are the artifact evaluation and communication, back to the phases two and three, which are the objective definition and artifact design. There is a need to combine both models for three reasons. First, the definition of an objective's solution should be separated from the iterative design and evaluation process, as proposed in the process model. We argue that a change of the objectives influences the artifact design and thereby they shouldn´t be changed in the iterative process. Second, the process model proposes the development of an initial model prior to the iterative artifact development. This initial model aggregates existing knowledge to explicitly include recent results from research and practice in the design process (Ahlemann and Gastl, 2007, p. 24-25). We argue that such an initial model is required to efficiently refine the model in iterative steps. Third, we see the need for iterations back from the phase of artifact evaluation to the artifact design, as proposed by the DSRM. We argue that obstacles that could not been foreseen in the design phase are only revealed in the artifact applying.

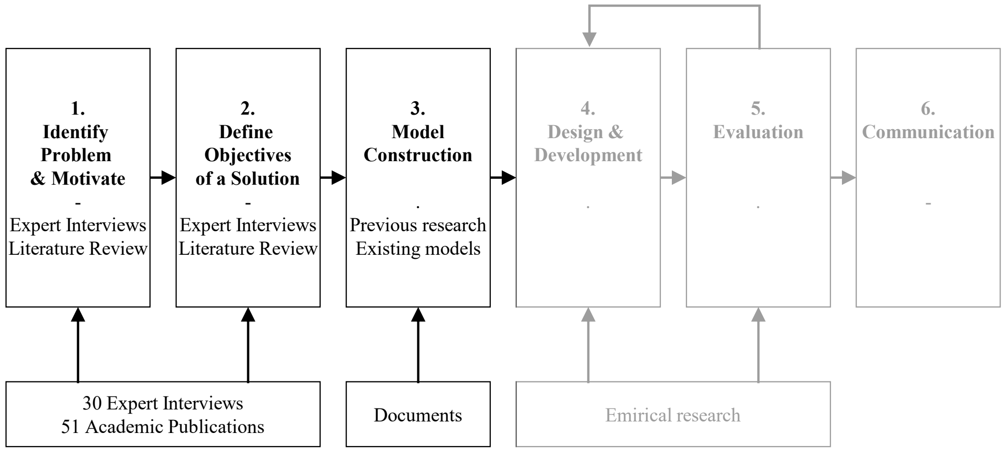


Figure 3-1 Systematic process to design the structure of the pattern language

The process to develop our structure of a pattern language is shown in Figure 3-1. For step 1 and 2 we start with an identification of the motivations, problems, and objectives. These factors were already covered in previous research projects, namely a systematic review of 51 academic publications and 30 expert interviews in the vehicle industry (Brandt and Ahlemann, 2020, p. 6). Thus, the motivation, problems and objectives are justified by empirical evidence and address the problem of unknown or misunderstood environments and intended uses that March and Smith (1995, p. 253) identified in many design science projects. We summarize the motivation (M) as follows:

(M) Recently, connectivity units are embedded into individual vehicles. If manufacturers could use these connectivity units to systematically gather, manage, store, and analyze these data, they could improve their lifecycle related tasks. Furthermore, benefits for different stakeholder become possible.

Based on the literature review and expert interviews we identified problems (P) that hinder the realization of the motivational factor M. The problems are further analyzed to derive the objectives (O) of a solution (Peffers et al., 2014, p. 57):

(P1) Existing vehicles do not enable vehicle manufacturers to systematically gather sensor data during the products' usage.

(O1) The solution needs to enable vehicle manufacturers to systematically gather vehicle instance data and manage the gathering along the vehicle lifetime.

(P2) Manufacturers are unable to store a vehicle digital twin from each individual vehicle in their IS.

(O2) The solution needs to offer a systematic method to store product instance data in a uniform format along the vehicle lifetime.

(P3) Existing IS and lifecycle applications analyze data from product classes and specific lifecycle phases only.

(O3) The solution needs to offer tools and methods to analyze usage data from product instances and offer an interface to other systems to enable applications with data from multiple lifecycle phases.

		Vehicle				**Device**			**Digital Twin**			**Service**		
		Sensor Data Gathering	**Edge Computing**	**Vehicle Exchange Unit**	**Offline Manager**	**Transfer Translator**	**Device Governance**	**Consent Manager**	**Anonym Digital Twin**	**Digital Twin**	**Static Digital Twin**	**Service Platform**	**Twin Configurator**	**Database Federator**
Commercial Platforms	Wireless Car		X	X	X	X					X		X	
	AERIS											X		
	Conweaver											X		X
	HiveMQ		X	X	X			X	X					
	Tantalum			X								X		
	OceanConnect IoT platform			X								X		X
	Watson IoT platform		X						X	X		X		X
	Internet of Things patterns					X								
	AIOTI HLA Ref. architecture			X										
	Contact Software			X							X	X	X	
Research Artifacts	PROMISE (Kiritsis, 2011)			X		X	X		X	X	X	X		X
	ICISP (Cai et al., 2014)											X		X
	Big data driven PLM (Yoo et al., 2016)	X										X		X
	FALCON (Hribernik et al., 2017)	X					X		X	X		X		X
	Service Architecture (Hribernik et al., 2013)	X					X	X	X	X		X		X
	Big BOM (Tao et al., 2014)	X					X		X	X	X			
	Framework of IoT in PLEM (Tao et al., 2016)	X		X		X			X	X		X		
	LEAP (Cerri et al., 2015)	X							X	X		X		
	Complex closed-loop PLM (Guo et al., 2014)	X	X									X		

Table 3-1 The structure of a pattern language

The identification of the motivations, problems and obstacles is followed by the construction of the initial model, which is the focus of the paper at hand. Ahlemann and Gastl (2007, p. 24-

25) suggest that an initial model might consider expert interviews, standards, norms, a literature review, and the domain knowledge of the researchers. For the development of our initial artifact, we consider the results from the 30 semi-structured expert interviews in the vehicle industry (Brandt and Ahlemann, 2020, p. 6), a review of 51 academic publications and the documentations of ten commercially offered IoV platforms. Table 3-1 depicts the sources of our pattern language structure. During the interviews, experts were questioned about the challenges that hinder the integration of vehicle data into the manufacturers' IS. Furthermore, we asked the experts about the measures that their employers take to gather, manage, store, and analyze IoV data. During the systematic review of existing academic literature and documentations of platforms, we identified 19 independent models that we analyzed for common elements. Overall, our initial model includes 21 cases. Based on these cases we develop a pattern language structure as an artifact of the type of model. We follow the process as proposed by Fettke and Loos (2003, p. 43 - 49) to aggregate the 21 cases with a classification scheme. Therefore, we create a longlist of characteristics that the cases have. Thereafter, we identify characteristics that are used by multiple cases and summarize similar characteristics. After prioritizing the characteristics, we test and apply them to all 21 cases and derive the elements of the pattern language structure. Unlike most other pattern languages that do not use a formalized modeling language (Matthes, 2009, p. 469), we use a component diagram as a structural unified modelling language diagram for the structure of the IoV pattern language. Component diagrams use components as subsystems of a larger system that perform self-contained tasks of the overall system. These diagrams depict interfaces between the components to show their context. Component diagrams are well suited for our pattern language, as the components are self-contained and exchangeable units such as the patterns for each element.

4 The Structure of a Pattern Language

In this paper we aim to guide manufacturers designing an IS that enables them to gather, manage, store, and analyze vehicle sensor data along the vehicles' use phase. Therefore, we develop a pattern language. Before the patterns can be defined, this paper identifies and specifies the sub-problems that a suitable IS needs to solve. In the following, we will call the identified and specified sub-problems as elements of the pattern language. According to our research methodology, we analyze expert interviews, existing reference models and documentations of IoV platforms to identify and specify the sub-problems and derive the elements of the pattern language. All elements in their context build the pattern language structure.

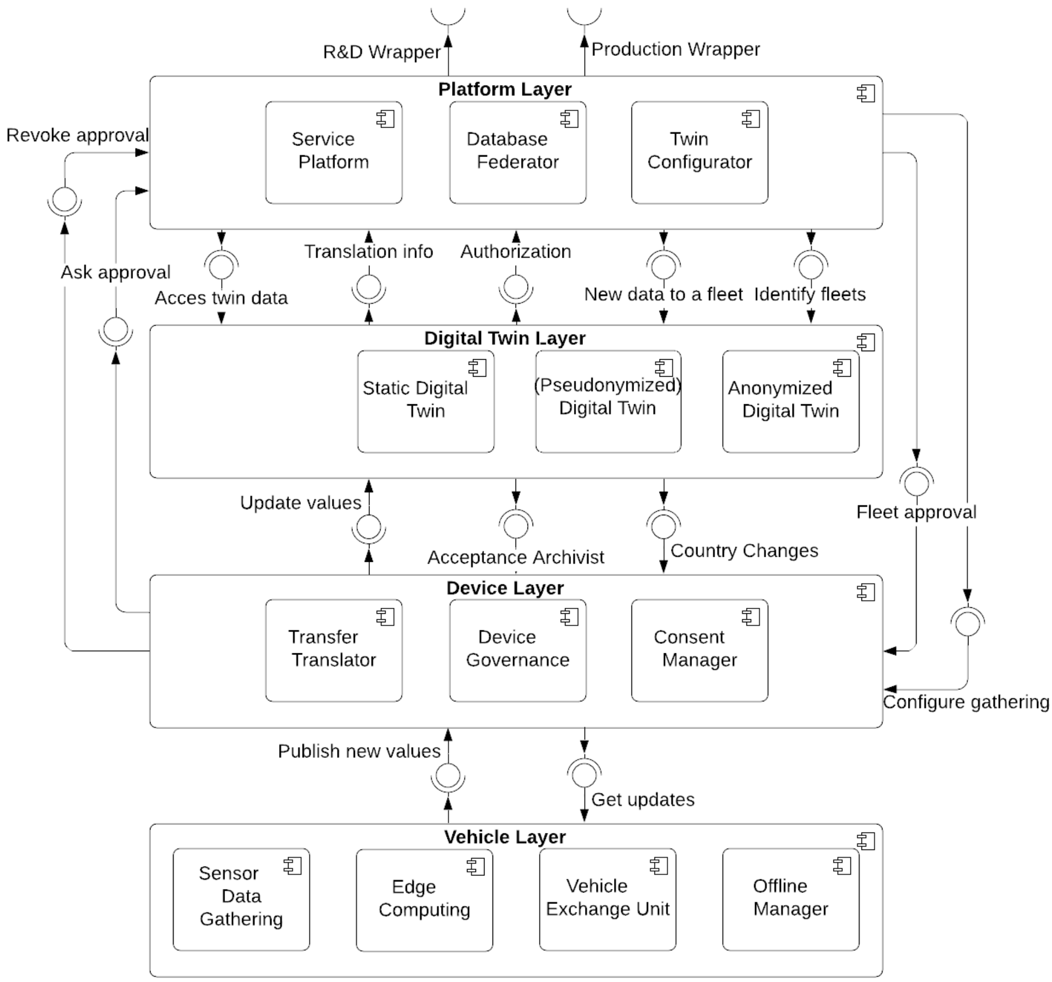


Figure 4-1 The structure of a pattern language with dependencies

Figure 4-1 gives an overview over the elements of the pattern language and their interdependencies. The elements on the vehicle layer need to be applied in each individual vehicle. They enable the vehicles to gather, preprocess, exchange, and buffer sensor data. The elements on the device layer provide functionalities to manage the gathering and preprocessing of sensor data in the individual vehicles. They might be applied in each vehicle or a product external IS.

Functionalities include the translation of data between the heterogeneous vehicles and the uniform format of the manufacturers IS, the governance of new or changed requests for data gathering and preprocessing and the management of customer approvals to gather sensor data from their vehicles. The elements on the digital twin layer provide functionalities to store product instance information. These include the storage of static and (anonymized/pseudonymized) dynamic product instances data and an authorization tool for data access. The elements on the platform layer finally enable the query and analysis of lifecycle data, which includes sensor data from the individual vehicles.

5 The vehicle layer

5.1 Sensor data gathering

Traditionally, sensors are embedded into vehicles to measure the status of the vehicles' parts and its' environment (Tao et al., 2016, p. 33) (Cerri et al., 2015, p. 127). Processing units analyze these measured statuses inside the vehicle to steer the internal processes and to enable driving functionalities (Hribernik et al., 2013, p. 8). As the sensor data unveil how individual vehicles are used and behave along their lives, they also enable the measurement and analysis of customer demand and product usage (Zhang et al., 2017, p. 11). The element sensor data gathering solves the problem that sensor data cannot be accessed to use it for other reasons than vehicle process steering, namely the analysis of the product status, customer demand and product usage. To do so, sensor data gathering needs to somehow intercept sensor data inside the vehicle (Cerri et al., 2015, p. 127). The element sensor data gathering also recognizes when an electronic component is physically exchanged (Tao et al., 2016, p. 33). Furthermore, the configuration, which sensor data should be gathered, might change along a products' life. Therefore, sensor data gathering can be updated by means of sensor data type and gathering frequency. Sensor data gathering transfers gathered data to edge computing, which is the element to preprocess the raw data and prepare the transfer. Thus, sensor data gathering requires some interface to edge computing.

5.1.1 Edge computing

The element sensor data gathering transforms the physical status of individual vehicles along their lives into measurable events (Marabelli et al., 2017, p. 352). As sensor data are generated during the vehicles' usage at high data rates and the data rates of wireless transmissions are limited or wired transmissions take place in long intervals, local preprocessing of raw measurements is required (HiveMQ, 2019) (Watson IoT Platform). The element edge computing solves the problem of high data rates. Furthermore, it ensures fast processing of data with low latencies (Watson IoT Platform). Therefore, it performs each vehicle's local data (pre-)processing. It processes raw sensor data based on given functions (Guo et al., 2014, p. 135). Such a processing includes recognizing pre-defined events, such as oil status warnings, and aggregating measurements over time, such as usage frequency counters. Although it is most common in the vehicle industry to perform such a rule-based preprocessing, technological achievements also enable the application of machine learning in edge devices (Wireless Car). Through the local application of machine learning inside the vehicles, machine learning can be applied to raw sensor measurements rather than to the preprocessed sensor data available outside the vehicle (Kaiser et al., 2018, p.8). One must differentiate between supervised machine learning and unsupervised machine learning. Supervised machine learning searches for known patterns in the raw sensor data, whereas unsupervised machine learning analyses a time series of raw data to detect new and previously unknown patterns (HiveMQ, 2019). Edge computing sends the preprocessed sensor data and newly detected patterns to the vehicle exchange unit to make them available outside the vehicle. Thereby, preprocessing lowers data rates and enables data exchanges in intervals. Besides transferring data outside of the vehicle, edge computing receives updated preprocessing functions (Wireless Car). For this bidirectional data transfer, edge computing offers an interface to the transfer unit.

5.1.2 Vehicle exchange unit

After sensor data are gathered and preprocessed, the data can be accessed and analyzed only with physical access to the vehicle. The element vehicle exchange unit solves this problem of decentral data (Tao et al., 2016, p. 33). It sends the existing data to an external product database for further processing with data exchange logic (Elloumi and Carugi, 2018) (Contact Software, 2020). Furthermore, the vehicle exchange unit offers an interface to receive new or changed commands for data gathering or preprocessing (Wireless Car). In addition, it receives updates to adjust the steering of vehicle processes (Huawei Technologies Co., 2019) (Contact Software, 2020). To change gathering configurations, the vehicle exchange unit has a function to update sensor data gathering and edge computing. The vehicle exchange unit generally ensures a secure data transfer and avoids unauthorized access to vehicle sensor data and vehicle functions (HiveMQ, 2019).

5.1.3 Offline management

The vehicle exchange unit can only transfer (preprocessed) sensor data if there is a connection to the product's external IS. In the case of a wireless transfer via the vehicle exchange unit, vehicles might partially drive in a region with no network coverage (HiveMQ, 2019). In addition, the offline manager buffers data in case of a wired transfer, as transfers occur in long intervals (Wireless Car). The element of offline management solves the problem of unavailable network connectivity. Therefore, the element integrates logic to transfer or buffer data in an offline mode. To perform offline management, the element has an interface to the vehicle exchange unit.

5.2 The device layer

5.2.1 Transfer translator

The element transfer translator reduces the impact of high vehicle heterogeneity on sensor data gathering and data exchange. A high vehicle heterogeneity exists for many manufacturers due to the long lives of manufactured vehicles (Statista, 2014) and the high variants. The general technological progress during a vehicle's long operation causes sensors and communication units embedded into the operated fleet to change constantly. In addition, the formats of sensor signals might differ between the manufacturer's models. One example is the fuel level measurement that sensors in some models might measure on a relative scale while others measure on an absolute scale. Furthermore, the encryption of sensor data on the internal communication busses might differ. Last, the communication might be different among the individual vehicles (Reinfurt et al., 2016, p. 3). Generally, models from one manufacturer often seem similar, but differ significantly from a technical perspective. To deal with vehicle heterogeneity and to reduce storage complexity, sensor data from the heterogeneous vehicles need to be translated into a standard format (Tao et al., 2016, p. 37). Furthermore, updated data gathering and preprocessing instructions need to be translated from the uniform manufacturers' format to the format of the individual vehicles (Wireless Car). Thus, the transfer translator works as a mediator and decouples the data storage from the heterogeneous vehicles (Kiritsis, 2011, p. 485). The element transfer translator enables the manufacturer to analyze vehicle sensor data and change instructions for data gathering and preprocessing in a standard format. Simultaneously, a transfer translator does not hinder the manufacturer from changing and further develop vehicle hardware as just the configuration of the transfer translator needs to be changed to deal with the extra complexity. For the translation of messages, the element needs access to the specification of the individual vehicles to use the correct translation (Reinfurt et al., 2016, p. 3).

5.2.2 Device governance

Often, the stakeholder that aims to gather sensor data has plenty of products that send data. The element of device governance solves the problem of managing and treating all products individually. Device governance sends updated data gathering and preprocessing instructions to the vehicles (Tao et al., 2014, p. 1254). This element changes sensor data gathering or preprocessing functions to the belonging vehicles (Kiritsis, 2011, p. 485). In case of a change request, device governance checks whether a customer approval for the changes already exists/is required or if the customer approves it (Hribernik et al., 2013, p. 5). In case of a given consent, device governance initiates the transfer translation to update edge computing and/or data gathering in the vehicle (Hribernik et al., 2017, p. 1410). In case a customer withdraws his approval to gather sensor data from his vehicle, device governance initiates an update of data gathering and preprocessing in the vehicle. The element device governance also redirects vehicle sensor data to the correct database. Thereby, an interface needs to be offered for the transfer translation to receive data from the vehicle in the uniform format of the manufacturer (Tao et al., 2014, p. 1254).

5.2.3 Consent manager

The sensor data of individual vehicles depict the customers' behavior in detail and in real-time (Watson IoT platform). In addition, the customer creates the product usage data, and thus, he owns them. The element consent manager solves the problem of data privacy. In case of a request to gather and analyze sensor data from a specific vehicle, the element consent manager checks whether the data protection regulations of the region where the product is operated require customer approval before gathering and analyzing (Kaiser et al., 2018, p. 8). In case approval is required, the consent manager checks whether approval was already given (Hribernik et al., 2013, p. 5). If customer approval is missing, the consent manager asks the customer for support. To gain customer approval, the consent manager might offer a (monetary) incentive (Hribernik et al., 2013, p. 5). The consent manager also monitors whether a vehicle is operated in a new region with changed regulations for data protection (HiveMQ, 2019). In case new data protection regulations become valid due to a modified region, customer approvals are requested. In case of missing approval, the gathering is stopped while the vehicle is operated in the specified region. Furthermore, the element archives all licenses and grants the customers access to this archive so that they might review their given approvals and make changes accordingly. The documentation of obtained approvals is also used to prohibit access to vehicle sensor data from unauthorized applications.

5.3 The digital twin layer

5.3.1 Static digital twin

Each vehicle has static properties that don´t or rarely change in a vehicle's life (Tao et al., 2014, p. 1255). Examples include in-built parts, the version of electronic components, and the owner information (Wireless Car). The element of static digital twin solves the problem that applications might need to access these static data about each vehicle (Contact Software, 2020). Although the properties might be relatively static, broken parts are replaced, electronic components are updated, or vehicles are sold. In case of such changes, the static digital twin is updated, and thereby, the static digital mirrors the properties of each physical product along its life (Kiritsis, 2011, p. 485). The static data need to be accessible to other elements and services. For instance, they serve as input for the data translator.

5.3.2 (Pseudonymized) digital twin

Along a product's life, the individual products send new sensor data. The element (pseudonymized) digital twin solves the problem that vehicle sensor data from the individual vehicles must be stored and updated along the vehicles' lives (Kiritsis, 2011, p. 485). Thereby, the digital twin depicts the status of individual physical products by gathering sensor data (Tao et al., 2014, p. 1252). As new sensor data are frequently collected during a vehicle's usage, the element stores and updates sensor measurements regularly (Cerri et al., 2015, p. 127). The (pseudonymized) digital twin stores both data streams and events (Hribernik et al., 2017, p. 1411). Data streams are stored at a high frequency and enable a more detailed analysis of the time series (Watson IoT platform). Events, on the other hand, store preprocessed data streams.

As the transfer translator translates all sensor data into the uniform format of the digital twin database, storage complexity is reduced. Furthermore, data in the (pseudonymized) digital twin can be deleted in case customer approval or service changes are changed.

5.3.3 Anonymized digital twin

While some applications require access to sensor data from a specific vehicle, others only need access to statistical data from a group of vehicles over their lifetime (Contact Software, 2020). For example, a customer service that updates the software in a vehicle requires access to sensor data from precisely one vehicle (Huawei Technologies). On the other hand, an application from a manufacturers' design department that analyzes the usage frequency of a function doesn´t require access to data from a specific vehicle but data from some vehicles of a predefined group (Holler et al., 2016, p. 485). The element anonymized digital twin virtually groups vehicles that match a new request for anonymized data (Contact Software, 2020). Thereby, it solves the problem of anonymity. To virtually group the vehicles, the anonymized digital twin needs to check for all vehicles of the potential group whether a customer approval is dispensable, already exists, or needs to be requested. Including product instances in the virtual fleets is documented and access is granted for authorized roles (Tao et al., 2016, p. 37). In case of changing customer approvals, the vehicle also needs to be deleted from the virtual groups in the anonymized digital twin. In case of changing virtual fleets, anonymized digital twin updates the gathering and preprocessing of product instance data accordingly.

5.4 The service layer

5.4.1 Service platform

Gathering sensor data by itself does not bring full benefits to manufacturers. Only when it is appropriately analyzed it might bring the benefits to justify the effort for data gathering. The element service platform solves this data analysis problem as it contains applications to analyze product (instance) data. The element needs access to product-related data from all lifecycle phases (Huawei Technologies Co., 2019). Differently from existing software, applications on the service platform offer standardized interfaces to other applications (Contact Software, 2020) to ensure interoperability (Tao et al., 2016, p. 37). Through such an interface, a data marketplace on the service platform makes product-related data from all lifecycle phases, including the digital twins, available for authorized roles (Tantalum, 2020). Service developers, customers, and other stakeholders might query this marketplace. To allow users without basic database knowledge to query the lifecycle data, data analysis can be achieved via a visual rules engine (Contact Software, 2020). This enables business users, customers, and other stakeholders to make visual evaluations and exports of data (Cerri et al., 2015, p. 129) (Conweaver, 2020) in a form in which they usually analyze data (Cerri et al., 2015, p. 129). Besides specific rule-based applications, applications on the service platform perform machine learning to detect patterns with customizable algorithms (Zhang et al., 2017, p. 15). With such a solution, real-time data streams and historical data can be processed (Watson IoT platform).

5.4.2 Database federator

PLM services might require access to product data from one or more lifecycle phases (Zhang et al., 2017, p. 15). These data are often distributed in different manufacturers' databases (Kiritsis, 2011, p. 485) with individual interfaces (Conweaver, 2020). The element database federator solves the problem of distributed data. It grants applications a uniform access to product-related lifecycle data. The element works as a single point of access for services to query product-related data (Hribernik et al., 2013, p. 8). Thereby, the element avoids that services establish individual interfaces to the databases, which eases maintenance and reduces complexity (Huawei Technologies Co., 2019). When an interface to the required database is established, the element normalizes and redirects the requests (Kiritsis, 2011, p. 485). Thus, data is not stored by the federator but an interface to all product related data is given (Watson IoT platform). Thereby, it prohibits storage redundancy is and ensures a single point of truth. Through the interface to other databases, the database federator neither requires nor inhibits manufacturers to build data lakes for product-related data, as legacy systems can be integrated (Hribernik et al., 2017, p. 1411).

5.4.3 Twin configurator

Along the lifetime of a vehicle it might change, which sensor data should be gathered and how it should be preprocessed inside the vehicle (Wireless Car). Therefore, the element twin configurator solves the problem that a stakeholder wants to change the configuration of sensor data gathering or preprocessing. Therefore, it creates and submits new or changed requests to individual vehicles. Subsequently, the element enables authorized users to construct a new or change an existing configuration file for data gathering and preprocessing (Contact Software, 2020). The element then sends the configuration files to the correct vehicles. Due to heterogeneous vehicles, available configurations need to be checked with the static digital twin so that only those configurations are made, which are technically feasible. Furthermore, existing configurations for data gathering are considered.

6 Discussion and Conclusion

Previous research has shown that manufacturers benefit from product sensor data, as these data unveil how individual products are used throughout their lives (Brandt and Ahlemann, 2020, p. 8). This is especially valid for vehicle manufacturers, who put high efforts into collecting sensor data from individual vehicles over the internet. The systematic gathering, management, storage, and analysis of sensor data is an unsolved problem in the vehicle industry (Brandt and Ahlemann, 2020, p. 8), and many reference models that don´t solve the problem in practice exist (Tao et al., 2014, p. 1255) (Kiritsis, 2011, p. 486) (Cerri et al., 2015, p. 128), we saw the need to harmonize the existing knowledge and derive guidance for manufacturers to overcome the obstacles. Therefore, we analyzed reference models from academic researchers, documentation from IoV platforms, and interviews with experts from the vehicle industry to identify and specify the problems that need to be addressed with a suitable IS. The identification and specification led to the definition of elements of a pattern language structure. In total, our structure of a pattern language identifies and contextualizes 14 elements that an IS to enable vehicle manufacturers to gather, manage, store, and analyze sensor data along the vehicles' lives needs to have. The 14 elements are grouped into four logical layers: gathering sensor data on a vehicle level, managing and steering the gathering, storing the instance data, and analyzing the instance data. Each element guides researchers and practitioners to design new IoV architectures as they summarize the functional requirements for a solution. This paper presents a first step to guide researchers and practitioners to create IS to gather, manage, store, and analyze sensor data from internet-connected vehicles.

Our contribution is twofold. First, we contribute to existing knowledge by developing a methodology to conduct DSR. Based on the DSRM and the process model for an empirically grounded reference model construction, we propose a methodology to build pattern languages and reference models. Such a methodology can be considered an artifact itself (March and Smith, 1995, p. 254). The first steps of the methodology are already validated by the main contribution of this paper, which is a pattern language structure that addresses the low adoption of existing reference models and IoV platforms. Based on our analysis, we found that vehicle manufacturers already use highly individualized IS to perform product-related tasks. Some of these IS might already solve parts of the overall problem to systematically gather, manage, store, and analyze sensor data from individual products. Thus, manufacturers need to supplement their existing IS gradually rather than adopting an entirely new IS. As many reference models (Schermann et al., 2007, p. 188) and software applications from vendors should be adopted as a whole, the hurdles of their adoption are high in practice (Aldin et al., 2009, p. 1340). Furthermore, it is generally challenging to adapt the existing models and interfaces from researchers or software vendors to the individual needs of the manufacturers (Keller, 2009, p. 472). With developing a pattern language structure, we address the missing adoptability of existing reference models and IoV platforms. We do so by identifying and specifying 14 elements that an IS needs to have. As we define the elements as self-contained elements, a manufacturer might replace some elements with their existing solutions, while they adopt others from the pattern language structure. Thereby, manufacturers are guided to gradually introduce more elements in their individual landscape of existing IS. The given elements identify and specify the requirements to a solution. By realizing the elements of the pattern language structure, manufacturers themselves develop patterns, i.e., best practices, that meet their individual requirements. As we do not propose patterns for each of the 14 elements, we do not limit the creativity of the designer to a limited number of patterns. Thus, manufacturers might create their own patterns for their context based on the elements' specification.

Although we followed the DSR research rigor, the paper at hand is not free of limitations. First, we limited our artifact to the vehicle industry. Thus, a transfer to other industries remains unanswered. Second, we did not involve practitioners to validate our results and to identify patterns for the elements. In further research, we aim to address both limitations. We will conduct an exploratory study that involves experts from the vehicle industry as well as experts from other technological companies that rely on product usage data. By including knowledge from different industries, we can evaluate how best practices from other industries might be adopted to the vehicle industry and vice versa. In this further study, we aim to define multiple patterns for each element in our structure of the pattern language.

7 References

Ahlemann, F. & Gastl, H. (2007). "Process model for an Empirically Grounded Reference Model Construction." In: Reference Modeling for Business Systems Analysis. Idea Group Publishing, Hershey, p. 77-97.

Aldin, L., De Cesare, S. & Lycett, M. (2009). "Semantic Discovery And Reuse Of Business Process Patterns." In: Mediterranean Conference on Information Systems 2009 Proceedings, p. 1340-1345.

Alexander, C., Ishikawa, S. & Silverstein, M. (1977). "A Pattern Language. Town – Buildings - Construction." Oxford University Press. New York.

Balasubramanian, J., Beiker, S. Chauhan, S., Colombo, T., Cornes, M., Hansson, F., Huddar, N., Jaarsma, R., Kässer, M. (2016). "Monetizing car data - New service business opportunities to create new customer benefits." McKinsey & Company.

BMI Lab (2020). "How does edge computing benefit connected cars?" URL: https://www.wirelesscar.com/how-does-edge-computing-benefit-connected-cars/ (visited on 20 August 2020).

Brandt, N. & Ahlemann, F. (2020). „How do you drive? Analyzing vehicle sensor data in product lifecycle management systems." In: Proceedings of the 28th European Conference on Information Systems (ECIS), p. 1-15.

Cai, H., Xu, L., Xu, B., Xie, C., Qin, S. & Jiang, L. (2014). "IoT-Based Configurable Information Service Platform for Product Lifecycle Management." In: IEEE Transactions on industrial informatics, Volume 10, Issue 2, p. 1558-1567.

Cerri, D., Taisch, M., Terzi, S., Buda, A., Framling, K, El Kaddiri, S., Milicic, A., Kiritsts, D., Parrotta, S. & Peukert, E. (2015). „Proposal of a Closed Loop Framework for the Improvement of Industrial Systems' Life Cycle Performances: Experiences from the LinkedDesign Project." In: Procedia CIRP 29, p. 126-131.

Contact Software (2020). "CONTACT Elements for IoT." URL: https://www.contact-software.com/de/produkte/iot-plattform-fuer-digitale-geschaeftsmodelle/ (visited on 20 August 2020).

Conweaver (2020). "ProStep iViP Symosium 2015." URL: https://www.conweaver.com/fileadmin/ www.conweaver.de/media/downloads/public/Cross_Domain_Engineering_%40_Bosch-CONWEAVER_%40ProSTEP_2015_Symposium.pdf (visited on 20 August 2020).

Dietz, M. & Pernul, G. (2019). „Digital Twin: Empowering Enterprises Towards a System-of-Systems Approach." In: Business & Information Systems Engineering 62 (2), p. 179-184.

Elloumi, O. & Carugi, M. (2018). „High Level Architecture – Release 4.0." In: Alliance for Internet of Things Innovation (AIOTI).

Enders, M. & Hoßbach, N. (2019). „Dimensions of Digital Twin Applications – A Literature Review." In: Twenty-fifth Americas Conference on Information Systems, p. 1-10.

Erikstad, S. (2018). "Design Patterns for Digital Twin Solutions in Marine Systems Design and Operations." In: COMPIT 2018, At Pavone, Italy, p. 354-363.

Feldhusen, J. & Bungert, F. (2007). "PLM Pattern Language: An Integrating Theory of Archetypal Engineering Solutions." In: 14th CIRP Conference on Life Cycle Engineering, p. 125-130.

Fettke, P. & Loos, P. (2009). „Morning Star, Evening Star, and Venus – On the Use of the Words 'Reference Model' and 'Pattern'." In: Business & Information Systems Engineering, p. 473-474.

Främling, K, Holmström, J., Loukkola, J., Nyman, J. & Kaustell, A. (2013). „Sustainable PLM through Intelligent Products.“ In: Engineering Applications of Artificial Intelligence Volume 26, p. 789-799.

Gamma, E., Helm, R., Johnson, R. & Vlissides, J. (1997). "Design Patterns: Elements of Reusable Object-Oriented Software." Addison-Wesley.

Gerla, M., Lee, E., Pau, G. & Lee, U. (2014). „Internet of Vehicles: From Intelligent Grid to Autonomous Cars and Vehicular Clouds.“ In: 2014 IEEE World Forum on Internet of Things, p. 241-246.

Gregor, S. & Hevner, A. (2013). "Positioning and Presenting Design Science Research for Maximum Impact." In: MIS Quarterly, Volume 37, Issue 2, p. 337-355.

HiveMq (2019). "Enabling the Connected Car with HiveMQ." URL: www.hivemq.com/downloads/ hivemq-enabling-the-connected-car.pdf (visited on 20 August 2020).

Hohpe, G. (2019). "Enterprise Integration Patterns." URL: https://www.enterpriseintegrationpatterns. com/patterns/messaging/ (visited on 20 August 2020).

Holler, M., Stoeckli, E., Uebernickel, F. & Brenner, W. (2016). „Towards Understanding closed-loop PLM: The Role of Product Usage Data for Product Development enabled by intelligent Properties.“ In: BLED 2016 Proceedings, p. 479-491.

Hribernik, K., Franke, M., Klein, P., Thoben, K. & Coscia, E. (2017). „Towards a Platform for Integrating Product Usage Information into Innovative Product-Service Design.“ In: 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), p. 1407-1413.

Hribernik, K., Wuest, T. & Thoben, K. (2013). „Product Avatar for Leisure Boats Owners: Concept, Development and Findings.“ In: IFIP Advances in Information and Communication Technology, p. 1-10.

Huawei Technologies Co. (2019). "IoT, Driving Verticals to Digitization." URL: https://www.huawei.com/minisite/iot/en/vehicle-networking.html (visited on 20 August 2020).

Huber, W. (2016). „Industrie 4.0 in der Automobilproduktion – Ein Praxisbuch.“ Springer Vieweg, Wiesbaden, Germany.

IBM Corp. (2020). "Product architecture." URL: https://www.huawei.com/minisite/iot/en/vehicle-networking.html (visited on 20 August 2020).

Johannesson, P. & Perjons, E. (2014). "An Introduction to Design Science." Springer Vieweg.

Kaiser, C., Steger, M., Dorri, A., Festl, A., Stocker, A., Fellmann, M. & Kanhere, S. (2018). „Towards a Privacy-Preserving Way of Vehicle Data Sharing – A Case for Blockchain Technology?“ In: International Forum on Advanced Microsystems for Automotive Applications, p. 111-122.

Keller, W. (2009). „Patterns sind mehr als nur Lösungen zu einem Problem in einem Kontext.“ In: Business & Information Systems Engineering, p. 471-472.

Kiritsis, D. (2011). „Closed-loop PLM for intelligent products in the era of the Internet of things.“ In: Computer-Aided Design 43, p. 479-501.

Maaß, W. & Storey, V. (2015). „ Logical Design Patterns for Information System Development Problems.“ In: Springer International Publishing Switzerland, p. 134-147.

Marabelli, M., Hansen, S., Newell, S. and Frigerio, C. (2017). „The Light and Dark Side of the Black Box: Sensor-based Technology in the Automotive Industry.“ In: Communications of the Association for Information Systems, Volume 40, p. 351-374.

March, S. & Smith, G. (1995). „Design and natural science research on information technology." In: Decision Support Systems 15, p. 251-266.

Matthes, F. (2009). „Capturing, Structuring, and Passing Knowledge for the Design of Complex Systems as Patterns ". In: Business & Information Systems Engineering, p. 468-470.

Paschke, A. (2008). "Design Patterns for Complex Event Processing.", p. 1-5.

Peffers, K., Tuunanen, T., Rothenberger, M. & Chatterjee, S. (2014). „A Design Science Research Methodology for Information Systems Research." In: Journal of Management Information Systems Volume 24 – Issue 3, p. 45-77.

Reinfurt, L., Breitenbücher, U., Falkenthal, M., Leymann, F. & Riegg, A. (2016). „Internet of Things Patterns." In: EuroPLoP'16, p. 1-21.

Schermann, M., Böhmann, T. & Krcmar, H. (2007). "Fostering the Evaluation of Reference Models: Application and Extension of the Concept of IS Design Theories." In: Wirtschaftsinformatik Proceedings 2007, p. 181-198.

Schlieter, H., Juhrisch, M. & Niggemann, S. (2010). „The Challenge of Energy Management – Status-Quo and Perspectives for Reference Models." In: Pacific Asia Conference on Information Systems 2010 Proceedings, p. 411-422.

Schönteich, F., Kasten, A. & Scherp, A. (2018). „A Pattern-Based Core Ontology for Product Lifecycle Management based on DUL." In: Workshop on Ontology Design and Patterns, p. 1-15.

Sprenger, M. and Mettler, T. (2016). „On the Utility of E-Health Business Model Design Patterns." In: Twenty-Fourth European Conference on Information Systems, p. 1-16.

Statista (2019). "Connected Car Report." URL: https://de.statista.com/statistik/studie/id/43051 /dokument/connected-car-2019/ (visited on 20 August 2020).

Tantalum (2020). "Internet of Things Platform." URL: https://www.tantalumcorporation.com/ (visited on 20 August 2020).

Tao, F., Wang, Y., Zuo, Y., Yang, H. & Zhang, M. (2016). „Internet of Things in product life-cycle energy management." In: Journal of Industrial Information Integration 1, p. 26-39.

Tao, F., Zuo, Y., Da Xu, L., Lv, L. & Zhang, L. (2014). „Internet of Things and BOM-Based Life Cycle Assessment of Energy-Saving and Emission-Reduction of Products." In: IEEE Transactions On Industrial Informatics, Vol. 10, No. 2, p. 1252-1261.

Terzi, S., Bouras, A., Dutta, D., Garetti, M. & Kiritsis, D. (2010). „Product lifecycle management – from its history to its new role." In: International Journal of Product Lifecycle Management, Vol. 4, No. 4, p. 360-389.

Watson IoT platform (2020). "Digital twins and the Internet of Things." URL: https://developer.ibm.com/technologies/iot/articles/digital-twins-and-the-internet-of-things/ (visited on 20 August 2020).

Guo, W., Zheng, Q, Zuo, B. and Shao, H. (2014). "A Closed-loop PLM Model for Lifecycle Management of Complex Product." In: Moving Integrated Product Development to Service Clouds in the Global Economy, Volume 1, p. 132 - 139.

Westerbeek, D. (2016). „The Internet of Things as a source for improving the product development process." In: 7th IBA Bachelor Thesis Conference, p. 1-15.

Winter, R., vom Brocke, J., Fettke, P., Loos, P. Junginger, S., Moser, C., Keller, W., Matthes, F. & Ernst, A. (2009). "Patterns in Business and Information Systems Engineering." In: Business & Information Systems Engineering, p. 468-474.

Wireless Car (2020). "How does edge computing benefit connected cars?" URL: https://www.wirelesscar.com/how-does-edge-computing-benefit-connected-cars/ (visited on 20 August 2020)

Xin, Y. & Ojanen, V. (2017). „The Impact of Digitalization on Product Lifecycle Management: How to Deal with it?" In: Proceedings of the 2017 IEEE IEEM, p. 1098-1102.

Yoo, M., Gronzel, C. and Kiritsis, D. (2016). „Closed-Loop Lifecycle Management of Service and Product in the Internet of Things: Semantic Framework for Knowledge Integration." In: Sensors, Edition 16, p. 1-26.

Zhang, Y., Ren, S., Liu, Y., Sakao, T. and Huisingh, D. (2017). „A framework for Big Data driven product lifecycle management." In: Journal of Cleaner Production Volume 159, p. 229-240.

5

PATTERN LANGUAGE TO GATHER, MANAGE, STORE AND ANALYZE VEHICLE SENSOR DATA

Brandt, Niklas, University of Duisburg-Essen, Essen, Germany

Abstract

Plenty of sensors are embedded into modern vehicles, creating detailed status data along individual vehicles' lives. Manufacturers, vehicle owners, and third parties could realize various benefits if they could systematically access, store and analyze these existing data. To do so, manufacturers undertake significant efforts to transfer vehicle status data into their information systems (IS) with vehicle-embedded internet connectivity units. In previous research, we identified that an IS to gather, manage, store, and analyze vehicle status data consists of thirteen elements. This paper is part of the overall design science research project and develops one to many patterns, i.e., best practices, for each of these elements. We conduct exploratory focus groups in different industries familiar with gathering product status data from individual products to explore and identify the patterns. After that, we assess the adaptability of the patterns to the vehicle industry with confirmatory focus groups.

Keywords: *Internet of Vehicles, Pattern Language, Information System, Design Science Research, Focus Groups*

Table of contents

List of Figures

List of Tables

1 Introduction

Sensors transform phenomena into electrical signals (Marabelli et al., 2017, p. 352). In modern vehicles, various embedded sensors create signals about the status of the vehicles' components (Gerloff and Cleophas, 2017, p. 1). These signals are processed in the vehicle to steer actors and enable driving functionalities (Holler et al., 2016, p. 486). Besides, these sensor signals could be further analyzed in a vehicle external information system (IS) to generate valuable insights into the status of individual vehicles throughout their lives (Stocker et al., 2017, p. 129) (Gerloff and Cleophas, 2017, p. 17). Applications for product status data include such to increase safety, enhancing travel comfort, and to reduce the environmental impact (Barbaresso et al., 2014, p. 6-7). Thereby, vehicle manufacturers might gain a competitive advantage (McAfee and Brynjolfsson, 2012) (Dremel et al., 2017, p. 82). Today, many manufacturers gather basic vehicle status data during service claims and various forms of market research (Newell and Marabelli, 2014, p. 6). Nonetheless, these data have a low level of detail, are gathered in intervals, and only include a subset of all products. Recently, the interest of vehicle manufacturers, customers and third parties in accessing and analyze status data throughout the whole life of individual vehicles increased (Newell and Marabelli, 2014, p. 6). Therefore, manufacturers and service providers aim to transfer the sensor signal that are traditionally only available inside the vehicle (Stocker et al., 2017, p. 126) to a vehicle external IS. Although the technology to gather and analyze product status data is available, researchers found that many vehicle manufacturers are not able to systematically gather, manage, store, and analyze product status data (Brandt and Ahlemann, 2020, p. 8) (Kaiser et al., 2019b, p. 8). Kammler et al. (2019, p. 4) observed that a standard procedure for manufacturers to integrate data-based insights into their value creation process is missing. Also, IS researchers paid little attention in the past to guide the design of IS to gather, manage, store, and analyze status data from individual internet-connected products (Vaia et al., 2018, p. 123) (Marabelli et al., 2017, p. 352). This is especially true for connected vehicles (Kaiser et al., 2018, p. 2), and just a few researchers and practitioners developed reference models to guide the design of such IS (Gerloff and Cleophas, 2017, p. 3). Nevertheless, many existing models lack an acknowledgment of the legacy IS of automotive manufacturers. Thereby, the inability to adopt and integrate parts of the models and adapt them to the context of the existing IS is a hurdle. In addition, current models often propose one specific solution for each component of the overall IS (Hribernik et al., 2017, p. 1410). Thereby, they do not acknowledge the different circumstances of manufacturers. Last, existing models do not explicitly consider the knowledge of other industries, like manufacturing or software (Teixeira et al., 2018, p. 1), that might have already solved the problem.

In this research essay, we build on a structure of a pattern language that proposes elements that an IS to gather, manage, store, and analyze status data from individual internet-connected vehicles needs to contain. We identify one to many best practices for each element of this structure. More precisely, we define our research question as:

Which are the patterns, i.e., best practices, to design an IS to gather, manage, store, and analyze sensor signal-based status data from individual internet-connected vehicles?

We conduct design-oriented focus groups to identify and evaluate the patterns. Focus groups are ideally suited for such interdisciplinary topics (Stocker et al., 2017, p. 128) and offer researchers the opportunity to explore, discuss and validate the patterns with multiple experts simultaneously. The paper at hand has both a theoretical and a practical contribution. It proposes a process to develop a pattern language and a template to document patterns. To solve the problem in practice, it adapts existing knowledge from one research domain and brings it to another.

The paper is structured as follows. After the general introduction in this chapter, we explain the concepts of this paper. After that, we give an overview of our research methodology in chapter 3. The patterns of the pattern language are finally presented in chapter 4 and discussed in chapter 5 before the paper concludes in chapter 6.

2 Conceptual Background

2.1 Processing Vehicle Sensor Data

Sensors are components that can transform physical, chemical, and biological real-world phenomena into electronic signals (Marabelli et al., 2017, p. 352). They are embedded into many technological products to measure the status of the products' parts and processes (Kammler et al., 2019, p. 3). Other electronic components within the products might consume these signals to monitor the real-world phenomena and respond with a pre-defined action that meets the signal (Holler et al., 2016, p. 486). As a complex technological product, a modern vehicle has plenty of embedded sensors that measure different phenomena to enable driving and feature functionalities (Kaiser et al., 2019a, p. 5). The raw signals cannot be understood in the format and frequency in which they occur within a vehicle (Kaiser et al., 2018, p. 5). Nevertheless, if they are systematically processed, they unveil the product status and customer driving style along individual vehicles' lives. Newell and Marabelli (2014, p. 1) consider this indirect communication of individuals by using connected products as implicit connectivity. Each driver becomes a continuous data generator (McAfee and Brynjolfsson, 2012). If the manufacturer knows the status of all individual vehicles, he might analyze the usage frequency of functions and lifetime loads on components to improve the design of future vehicles (Holler et al., 2016, p. 486). Furthermore, he can offer fast remote assistance to enhance customer satisfaction (Kuschel and Dahlbom, 2007, p. 1869). In addition, knowing the product status and customer behavior enables new promising business models that consider the product status and customer behavior. Examples are predictive maintenance services (Schoch 2017, p. 1600) and pricing models for insurance based on the customer driving style. All in all, manufacturers, customers, insurers, service developers, and other stakeholders gain an opportunity to innovate value creation when they systematically gather and analyze sensor signals from individual vehicles in a product external IS (Vaia et al., 2018, p. 114).

2.2 The Vehicle Information System

Although researchers and practitioners have already proven that there are many beneficial applications to analyze sensor signals from vehicle-embedded sensors, these data are traditionally only accessible inside the vehicles. Thereby, manufacturers, customers, insurers, service developers, and other stakeholders cannot analyze them remotely. To overcome this issue, manufacturers and third parties equip individual vehicles with internet connectivity units (Vaia et al., 2018, p. 118) and design IS to gather these data for specific applications (Marabelli et al., 2017, p. 355). Nevertheless, most applications today only access a subset of all available sensor signals (Kaiser et al., 2018, p. 6). Furthermore, the demand for data gathering might change over time and require a reconfiguration of data gathering and processing within the vehicle, which most existing IS do not holistically support. Kaiser et al. (2018, p. 1) call such a holistic IS to gather, manage, store, and analyze sensor data from individual vehicles a Vehicle Information System (VIS). Although IS researchers devote great attention to the design of various IS, Kaiser et al. (2018, p. 10) found that IS researchers pay little attention to such VIS. Marabelli et al. (2017, p. 351) observed this lack of IS maturity and sufficient research even for sensor-embedded products in general. According to the definition from Kaiser et al. (2018, p. 6), the VIS includes sensor signal generation, acquisition, preprocessing, storage, and analysis. In modern vehicles, plenty of embedded sensors already generate sensor signals. To acquire them wirelessly, one needs to filter or preprocess the high amount of sensor signals produced while the vehicle is operated (Vaia et al., 2018, p. 118). The subsequent transfer can either be performed by manufacturers with embedded connectivity units or by third parties with retrofitting connectivity units on standard vehicle interfaces (Gerloff and Cleophas, 2017, p. 17). Kaiser et al. (2019b, p. 8) conducted a qualitative study and unveiled that manufacturers will be the leading providers of vehicle sensor data. Nevertheless, the experts interviewed do not think manufacturers are the providers of data-driven applications due to their missing preprocessing and analysis competencies. This strengthens the need for a holistic and technology-independent guideline, which enables manufacturers to overcome this lack with a suitable VIS. Besides the technological aspects, the social implications and privacy concerns need to be considered that arise when data is gathered that might unveil the drivers' behavior (Newell and Marabelli, 2014, p. 8).

3 Research Design

IS researchers conduct design science research (DSR) to design IT artifacts that solve common problems in practice (Brandtner et al., 2015, P. 26) (Hevner et al., 2004, p. 77). DSR describes the process that guides researchers to develop such artifacts of type construct, model, method, and instantiation (Johannesson and Perjons, 2014, p. 29). We conduct DSR to unveil patterns, i.e., best practices, for an existing pattern language structure, which is an artifact of the type method. Our pattern language solves the problem with knowledge from industries that have solved the problem already, and thus, it constitutes an innovative artifact according to the DSR definition (Hevner et al., 2004, p. 78).

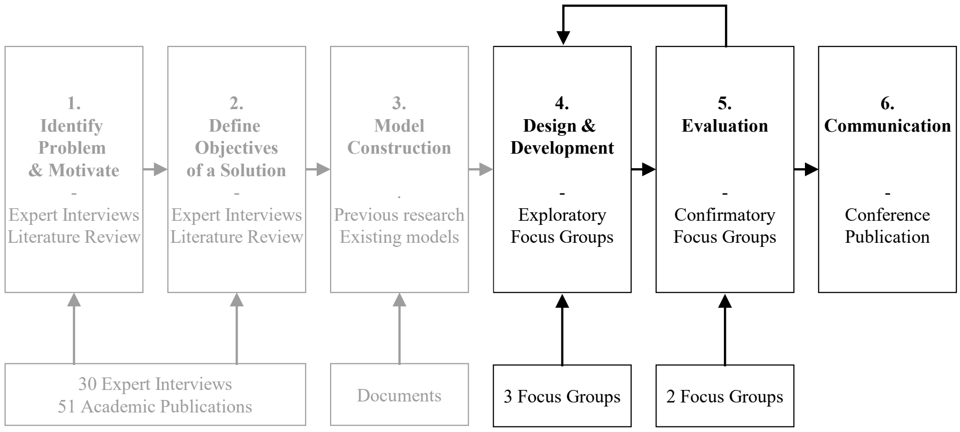


Figure 3-1 Process to develop a pattern language to gather, manage, store, and analyze data from connected vehicles

Figure 3-1 depicts our DSR process, which is based on the design science research methodology (Peffers et al., 2014, p. 43-44) and the process model for an empirically grounded reference model construction (Ahlemann and Gastl, 2007, p. 12). This paper builds on a previous research project (step 1-3) and starts in step 4.

Layer	Element	Description
Product	1.1 Data gathering	*Data gathering* summarizes the need to gather usage data in each product via sensors and input devices.
	1.2 Edge computing	Local *edge computing* preprocesses data in some kind to enable a systematic gathering of usage data.
	1.3 Exchange unit	Some wired or wireless *exchange unit* is required to exchange the sensor data that are locally available.
	1.4 Offline manager	If no transfer of data is possible, an *offline manager* enables the systematic buffering.
Device	2.1 Transfer translator	A *transfer translator* mediates data formats between the heterogeneous products and the uniform digital twins.
	2.2 Device governance	Along of a products' life, data that a stakeholder intends to gather might change. With *device governance*, gathering of each product can be steered.
	2.3 Consent manager	To gain approval for data gathering, *consent manager* maps the products and given customer approvals.
Digital twin	3.1 Static digital twin	The *static digital twin* stores those data that remain almost static along the product lifecycle such as the products inbuilt parts or design files.
	3.2 (Pseudonymized) digital twin	The *(pseudonymized) digital twin* stores dynamically changing lifecycle data with a reference to the product instance that has created these data.
	3.3 Anonymized digital twin	Depending on the legal restriction the product is operated, the *anonymized digital twin* stores data without a reference to a specific product.
Platform	4.1 Service platform	The *service platform* offers tools for diverse stakeholders to analyze product lifecycle data, including product usage data.
	4.2 Database federator	As the product related data from the lifecycle phases are stored in diverse organizations' databases, the *database federator* offers a general interface to query lifecycle data.
	4.3 Twin configurator	The *twin configurator* enables stakeholder to (re-)configure the data that is gathered from the individual products along the products' lives.

Table 3-1 The pattern language with the 14 elements

In the previous project (step 1-3 on Figure 3-1) we developed a structure of a pattern language that identifies and specifies 14 elements that an IS needs to contain to gather, manage, store, and analyze vehicle sensor data during the products usage. Table 3-1 summarizes the elements and describes their content. Based on these elements of the initial model, this paper derives the patterns, i.e., best practices, in the iterative design (step 4) and evaluation (step 5) steps for each element (March and Smith, 1995, p. 254). More precisely we develop one to many patterns for each of the elements of the pattern language structure. As we aim to solve a practical problem, we involve practitioners in the iterative phases (Ahlemann and Gastl, 2007, p. 4). Therefore, we use focus groups as a research methodology. Traditionally, focus groups are used to study cross-cultural topics (Kitzinger, 1995, p. 299-300). Nevertheless, IS researchers already utilized focus groups to design IT research artifacts (Smolander et al., 2008, p. 580)

(Brandtner et al., 2015, p. 27). To establish design-oriented focus groups as research methodology, Tremblay et al. (2010, p. 602) propose a systematic process to conduct focus groups in DSR projects. They divide focus groups into exploratory (EFG) and confirmatory focus groups (CFG). EFG enrich the design process through the engagement of domain experts (Tremblay et al., 2012, p. 332). Thereafter, experts validate the artifacts' utility and recommend changes for future designs in the CFG (Tremblay et al., 2010, p. 602). Sprenger and Mettler (2016, p. 7) utilized the process of EFG and CFG successfully to validate patterns for e-health business models. Kuschel and Dahlbom (2007, p. 1869) conducted a workshop that is like the design-oriented focus groups to unveil business models that use sensor data and thus, they proofed the suitability of the method for such a design project. A major advantage of focus groups for the design of DSR artifacts over other research methods lies in the interaction between the participants. Design ideas or concerns from individual participants can be discussed directly with multiple experts at a time (Nili et al., 2017, p. 2). Furthermore, focus groups involve experts from multiple domains and ensure that overlapping and interdependent topics are designed by all related experts together (Belanger, 2012, p. 109). Another advantage of focus groups lies in the participation of at least three to four experts per group (Brandtner et al., 2015, P.37) that reduces the effort for data gathering and validation (Kitzinger, 1995, p. 299).

According to Tremblay et al. (2010, p. 603) at least two EFG and two CFG should be conducted in a DSR project. We started with two EFG in industries that already gather, manage, store, and analyze data from internet-connected products. As we still explored many new patterns in the second EFG, we conducted a third EFG. Thereafter, we evaluate the ability of the patterns to solve the problem in the vehicle industry in one CFG with experts from a worldwide leading carmaker. Thereafter, we identified blind spots and addressed these with another EFG with experts from industrial company. In the last CFG with other experts from the same carmaker, we evaluated the patterns again and designed the final pattern language. Each focus group was conducted with three to four experts in a homogeneous environment, i.e., with experts from one company, to maximize interaction between the participants (Kitzinger, 1995, p. 300). On average, the focus groups lasted 102 minutes. The recruiting, which is reported to be a difficult task due to a lack of experts and missing incentives (Tremblay et al., 2012, p. 340), was achieved over direct and indirect work and personal relationships of the researchers. Both the EFG and CFG followed a similar procedure (Tremblay et al., 2010, p. 604). Each EFG started with an introduction to the topic and a presentation of the existing elements of the pattern language structure. Thereafter, the participants used brainstorming to identify those patterns for each element that are familiar to them based on their experiences (Tremblay et al., 2010, p. 604). Similar, each CFG started with a description of each element of the pattern language and the corresponding patterns from the EFG. The CFG discussed the ability to transfer each pattern from the foreign industry to the automotive industry. Furthermore, the experts in the CFG ideated further patterns for each element of the pattern language that might be already common in the automotive industry. Due to travel restrictions and the ability to conduct focus groups online, we conducted all focus groups virtually via a video conference application. All participants were asked to enable their cameras to increase interaction (Nili et al., 2017, p.3). One researcher guided each focus group as a moderator and structured the sessions (Brandtner et al., 2015, p. 34). With the permission of the participants, all focus groups were voice recorded and transcribed anonymously (Kitzinger, 1995, p. 301). Non-verbal communication that is considered in most focus groups (Nili et al., 2017, p.3) was ignored due to the lack of presence that might lead to unnatural and incomplete observations. As supposed by Tremblay et al. (2010, p. 605) we analyzed the transcriptions of the EFGs with the template analysis, which can be applied to analyze any kind of textual data. Template analysis is a structured coding process, where a coding template is defined and applied to the transcript accordingly (Brooks et al., 2015, p. 203). Therefore, a template needed to be defined, which is described in the next chapter. The template requires certain information per pattern. Therefore, the transcriptions of the interviews were coded in a dual approach. First, concepts that might lead to a pattern were coded in each EFG transcription. Within each concept, the information that are required per template were extracted with another set of codes. After each coding process, the identified concepts were compared to concepts from previous focus groups and the information eventually merged with existing ones. If no similar concept existed before, a new template was prepared. In the CFGs, these concepts were then evaluated, if they might work in the automotive industry and required changes were discussed. After confirmation in the CFGs, the concepts became patterns.

4 The Pattern Language

4.1 Template for the pattern documentation

In general, pattern languages are not documented with a formal language (Matthes and Ernst, 2009, p. 469). Rather, authors define a template with individual categories that they use to document the patterns of their language (Wellhausen and Fießer, 2011, p. 1). A common structure for documentation is essential for convenience and clarity (Alexander et al., 1977, p. 10). The design of a pattern documentation template is of high importance as it is the basis of the pattern language and thereby, it defines its' quality. Furthermore, the template is a research result of type construct itself, according to March and Smith (1995, p. 256).

Pattern Type	**Author**	**Headline**	**Number**	**Alias**	**Exemplary Image**	**Context**	**Summary**	**Problem**	**When to use**	**Solution description**	**Solution schematic**	**Instruction**	**(Dis-)advantages**	**Related patterns**	**Examples**	**Forces**	**Variants**
1	Gassmann	X	X		X					X				X	X		
1	Sprenger	X						X		X	X				X		
1	Feldhusen	X			X	X		X		X	X			X	X		
1	Goodyear	X				X		X		X				X	X		
2	Alexander	X	X		X	X				X	X	X		X			
2	Schermann	X				X		X			X		X		X		
2	Köhne	X				X		X		X			X			X	
2	Mezaros	X	X			X		X		X					X	X	
2	Paschke	X						X		X			X		X		
2	Schmidt	X				X		X		X		X					
2	Reinfurt	X		X	X	X		X		X			X	X	X	X	X
3	Hohpe	X				X		X		X	X	X		X			
3	Gamma	X		X	X			X	X	X	X	X	X	X	X		
3	Reinfurt	X		X	X	X				X	X		X	X	X	X	
	Buckl	X	X	X	X		X			X	X	X	X				
	Sum	**15**	**4**	**4**	**7**	**10**	**1**	**9**	**1**	**14**	**8**	**5**	**7**	**8**	**9**	**4**	**1**

Table 4-1 Comparison of existing pattern language documentation structures

As no standard pattern documentation template exists, we reviewed 15 pattern languages and summarized the components of each template in Table 4-1. To derive our template, we discuss the need of each component. As a result, we derive a standard template that other researchers might also use for their pattern language.

All the existing templates have a self-explaining headline, while only a few included a number and an alias under which experts might also know the pattern. We will also use a headline to improve the understanding and communication of each pattern and a number to structure the patterns. An alias will not be used, as the headline will be self-explaining. Furthermore, most templates contain a verbal description of the underlying problem, the context in which a problem occurs and the subsequent solution for the problem in the specified context. The problem might be like several patterns that differentiate in their context and solution. In our template, we will use such a verbal description for these components to make each pattern self-explanatory. In addition, authors often used a schematic to explain and visualize details, which we will adopt to ease the understanding for the pattern. As we include a schematic for visualization, we will neglect the use of an exemplary image. To enable practitioners to assess the impact of any pattern, we will include a list with advantages and disadvantages as well as examples, which almost half of the existing templates also did. We will also state related patterns, as the users need to be guided to consider interdependencies. As these components already guide users to apply the patterns, it is not required to state forces that hinder or support the pattern usage and instructions on how to use and implement a pattern. The final template is shown in Table 4-2.

<table>
<tr><th>No. #</th><th colspan="2">Title: Self-explaining heading</th></tr>
<tr><td colspan="3">Problem:
Statement of the precise problem and a justification why an element is needed.</td></tr>
<tr><td colspan="3">Context:
Context, in which the problem might occur.</td></tr>
<tr><td colspan="3">Solution Schematic:
Logical schematic of the solution by using a component diagram.</td></tr>
<tr><td colspan="3">Solution description:
Verbal description of the proposed pattern.</td></tr>
<tr><td colspan="2">Advantages:
Positive aspects of the pattern.</td><td>Disadvantages:
Negative aspects of the pattern.</td></tr>
<tr><td colspan="3">Relation to other patterns:
Direct relations and interdependencies with other patterns.</td></tr>
<tr><td colspan="3">Examples:
Where and how is the pattern already used?</td></tr>
</table>

Table 4-2 Template for the pattern documentation

4.2 The patterns of the pattern language

Layer	Element	Pattern	EFG 1	EFG 2	EFG 3	EGF 4
Product Layer	1.1 Sensor Data Gathering	1.11 Passive Wiretapper	X	X		
		1.12 Active Wiretapper			X	X
		1.13 Decentral Agents				
	1.2 Edge Computing	1.21 Transfer pipe	X	X		X
		1.22 Sensor Signal Filter		X	X	
		1.23 Rule-based Engine	X		X	
		1.24 Machine Learning Engine	X			X
	1.3 Vehicle Exchange Unit	1.31 Asynchronous Communication	X	X	X	X
		1.32 Synchronous Communication			X	X
		1.33 No Communication				X
	1.4 Offline Manager	1.41 Wireless ad hoc network	X	X		
		1.42 Circular Buffer		X		X
		1.43 Transfer Prioritizer				X
Device Layer	2.1 Transfer Translator	2.11 Product-sided Translator		X	X	X
		2.12 Central Translator				X
	2.2 Device Governance	2.21 Message-oriented middleware		X	X	X
		2.22 Connection-oriented Communication			X	
	2.3 Consent Manager	2.31 Mandatory Consent	X			X
		2.32 Categorized Consent		X		
		2.33 Purpose Related Consent				
Digital Twin Layer	3.1 Static Digital Twin	3.11 As-Configured Digital Twin	X	X	X	X
		3.12 As-Built Digital Twin		X		X
		3.13 As-Maintained Digital Twin				X
	3.2 (Pseudonymized) Digital Twin	3.21 Status Tracker		X		X
		3.22 Data Lake	X		X	X
		3.23 Data Warehouse		X	X	X
	3.3 Anonymized Digital Twin	3.31 Identity Masker				X
		3.32 Asymmetric Identity Encryption		X	X	X
		3.33 Event logger			X	
		3.34 Data shifter				

Table 4-3 Overview over the patterns of the pattern language and the associated focus group (1/2)

	Element	Pattern	EFG 1	EFG 2	EFG 3	EGF 4
Platform Layer	4.1 Service Platform	4.11 Visual Rules Engine			X	
		4.12 Descriptive Application		X		X
		4.13 Predictive Application	X			X
		4.14 Prescriptive Application				X
	4.2 Database Federator	4.21 Data Query Language		X		
		4.22 Programming Interface	X			X
	4.3. Twin Configurator	4.31 As is configuration		X	X	X
		4.32 Configuration Management			X	

Table 4-4 Overview over the patterns of the pattern language and the associated focus group (2/2)

In the EFG, we identified two to four patterns for each element of the pattern language. Based on these focus groups, the pattern language contains 36 patterns to gather, manage, store, and analyze sensor data from internet-connected products. In the CFG, we validated the usefulness of the identified patterns for the automotive industry. In addition, we explored two more patterns in the CFG that were not mentioned by experts in the EFG. Table 4-4 gives an overview over the patterns that were identified for the elements of the pattern language. Furthermore, Table 4-4 gives an overview over the focus groups that mentioned the patterns. 24 out of the 38 patterns were mentioned by at least two EFG independently. Overall, the CFG confirmed that all patterns from the EFG are valid in automotive industry in individual contexts. Therefore, we added the context, the advantages and disadvantages based on the results from the CFG. In the following sub-chapters, we summarize the patterns. A detailed documentation of the patterns in the template format can be found in the attachment.

4.2.1 Sensor data gathering patterns

Sensors are devices that transform real world phenomena, like the physical status of a product, into electrical signals (Marabelli et al., 2017, p. 352). Sensor signals depict the phenomena that they monitor objectively and in high detail (Schoch 2017, p. 1592). The signals are often used in technological products to steer actuators that enable the product to react to pre-defined situations. When these signals are recorded systematically, they create transparency about the product status along the products' life. Therefore, sensor signals are the basis for a digital twin of individual products and the applications accordingly (Gerloff and Cleophas, 2017, p. 3). Sensors are distributed inside technological products, as they are physically placed where the phenomena occur. This leads to the problem that the distributed signals need to be systematically accessed. We identify the patterns **passive wiretapper**, **active wiretapper** and **decentral agents** that enable sensor data gathering.

Within the product, sensor signals are exchanged via one-to-many communication networks between sensors, actuators, processing units and other components. A **passive wiretapper** is an additional component that is installed on these networks that reads all signals that are exchanged. The **passive wiretapper** can only read traffic in the format and frequency that the signals occur on the communication network. Furthermore, it cannot request any preprocessed data, such as the diagnosis data, from connected processing units.

"The typical solution is at the vehicle you don't ask anything. Very important that you may not write any questions, any things. This is a security of the car. That is why you read the data on the CAN bus, you can read quite normally what is written on there." – CFG 1

Like a **passive wiretapper**, an **active wiretapper** is a component that is installed on the communication network. The **active wiretapper** can listen to the traffic on the network. In addition, it can request data from components independently from the formats and frequencies of the exchanged signals. It can even request data that is not exchanged on the network at all, such as preprocessed diagnosis data. Nevertheless, by actively communicating with other components, an active request for data creates a high back- and forth-traffic that might disturb other components.

"But the problem (with a passive wiretapper) that really hurts is you get certain information; you don't get it by just reading. Therefore, you do not just get it by reading it. (...) If you now want to query more complicated things, then you can actively ask the components to send the information." – CFG1

The **decentral agents** act like the **active wiretapper**. Some or all components on the communication network have an agent slave. The agent master might just listen to the traffic on the network or request additional data from any components' agent slave. In addition, the agent master might reprogram the agent slaves. Thereby, the master can reprogram the type and frequency that the slaves send data on the communication network regularly.

"(The passive wiretapper) is of course also very efficient if you know in advance that there is data that I need anyway. (...) If I know this in advance anyway, I see the possibility of knowledge that can still be changed in the several years of development time. If one can change such a requirement. So, you need an agent master system." – CFG2

4.2.2 Edge computing patterns

Sensors create electrical signals that depict the status of a single measurable phenomenon. This means that information of interest cannot always be measured directly, as multiple phenomena could be involved. Therefore, the information of interest requires the fusion of multiple signals (Dibaei, 2020, p. 402). Furthermore, sensor signals might be created at high rates. Nevertheless, such high data rates might not be needed for the analysis (Schoch 2017, p. 1592). As a result, some degree of edge computation might be needed (Gerloff and Cleophas, 2017, p. 16). By performing edge computing, sensor signals become status data as they depict the status of the product or components. We identify the patterns “transfer pipe”, “sensor signal filter”, **rules-based engine** and **machine learning engine** that enable edge computing.

A **transfer pipe** transfers all data without preprocessing. Thus, the **transfer pipe** makes no preprocessing. Thereby, it avoids that relevant details are deleted by any preprocessing. Nevertheless, this might lead to a high volume of data and high data rates.

“So, for example let’s say you have an approach where you look at a lot of different data types and you asking a question like how we can predict when this machine breaks down. In addition, you don´t know what data, what sensor type influences or can tell you when this machine is due to break down. Therefore, you kind of want to make sure that data is available when you need it. Therefore, you just transfer all data without preprocessing.” – EFG1

The **sensor signal filter** systematically reduces data rates and complexity by filtering out irrelevant signals. This might be a filter in terms of fixed intervals, signal frequencies, state changes or others. Nevertheless, data are not processed and thus, even basic data analytics algorithms cannot be performed inside the vehicle.

“The bottleneck is always the data rate over the air. And in some cases, sensor signal filtering or rules-based engines are simple mathematical operations (that can help to reduce data rates).” – CFG2

A **rule-based engine** brings parts of the data analysis as a preprocessing task to the edge. By aggregating and transforming the sensor signals within the product, a transfer of raw signals can be avoided, and data rates are reduced accordingly.

“If its alight to just get the standard deviation and say if the standard deviation of the last ten or twenty measurement rows or the measurement cycles or whatever you use. Is out of a certain boundary then send this data to the cloud.” – EFG1

Detecting patterns might require machine learning that processes highly detailed sensor signals to detect the patterns. By using a **machine-learning engine**, sensor signals can be analyzed for patterns locally inside the product. Thereby, the high amount of (raw) data does not need to be transferred outside of the product. Instead, the results of the pattern detection can transferred.

“I'll create a trained model across all robots on the platform there, so we have to derive all of the information from the robots. But obviously, if it comes to edge computing, then I mean, based on the latency requirement of the product regarding, you know the use case, then there is obviously the ability to push down the trained model” – EFG3

4.2.3 Vehicle exchange unit patterns

When sensor data is gathered and preprocessed, the product status data of interest are available in the required format and frequency. Nevertheless, the data are only available inside the product. To store and analyze it in the digital twin, it needs to be transferred out of the product. In addition, the product might need an interface to receive new instructions for data gathering and preprocessing (Newell and Marabelli, 2014, p. 6). As a result, a concept to transfer the data is needed (Wang, 2015, p. 175). We identify the patterns **asynchronous communication**, **synchronous communication** and **no communication** that enable the exchange of data.

The transfer of data between the product and a product external IS requires some communication. As the product might not always have access to a gateway to transfer the data, an **asynchronous communication** can be used. Thereby, the product can send data that IS can request asynchronously. Real-time transfer and streaming of data cannot be realized.

„It goes just more and more in the direction of MQTT and just in the direction of asynchronous processing! Therefore, we do not always need everything online from the car. That is an important insight that we now have. " – CFG1

To grant applications a real-time access to data from the product, a **synchronous communication** between the product and the IS can be realized. Thereby, data is consumed directly. This enables real-time data access and analysis.

„The only thing where you really need some new things, like streaming directly from the car, there are your own use cases for that, especially in the context of autonomous driving." – CFG1

There might be situations where no data, neither synchronously nor asynchronously, should be exchanged. In such cases, **no communication** can be used. With **no communication**, existing data are (temporarily) not transferred between the product and the IS.

"So, for example, if you have a fighter jet that goes into stealth mode, nothing comes out and nothing goes into that fighter jet. So, your offline manager probably needs a check offline, and then nothing comes in and nothing goes out until someone's taken it out of that mode." – EFG3

4.2.4 Offline manager patterns

Product status data are generated by a product and transferred to a product external IS. This transfer requires an active communication between the product and the external IS. (Temporarily) there might be no active communication between the product and the product external IS. Furthermore, there might be situations when available data rates are lower than the volume of data that should be transferred (Schoch 2017, p. 1592). Therefore, data that cannot be sent at the time when it is available for sending need to be managed. We identify the patterns **wireless ad hoc network**, **circular buffer**, **memory management** and **transfer prioritizer** that enable the offline management.

When multiple products are operated at different locations, only a subset of these products has access to the central gateway. If the products can communicate with each other, they can build a **wireless ad hoc network**. Data can be transferred via the product-to-product communication until a product with an access to the central gateway transfers the data to the gateway.

"So, you have the different things that can interconnect to themselves and have different transmission protocols like Bluetooth or ZigBee with each of them have a very limited transition range. But then you use the range of each different thing to connect to a gateway." – EFG1

When data needs to be buffered, if no or a limited transfer is possible, the **circular buffer** can store new data. Whenever possible, it transfers the oldest available data. In case of an overloaded **circular buffer**, it overwrites the oldest data. Thereby, data might be lost.

"We have built a special non-volatile memory. It costs a lot of money, somehow in the good double-digit range, and we are on the road with it. (...) For two weeks to operate the vehicle offline and we collect over this period of time we collect the data and then we derive them so far was again the online connection have again from." – CFG1

If data shouldn´t be lost and saved along the life of a product, aggregating status data to countable events might be used. This avoids non-prioritized overwriting of data. Thereby, a **memory management** assigns each data type a specific storage space. Whenever there are new data of a specific type, the old data are updated by aggregating the new and old data.

"If we talk about offline, how do I manage to store the history? With less volume than for these circular buffers. Since there is just in the control units that already have a long history to have there is a so-called history memory. Where then really very special events, maneuvers, states are counted. And they lie for it to the end of all days." – CFG2

In case of further limitations regarding data transfer rates or storage capacities, the **transfer prioritizer** assigns each data a certain priority. Thereby, data can be transferred or stored according to the assigned priority.

"You need to understand the priority of the data coming in and coming out so it can select or assistance selecting the correct communication protocol and speed. (...) Do you know where the machine data will have a priority over the ERP driven tablet that someone's wandering around with, or a voice core that you see here are prioritized and all the machine dates could go 5G and everything else is going to go LTE, and you can even do it by zones." – EFG3

4.2.5 Transfer translator patterns

The products might have varying functionalities and use different hardware platforms, sensors, software versions and others that hinder interoperability (Barbaresso et al., 2014, p. 6). Thereby, even the same variant of a product might use different technology due to varying component suppliers or software versions that are built into the products along the product manufacturing lifecycles. As a result, the formats of status data might differ between the products and variants of a manufacturer. For data analysis, there might be the need to harmonize product status data (Kammler et al., 2019, p. 5). To ease data governance of stored product status data in the product external IS, a translation of product status data is required prior to data storage. We identify the patterns **product-sided translator** and **central translator** that enable the transfer translation.

With a **product-sided translator**, the translation of the heterogeneous product status data takes place locally inside the product. Therefore, each product performs the translation. Hence, data from all products are transferred in a harmonized format.

"Now we saw it at a big German carmaker for instance they push the standardization layer or this standardization level as far down as possible. But this resulted in factories where you have old machinery and they built individual edge devices (...). So, the new ones are usually aligned to standards and the old ones, if you need to integrate them, then they try to standardize as early as possible." – EFG3

A **central translator** on the other hand translates the heterogeneous product status in the central IS. Hence, data from all products are transferred in a heterogeneous format and translated to the harmonized format centrally.

"If you want to translate it to another format on demand (...) we put a translator on the cloud to translate from all cloud information model to another information model." – EFG4

4.2.6 Device governance patterns

Along the life of a product, data is not always exclusively exchanged in a one-to-one relation between a sender and a receiver. Rather, data might be exchanged between multiple products and / or product external IS (Gerloff and Cleophas, 2017, p. 6). We identify the patterns **message-oriented middleware** and **communication-oriented communication** that enable the device governance.

With a **message-oriented middleware,** data can be transferred between one sender and one to many recipients. Thereby, the sender splits the data into messages and sends them to the middleware. The middleware forwards each message to one or many recipients. If the data is mostly shared between one sender and one recipient, the indirect exchange via the middleware might be in efficient.

“And any implementation alternatives that we are seeing a lot is obviously having a kind off IoT hub. That underneath is leveraging an MQTT broker, (...) you're defining topics that providers can provide data and consumers can consume data on. (...) They are just provisioning or brokering the incoming information to subscribers” – EFG3

An efficient exchange of data between one sender and one recipient can be established with a **connection-oriented middleware**. Via this connection, data is directly exchanged, and no middleware is involved to forward the data.

“The vehicles then collect the data. The data can in turn be compressed again by these additionally available small evaluation mechanisms, trigger mechanisms. And these are then derived and sent to the back end as a data packet in connection with the vehicle data, because this is required for normalization.” – CFG1

4.2.7 Consent manager patterns

Product status data unveil how a product is used (Newell and Marabelli, 2014, p. 6). Therefore, it might unveil the behavior of the person that operates the product. In addition, product status data might unveil confidential tasks that the user performs with the product. Depending on the legal regulations of the region where the product is operated and / or the companies’ and customers’ privacy policy, the storage of status data might be restricted without customer consent (Dremel et al., 2017, p. 82). Therefore, the customers or the user of the product might need to give consent, before data can be transferred (Newell and Marabelli, 2014, p. 8). We identify the patterns **mandatory consent**, **categorized consent** and **purpose related consent** that enable the consent management.

With a **mandatory consent**, the user or customer has no option to disable the consent. By using the product, he explicitly or implicitly agrees that his data is gathered, stored, and analyzed. Refusing to gather data for specific purposes lead to service shut down.

“When you buy a car of this one brand, you get a phone book with 6000 pages and a signature field on the last page. And you confirm everything with that. There is no yes or no. And if you don't sign it, you can't use this car. That’s it.” – CFG2

Purposes to analyze product status data can be grouped into categories. With a **categorized consent**, the user or customer has the possibility to give or refuse the consent to categories. Consent cannot be given to specific purposes, but categories.

“Making it possible to opt out for customers. So some apps they have the yes I am ready to supply my statistics box selected but you can unselect it.” – EFG1

Before new product status data can be gathered or a new application can analyze these data, the **purpose related consent** requires a user or customer consent. Therefore, the user or customer can approve or refuse any data usage depending on the intended purpose with a *purpose related consent*.

"I don't give you permission to use my data. But now you have offered me with quite a nice offer off. Something pay as you drive insurance. And part of that my accepting that offer. I am changing my permissions." – EFG3

4.2.8 Static digital twin patterns

The components that the products of a manufacturer consist of might differ due to product variants. Furthermore, the component versions or suppliers might change along the time that a product is manufactured. Therefore, the bill of materials might differ between products. Nevertheless, this information can be important to interpret product status data. In addition, this information might be needed in order communicate with the product. We identify the patterns **as-configured digital twin**, **as-built digital twin** and **as-maintained digital twin** that enable the static digital twin.

The **as-configured digital twin** is the most basic static twin as it stores a list of components that an individual product consists of. It does not contain information about the version or any manufacturing information.

"Internally, we have something what we call master data where we store product specifications in there. And these are accessible via our normal windows lock in data from there." – EFG4

The **as-build digital twin** builds on the **as-configured digital twin**. In addition to the configuration data, the *as-build digital twin* stores details that could identify the unique manufactured components and product. There might be multiple products with the same **as-configured digital twin**. However, the **as-build digital twin** is unique. Nevertheless, the **as-build digital twin** might not be correct over the life of a product due to missing updates in case of maintenance, repair, and overhaul.

"This car may or may not have an air conditioning system, but it does have four wheels on the steering or five wheels still may go away. So that's the then is the as built, as designed, is associated with a model number. (...) And it puts the serial number of the part of the options in there." – EFG3

Some components might be replaced in case of maintenance, repair, overhaul, or other reasons along the life of a product. Therefore, the **as-maintained digital twin** builds on the **as-build digital twin**. Whenever a component of the product is changed or modified, these changes are documented in the **as-maintained digital twin** so that it documents the most recent status of the product along its whole life.

"The serial number of the final product. That doesn't change. Everything else in that can change. And you need to track those changes because that comes highly relevant when you're calculating things meantime, before failure." – EFG3

4.2.9 (Pseudonymized) digital twin patterns

A product might continually generate new product status data along its' life. When these data are sent to the product external IS, the data can be analyzed to enable applications (Wang, 2015, p. 175). Therefore, the raw or processed data often need to be stored (Gerloff and Cleophas, 2017, p. 3). The demand to store data exists to analyze data over a longer period, to track changes of the product status, to access data later to detect casual relations in retrospect (Demel et al. 2017, p. 97) and others. We identify the patterns **status tracker**, **data lake** and **data warehouse** that enable the digital twin.

The **status tracker** stores just the most recent data for each data type. Older data are overwritten. This means that historical data on the other hand are not stored for further analysis. Some other method to store data on a long-term basis for other applications might be needed.

"So, what you have now is the current state. Therefore, the data comes via a stream into the platform and there the latest state is maintained. And periodically we also make backups of that state. But what we store is the current state." – EFG2

The long-term storage for historical product status data can be realized by using a **data lake**. As the purpose of the stored data might be unknown, the data can be stored in an unstructured or raw format with a low access performance. The **data lake** does not focus on data storage when the purpose of the analysis is clear up front.

"A separate layer that stores the historic information that's coming from the devices and the changes in the device configuration during time and so on. And for that, you typically have (...) the Data Lake where you're gathering the changes in time. (...) So there is a subsequent storage environment that enables you to drive Analytical process is based on the historic changes or streams and events of information from the devices. (...) First of all, you have to store the historic information somewhere so that you can, you know, analyze it in a performance away." – EFG3

A **data warehouse** is a long-term storage and stores data in a structured format for a specific purpose. Stored data might also be processed already. As data might be accessed and added frequently, it has a higher access performance compared to the **data lake**.

"Not all data is directly shipped into this data warehouse. It's quite expensive, and therefore they only shift data in to that one where it makes sense for that one form or product related development." – EFG4

4.2.10 Anonymized digital twin patterns

Product status data might unveil how a product is used and how the user behaves (Newell and Marabelli, 2014, p. 6). Therefore, there might be the need to anonymize the data and cover personal and / or confidential information. As a result, anonymized status data cannot be related to any product or customer. After anonymization, data might be stored permanently and analyzed with fewer restrictions than data that is not anonymized (Newell and Marabelli, 2014, p. 6). We identify the patterns **identity masker**, **asymmetric identity encryption**, **event logger** and **data shifter** that enable the anonymized digital twin.

By using an **identity masker**, status data are stored without anonymization. The anonymization takes place by restricting the access to the subset of the stored data that does not unveil the identity.

"Instead of having to store data and then duplicate that data for anonymization, you can actually do dynamic data masking. (...) My role is to do vehicle quality, root cause analysis and so forth. Then I can get all the anonymous digital to in data. I don't actually care who's driving the vehicle." – EFG3

Like the **identity masker**, data is not completely anonymized by an **asymmetric identity encryption** when it is stored. Instead, the encryption uses a key to encrypt critical status data before they are stored. Other data remain unencrypted. The key to decrypt the data is hold by the product user so that it cannot be decrypted.

"So, you store, that is you could encrypt the data. (...) What data needs to be encrypted? Maybe the vehicle identification number needs to be encrypted with the rest is fine." – EFG3

If data should be completely anonymized, the event logger ensures anonymity of stored data. An **event logger** stores only uncritical product status data. Therefore, critical status data are removed prior to its' storage. Then, the remaining status data are stored in a pool with data from other products. After the data are stored, the identity of the product cannot be traced back anymore.

"I am now thinking specifically of a weather map. For example, only weather-relevant data should be included, but no park information, for example. And if the data is now anonymized, it is detached from the other use cases and can no longer be merged in a simple manner. In other words, you can imagine that there are individual silos for weather maps and then individual silos for parking tickets." – CFG1

Product status data might contain data that could unveil the products identity. If critical data needs to be stored that would get lost during anonymization, the **data shifter** can be used. Therefore, it brings some degree of randomness to the status data. This randomness can be chosen according to the demand to analyze the data, so that it does not disturb the analysis.

"So, what I know, there's a random generator that cuts a new ID. For further processing. Then the data may change once a week." – EFG2

4.2.11 Service platform patterns

Product status data depict the status of individual products. By just storing status data, no value is created. Instead, data needs to be systematically visualized (Wang, 2015, p. 178) and analyzed (Kammler et al., 2019, p. 5). Thereby, data analysis creates benefit such as supporting decision making, increasing customer satisfaction and others (Dremel et al., 2017, p. 84). We identify the patterns **visual rules engine**, **descriptive application**, **predictive application,** and **prescriptive application** that enable the product external data analysis.

Users without knowledge in data science might need to analyze product status data. The **visual rules engine** offers users pre-defined data analysis components that can be combined and applied on status data. Thereby, users without knowledge in data science can flexibly access and analyze these status data.

"It's very simple, mean, like a node red or Grafana or similar so that you can do some local flow configuration, some basic calculation in python embedded etcetera and some basic visualization just to make life easier." – EFG4

In case of a demand to frequently perform a similar analysis, standard applications are needed. A **descriptive application** depicts the as is state of a product status in a descriptive manner. Therefore, the application describes the history and present.

"I mean, you typically start with, you know, traditional reporting in order to understand, you know what happened in history than you obviously want to understand. Why did it happen? So, you needed more detailed information." – EFG3

A **predictive application** analyzes historical data to detect patterns in the data. Based on these patterns, the application predicts the future.

"These services that you mentioned for example predictive maintenance. These are services which I developed and then just run as a service. So that people are dragging components together and building their own services based on some components." – EFG1

A **prescriptive application** is based on a *predictive application*. It uses the predictions about the future to react to the expected predictions and influences them. Therefore, the application predicts and influences the future.

"We want the data not only for future products or predictive services, but also to be able to put new software on the vehicle much faster." – CFG2

4.2.12 Database federator patterns

Product status data are stored in one-to-many databases. Furthermore, status data might be analyzed together with other product related data from distributed (legacy) data sources (Barbaresso et al., 2014, p. 32). Thus, applications might need to access several databases. Therefore, a method to query product related lifecycle data is needed. We identify the patterns **data query language** and **programming interface** that enable the database federation.

The **data query language** is a method to directly access stored data. The method to access the data is based on the **data query language** of the data source. Applications that access multiple data sources might need to use multiple different methods when using the **data query language**.

"If the user wants something, then an IT solution architect make a direct connection to the database. And then there are 20000 of them to the many databases." – CFG2

The **programming interface** avoids using different methods for different data sources by requesting product status via a middleware. The method to access the data is based on the middleware and independent from the data source. The middleware abstracts the different data sources and enables the access to pre-defined data with one harmonized method.

"If you are the one who is owning the data you usually write endpoints or queries which you can we can search the database for very specific reasons or very specific functions." – EFG1

4.2.13 Twin configurator patterns

The demand for product status data might change along the life of a product (Kammler et al., 2019, p. 11). Such a change can be driven by new or changed applications that analyze the status data, by a changed customer consent and other reasons. Whenever there is such a demand, the data gathering, preprocessing and transfer in the individual products might need to be changed. Therefore, the individual products might need to be reconfigured. We identify the patterns **as is configuration** and **configuration management** that enable the reconfiguration.

The manufacturer defines the status data that should be sent along the products life in the design and manufacturing phase. With an **as is configuration**, this configuration cannot be changed along the life of a product. Instead, the product transfers the configured data in the pre-defined format along its whole life.

"There's no configuration possible at all, because that's basically a box you buy, and then you're put it into place and it runs does what it should do, but it's not really intelligent." – EFG4

As the intend to gather and analyze product status data might change along the life of a product, the **configuration management** enables one to change the configuration. Thereby, the configuration can be dynamically changed to address a changed demand.

"So, let's say we had something where we didn't collect vibration before, but now we decided we collect noise. Now we actually realized that noise or vibrations a leading indicator failure. So, we have to go back and put an additional sensor and then that twin configure it." – EFG3

5 Discussion

Vehicles as technological products are equipped with numerous sensors. These sensors measure the status of the vehicles' components along the life of an individual vehicle and enable an efficient steering of the vehicle functionalities. Driven by the availability of IoT technology, vehicle manufacturers aim to embed communication technology to their products and create a digital twin of the individual vehicles in a vehicle external IS. With the study at hand, we aim to enable vehicle manufacturers to design a holistic IS to gather, manage, store, and analyze sensor-based status data and create digital twins of the vehicles. To guide them to systematically gather vehicle status data, we developed a pattern language for the design of a suitable IS. By conducting four EFG we derived patterns, i.e., best practices, to gather, manage, store, and analyze sensor-based status data in other industries that already use such IS for their internet connected products. With two subsequent CFG, we transfer the patterns from the other industries to the automotive industry.

For each element of the pattern language structure from the previous research essay, we identified two to four patterns that solve the described problems already in a non-automotive context by conducting EFG. Thus, the pattern language consists of 38 patterns in total. Although most patterns were mentioned by multiple EFG, some were mentioned in one focus group exclusively and two were only mentioned in the CFG. The fact that most patterns were mentioned in multiple focus groups shows that these patterns are best practices in multiple solutions and contexts already. Nevertheless, only two patterns were mentioned in all focus groups. The fact that not all patterns were mentioned in all focus groups might indicate that the pattern is not valid for all industries. On the other hand, it might show that the pattern is not known in the other industries and an overview of available solutions to a problem is missing. In the CFG with the automotive experts, we found that all patterns of the EFG are either known already or could be a pattern in the automotive context. Therefore, the experts in the CFG did not exclude any patterns from the EFG. Nevertheless, the automotive experts clearly defined the contexts, where the patterns could be helpful. In addition, they defined advantages and disadvantages of the patterns in the defined contexts.

6 Conclusion and Outlook

Analyzing status data from internet-connected vehicles promises high potentials for vehicle manufacturers to gain a competitive advantage (Dremel et al., 2017, p. 82). Nevertheless, many vehicle manufacturers do not have IS to systematically gather, manage, store, and analyze product status data from individual internet connected vehicles (Brandt and Ahlemann, 2020, p. 8). As IS researchers paid little attention to such VIS in the past (Kaiser et al., 2018, p. 2), we addressed this lack of research by developing a pattern language to guide practitioners in the design of a suitable IS. Therefore, we derived patterns, i.e., best practices, to design a holistic IS to gather, manage, store, and analyze vehicle status data. We conducted EFG with experts from industries that already operate internet-connected products and gather their status data systematically. By conducting subsequent CFG with automotive experts, we assessed the transferability of the derived patterns to the automotive industry. With the pattern development, we address the lack of existing models, which are (1) the missing acknowledgement of the varying implementation contexts, (2) the missing ability to adopt parts of a solutions and (3) the missing consideration of other industries that have already solved the problem. We do so by (1) providing vehicle manufacturers with a set of multiple alternative solutions for each element that can solve the problem in different contexts. As our patterns are (2) loosely linked to each other, manufacturers can integrate a subset of the patterns into their legacy systems and complement their existing systems. Furthermore, we (3) adapt existing and tested solutions from other industries to the automotive industry with EFG and CFG.

Although we followed the DSR process rigorously, our paper is not free of limitations. While we conducted the EFG in four different companies, we conducted both CFG at the same vehicle manufacturer due to a lack of access to experts from other automakers. As a result, other automotive manufacturers might assess the usefulness of the patterns from the EFG differently. Furthermore, we conducted qualitative research with a limited number of experts. As a result, some existing patterns might not have been discovered. To overcome this limitation, we call other researchers to conduct further qualitative and quantitative research to explore more patterns and assess whether the patterns are suitable for other automotive companies.

7 References

Ahlemann, F. and Gastl, H. (2007). „Process model for an Empirically Grounded Reference Model Construction.“ In: Reference Modeling for Business Systems Analysis, p. 77-97.

Alexander, C., Ishikawa, S. and Silverstein, M. (1977). „A Pattern Language – Towns, Building, Construction.“ Oxford University Press, New York.

Barbaresso, J., Cordahi, G., Garcia, D., Hill, C., Jendzejec, A. and Wright, K. (2014). “USDOT’s Intelligent Transportation Systems (ITS) ITS Strategic Plan 2015-2019.” U.S. Department of Transportation.

Belanger, F. (2012). „Theorizing in Information Systems Research using Focus Groups.“ In: Australasian Journal of Information Systems, Volume 17 Issue 2, p. 109-135.

Brandt, N. and Ahlemann, F. (2020). „How do you drive? Analyzing vehicle sensor data in Product lifecycle management systems.“ In: Twenty-Eighth European Conference on Information Systems, p. 1-15.

Brandtner, P., Helfert, M., Auinger, A. and Gaubinger, K. (2015). „Conducting focus group research in a design science project: Application in developing a process model for the front end of innovation.“ In: An International Journal on Information Technology, Action, Communication and Workpractices, Volume 9, Issue 1, p. 26-55.

Brooks, J., Bartys, S., Turley, E. and King, N. (2015). „The Utility of Template Analysis in Qualitative Psychology Research.“ In: Qualitative Research in Psychology, Volume 12 Issue 2, p. 202-222.

Dibaei, M., Zheng, X., Jiang, K., Abbas, R., Liu, S., Zhang, Y., Xiang, Y. and Yu. S (2020). “Attacks and defences on intelligent connected vehicles: a survey.” In: Digital Communications and Networks, Volume 6, p. 399-421.

Dremel, C., Herterich, M., Wulf, J., Waizmann, J. and Brenner, W. (2017). „How AUDI AG Established Big Data Analytics in Its Digital Transformation.“ In: MIS Quarterly Executive, Volume 16, Issue 2, p. 81-100.

Gerloff, C. and Cleophas, C. (2017). “Excavating the Treasure of IoT Data: An Architecture to Empower Rapid Data Analytics for Predictive Maintenance of Connected Vehicles.” In: Thirty Eighth International Conference on Information Systems, p. 1-21.

Hevner, A., March, S. and Park, J. (2004). “Design Science in Information Systems Research.” In: MIS Quarterly Vol. 28 No. 1, p. 75-105.

Holler, M., Stoeckli, E., Uebernickel, F. and Brenner, W. (2016). „Towards Understanding closed-loop PLM: The Role of Product Usage Data for Product Development enabled by intelligent Properties.“ In: BLED Proceedings, p. 479-491.

Hribernik, K., Franke, M., Klein, P., Thoben, K. and Coscia, E. (2017). „Towards a Platform for Integrating Product Usage Information into Innovative Product-Service Design.“ In: International Conference on Concurrent Enterprising, p. 1407-1413.

Johannesson, P. and Perjons, E. (2014). „An Introduction to Design Science.“ Springer Vieweg, Wiesbaden.

Kaiser, C., Festl, A., Pucher, G., Fellmann, M. and Stocker, A. (2019a). „The Vehicle Data Value Chain as a Lightweight Model to Describe Digital Vehicle Services.“ In: 15th International Conference on Web Information Systems and Technologies, p. 1-12.

Kaiser, C., Stocker, A. and Fellmann, M. (2019b). „Understanding Data-driven Service Ecosystems in the Automotive Domain.“ In: American Conference on Information Systems, Volume 25th, p. 1-10.

Kaiser, C., Stocker, A., Festl, A., Lechner, G. and Fellmann, M. (2018). „A Research Agenda for Vehicle Information Systems.“ In: European Conference on Information Systems Research-in-Progress Papers, p. 1-12.

Kammler, F., Hagen, S., Brinker, J. and Thomas, O. (2019). „Leveraging the value of data-driven service systems in manufacturing: a graph-based approach.“ In: Twenty-Seventh European Conference on Information Systems, p. 1-15.

Kitzinger, J. (1995). „Introducing focus groups.“ In: BMJ Clinical Research, Volume 311, p. 299-302.

Kuschel, J., and Dahlbom, B. (2007). „Mobile Services for Vehicles.“ In: European Conference on Information Systems, p. 1863-1874.

Marabelli, M., Hansen, S., Newell, S. and Frigerio, C. (2017). „The Light and Dark Side of the Black Box: Sensor-based Technology in the Automotive Industry.“ In: Communications of the Association for Information Systems, Volume 40, p. 351-374.

March, S. and Smith, G. (1995). „Design and natural science research on information technology.“ In: Decision Support Systems, Volume 15, p. 251-266.

Matthes, F. and Ernst, A. (2009). „Capturing, Structuring and Passing Knowledge for the Design of Complex Systems as Patterns.“ In: Business & Information Systems Engineering Issue 6, p. 468-470.

McAfee, A. & Brynjolfsson, E. (2012). “Big Data: The Management Revolution”. https://hbr.org/2012/10/big-data-the-management-revolution. Last visited on 20.02.2021.

Newell, S. and Marabelli, M. (2014). “The Crowd and Sensors Era: Opportunities and Challenges for Individuals, Organizations, Society, and Researchers.” In: Thirty Fifth International Conference on Information Systems, p. 1-17.

Nili, A., Mary, T. and David, J. (2017). „A framework and approach for analysis of focus group data in information systems research.“ In: Communications of the Association for Information Systems, Volume 40, p. 1-21.

Peffers, K., Tuunanen, T., Rothenberger, M. and Chatterjee, S. (2014). „A Design Science Research Methodology for Information Systems Research.“ In: Journal of Management Information Systems, Volume 24 Issue 3, p. 45-78.

Schoch, J., Staudt, P. and Setzer, T. (2017). „Smart Data Selection and Reduction for Electric Vehicle Service Analytics.“ In: Proceedings of the 50th Hawaii International Conference on System Sciences, p. 1592-1601.

Smolander, K., Rossi, M. and Purao, S. (2008). „Software architectures: Blueprint, Literature, Language or Decision.“ In: European Journal of Information Systems, Volume 17, p. 575-588.

Stocker, A., Kaiser, C. and Fellmann, M. (2017). „Quantified Vehicles – Novel Services for Vehicle Lifecycle Data.“ In: Business and Information Systems Engineering, Volume 59, p. 125-130.

Teixeira, C., Braga de Vasconcelos, J. and Pestana, G. (2019). “A Knowledge Management System for Analysis of Organisational Log Files.” In: 13th Iberian Conference on Information Systems and Technologies (CISTI). Spain: Casceres, p. 1-4.

Tremblay, M., Hevner, A. and Berndt, D. (2012). „Design of an information volatility measure for health care decision making.“ In: Decision Support Systems, Volume 52, p. 331-341.

Tremblay, M., Hevner, A., Berndt, D. and Chatterjee, S. (2010). „The Use of Focus Groups in Design Science Research.“ In: Communications of the Association for Information Systems, Volume 26, p. 599-618.

Vaia, G., Carmel, E., DeLone, W., Trautsch, H. and Menichetti, F. (2018). „Vehicle Telematics at an Italian Insurer: New Auto Insurance Products and a New Industry Ecosystem.“ In: MIS Quarterly Executive, Volume 11 Issue 3, p. 113-125.

Wang, Y., Li, Y., Tian, D., Wang, J. and Wang, J. (2015). “A method of visual management platform for connected vehicles data.” In: International Conference on Connected Vehicles and Expo, p. 175-179.

Wellhausen and Fießer (2011). „How to write a pattern?: a rough guide for first-time pattern authors.“ In: EuroPLoP '11: Proceedings of the 16th European Conference on Pattern Languages of Programs, p. 1-14.

8 Appendix

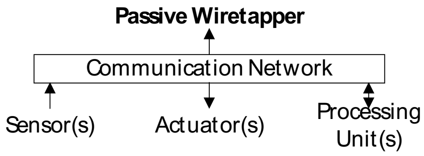
8.1 Sensor data gathering patterns

1.1	Passive Wiretapper

Problem:
See chapter 8.1

Context:
Electrical signals are transferred between the distributed sensors, processing units, actuators, and other components on one or multiple communication networks inside the product. The signals are already transferred on this network in the format that is needed for the digital twin. In addition, the frequency of the signal availability is sufficient. Furthermore, a component that can read the traffic on the *communication network* can be added to the *communication network*.

Solution Schematic:



Solution description:
Install a *passive wiretapper* to the existing communication network. The *passive wiretapper* reads all signals that are transferred over the network. It processes all signals on the communication network by means of identifying each signal to determine its' relevance. Then, the *passive wiretapper* extracts only the information relevant. The *passive wiretapper* can only read the signals that are already exchanged on the communication network. Thereby, it cannot actively request a signal from a specific sensor, actuator, or processing unit. As a result, signals are only available when another component that is connected to the communication network requests it, or the sensor publishes the signal due to an internal rule.

Advantages:	Disadvantages:
+ Can be installed to existing products as retrofit solution without influencing other components + Existing communication on the network is not disturbed + Low traffic on the network	- Signal intervals cannot be influenced - Other components (sensors, processing units) cannot be updated - Not all signals can be read, as some signals need to be actively requested - Signals with a high frequency might block the wiretapper

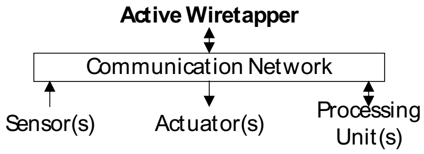
Relation to other patterns:
Sensor Signal Filter, Rule-Based Engine, Machine Learning Engine, As Is Configuration

Examples:
An industrial robot has multiple embedded sensors that measure the robots' status to enable its' operation. The sensors exchange the signals on an internal communication bus so that the processing units can analyze the data and steer the robot. To extract the existing signals, a *passive wiretapper* is installed on a standardized interface of the robot as a retrofit data gathering solution. Thereby, all data that is exchanged on the communication bus anyway can be extracted.

1.2	Active Wiretapper

Context:
Electrical signals are transferred between the distributed sensors, processing units, actuators, and other components on one or multiple communication networks. The signals are not necessarily transferred on this network in the format and/or frequency that is needed. Nevertheless, the signals can be provided by components on the communication network when actively requested by another component. Furthermore, a component on the communication network can read the traffic and write requests to other components on it.

Solution Schematic:



Solution description:
Install an *active wiretapper* on the existing communication network. The *active wiretapper* processes all signals that are exchanged over the communication network by means of identifying each signal to identify its' relevance. Then, the *active wiretapper* extracts only this information that are relevant to extract. Besides reading existing signals, the *active wiretapper* can also request a signal from a specific sensor or any other component that is connected to the communication network. Whenever a signal is needed, the *active wiretapper* sends a request du the corresponding component.

Advantages:	Disadvantages:
+ Can be installed to existing products as retrofit solution without influencing other components + In the passive state, existing communication on the communication network is not disturbed + Required for maintenance procedures, where product internal data is requested	- Other components (sensors, processing units) cannot be updated - Requests for additional signals might disturb other communication and cause additional traffic - The active wiretapper might need to know the software status of the components to send correct requests to the components

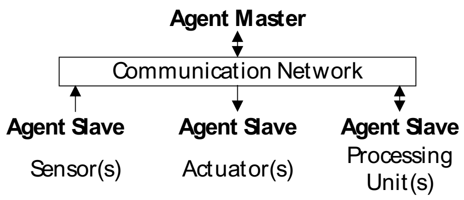
Relation to other patterns:
Transfer Pipe, Rule-Based Engine

Examples:
The OBD-2 interface of vehicles is an interface to access the CAN bus of a vehicle, which is the communication network of sensors, actuators, and processing units within vehicles. Traditionally, this interface is used at service stations to read diagnosis data. These diagnosis data are sensor signals that are preprocessed by the vehicle processing units and stored internally. By installing an *active wiretapper* on this interface, existing signals on the CAN bus can be read. In addition, diagnosis data might be requested. Thereby, the *active wiretapper* receives additional signals that can be further processed.

1.3	Decentral Agents

Context:
Electrical signals are transferred between the distributed sensors, processing units, actuators, and other components on one or multiple communication networks. The signals are not necessarily transferred on this network in the format and/or frequency that is needed. Nevertheless, the components on the communication network could technically provide the signals. There is a need to gather the new or changed data periodically and not only once. Last, there might be a need to update sensors, actuators, or processing units over time.

Solution Schematic:



Solution description:
Install *decentral agents* on the relevant components of the communication network. Each sensor, actuator or processing unit has an agent slave. A central processing unit works as an agent master and can send instructions to the agent slaves. These instructions include updated rules that define the data that the agent master wants to receive from the individual components. These instructions might be one time, temporary or permanent requests. Accordingly, the agent slaves send the signals to the agent master over the communication network. Furthermore, the agent master might read other traffic on the communication network if some of signals that the agent master extracts are transmitted over the network anyway.

Advantages:	Disadvantages:
+ Which and how status data are gathered can be flexibly reprogramed + Except for updating components, the agent master is passive (communication network is not disturbed) + Software updates of single components can be realized	- Requires distributed computation units (agent slaves) - The product cannot be used during updates of the agent slaves

Relation to other patterns:
Transfer Pipe, Rule-Based Engine, Machine Learning Engine, Configuration Management

Examples:
Wind turbines use *decentral agents* to update the embedded sensors, actuators, and processing units. Wind turbines are optimized by the manufacturer along the whole life of a turbine. To optimize turbines that are already operated, the manufacturer sends updates to the turbines. The turbine operation is then paused, and the components are reprogrammed via the *decentral agents*.

8.2 Edge computing patterns

2.1	Transfer pipe

Problem:
See chapter 8.2

Context:
This pattern solves the problem, if sensor data gathering already provides the sensor signals at the rate and format that is needed for the analysis in the digital twin. This might be the case, if the data frequency that is needed for the analysis is the same as it is for the internal process steering. Furthermore, the analysis might require raw sensor signals. In addition, this pattern is applied, when any edge computing would drop important information.

Solution Schematic:

Solution description:
Install a *transfer pipe* to the product. The *transfer pipe* receives the raw sensor signals. It forwards all signals without preprocessing. The format and frequency of the raw signals remains unaffected.

Advantages:	**Disadvantages:**
+ High fidelity of data available for analysis due to no preprocessing	- High data rates - High data variety - No control over the frequency and format

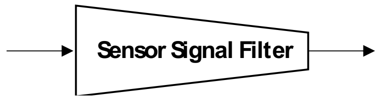
Relation to other patterns:
Active Wiretapper, Decentral Agents, Synchronous Communication, Connection-Oriented Communication, Data Lake, Predictive Application, Prescriptive Application

Examples:
A squeezing machine produces pressed parts for car bodies. This machine is designed by its' manufacturer to control the manufacturing process. Nevertheless, the machine does not have the computation capabilities to perform unsupervised machine learning to predict machine failures. Therefore, the sensor signals that are used to predict the failure are transferred by a transfer pipe without any preprocessing. By forwarding the signals, the limited processing capabilities of the machine is not a hurdle to create insights.
In a vehicle, local processing units analyze sensor signals and store the results of the analysis as diagnosis data. When the vehicle arrives at a service station, an *active wiretapper* is applied to the OBD-2 interface of the vehicle. This connector requests diagnosis data that the processing units created during the vehicles' usage. Then then connector acts as a *transfer pipe*, as all diagnosis signals are extracted and forwarded.

2.2 Sensor Signal Filter

Context:
This pattern solves the problem, if sensor data gathering already provides the sensor signals in the format that is needed for the analysis in the digital twin. Just the frequency of the signal availability is higher than it is needed.

Solution Schematic:



Solution description:
Install a *sensor signal filter* to the product. The *sensor data filter* receives raw sensor signals. It filters the signals based on pre-defined rules. By applying a rule-based filtering, data rates might be reduced. The *sensor signal filter* reads all signals and drops irrelevant ones by comparing the received signal with the last relevant signal under the consideration of the filtering rule. Rules include time intervals, signal frequencies and state changes. The pattern does not perform any other preprocessing.

Advantages:	**Disadvantages:**
+ High fidelity of data available for analysis due to no preprocessing + Controlled and reduced data rates	- High data volume due to unprocessed sensor signals - High data variety - No control over the format - Details might be filtered out

Relation to other patterns:
Passive Wiretapper, Rule-Based Engine, Asynchronous Communication, Data Lake

Examples:
A mobile application tracks the position of the user. Therefore, the positioning sensor measures the position in pre-defined intervals. Accordingly, sensor data gathering makes the signals that unveil the position, available to in these intervals. By then, the *sensor signal filter* uses pre-defined rules to compare the last result with the actual result. In case that the deviation is below a defined threshold, the recent position is dropped.

2.3 Rule-based Engine

Context:
Sensor signals are available in the frequency that is needed for the analysis. This pattern solves the problem if these sensor signals are not needed in a raw format. Thus, the signals can be preprocessed by applying pre-defined rules to analyze single or multiple signals to increase the information content. Furthermore, the result of the signal analysis might be used inside the product to steer product functionalities.

Solution Schematic:

Solution description:
Install a *rule-based engine* in the product. The engine uses raw sensor signals as input. The *rule-based engine* does not perform any filtering, as signals are provided in the frequency that is needed for the analysis. Based on pre-defined rules, the *rule-based engine* processes the raw signals and outputs processed data. Processing steps include monitoring if a limit value is exceeded, aggregations of single signals as well as an analysis and combination of multiple signals. Hence, the information content of the signals is increased, and the format of data transfer can be changed. After processing, data can be used either to transfer it outside of the product or to use it for product steering purposes.

Advantages:
+ Controlled and reduced data rates
+ Controlled data format
+ Low latency
+ Avoid the transfer of irrelevant details

Disadvantages:
- Details might be filtered unavailable after processing

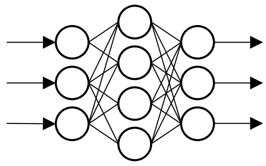
Relation to other patterns:
Passive Wiretapper, Active Wiretapper, Decentral Agents, Sensor Signal Filter, Asynchronous Communication, No Communication, Status Tracker

Examples:
A manufacturer that provides sensor systems to retrofit legacy machines uses the *rules-based engine*. Therefore, they process all sensor signals on the edge to reduce data rates and decrease complexity of central data handling. Besides, they decrease latency so that the sensor systems can trigger an alarm and stop the machine in case of critical signals. As a result, only highly, aggregated data are sent to the digital twin.
The fuel level in vehicles cannot be measured directly by sensors. Instead, multiple phenomena are measured that are analyzed together to derive the fuel level. Furthermore, the fuel level is further processed with other sensor signals to derive the fuel consumption. Thereby, a *rules-based engine* in processes the signals on a vehicle level. Thus, data of interest is transferred outside the vehicle instead of irrelevant signals that would need to be analyzed anyway to generate insights. Thus, the raw measurements itself are not worth its' gathering.

2.4 Machine Learning Engine

Context:
To some degree, the physical product instances are used differently. This means that the sensor signals of multiple similar products differ, based on the products' usage. Therefore, not all behavior can be predicted and covered by (static) rules. Rather than identical sensor signals, similar patterns of sensor signals might occur.

Solution Schematic:



Machine Learning Engine

Solution description:
Install a *machine-learning engine* in the product and apply it to the sensor signals. The *machine-learning engine* detects patterns that are already known with supervised machine learning. Furthermore, it can monitor the sensor signals to identify unknown patterns with unsupervised machine learning to find anomalies. Thereby, the *machine learning engine* outputs highly aggregated results about detected patterns that are already known and new patterns that were observed.

Advantages:
+ Controlled and reduced data rates
+ Controlled data format
+ Avoid the transfer of irrelevant details
+ Detection of patterns

Disadvantages:
- Finding correlations between signals from multiple products is impossible on the edge
- High computation power on the edge required

Relation to other patterns:
Passive Wiretapper, Decentral Agents, No Communication, Predictive Application, Prescriptive Application

Examples:
During the stamping process, material faults like cracks or inclusions might occur. The manufacturer expects that the embedded sensors measure such faults when they occur. Such a pattern detection would reduce costs for manual inspection of the pressed parts for fault. As the manufacturer operates only a few different stamps with different sensors, he does not expect any correlations between the different stamps. Therefore, he applies machine learning to detect patterns between any quality issues and the recorded sensor signals in the product. Data are only transferred when patterns are detected. Thereby, data rates and central data governance are reduced.

8.3 Vehicle exchange unit patterns

3.1	Asynchronous Communication

Problem:
See chapter 8.3

Context:
Data between the product and the product external information system is exchanged in intervals or sporadically rather than in real-time. Thereby, no streaming of data takes place. Furthermore, no processing of exchanged product status data in (close to) real-time is planned and high latencies are not a hurdle. Therefore, exchanged data might not be used to steer product functionalities that require fast responses to observed phenomena.

Solution Schematic:

Solution description:
Realize an *asynchronous communication* between the product and the product external information system. Thus, either the product and/or the information system initiates the communication at any time. Therefore, the sender sends a message. Thereafter, the other stakeholder might receive the message later. The receiver can make an active request to receive data that was previously sent to it earlier. Alternatively, data is sent to the passive receiver in intervals or under certain conditions. All data is buffered in the time between sending and receiving.

Advantages:	Disadvantages:
+ Communication is not blocked-in case of parallel communication + The product does not need to be available	- No guaranteed data processing in real time

Relation to other patterns:
Sensor Signal Filter, Rule-Based Engine, Wireless Ad Hoc Network, Circular Buffer, Message-Oriented Middleware

Examples:
MQTT protocol: A device publishes sensor data. This message is sent to a message broker who distributes the data to all subscribers of this data. Besides direct processing, the broker might forward data to subscribers later in case of the subscribers' unavailability. Furthermore, the broker stores messages that it forwards in a query and avoids that communication is blocked. Therefore, an *asynchronous communication* is used for the data transfer.

3.2	Synchronous Communication

Context:
Data between the product and the product external information system is exchanged in (close to) real-time. Furthermore, data might be streamed. The fact that other communication is blocked during the communication is not a hurdle.

Solution Schematic:

Solution description:
Realize the concept of *synchronous communication* between the product and the product external information system. Thus, either the product and/or the digital twin might initiate the communication at any time. When the connection between the recipient and the sender is initiated, other communication is blocked. During the data transfer, the recipient listens to the data in close to real-time. The only delay is caused by physical parameters for data transmission. To transfer data, the recipient and the sender need to be available at the same time.

Advantages:	Disadvantages:
+ Data processing in real time	- Communication to other products is blocked until the original communication ends - Both product and the digital twin need to be available at the same time

Relation to other patterns:
Transfer Pipe, Circular Buffer, Transfer Prioritization, Connection-Oriented Communication

Examples:
Http protocol: Although the most popular applications of the http protocol is the delivery of websites in the world wide web, the http protocol can also be used to transfer data between a product and another information system. Therefore, either the receiver requests or the sender actively sends data. During the data transmission, the sender and the receiver cannot be involved in other communication. Furthermore, both the recipient and the sender need to be available at the same time as there is a *synchronous communication*.

3.3	No Communication

Context:
(Temporarily) no exchange of status data is needed. Therefore, no communication between the product and the digital twin takes place.

Solution Schematic:

Solution description:
Block all data transfer between the product and the digital twin information system with *no communication* when no communication should take place. The offline mode might be switched on and off. Data might be handled with any offline management.

Advantages:	**Disadvantages:**
	- Delay between data gathering and transfer - Data might be lost or abstracted in case of long intervals

Relation to other patterns:
Rule-Based Engine, Machine Learning Engine

Examples:
When a fighter jet is on a mission, it might be a priority to hide. Therefore, it can be required to block all communication to avoid the risk of being found. Therefore, data that is gathered is temporarily not exchanged.

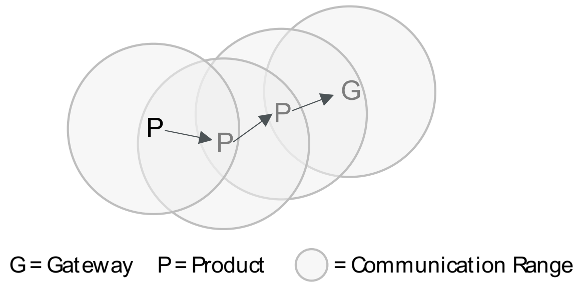
8.4 Offline manager patterns

4.1	Wireless ad hoc network

Problem:
See chapter 8.4

Context:
Multiple products transfer product status data to a central gateway. Furthermore, the products themselves can communicate with each other. There might be situations, where only some of the products have access to the central gateway due to its limited network coverage. Nevertheless, the product-to-product communication range of products within the network coverage of the central gateway is higher than the distance between the products. In addition, other products that are also out of this range are in the product-to-product communication range of any other product that has an indirect communication established with a product in the range of the central gateway.

Solution Schematic:



Solution description:
Use wireless a *wireless ad hoc network*. The products without a connection to the central gateway search for other products that that have a direct or indirect connection to the central gateway. By establishing product-to-product connections, the products extent the network coverage of the central gateway. Products outside of the network coverage of the central gateway transfer the product status data to other products, which then forward the data. Finally, the status data is sent to the central gateway, when a product within the network coverage of the central gateway receives the data.

Advantages:	Disadvantages:
+ Messages from devices that are outside of the network coverage of the central gateway can send data	- Complex device to device communication - High latency - Data are transmitted multiple times - Legal obstacles when one product stores data from other products

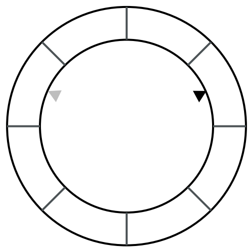
Relation to other patterns:
Asynchronous Communication, Message-Oriented Middleware

Examples:
A mobile phone is operated in an area where one of the multiple network carriers has no network coverage. If a mobile owner temporarily operates the mobile in an area with no or a bad network coverage, he cannot use all services appropriately. If a mobile that uses another network carrier that has a well access to the central gateway in this area is nearby, the mobile that has limited service could send his or her data to the mobile with a good access to its central gateway. Thus, the mobile with a good access to the central gateway forwards the data.

4.2	Circular Buffer

Context:
Temporarily, a product has no access to a central gateway. Along this time, product data cannot be transferred out of the product. Data that is gathered along this period should not be deleted. Furthermore, the amount of product data that is generated along the time where the product has no network coverage is mostly similar. In addition, data that might be deleted without being transferred is not an issue.

Solution Schematic:



► = Read Position ► = Write Position

Solution description:
Use a *circular buffer* to buffer data locally. When there is data available for transfer, but no (sufficient) connection to the gateway is existing, data is stored in the write position of the *circular buffer*. As soon as a sufficient connection to the central gateway is established, stored data is read and sent. If the *circular buffer* is full, the earliest data is overwritten, even if it was not transmitted. Thereby, the buffer size specifies the duration that data can be buffered before data is overwritten.

Advantages:	Disadvantages:
+ Simple offline management	- No real-time access to data in offline phases - Data might be lost due to the rolling over-writing - Data is overwritten without prioritization rules

Relation to other patterns:
Asynchronous Communication, Synchronous Communication, Transfer Prioritization, Connection-Oriented Communication

Examples:
A machine is operated without an access to a central gateway. Only in case of maintenance activities, manual transfer of machine data is performed. Therefore, the *circular buffer* works as a storage until they are exchanged in the maintenance intervals. As both the maintenance intervals and the machine data that is generated and stored depend on the machine's usage time, the buffer size can be calculated so that no data is dropped.

4.3	Transfer Prioritization

Context:
A product generates status data at a higher rate than it is possible to transfer or store it. It is possible to assign a level of criticality to each data that should be transferred / stored. Therefore, data can be prioritized.

Solution Schematic:

Criticality	Data
1	…
2	…
…	…
n	…

Solution description:
Each data type that should be transferred has a pre-defined level of criticality assigned. In case of limited data rates for data transfer or storage, the *transfer prioritization* prioritizes the transfer or storage of data according to its criticality. Nevertheless, ongoing data transferring, or storage processes are not stopped, when data with a higher priority is available. Nevertheless, few exceptions can be defined that even overrule and stop existing processes.

Advantages:
+ Critical data will be available in case of limited transfer or storage rates
+ Stored data is only overwritten by data with higher priorities

Disadvantages:
- Maintenance of criticality levels

Relation to other patterns:
Synchronous Communication, Circular Buffer, Connection-Oriented Communication

Examples:
In a factory, machines communicate with each other and the central steering system. When a client is integrated into the factory network that is not critical in terms of ensuring process stability, this client has a lower priority. In case of bandwidth issues, the data from this client is deprioritized and not sent in real-time but stored locally until a higher bandwidth allows sending it.

4.4	Memory Management

Context:
For an unlimited time, a product has no access to a central gateway. Along this time, product data cannot be transferred out of the product. Data that is created along this period should not be deleted. Furthermore, detailed data are not necessarily needed for the analysis. Therefore, data might be aggregated. In addition, data that might be deleted without being transferred is an issue.

Solution Schematic:

Physical address	Aggregated Value
#1	…
#2	…
…	…
#n	…

Solution description:
A non-volatile memory is installed. The *memory management* defines specific data points that are of interest and reserves a storage capacity for these data points. Along the life of the product, new data is added to the existing data point by aggregating the data. Therefore, the size of the data might stay constant along the life, although data might be added.

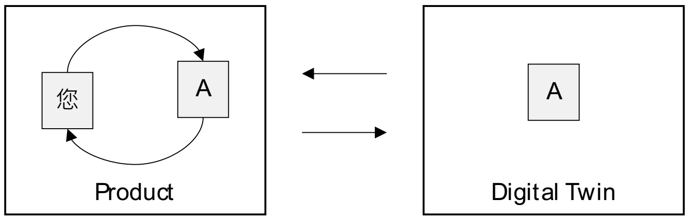
Advantages:
+ Simple offline management
+ Lifelong storage of critical data

Disadvantages:
- No real-time access to data in offline phases
- Data is highly aggregated

Relation to other patterns:
Asynchronous Communication, Synchronous Communication, Transfer Prioritization, Connection-Oriented Communication

Examples:
The historical battery storage of a vehicle stores the energy throughput of the battery along the vehicles' life. Therefore, the energy throughput is summed up. This value of the energy throughput can access whenever needed along the vehicles' life. Thereby, a very limited storage capacity is sufficient for the storage of lifelong information.

8.5 Transfer translator Patterns

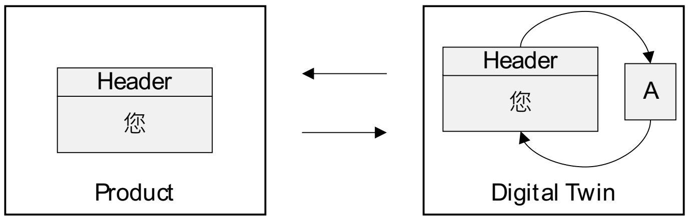
<table>
<tr><th>5.1</th><th>Product-sided Translator</th></tr>
<tr><td colspan="2">Problem:
See chapter 8.5</td></tr>
<tr><td colspan="2">Context:
Product status data should be harmonized as early as possible to avoid complex data governance in the central information system. Thereby, all products should exchange status data in the common format of the central information system.</td></tr>
<tr><td colspan="2">Solution Schematic:


</td></tr>
<tr><td colspan="2">Solution description:
Install a device-sided translator in each product. To perform the translation of data formats from the heterogeneous formats of the individual products to the common data format, product status data are translated into the common format in the individual products. Before sending data to the information system, all data is translated to the common format. Thereby, data is only heterogeneous on a product level. In case that the common data format changes, instructions for the translation are changed in each product. The device-sided translator can either already be installed in the design of new devices or retrofitted with additional components to legacy devices.</td></tr>
<tr><td>Advantages:
+ Data transfer in a common format
+ Interoperability of multiple products</td><td>Disadvantages:
- Installation needed in each product
- Update of the common data format needs updates of each product</td></tr>
<tr><td colspan="2">Relation to other patterns:
Message-Oriented Middleware, Connection-Oriented Communication, As-Configured Digital Twin, As-Built Digital Twin</td></tr>
<tr><td colspan="2">Examples:
In a highly automated production facility, machines communicate with each other and with the central manufacturing system. To ensure interoperability of the machines, the standardization is realized within the products. Legacy machines with different standards are equipped with a, additional device to enable the device-sided translation.</td></tr>
</table>

5.2	Central Translator

Context:
Product status data should be harmonized in the central information system to avoid local preprocessing. Furthermore, the common format is changed on a regular basis.

Solution Schematic:



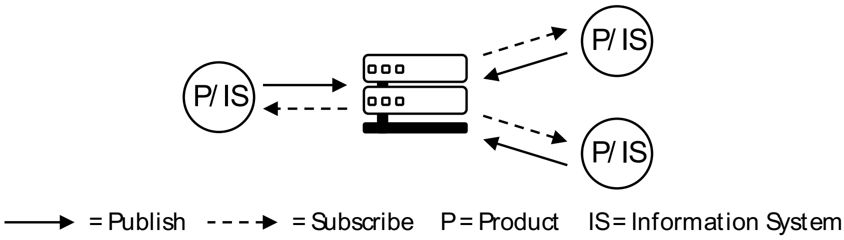
Solution description:
Install a *central translator* to the information system. To perform the translation of data formats from the heterogeneous formats of the individual products to the common data format, instructions for the translation into the common format are stored in the information system. The instructions might depend on the product variant, sensor status and other details. To perform the central translation, all the translation relevant information is included in the header of the data. Based on the header, the correct instructions are used to translate the heterogeneous data to the common format. In case that the common data format is changed, instructions for the translation are changed in the *central translator*.

Advantages:	Disadvantages:
+ Single point of data translation + The product hardware design is independent from the translation	- Limited interoperability between multiple products - Overhead that specifies the components and translation needed - In case that the header only contains the product ID: High effort needed to maintain the product specifications along their lives as parts of the product might be reconfigured

Relation to other patterns:
Connection-Oriented Communication, As-Maintained Digital Twin

Examples:
A vehicle continually generates diagnosis data about the status of different components along their lives. Based on these diagnosis data, a components' status can be determined. When these diagnosis data are accessed at a service station, the technician requests the assembly status of the vehicle in a central database. Based on the assembly status and in-build parts, he queries the diagnosis data from the vehicle and translates the received data based on the known assembly status. Without access to the assembly status of the individual vehicle, the technician would neither be able to query the diagnosis data in the vehicle nor would he be able to interpret the request results.

8.6 Device governance patterns

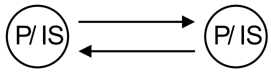
<table>
<tr><th>6.1</th><th>Message-oriented middleware</th></tr>
<tr><td colspan="2">Problem:
See chapter 8.6</td></tr>
<tr><td colspan="2">Context:
Multiple products and / or product-external information systems regularly exchange data. Furthermore, the same data is exchanged between multiple stakeholders. Last, data is exchanged in the form of events rather than streams.</td></tr>
<tr><td colspan="2">Solution Schematic:


</td></tr>
<tr><td colspan="2">Solution description:
Use a message-oriented middleware to distribute data between the product(s) and / or the information system(s). The message-oriented middleware has an interface to receive product status data or any other kind of data. It stores the data temporarily and forwards it to other product(s) or information system(s) that fulfill pre-defined rules. Therefore, there is no direct communication between two stakeholders. Rather, the message-oriented middleware distributes the data.</td></tr>
<tr><td>Advantages:
+ Bidirectional data transfer
+ Efficient in case of a one-to-many communication</td><td>Disadvantages:
- High data rates in case of a one-to-one communication
- No streaming of data</td></tr>
<tr><td colspan="2">Relation to other patterns:
Asynchronous Communication, Wireless Ad Hoc Network, Product-Sided Translator, Event Logger</td></tr>
<tr><td colspan="2">Examples:
A weather station gathers sensor data about the temperature, humidity, and others. To transfer the weather data to all research institutes and weather forecasting companies, the weather station uses the MQTT protocol to distribute the sensor data. Therefore, it sends all data to a central message broker that acts as a message-oriented middleware. This broker forwards the data to all authenticated subscribers of the subject.</td></tr>
</table>

6.2 Connection-oriented Communication

Context:
Multiple products and / or product-external information systems regularly exchange data. Nevertheless, the same data is only rarely exchanged between multiple products and information systems. In addition, (close to) real-time access to data might be critical. Last, data might be exchanged as a data stream rather than events.

Solution Schematic:



P = Product IS = Information System

Solution description:
Use a *connection-oriented communication* to exchange data between the product(s) and / or the information system(s) directly. Whenever a product or an information system wants to initiate a data exchange with another product or information system, it requests a connection. When a connection is established, a direct communication can take place. For the duration of the connection, no other connection can be established with the same interface.

Advantages:	Disadvantages:
+ Bidirectional data transfer + Streaming of data + Low data rates in case of a one-to-one communication	- High data rates in case of a one-to-many communication

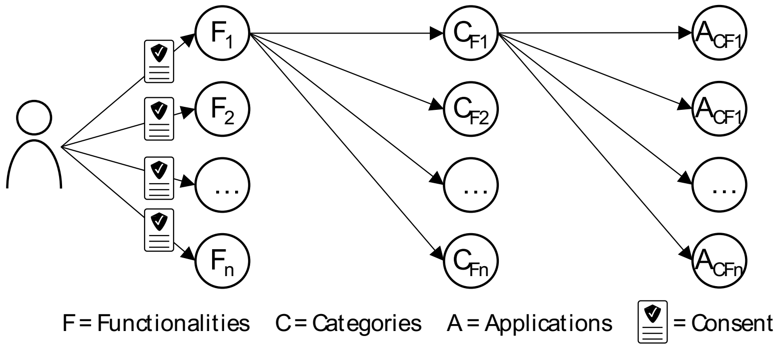
Relation to other patterns:
Transfer Pipe, Synchronous Communication, Circular Buffer, Transfer Prioritization, Product-Sided Translator, Central Translator, Data Lake

Examples:
A manufacturer analyses machine data in real time to detect anomalies and predict failures. Therefore, the manufacturer uses a TCP based protocol. Thereby, the machine establishes a connection to the information system. After the connection is established, the machine streams product status data as a time series of data to the information system.

8.7 Consent manager patterns

7.1	Mandatory Consent
Problem: See chapter 8.7	
Context: It is possible to restrict the product or a functions' usage to customers that agree to data gathering. This restriction might exist for some customer groups or some data only.	

Solution Schematic:



Solution description:
Use a *mandatory consent*. Before the product and its' functions can be used, the owner, user or any other need to confirm that data can be gathered, stored, and analyzed. This consent might be given with an active agreement or by starting the products' usage. Therefore, product functionalities are blocked if no consent is actively or passively given. The *mandatory consent* might be restricted to some data and / or some intended uses. As the consent is mandatory, it is not possible to modify the consent.

Advantages:	**Disadvantages:**
+ Data can be gathered from all products without restrictions	- Does not meet all legal or company private regulations

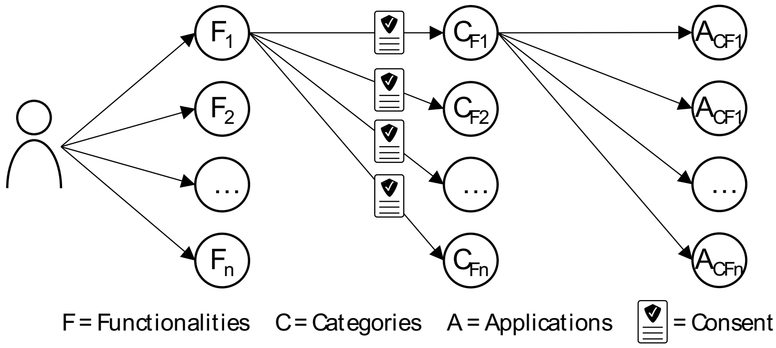
Relation to other patterns: -

Examples:
The Chinese government demands that all vehicle manufacturers supervise the status of their electric vehicles. Furthermore, the government collects these data. As there is a legal demand to gather product status data whenever the vehicle is operated, a *mandatory consent* is chosen. Thereby, the driver agrees that certain data is gathered, stored, and analyzed whenever he uses the vehicle. The driver has no opportunity to refuse.

7.2 Categorized Consent

Context:
Product status data is analyzed for different specific purposes. These purposes might be product improvements, customer applications and others. These specific purposes can be logically group into categories. It is sufficient to gain customer approval for categories to gather, store and analyze product status data for all purposes of one category.

Solution Schematic:



Solution description:
Use a *categorized consent*. The usage of the product and its functionalities is independent from a given consent. The only restriction might be functions that require product status data and therefore, the usage is blocked until a consent is given. Before any product status data is gathered, the customer might approve or decline data gathering for a purpose (e.g., services, marketing, …) in one consent. Whenever there is a request du gather new or changed data, the customer consents in the category of the new or changed data is checked. Although a categorized consent is given, a request to gather data might not only be revoked for categories but also specific data types or uses. Therefore, data gathering can be restricted after a consent is given to a complete category.

Advantages:	**Disadvantages:**
+ Consent for new applications clear (according to the category)	- Disagreement to specific applications blocks all applications from the category - Might problematically according to legal or company restrictions

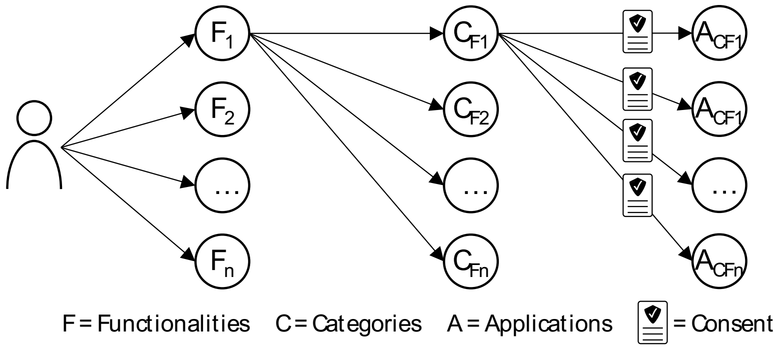
Relation to other patterns:

Examples:
Website use cookies to track visitors if they use the website. Furthermore, cookies track the behavior over websites. Some countries legally require website owners to get the visitors' approval to store cookies on the computers. Therefore, cookie managers commonly group cookies according to their intended use into categories. These categories might be statistical, marketing or technically required cookies. The user might then agree or disagree that certain cookies are used.

7.3	Purpose Related Consent

Context:
Product status data is analyzed for different specific purposes. Before data might be gathered, stored, or analyzed for the individual purposes, customer approval to do so is needed for each of the intended purposes.

Solution Schematic:



Solution description:
Use a *purpose related consent*. The usage of the product and its functionalities is independent from a given consent. The only restriction might be functions that require product status data and therefore, the usage is blocked until a consent is given. Before any product status data is gathered, consent need to be given. Furthermore, before the purpose of data analysis can be changed, user is also asked for his consent. Last, the consent for individual purposes can be rejected at any time.

Advantages:	**Disadvantages:**
+ The customer is actively involved and has control about his data	- Customer trust in the data analysis purposes is needed

Relation to other patterns:
Identity Masker

Examples:
A provider of cabs found that customers order more cabs in parts of a big city, when it is raining. The manufacturer of the cabs provided a solution for the cab provider that the cabs send an information to the central information system when the windscreen wiper is enabled. Therefore, the cab provider agreed that all cabs could send status data of the windscreen wiper in case of rain. Therefore, he approved the data gathering for the one individual purpose. After the approval is given, the status data about the windscreen wipers should only be analyzed for the purpose of measuring the weather.

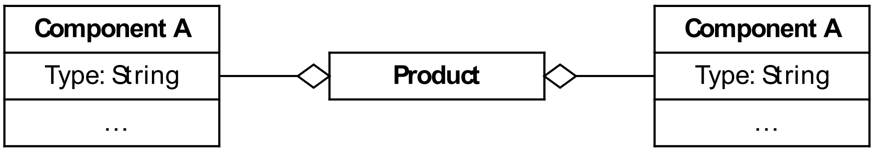
8.8 Static digital twin patterns

8.1	As-Configured Digital Twin

Problem:
See chapter 8.8

Context:
The products and variants of a manufacturer have different in-built components. To gather, store and / or analyze status data from individual products, the type of parts that are embedded to each product need to be known. Nevertheless, there are no different product versions that have an influence on data gathering, storing or analysis.

Solution Schematic:



Solution description:
Create a virtual *as-configured digital twin* for each product. This *as-configured digital twin* stores the configuration of parts that the product is assembled of. Therefore, it might use a reference to other data sources that contain more detailed information that is valid for multiple other products. The *as-configured digital twin* remains static, even if parts are replaced or other parts added along the life of the products.

Advantages:	Disadvantages:
+ Product configuration is known	- Versions of the components are not known - Part replacements are not tracked along the life

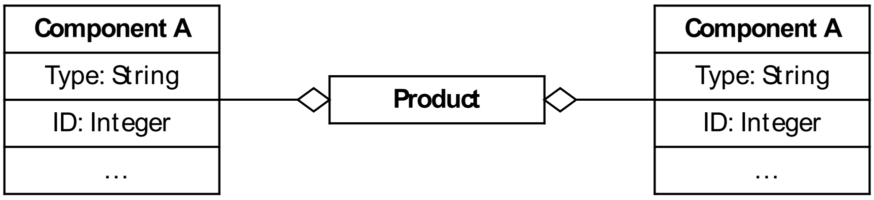
Relation to other patterns:
Product-Sided Translator, As-Built Digital Twin, As-Maintained Digital Twin, Event Logger, As Is Configuration

Examples:
A manufacturer of a sensor system that can be installed at manufacturing machines uses the *as-configured digital twin*. Each sensor system contains individual configurations of sensors. These sensor system configurations are stored in the *as-configured digital twin*. There is no need to know the individual components, as all sensors of the sensor system share the same interfaces and syntaxes for communication.

8.2	As-Built Digital Twin

Context:
The products and variants of a manufacturer have different in-built components. Although many products might seem similar, each product might consist of unique parts. This might be the case due to different production times, suppliers, software version, …. To gather, store and / or analyze data from individual products, information about the individual parts that are embedded to the individual product need to be known.

Solution Schematic:



Solution description:
Create a virtual *as-built digital twin* for each product. The *as-built digital twin* uses the *as-configured digital twin* as a basis. For relevant parts of the *as-configured digital twin*, the *as-built digital twin* adds additional data about the physical instance that might be unique to the product instance. Therefore, data about suppliers, software versions of components and other data about the individual components might be stored. The *as-built digital twin* remains static, even if parts are replaced or other parts added along the life of the products.

Advantages:	Disadvantages:
+ Product configuration is known + Versions of components are known	- Part replacements are not tracked along the products' life

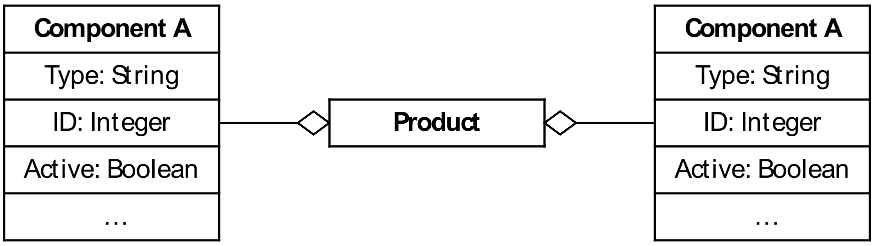
Relation to other patterns:
Product-Sided Translator, As-Configured Digital Twin, As-Maintained Digital Twin

Examples:
Vehicle sensors create electrical signals that need to be interpreted to understand it. To perform the interpretation, the vehicle configuration, software versions of the control units need to be known. Therefore, an *as-built digital twin* is used to understand the signals.

8.3 As-Maintained Digital Twin

Context:
The products and variants of a manufacturer have different in-built components. Although many products might seem similar, each product might consist of unique parts. This might be the case due to different production times, suppliers, software version, …. To gather, store and / or analyze data from individual products, information about the individual parts that are embedded to the individual product need to be known. Furthermore, some components might be replaced along the life of a product. To enable data gathering, storage and analysis even if components are replaced, data about the replaced parts need to be updated.

Solution Schematic:



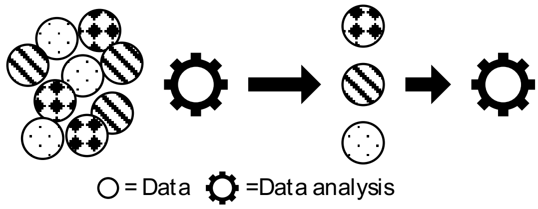
Solution description:
Create a virtual *as-maintained digital twin* for each product. The *as-maintained digital twin* uses the *as-built digital twin*. The twin contains all in-build parts with the individual and unique identification of the in-build parts. In case that a component that is documented in the *as-maintained digital twin* is replaced, the maintainer of the *as-maintained digital twin* updates the information.

Advantages:	Disadvantages:
+ Product configuration is known + Versions of components are known + Part replacements are tracked along the products' life	- Updating digital twin requires high efforts - Not all changes might be mentioned to the maintainer of the *as-maintained digital twin*

Relation to other patterns:
Central Translator, As-Configured Digital Twin, As-Built Digital Twin, Configuration Management

Examples:
In predefined intervals, the engines of a plane are maintained. To increase the planes' overall utilization, engines that were already overhauled replace the engines of a plane. The engine changes are documented and so the *as-maintained digital twin* keeps track of these changes. During the analysis of the digital twin, data the documentation of these changes are used to understand the behavior of the engine over time. Furthermore, recalls in case of failures of specific hardware or software can be steered as the identification of failed products can be performed.

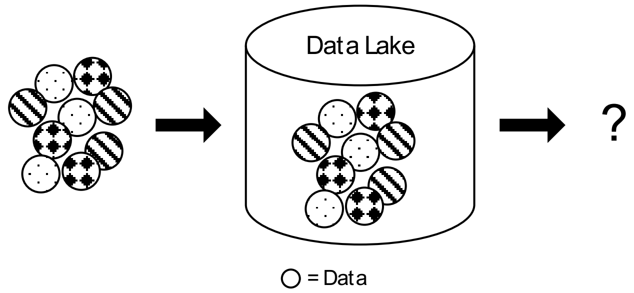
8.9 (Pseudonymized) digital twin patterns

9.1	Status Tracker
Problem: See chapter 8.9	
Context: No historic product status data are needed for the analysis. Instead, knowledge about the most recent product status is sufficient. Therefore, a method to store the last available data point per type is necessary. There is no need to analyze store older data.	
Solution Schematic: 	
Solution description: Use a *status tracker* to keep track of the latest product status(es). The *status tracker* stores either unprocessed or processed status data. There might be different triggers that initiate the deletion of the oldest data. Triggers are new data, pre-defined events, time, and others. In case that such an event occurs, the data is deleted and might be replaced. The *status tracker* might not only store the one latest data per type, but a fixed number of latest data.	
Advantages: + Simple data governance + The intended use of data is clear due to the trigger definition	**Disadvantages:** - No / few historic data available for later analysis
Relation to other patterns: Rule-Based Engine, Event Logger, Descriptive Application, Programming Interface	
Examples: An internet-connected vehicle uses a *status tracker* for a parking application. As soon as the vehicle is locked, it sends its' GPS position to the *status tracker*. The data is stored if it is overwritten by a new GPS position, when the vehicle is locked again. As it is the intended use to guide the driver to the vehicle, only the most recent data are stored. The *status tracker* is realized in a MQTT based IoT hub. The aim of the MQTT broker is to forward data to all subscribers as soon as he receives data from the sender. The broker stores the received data for a limited time until it is forwarded and received by all transcribers. Thereafter, the deletion of older data is triggered in time intervals or by the event that all subscribers received the data.	

9.2	Data Lake

Context:
At the time when the product status data is generated, there is no application that consumes the data. Rather there might be applications in the future that need the data. Nevertheless, it is unclear whether the data is needed, and which format is required. Furthermore, a long-term storage is needed to cope with uncertainty about the future data usefulness.

Solution Schematic:



Solution description:
Use a *data lake* to store data that might be analyzed in the future. The *data lake* stores data in the raw format, as the format of future analysis is unknown. Alternatively, data can be stored in a common format per data type. Besides product status data that is based on sensor signals, other data sources might be added to the *data lake*. Although data is not preprocessed, the *data lake* has a consistent structure to ensure that data can be accessed. Furthermore, the *data lake* has retention policies to steer the duration that each data type is stored. There might be multiple data lakes for different purposes. Data that is stored in the *data lake* has low requirements to access performance and is only invoked irregularly.

Advantages:	**Disadvantages:**
+ Availability of data in case of future applications	- High data volume of unprocessed raw data - Unclear applications

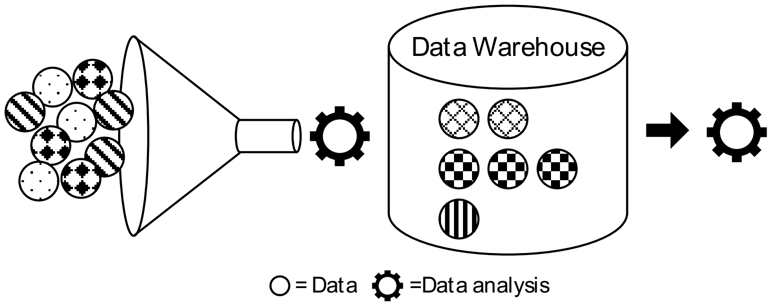
Relation to other patterns:
Transfer Pipe, Sensor Signal Filter, Connection-Oriented Communication, Data Warehouse, Event Logger, Visual Rules Engine, Predictive Application, Prescriptive Application, Data Query Language

Examples:
A manufacturer of consumer products gathers product status data to analyze, how the products are used. The manufacturer focuses on a few functions first. Nevertheless, in case of success he wants to extent the analysis to further functions. Therefore, he stores raw product status data in the *data lake* for functions that he might want to focus on in the future.
A manufacturer of industrial robots uses a *data lake* to store sensor data from multiple robots. In case of failures of some robots, he uses the stored data to detect previously unknown patterns in historic data by comparing the state of multiple similar products. When detailed historic data are available, one can detect patterns of failed products. By detecting these patterns in other products, relevant signals to predict failures in the future can be identified.

9.3	Data Warehouse

Context:
At the time when the product status data is generated, there is an application that consumes the data, or any other intended use known. Therefore, it is clear which data format is required. Furthermore, a long-term storage might be needed to analyze or view the data in the future.

Solution Schematic:



Solution description:
Every time new sensor data arrives, the existing dataset is extended. The *data warehouse* performs some analysis of the new data in combination with other new data from the device or some data that is already stored. After the analysis, data is stored in an aggregated format in the *data warehouse*. Through the early binding of the data, analyzed data that frequently needs to be accessed is only analyzed once and offered to other applications.

Advantages:	Disadvantages:
+ Clear intended use of data + Data are stored in the format that is needed + Historic data are available + Potentially low data volume	- Data are only available when the application is known at time when data is created

Relation to other patterns:
Data Lake, Identity Masker, Asymmetric Identity Encryption, Descriptive Application, Programming Interface

Examples:
A manufacturer aims to increase the efficiency of the routes of his assets. Therefore, the assets are equipped with a tracking sensor and the positions are sent to the *data warehouse*. Before the positions are stored in the *data warehouse*, the asset positions are filtered for changed locations and only relevant data are stored. By providing time series data about the positions, the movement of assets can be derived, and the process optimized.

8.10 Anonymized digital twin patterns

10.1	Identity Masker

Problem:
See chapter 8.10

Context:
It is necessary to track the product status over time. Therefore, new product status data need to be added to the existing data set. There is no need to anonymize stored data. Instead, it is sufficient to restrict access to those parts of the data sets that would unveil the products identify.

Solution Schematic:

Product-ID	ID
Data type a	Data
Data type b	Data
…	…
Data type n	Data

= Mask

Solution description:
Use an *identity masker* to mask parts of the data set. Product status data are stored without anonymization. Instead, data that would unveil the product are marked as such. As the product status data are stored without anonymization, new product status data can be added to the correct data set. Therefore, the product status can be investigated over time. Whenever the pool of masked data is accessed, the masked data cannot be accessed. Therefore, anonymization takes place when data is accessed.

Advantages:	Disadvantages:
+ Tracking of data over time + Data that unveil the identity cannot be accessed + Data from a specific product can be deleted after it is stored	- Data are stored without anonymization - The combination of multiple data might indirectly uncover the product and / or customer identity

Relation to other patterns:
Purpose Related Consent, Data Warehouse, Data Shifter

Examples:
A manufacturer of trucks wants to improve the design of his future trucks. Therefore, he intends to analyze truck status data that is created while the trucks are operated by the customer / driver. The non-anonymized data about the trucks' status is stored by the customer as he analyzes the data to optimize the utilization. Whenever the truck manufacturer intends to access the data, an *identity masker* is applied to critical data that could unveil the driver's identity. Data might be the vehicle identification number and positions. As the truck manufacturer does not need detailed product status data from a single truck but wants to analyze the general product status of fleets, hidden data does not block their analysis.

10.2 Asymmetric Identity Encryption

Context:
It is necessary to track the product status over time. Therefore, new product status data need to be added to the existing data set. Nevertheless, there is a need to anonymize data before storing it.

Solution Schematic:

Product-ID	Encrypted ID
Data type a	Data
Data type b	Data
…	…
Data type n	Data

Solution description:
Use an *asymmetric identity encryption* to encrypt data that could unveil the products' and / or users' identity. Before status data is stored, critical data such as the product-id is encrypted. This might be single data or combinations of multiple data that could unveil the identity. Each time when new data about a product should be stored, the critical data is encrypted. As the encryption remains constant along the life of a product, the encrypted product identification remains constant. The key for decryption of the anonymized data is deleted or handed to trustful entity. Thereby, data is anonymized when it is accessed.

Advantages:
+ Tracking of data over time
+ Data that unveil the identity cannot be accessed
+ No critical data are stored without encryption
+ If the key for decryption is stored by a third instance, data might be unveiled
+ Data from a specific product can be deleted after it is stored

Disadvantages:
- The product identity might be unveiled
- The combination of multiple data might indirectly uncover the product and / or customer identity

Relation to other patterns:
Data Warehouse, Data Shifter

Examples:
A manufacturer of wind turbines improves the operation of existing wind turbines with software updates. To receive real data from existing wind turbines, he uses an *asymmetric identity encryption*. Thereby, critical data that would unveil the owner and / or location of the turbines are encrypted. Nevertheless, data from individual turbines might be gathered along individual turbines lives, as the encrypted identity of the turbines remains constant. In addition, the owner of the turbines has the private key so that he might access detailed data about his wind turbines that are gathered anyway.

10.3 Event logger

Context:
It is not necessary to track the product status from one specific product over time. Rather, single data points from products that fulfill specified characteristics (e.g., model, location, status, …) are needed.

Solution Schematic:

Data type a	Data
Data type a	Data
…	…
Data type a	Data

Solution description:
Use an *event logger* to store data points or combination of data points that do not unveil the products' and / or users' identity. In case that a product submits new data, the *event logger* adds the relevant data to the existing pool of data. The pool contains status data from multiple products that are grouped together. As the *event logger* stores events from multiple products, the identity of the individual products that have created the data cannot be unveiled in the future.

Advantages:
+ Data that unveil the identity cannot be accessed
+ No critical data are stored without anonymization

Disadvantages:
- No tracking of data over time
- Data might be decrypted
- Data from a specific product cannot be deleted after it is stored

Relation to other patterns:
Message-Oriented Middleware, As-Configured Digital Twin, Status Tracker, Data Lake, Data Shifter

Examples:
A cab company operates multiple cabs in a big city. Whenever there is rain in parts of the city, the cab company knows that there will be a higher demand for cabs in these parts of the city. One indicator for the start of rain is the start of the windscreen wipers in the individual cabs. Therefore, the cab company gathers product status data to track when a cab starts the windscreen wipers. As the cab company is not interested in any other data but the cabs' position and the status of the windscreen wipers, the company stores these events without any relation to a specific cab or other data. Nevertheless, a full anonymization is only feasible, when multiple cabs are operated in the same area. Otherwise, the cabs' position can be recalculated.

10.4	Data shifter

Context:
It is necessary for an application to have access to types of product status data that would unveil the identity of a product. Data might not directly unveil the identity, but the tracking over time could unveil products identify.

Solution Schematic:

Product-ID	Shifted ID
Data type a	Shifted data
Data type b	Data
…	…
Data type n	Shifted data

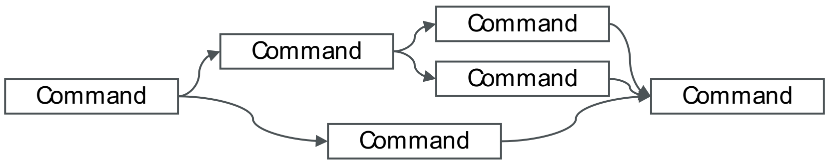
Solution description:
Use a *data shifter* to change product status data before it is stored. In case that a product submits new data, the *data shifter* applies an algorithm that changes the data. Such an algorithm applies randomness to the data that makes it impossible to track the changes back. As a result, the anonymized data that is stored is not exactly the data that is gathered. Nevertheless, the randomness does not disturb the analysis.

Advantages:	Disadvantages:
+ Identity data or combinations of data that would unveil the product identity can be anonymized	- Avoids that data might be decrypted - Shifting data might hinder the analysis

Relation to other patterns:
Identity Masker, Asymmetric Identity Encryption, Event Logger

Examples:
A machine manufacturer wants to gather data from the machines that are operated by its' manufacturers. As data about production volumes and other facts might be confidential and status data of the individual machine could unveil these confidential facts, there is a need for anonymization. Therefore, product status data are shifted in time and value. Thereby, key facts of interest might be changed a little. Nevertheless, the main statement of the data remains.

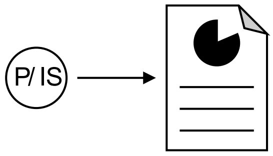
8.11 Service platform patterns

11.1	Visual Rules Engine
Problem: See chapter 8.11	
Context: It is not sufficient to query and analyze existing data with static algorithms. Instead, demand to query and analyze data individually exist. Therefore, business user that aim to solve the business problems and do not have programming knowledge should be able to flexibly query and analyze existing data to generate insights.	
Solution Schematic: 	
Solution description: Use a *visual rules engine* to enable stakeholders to individually analyze product status data. A *visual rules engine* offers users methods to perform pre-defined commands on stored product status data without using traditional programming syntax. Therefore, the commands are represented by visual components. Each visual component represents a specific command. Product status data are then queried in the databases or analyzed by combining the components in a specific sequence. Thereby, flexible programs to analyze product status data can be developed.	
Advantages: + Users with no programming knowledge (like business experts) can access and analyze data + Flexible relations between data processing steps	**Disadvantages:** - Risk that data are misunderstood
Relation to other patterns: Data Lake, Data Query Language	
Examples: In a manufacturing environment, high volume of data from individual machines are gathered and stored. Nevertheless, the manufacturer does not have applications for all the data at the time that he gathers the data. Nevertheless, in case of an event like in case of a machine failure, product status data need to be analyzed. As the manufacturing experts are not necessarily IT-experts, a tool for the business experts to query stored data is needed. Therefore, business experts might use a *visual rules engine* to combine certain querying and analyzing commands. Thereby, business experts are flexible to analyze data without having the effort to involve IT experts for their tasks.	

11.2 Descriptive Application

Context:
Product status data describes how a product is used and how it behaves. To create a benefit, status data need to be processed by an application to depict the status of individual products or statistics about a group of products.

Solution Schematic:



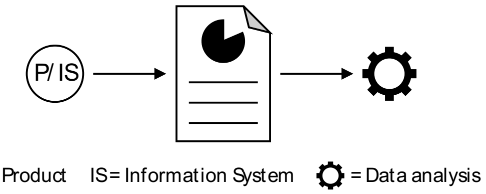
P = Product IS= Information System

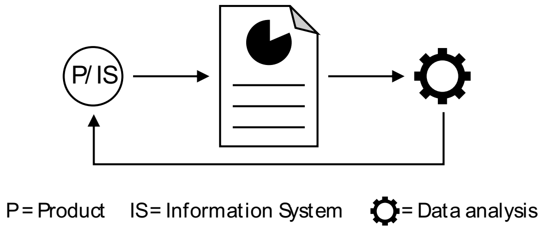
Solution description:
Use a *descriptive application* to depict the products' status. Such an application describes and summarizes status related information. This might be from individual products or a group of products. In addition, the *descriptive service* might depict product data in detail or summarize it in statistics. It helps one to understand situations that occur and to justify decisions.

Advantages:	**Disadvantages:**
+ Enables basic understanding + Detection of previously unknown facts	- Requires manual interpretation

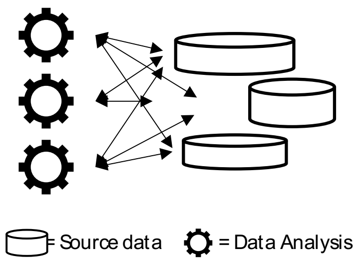
Relation to other patterns:
Status Tracker, Data Warehouse, Programming Interface

Examples:
The developer of a mobile application wants to measure the engagement rate of the people that have installed the application. Furthermore, they intend to measure the effectiveness of their promotions. Therefore, they use a *descriptive application* that regularly exports a report about the customer engagement. These results are used to make decisions about and steer future promotions.
Traditionally, searching for a parking space is a manual searching process. A vehicle manufacturer uses vehicle-embedded sensors in their vehicles to detect a free parking lot while a vehicle passes this parking lot. The vehicle sends this information about a free parking lot to the digital twin. The *descriptive application* uses this information to create a virtual map that depicts the free parking lots. In case of that a vehicle requests this map, free parking lots are depicted. Therefore, the service is used to monitor free parking lots and send according to information to individual vehicles in case of a free slot.

11.3	Predictive Application
Context: Product status data depict the as is state of a product. Nevertheless, the data might be suitable to make predictions about the future events of product statuses.	
Solution Schematic:  	
Solution description: Use a *predictive application* to predict the product status or other events based on product status data. The prediction might be based on product status data from a single product or a group of products. The *prediction service* might use (un-)supervised machine learning to detect patterns in the data that lead to the events of interest. By including multiple products, patterns that are detected in data from one product might searched for in the data of another product to predict a similar event in the future.	
Advantages: + Pattern detection (between multiple products) + Detection of previously unknown facts + React to future events	**Disadvantages:** - High detail of sensor data required - High computation required
Relation to other patterns: Transfer Pipe, Machine Learning Engine, Data Lake, Prescriptive Application, Configuration Management	
Examples: Most often, a turbine of a plane is maintained in fixed intervals that depend on the turbines' hours of operation. To assure security, turbines are maintained even if the status of the turbine does not require maintenance activities. To reduce the maintenance activities while assuring safety in parallel, an airline uses a *predictive application* on the product planes' status data to predict failures. The airline benefits from the fact that there are many sensors embedded to each turbine and so they depict the turbines' status in detail.	

11.4	Prescriptive Application
Context: Product status data depict the as is state of a product. Nevertheless, the data might be suitable to make predictions about the future events of product statuses. Furthermore, these results might be used to steer product functionalities.	
Solution Schematic: 	
Solution description: Use a *prescriptive application* to analyze product status data (in real time). The result of the analysis is used to predict whether the product status is in a pre-defined status range. In case of derivations, the *prescriptive application* might react with feedback towards the product or any other action.	
Advantages: + Pattern detection (between multiple products) + Detection of previously unknown facts + Avoid future events	**Disadvantages:** - High detail of sensor data required - High computation required
Relation to other patterns: Transfer Pipe, Machine Learning Engine, Data Lake, Predictive Application, Configuration Management	
Examples:	

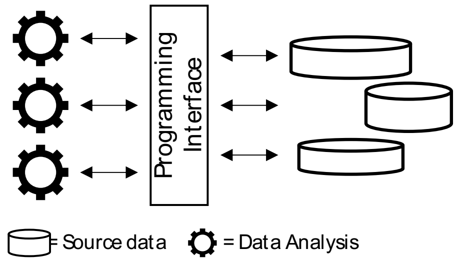
8.12 Database federator patterns

12.1	Data Query Language
Problem: See chapter 8.12	
Context: Besides other, software applications that analyze product status data need access to stored data. Therefore, they need a standardized method to do so.	
Solution Schematic: 	
Solution description: Use a *data query language* to systematically access product status data. Depending on the data source, such a language provides pre-defined methods to query and retrieve data. A *database query language* can be used by applications to automate the data retrieval or by users to manually query the data individually.	
Advantages: + Stored data can be queried and retrieved automatically + Stored data can be queried and retrieved manually and flexible + Fast realization	**Disadvantages:** - Complex querying - Data query language depends on the technical specification of the source data
Relation to other patterns: Data Lake, Visual Rules Engine	
Examples: Product status data as source data is stored in relational databases. The popular *data query language* SQL can be used to query and retrieve these data. SQL offers pre-defined methods that might be used. To perform certain results, multiple queries need to be combined.	

12.2 Programming Interface

Context:
Besides other, software applications that analyze product status data need access to stored data. Data might be stored in multiple different ways, which increases the complexity to access data. Furthermore, custom functionalities to query and retrieve stored data might be needed.

Solution Schematic:



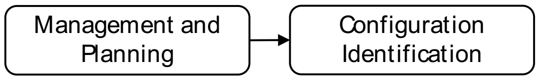
Solution description:
Use a *programming interface* to harmonize, how data is accessed. The *programming interface* defines methods that can be used to query and retrieve data. Thereby, new methods might be added, or existing methods changed, depending on the individual needs. Furthermore, the *programming interface* offers a technology-independent access to stored data.

Advantages:	**Disadvantages:**
+ Stored data can be queried and retrieved automatically + Stored data can be queried and retrieved manually and flexible + Hides complexity of heterogeneous storages + Flexible definition and change of methods to query and retrieve data	- Maintenance of the programming interface - Slow realization due to high effort for data definition

Relation to other patterns:
Status Tracker, Data Warehouse, Descriptive Application

Examples:
Product status data as source data is distributed in multiple databases. Each of the databases might use a different querying language. To reduce complexity, an application-programming interface (API) as a *programming interface* is used. The API translates the requests to access from the common format of the API to the individual format of the databases. Thereby, all databases can be queried with the same methods. Furthermore, the API reduces complexity as if might offer frequently used combinations of queries as a method.

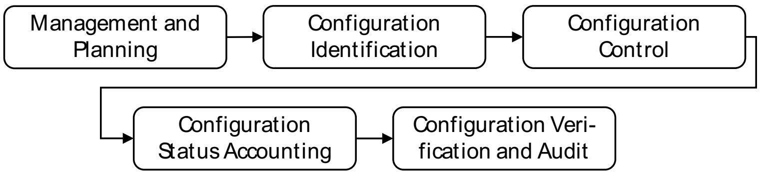
8.13 Twin configurator patterns

<table>
<tr><th>13.1</th><th>As is configuration</th></tr>
<tr><td colspan="2">Problem:
See chapter 8.13</td></tr>
<tr><td colspan="2">Context:
The product designer can foresee the demand for data gathering along the products' life in the product design phase already. This is the case when the designer has a precise application for the status data. Therefore, there is no need to change the configuration at a later point in time.</td></tr>
<tr><td colspan="2">Solution Schematic:

</td></tr>
<tr><td colspan="2">Solution description:
Use an as is configuration. Thereby, the data gathering and preprocessing on the product layer is defined in the design phase of the product. Therefore, the management and planning are performed prior to the products' manufacturing. Along the whole life of the product, the pre-defined product status data are gathered in pre-defined formats and intervals. The configuration cannot be changed along the life of the product but only identified.</td></tr>
<tr><td>Advantages:
+ Simple architecture if the applications to analyze product status data are known in the design phase</td><td>Disadvantages:
- No new product status data can be gathered after product manufacturing
- Existing configuration of product status data that is gathered cannot be changed after product manufacturing
- Gathering of product status data that is no longer needed cannot be stopped after product manufacturing</td></tr>
<tr><td colspan="2">Relation to other patterns:
Passive Wiretapper, As-Configured Digital Twin</td></tr>
<tr><td colspan="2">Examples:
A low-cost consumer product has embedded IoT technology that gathers data about the users' behavior. The data are gathered to comply with legal regulations. Thereby, the application is clear, and no further applications are planned to keep the costs low. Therefore, the as is configuration fulfills the demand and no changes of the re-configuration along the life is needed.</td></tr>
</table>

13.2	Configuration Management

Context:
The product designer cannot foresee during the design phase, which and how product status data should be gathered along the life of a product. Instead, new applications might be created, or existing applications might be changed or stopped.

Solution Schematic:



Solution description:
Use a *configuration management*. During the management and planning phase, new configurations for individual products are planned to meet new or changed demand for product status data. Before the changed configurations can be sent to the individual products, the configuration of the product is accessed to validate the technological feasibilities. During the configuration control, the compatibility of the new configuration with the product is checked. This includes technical constraints and other restrictions such as the consent. After the new configuration is sent to the individual products, the changed configuration is documented during the configuration status accounting and maintained during the configuration verification and audit.

Advantages:	**Disadvantages:**
+ New product status data can be gathered after product manufacturing + Existing configuration of product status data that is gathered cannot be changed after product manufacturing + Gathering of product status data that is no longer needed cannot be stopped after product manufacturing	

Relation to other patterns:
Decentral Agents, As-Maintained Digital Twin, Predictive Application, Prescriptive Application

Examples:
A manufacturer that uses connected machines in his manufacturing process aims to predict machine failures. Therefore, he already knows influential factors that will be analyzed to predict the failure. Nevertheless, the manufacturer might recognize along the machine's operation, that other data might also have an influence. Therefore, the manufacturer uses a *configuration management* so that he is flexible and might change the data gathering and / or preprocessing at any time along the machines' life.

6

SUMMARY, DISCUSSION AND CONCLUSION

Brandt, Niklas, University of Duisburg-Essen, Essen, Germany

Abstract

The objective of this thesis was to develop an artifact that systematically guides vehicle manufacturers to gather, store, manage and analyze vehicle sensor data along the individual vehicles' lives. Therefore, the central research problem was divided into separate research questions that were answered in the research essays of this thesis. This chapter reviews the individual essays and summarizes their main results. Furthermore, this chapter reflects the central artifact of this thesis, i.e., the pattern language, critically. Based on the reflection, it presents limitations and gives an outlook over future research possibility.

Keywords: *Summary, discussion, future research, limitations*

Table of contents

1 Summary of Key Findings

This thesis focuses on design recommendations for Vehicle Information Systems (VIS) that enable internet-connected vehicles to participate in the Internet of Vehicles (IoV) and share data about their status along their individual lives. Observations in the automotive industry showed that practitioners have already realized some IoV applications (Bilgeri et al., 2019, p. 191). Nevertheless, using a holistic and flexible VIS to gather, manage, store, and analyze data from individual vehicles throughout their lives is not state-of-the-art in the automotive industry. In addition, commercially offered VIS and existing models from researchers do not solve the problem holistically. Existing models lack the flexibility to integrate them into legacy information systems (IS) of the individual vehicles and the vehicle manufacturers (Kaiser et al. 2018, p. 8). Furthermore, most existing models do not acknowledge the particular context of the manufacturers and do not enable users to adapt some parts of the model. Instead, current models are often intended to be adopted without possibilities of adapting them to existing IS (Hribernik et al., 2017, p. 1410) (Huawei Technologies Co., 2019). To overcome these existing models' limitations and enable vehicle manufacturers to gather, manage, store, and analyze data from individual vehicles, this thesis developed an IoV pattern language for VIS. A pattern language is ideally suited to overcome the limitations of the existing models. It splits the overall problem, i.e., the VIS design, into smaller and independent sub-problems (Mears and Zhu, 2017, p. 3-4). It defines one to many patterns (DeHon et al., 2004, p. 7), i.e., best practices, for each sub-problem in a specific context (German and Cowan, 2000, p. 4). These patterns are identified from existing and recurring solutions in practice. Hence, it is known that the patterns have already solved similar problems in similar contexts (Voit et al., 2020, p. 330). To develop the pattern language, this thesis is guided by an overarching process derived from the design science research methodology from Peffers et al. (2007, p. 54) and the process model from Ahlemann and Gastl (2007, p. 89). Following this process, this thesis answers four research questions to address the overall research problem. Each research question answers different aspects of the research problem.

Research Question 1: Which PLM applications could benefit from information about individual product instances and what hinders organizations to realize them? What are the limitations of existing PLM IS that were designed to support the lifelong gathering and analysis of product instance data?

The focus of the first essay is to create transparency about the status of the Internet of Things (IoT) and to answer the first research question accordingly. Therefore, a systematic literature review unveiled barriers that hinder industrial companies to systematically gather, manage, store, and analyze sensor data from their products along their individual lives. These barriers include privacy & data security risks, the missing capabilities to gather, transfer and analyze the data, the existing IS silos of manufacturers and the heterogeneous product data. Furthermore, drivers, i.e., technological progress, other than IoT sources and more existing cyber physical systems were identified to drive the use of the IoT. Existing literature stated that the IoT could enable applications that were not able before, i.e., demand-based design, customer services, estimation of remaining lifetime and further increasing sustainability. By contextualizing these factors in a topic map, a theory for explaining was created. This can be considered as a class one theory, according to Gregors' (2006, p. 622-623) theory classification.

While analyzing the IoT barriers, drivers, and applications it became clear that researchers often propose new models to guide manufacturers overcoming the barriers and to enable the applications. Hence, the first essay also analyzed existing reference models. Most models that were analyzed do not cover the complete lifecycle of a product or they are intended for a specific application. Besides these specific models, there have been also some holistic models that are independent from any industry. Nevertheless, these few holistic models are not suited to overcome the barriers of the manufacturers holistically. The models were found to lack adaptability and integrability into existing legacy IS of manufacturers. Hence, the first essay showed a need for a holistic guidance for manufacturers to design future IS to manage IoT data.

Research Question 2: Which factors influence the decision of vehicle manufacturers to adopt the concept of the vehicle digital twin on a firm level?

Whenever it comes to innovations, multiple factors influence, whether and how one adopts these innovations. The technology-organization-environment (TOE) framework is one of the only frameworks for innovation adoption on a firm level (Tornatzky and Fleischer, 1990, p. 154). Focus of the second essay was to adapt the TOE to unveil the factors that influence the innovation decision of vehicle manufacturers to adopt the concept of the vehicle digital twin, i.e., gathering, managing, storing, and analyzing IoV data along individual vehicle lifecycles. Thereby, essay two answers research question two. Based on 30 interviews in the automotive industry, fifteen factors in the technology, organization and environment dimension were identified. Depending on the characteristic of each factor in their organizations, some automakers are more likely to adopt the concept of the vehicle digital twin while others are less likely to adopt the concept. Besides guiding the evaluation of the adoption likeliness of vehicle manufacturers to adopt the vehicle digital twin or not, the adapted TOE also extends the topic map of the first essay. It does so, as each factor of the TOE has a negative or positive influence on the adoption decision. Thereby, potentially negative factors can be seen as barriers and positive factors as drivers or applications. As a result, essay one and two together aggregate academic (literature review) and practitioners' (expert interviews) knowledge and define the demand and objectives of a future VIS.

Research Question 3: What are the elements of a pattern language that guides the design of an IS to gather, manage, store, and analyze sensor data from individual vehicles?

After the research was motivated in the first two research essays, the third essay answers research question three by focusing on the construction of an initial model. Therefore, academic models, commercially offered IS and the expert interviews from essay two were analyzed to identify the elements that existing IS for IoV sensor data gathering, managing, storing, and analyzing contain. Based on this aggregation of existing knowledge, common and unique elements of the IS were found. By discussing the need for each element, thirteen elements were identified and contextualized that an IS to gather, manage, store, and analyze IoV sensor data from individual vehicles along their lives' needs to consist of. This identification and contextualization of self-contained elements of an IS divides the overall VIS design problem into thirteen sub-problems. This initial model constitutes the structure of a pattern language, i.e., the structure of the central artifact of this thesis. Each of the elements is intended to be solved by one-to-many pattern, i.e., best practices, later. Hence, the structure of a pattern language initiates the iterative design and evaluation loops.

Research Question 4: Which are the patterns, i.e., best practices, to design an IS to gather, manage, store, and analyze sensor signal based status data from individual internet-connected vehicles?

Essay four focuses on the identification of patterns, i.e., best practices, to overcome the individual sub-problems that were defined in the pattern language structure of research essay three. Hence, essay fours answers the fourth research question. To identify the patterns, six design oriented focus groups were conducted according to the methodology of Tremblay et al. (2010, p. 602). In total, four exploratory focus groups were conducted with experts from industries that already gather data from IoT products along the products' lives. In the exploratory focus groups, experts from manufacturing equipment manufacturers, from cloud solution providers and IoT researchers were included. To evaluate the usefulness of the identified patterns for the automotive industry, experts from an automaker were included in two separate confirmatory focus groups. Besides the evaluation of the patterns' usefulness, the confirmatory focus groups unveiled known patterns in the automotive industry. In total, 38 patterns were identified for the elements of the pattern language structure. For each element, multiple patterns were identified and adapted to the automotive industry. Depending on the individual context and demand of an automaker, one of the patterns might be chosen for each element. Therefore, the pattern language offers users maximum flexibility to adapt these patterns or to develop own patterns that solve the sub-problems in their context best.

Research Problem: There is no suitable guidance for vehicle manufacturers to design a holistic information system to gather, manage, store, and analyze data from individual internet-connected vehicles along their lives.

By following the DSR process, this thesis systematically answers the research questions to address the overall research problem. The pattern language, which is the central artifact of this thesis, highlights that a pattern language is well suited to guide others in the design of an IS. By identifying existing solutions in practice, rather than developing new ones, the practical relevance of the patterns to solve the problem is already known. In the case of this thesis, the solutions are already known in other industries and therefore, just the adaptability to the automotive industry needed to be evaluated. Rather than just applying the patterns, researchers and organizations need to evaluate, whether one of the patterns solves their individual problem in their individual context. In case that no satisfactory patterns exist, new patterns can be developed and added to the existing pattern language.

2 Discussion

This dissertation contributes to both research and practice. To sum up, the overall contribution of this dissertation is the design science research artifact that guides vehicle manufacturers to design VIS for internet-connected vehicles. In the past, researchers and practitioners have already developed different kind of models to guide manufacturers in the design of such IoV VIS. Nevertheless, these models did not solve the problem in practice holistically (Brandt and Ahlemann, 2020, p. 12). Hence, this dissertation addresses an unsolved problem in the automotive industry. By developing a pattern language as an artifact type for the guidance of the design of an IS, this thesis addresses the existing problem with an innovative solution. The reason for the innovation character of the solution is that to the best of our knowledge, a pattern language was never used before to guide the design of a VIS or even an IoT IS. With the development of an innovative solution to an unsolved problem (Gregor and Hevner, 2013, p. 347), this dissertation conducts design as a science (Simon, 1996, p. 111).

Research essay one builds the foundation of this dissertation and primarily enables a sound foundation for the following research essays. Besides, the first research essay makes own contributions. It analyzed existing knowledge in the literature to develop a class one theory, which is a theory for explaining, according to Gregors' (2006, p. 622-623) classification of theories. This theory aggregates existing knowledge and contextualizes the findings, why organizations aim to gather IoT data and what hinders them. Hence, it supports others to understand the adoption of the IoT in organizations. In addition, the first research essay analyzed existing models that guide organizations to design IS to gather, manage, store, and analyze IoT data. Such an aggregation and analysis build the basis for designers of future IS. The first research essay creates transparency about existing and previous research achievements in the field of study.

Both practitioners and researchers observed already that holistically gathering, managing, storing, and analyzing IoV data from individual vehicles to realize the vehicle digital twin is not state of the art for automakers. Nevertheless, few systematically studied the reasons for that in the past. In this sense, the second research essay contributes to research and practice as it identifies factors that influence the decision of vehicle manufacturers to adopt the concept of the vehicle digital twin on a firm level. Therefore, the technology-organization-environment framework from Tornatzky and Fleischer (1990, p. 154) was adopted in the second research essay to conceptualize these factors. The framework guided the identification of those factors that influence the adoption decision of manufacturers. Based on the factors, the individual likeliness of manufacturers to adopt the concept of the vehicle digital twin can be determined. The other way around, manufacturers that have adopted the vehicle digital twin already can use the framework to identify measures, i.e., other factors, to gain more benefits from the individual existing vehicle digital twins.

In the past, researchers and practitioners developed multiple models to guide others in the design of an IoT IS for products in general and vehicles specifically. As an artifact of type model, the development of a pattern language was initiated in the third research essay. Pattern languages were already used in multiple domains like architecture (Alexander, 1977, p. 10), computer science (Gamma et al, 1997, p. 394) and even information systems (Sprenger and Mettler, 2016, p. 8). By using a pattern language to guide others in the design of an IS, research essay three contributes to the body of research as it defines a new artifact type in IS research for the guidance of an IS design. Prior, IS researchers mostly did not explicitly name their artifacts for similar problems a pattern language but a (reference) model. In addition to this methodological contribution, research essay three aggregates existing VIS and defines core problems that need to be solved by an IoV VIS. This contribution to practice specifies the demand for future IoV VIS.

The fourth research essay builds on the results of the third one and contributes to research and practice by defining patterns for the pattern language structure. For the documentation of the individual patterns, a template was developed. This general template is designed by analyzing existing pattern documentation templates and might be used by other researchers to document their patterns. Furthermore, the fourth research essay introduces a novel approach to identify and confirm patterns with the use of exploratory and confirmatory focus groups. Future researchers might take this as an example to follow up in a similar approach. Last, the fourth research essay finally addresses the research problem by systematically guiding researchers and practitioners to design IoV IS. Guiding the design with a pattern language addresses the limitations of previously existing models by proposing one to many solutions to the individual IS design sub-problems that solve the sub-problems in different contexts. By defining both the sub-problems and the solutions in a self-contained manner, the pattern language enables organizations to better integrate the solution into their existing IS landscape. Hence, research essay four eases the adoption compared to the existing models. In addition, the pattern language builds on knowledge from other industries that have already solved the problem in a similar context. Thereby, the suitability of the patterns to solve the sub-problems in a similar context is already proven.

3 Limitations and Outlook

The previous chapter discussed the contribution of this thesis. Nevertheless, neither any of the manuscripts nor the overall research are free of limitations. Besides other reasons, the limitations can be traced back to the general limitations of the chosen research methods and the focus that was chosen in this thesis to handle the research. This very last chapter summarizes the limitations. As the unveiled limitations offer further research opportunities, this chapter is a call to others to conduct further research in the field of the identified limitations.

The first research essay analyzes existing scientific literature to unveil the existing body of knowledge in a concept centric approach with a systematic literature review. One outcome is a topic map that depicts and contextualizes the IoT barriers, drivers and benefits from a data gathering perspective. Nevertheless, the propositions about the relations of the factors to each other are not empirically grounded but taken out of context of the analyzed academic literature. Therefore, the propositions about the factors' context need to be further investigated. Regarding to the second outcome, which is the analysis of existing models that guides the gathering, managing, storage and analysis of IoT data, the literature review focuses on existing academic literature only. Hence, existing commercially offered IS are not considered in the review.

The second research essay partially takes up the limitations of the first research essay by including experts' knowledge to investigate the factors that influence the decision of vehicle manufacturers to adopt the concept of the vehicle digital twin. These factors confirm, complement, or neglect the barriers, drivers, and applications from the literature review with a focus on the automotive industry. By focusing on the automotive industry, other industries that could make an additional contribution were not considered. In addition, experts from just two companies were interviewed for the study. As the second research essay focused on qualitative exposure of factors only, future research might validate the factors with a quantitative approach in multiple organizations and industries. With a quantitative approach, future research could also weight the importance of an influencing factor on the innovation decision and investigate the influence of the identified factors on each other.

The third research essay develops the initial model. The initial model was developed based on existing academic literature, commercially offered IoV IS from software vendors and expert interviews in the automotive industry. Hence, the initial model does not consider commercially offered IS and expert interviews from other industries than the automotive one. Furthermore, the interdependencies between the identified elements were not identified in existing models or IS. Rather, these interdependencies are build based on the description of possible IoV applications. Therefore, these interdependencies might be a topic of future research. Last, the research essay contributes to the body of knowledge by presenting a structure of a pattern language. Nevertheless, this structure is only breaking down the overall IS design problem into smaller sub-problems. By doing so, it describes the problems and the demand to a solution subsequently. Solutions by means of patterns are not part of the pattern language structure. This offers future researchers the opportunity to develop patterns, i.e. best practices, for the elements of the structure.

The fourth research essay addresses one limitation of the third research essay. It does so by developing one to many patterns for each element of the pattern language structure. The patterns were identified by conducting focus groups in different industries. Although experts from four different organizations were involved in the exploratory focus groups, experts from just one automaker were involved in the confirmatory focus groups. This limitation occurred due to limited access to other automakers and can be addressed in future research. Although a total number of 21 experts was involved in the exploratory and confirmatory focus groups, a quantitative approach could support the discovery and validation of more patterns. Last, the patterns were identified in diverse industries. As some patterns were mentioned in multiple focus groups with experts from different industries, the patterns, or a sub-group of them might be useful in other industries, too. Therefore, future research could investigate the usefulness of the pattern and the overall pattern language in other industries.

4 References

Ahlemann, F. and Gastl, H. (2007). „Process model for an Empirically Grounded Reference Model Construction.“ In: *Reference Modeling for Business Systems Analysis*, p. 77-97.

Alexander, C., Ishikawa, S. & Silverstein, M. (1977). „A Pattern Language. Town – Buildings - Construction.” Oxford University Press. New York.

Bilgeri, D., Fleisch, E., Gebauer, H. and Wortmann, F. (2019). „Driving Process Innovation with IoT Field Data.“ In: *MIS Quarterly Executive*, p. 191-207.

Brandt, N. and Ahlemann, F. (2020). "How Do You Drive? Analyzing Vehicle Sensordata In Product Lifecycle Management Systems." In: *Proceedings of the 28th European Conference on Information Systems (ECIS)*, p. 1-23.

DeHon, A., Adams, J., DeLorimier, M., Kapre, N., Matsuda, Y., Naeimi, H., Vanier M. and Wrighton M. (2004). “Design Patterns for Reconfigurable Computing.” In: *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, p. 13-23.

German, D. and Cowan D. (2000). “Towards a unified catalog of hypermedia design patterns.” In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, p. 1-8.

Gamma, E., Helm, R., Ralph, E. and Vlissides, J. (1997). ”Design Patterns Elements of Reusable Object-Oriented Software.” Prentice Hall; 1st ed.

Gregor, S. (2006). “The Nature of Theory in Information Systems.” In: *MIS Quarterly* (30:3): p. 611-642.

Gregor, S. & Hevner, A. (2013). “Positioning and Presenting Design Science Research for Maximum Impact.” In: *MIS Quarterly*, Volume 37, Issue 2, p. 337-355.

Hribernik, K., Franke, M., Klein, P., Thoben, K. & Coscia, E. (2017). „Towards a platform for integrating product usage information into innovative product-service design.” In: *International Conference on Engineering, Technology and Innovation (ICE/ITMC)*: p. 1407-1413.

Huawei Technologies Co. (2019). “IoT, Driving Verticals to Digitization.” URL: https://www.huawei.com/minisite/iot/en/vehicle-networking.html (visited on 20 July 2021).

Kaiser, C., Stocker, A., Festl, A., Lechner, G. and Fellmann, M. (2018). „A Research Agenda for Vehicle Information Systems.“ In: *Proceedings of the Twenty-Sixth European Conference on Information Systems*, p. 1-12.

Mears, B. and Zhu, J. (2017). “Design Patterns for Silent Player Characters in Narrative-Driven Games.” *Proceedings of the 12th International Conference on the Foundations of Digital Games*, p. 1-4.

Peffers, K., Tuunanen, T., Rothenberger, M. and Chatterjee, S. (2007). “A design science research methodology for information systems research.” In: *Journal of Management Information Systems*, 24(3), p. 45-77.

Simon, H. (1996). „The Sciences of the Artificial”. *MIT Press*, Cambridge, p. 1-231.

Sprenger, M. and Mettler, T. (2016). “On the utility of e-health business model design patterns.” In: *Twenty-Fourth European Conference on Information Systems*, p. 1-16.

Tornatzky, L.G. and Fleischer, M. (1990). “The Processes of Technological Innovation.” Lexington Books, Lexington.

Tremblay, M., Hevner, A., Berndt, D. and Chatterjee, S. (2010). „The Use of Focus Groups in Design Science Research." In: *Communications of the Association for Information Systems*, Volume 26, p. 599-618.

Voit, T., Schneider, A. and Kriegbaum, M. (2020). „Towards an Empirically Based Gamification Pattern Language Using Machine Learning Techniques." In: *IEEE 32nd Conference on Software Engineering Education and Training*, p. 1-4.

Appendices

Appendix A. Essay on Pattern Languages.

A

DELIMITATION OF PATTERN LANGUAGES AND REFERENCE MODELS AS DESIGN ARTIFACTS

Brandt, Niklas, University of Duisburg-Essen, Essen, Germany

Abstract

Reference model and pattern language are two common notions for design science artifacts that guide others in the design of a good solution to their common problem. With a systematic literature review, I study, whether the characteristics of the differently called artifacts differ. In order to do so, I define a classification scheme that summarizes all characteristics that such an artifact might have. Based on the application of the classification scheme to the artifacts in the literature pool I find that there are no characteristics that clearly differentiate all existing artifacts that are called reference model from the ones that are called pattern language. Nevertheless, there are characteristics that tend to differentiate the differently called artifacts. As a result, I propose to use the notions reference model and pattern language for different artifact types depending on the artifacts' characteristics. The strict separation and the definition of two artifact types offers future researchers the chance to choose that artifact type that fits their goals best and to clearly communicate their artifacts. As a result, this systematic literature review guides future design science researchers in their design and documentation process.

Keywords: *Pattern Language, Design Patterns, Reference Model, Research Artifact, Design Science Artifact*

This essay was already graded in a seminar of the University of Duisburg-Essen. Nevertheless, it is attached to this dissertation to ease the access to it due to relevance for this dissertation. It will not be graded again in this dissertation.

Table of contents

List of Figures

List of Tables

1 Introduction

Design oriented researchers often design solutions to solve problems that are either unsolved or can potentially be solved more efficiently (Hevner, March and Park, 2004, p. 81). More specifically, researchers define abstract solutions that can be reused and adapted to a class of similar problems (Schlieter, Juhrisch and Niggemann, 2010, p. 413). Through abstraction of the problem and solution from one instantiation to a class of instances, researchers increase the value of their contribution and build the basis for theorizing (Gregor and Hevner, 2013, p. 341). In the IS domain, researchers often follow the design science research (DSR) approach to develop reusable solution to a recurring problem. With DSR, researchers design solutions of type constructs, models, methods, and instantiations (March and Smith, 1995, p. 251). IS researchers often use the term reference model for their DSR artifact of type model (Junginger, 2009, p. 472). Such reference models contain knowledge that can be used to guide the design of a solution to a recurring problem in multiple contexts (Guerra and Nakagawa, 2015, p. 2). In other domains, researchers call their abstract solutions to common problems differently. A popular term in architecture (Alexander, 1977, p. 10) and computer science (Gamma et al, 1997, p. 394) is pattern language. Pattern languages consist of one to multiple design patterns, while each pattern is a part of the solution to the overall problem (Guerres and Nakagawa, 2015, p. 3). A design pattern is defined as a self-contained best practice that has a relation to other patterns. Thereby, a pattern can be compared to a word of a natural language that is brought into a logical context with other words to build a language.

Although reference models and pattern languages have their origin in different research domains, they seem to share the same idea of guiding the design of a solution to a recurring problem in different contexts. This raises the question whether there is a difference between artifacts that use the two different notions. There is only limited research available about the similarities and differences (Winter et al., 2009, p. 468 - 474; Guerra and Nakagawa, 2015, p. 1 - 12). Matthes and Ernst raise the question, whether one can translate reference models into pattern languages and vice versa due to the similarities (2009, p. 470). Vom Brocke (2009, p. 470) goes even further and denies the need for a translation as he implies that pattern languages and reference models might be the same and both are just called differently. Fettke and Loos (2009, p. 473) share this thought and propose to not differentiate between the two notions. They argue that the initial ideas of both concepts were different, but these differences between reference models and pattern languages have been gone. Also, according to the DSR classification, both results are of type method and are classified as a level 2 research result (Gregor and Hevner, 2013, p. 342). This uncertainty raises the question, whether there is a fundamental difference between the artifacts that are called reference models and the artifacts that are called pattern languages. Guerra and Nakagawa (2015, p. 1) found that little research explores the relation and differences between the two types. Therefore, I define the research question as follows:

What are the similarities and differences between the notions of a reference model and a pattern language?

To answer the research questions at hand, I conduct a systematic literature review. Such a review of existing literature is promising, as reference models and pattern languages are mature concepts, and much literature already exists. With a review, knowledge about the origin and practical application of the two concepts that is separated in the literature of different domains can be synthesized in a concept centric approach (Webster and Watson, 2002, p. 14 - 16). Thereby, I aim to re-conceptualize the knowledge that existing artifacts contain (Torraco, 2005, 357) and pave the way for more precise and harmonized definitions of design-oriented research results over domain borders. Based on the analysis of relevant literature I develop a theoretical framework that summarizes similarities and differences of reference models and pattern languages.

The paper at hand is structured as follows. After this motivational chapter I clarify the underlying concepts in chapter 2. The methodology of this systematic literature review is explained in chapter 3, before the results are aggregated in chapter 4. Chapter 5 discusses the results, and the paper concludes with a summary in chapter 6.

2 Theoretical Background

2.1 Design of an Artificial as a Science

Natural science is the study of natural phenomena (March and Smith, 1995, p. 253). Design science on the other hand creates and evaluates artificial artifacts (Hevner et al., 2004, p. 77) that support the user to reach human goals under the constraints of the natural phenomena (March and Smith, 1995, p. 254). Artificial artifacts change regularly with the goals that they aim at, which is opposite to natural laws that stay constant over time (Simon, 1996, p. 3). The demand for regular change and the fact that our lives are more artificial than natural (Elhacham et al., p. 2020) shows the need for a systematic study of the artificial (Simon, 1996, p. 2). Walls et al. (1992, p. 38) argues that the study and the design of the artificial can be a science, as the designer directly or indirectly applies or tests scientific theory. It is important to note that one must differentiate design science from routine design (Hevner et al., 2004, p. 81). In routine design, both the application domain and the solution maturity are high, and the scientific theories have already been applied in a previous design. In design science on the other hand are the application domain and the solution maturity or either one low (Gregor and Hevner, 2013, p. 345). In addition, scientific artifacts have some degree of abstraction to solve classes of problems (Kuechler and Vaishnavi, 2012, p. 396), while routine design often addresses single and specific problems. Furthermore, design science might be a science as natural science is often not well understood or lacks behind constructed artifacts in time. Then, the justification of an artifact itself explains the natural phenomena and thereby, design science contributes natural science (March and Smith, 1995, p. 254). Design science is not exclusively conducted in any domain, rather it is conducted in engineering (Simon, 1996, p. 111), architecture (Alexander, 1977, p. 10), information system (IS; March and Smith, 1995, p. 254) and other research domains. In general, design science researchers aim to build and evaluate artifacts that solve unsolved problems in practice (Hevner et al., 2004, p. 78) by transforming an as-is state into the preferred state (March and Storey, 2008, p. 725). The design of artifacts in design science is achieved in build and evaluate activities, which are parallel to the discover and justify activities in natural science (March and Smith, 1995, p. 258).

2.2 Reference Models and Pattern Languages

Design science researchers create and evaluate artificial artifact, i.e., solutions to real world problems. March and Storey (2008, p. 726) created a broadly accepted classification scheme for such DSR artifacts. According to that, DSR artifacts might constitute constructs, models, methods, or instantiations. The term pattern language is a popular term for such artifacts. Alexander et al. (1977, p. 10) were the first who used the term pattern language for their well acknowledged artifact. Their artifact consists of 253 patterns, i.e., best practices, for the construction of buildings and towns (Alexander et al. 1977, p. 11). Each pattern summarizes the solution to a sub-problem of the overall problem to design a good town. Alexander et al. (1977, p. 11) calls the collection of their 253 patterns a pattern language, as the patterns might be modified and combined in numerous ways. This is like vocabularies that are combined and modified to the individual context of a sentence and create a language. After the origin of pattern languages in the domain of architecture, Beck and Cunningham (1987, p. 1) adopted the artifact type to the domain of computer science. Building on this, the gang of four (Gamma et al. 1997, p. 93) developed their well acknowledged creational, structural, and behavioral patterns to develop object-oriented software. Ever since, design patterns play an important role in computer science research (Hughes and Duncan, 2012, p. 30; Heer and Agrawala, 2006, p. 859), while researchers in other domains like IS research (Sprenger and Mettler, 2016, p. 8) partially adopted the artifact type as well. Besides pattern languages, other researchers commonly call their DSR artifacts as reference models. Like pattern languages, the term reference model is often used for artifacts that are used to guide practitioners in the design of solutions to their problems (Legner et al., 2020, p. 739). This term is especially popular in IS research. Nevertheless, also researchers in other domains such as supply chain management call their well acknowledged research results as reference models (Stewart, 1997, p. 65). Other than for pattern languages, the origin of the reference model term for an artifact type is unclear (Thomas, 2006, p. 486). To summarize, both terms pattern language and reference model are used by researchers for artifacts that have a similar purpose - guiding others to design and evaluate good solutions to common problems in practice (Lewis et al., 2002, p. 4; Zimmermann, 1980, p. 431). Furthermore, both terms describe abstract artifacts that can be applied to multiple instances of a problem. Thus, the artifacts summarize the core of a solution that can be reused in multiple contexts. As there are no commonly accepted definitions for both terms (Thomas, 2006, p. 484), the delimitation of both artifact types, if any, is not clear. This leads to the question, whether both terms are different terms for the same concept.

3 Research Methodology

3.1 Literature Review

To answer the research question, I conduct a systematic literature review. Therefore, I perform a keyword search in relevant journals and proceedings of leading conferences as proposed by Webster and Watson (2002, p. 16). My research question deals with the origin and use of the notions pattern language and reference model. While the term pattern language has its' origin in architecture and is popular in computer science (Guerra and Nakagawa, 2015, p. 3), the term reference model is characterized by IS researchers. Therefore, I include the ACM digital library (www.dl.acm.org) and the IEEE Xplore digital library (www.ieeexplore.ieee.org) due to their contributions in computer science and IS. Furthermore, I include the AIS eLibrary (www.aisel.aisnet.org) due to many IS sources. In order to acknowledge contributions from other domains, including architecture, I perform a back- and forward search. I query the databases with two categories of key words. In the first category I use the terms "reference model", "pattern language" and "design patterns" that I link with a logic OR. I purposefully only included the plural term "design patterns" in the systematic search, as I am studying complete solutions and single patterns constitute only one part of a solution (Guerres and Nakagawa, 2015, p. 3). I search for the terms of the first category only in the titles to find publications that have a clear focus to pattern languages or reference models. Furthermore, I link the first category with the second category with a logical AND. The second category links the strings "develop", "identify" and "towards" with a logical OR and search for these strings in title, abstracts and keywords. I include the second category to reduce the literature pool to those publications that develop either a reference models or a pattern language. I expect that these publications contain the most valuable information to understand the origin and practical application of the two artifacts and thereby, those publications are best suited to answer our research questions. I only include peer-reviewed publications that are written in German or English language.

After removing exact duplicates, I identified a total number of 267 publications by performing the literature search according to our search strategy. After reading the titles, abstracts and partially the results, I removed 186 papers so that 81 papers remained in our literature pool. Thereby, I excluded publications that analyze existing artifacts, propose methods to instantiate artifacts and introduce tools to manage artifacts. Instead, I only include publications that develop an artifact, as these paper contain holistic information about the artifacts that are needed to analyze the artifacts' notions in detail. By conducting a backward search I included an additional number of six publications, as these publications were mentioned to be of major relevance (Alexander, 1977; ISO Standard, 1994; Stewart, 1997; Gamma et al., 1997; Hohpe et al., 2002; ISACA, 2019). In total, our literature analysis includes 87 publications. Out of these, 48 name the artifacts design patterns / pattern languages and 39 name the artifacts reference models. In order to simplify communication, I will call both concepts of design patterns and pattern languages as design patterns in the rest of the paper.

3.2 Classification scheme

I analyze the similarities and differences of differently called research artifacts. Therefore, I need a classification scheme with categories that are feasible to describe the similarities and differences (Fettke and Loos, 2003, p. 43 - 49). As there is only limited research on the similarities and differences of both artifact types available (Winter et al., 2009, p. 468 - 474), I see the need to develop a new framework that considers existing frameworks as a basis.

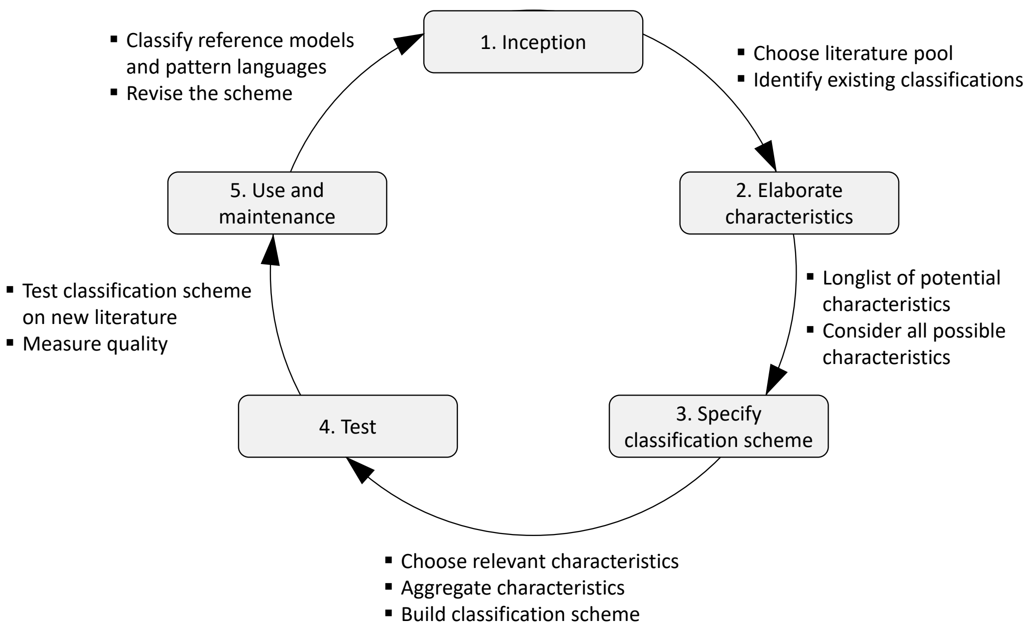


Figure 3-1 Process to derive the classification scheme (Fettke and Loos, 2003, p. 43 - 49)

I develop a classification scheme according to the process model from Fettke and Loos (2003, p. 43-49). Figure 3-1 depicts the five steps that I took to derive the classification scheme (Fettke and Loos, 2003, p. 43 - 49). I start with an identification of relevant literature according to the paper selection process in chapter 3.1. Therefore, I start with the two studies that already researched the similarities and differences of artifacts that are called design patterns and reference models / architectures (Winter et al., 2009, p. 468 - 474; Guerres and Nakagawa, 2015, p. 4). By applying forward search in these two publications I include two more publications in the first step. Both for patterns and reference models, I identify the five oldest and most recent relevant publications in the literature pool. I explicitly considered the oldest and most recent publications as I aim to consider changes of the artifact types over time in the classification scheme. Based on these 20 sources in addition to the four sources that already contain a classification scheme, I create a longlist of 73 potential categories. After selecting relevant categories and aggregating similar ones, the final classification scheme consists of 11 categories and 25 characteristics.

Category	Characteristic		
Claim	Artifact for explaining	Artifact for analyzing	Artifact for designing
Abstraction	Strategic		Operative
Granularity	System		Aspect of a system
Domain	Technology	Organization	Methodology
Methodology	Problem experts	Literature	Theory
Language	Modeling language		Template
Design Process	Recurring design	Novel design	Theoretical foundation
Adaption	Single solution		Alternative solutions
Adoption	Independent parts	Dependent parts	Fixed system

Table 3-1 Classification scheme to analyze the reference models and design patterns

Our classification scheme to analyze the similarities and differences of design oriented artifacts is depicted in Table 3-1 The first category analyzes the claim. This category describes the degree that the artifact claims to scientifically create knowledge (Fettke and Loos, 2009, p. 473). Artifacts for explaining substantially describe a domain (Frank, 2006, p. 118) and explain the addressed topic with its properties (Wang et al., 2003, p. 1; Hall and Rapanotti, 2003, p. 1). Thereby, such an artifact provides basic knowledge that is needed for the design of a solution (Legner et al., 2020, p. 739). Artifacts for analyzing evaluate existing solutions for their similarity to a proposed design (Romero et al., 2019, p. 230). Hence, existing solutions are analyzed for their accordance with design knowledge (Hall and Rapanotti, 2003, p. 1). Last, artifacts for designing summarize useful knowledge to develop a solution (Fettke and Loos, 2009, p. 473). Such artifacts act as blueprints (vom Brocke, 2009, p. 470-471) for multiple solutions to similar problems (Frank, 2006, p. 123). The second category analyzes the artifacts' level of abstraction (Fettke and Loos, 2009, p. 473). Strategic artifacts address strategically important problems and are rather abstract with no recommendation of specific decisions (vom Brocke, 2009, p. 471). Rather they summarize knowledge and indicate important factors that need to be considered (Legner et al., 2020, p. 739). Contrary, operative artifacts summarize precise recommendations (Noble, 1998, p. 3) that might be directly applied in practice without much elaboration (Lewis et al., 2002, p. 1). The addressee of such an artifact are the designers that the artifact guides (Beck et al., 1996, p. 103). With the category of granularity I analyze the extent of the problem that is solved by artifact. Artifacts with the characteristic system consider multiple (sub-)systems or system-overarching problems (Zimmermann, 1980, p. 425). These artifacts address complete systems or major parts of it (Fettke and

Loos, 2009, p. 473). Other artifacts with the characteristic aspect of a system focus on specific parts of a system that contribute to an overall system (Baudry et al., 2001, p. 325). Due to their focus on specific aspects, these artifacts are closer to the implementation (Beck et al., 1996, p. 107). The fourth category, which is the artifact domain, relates to the problem that is addressed by the artifact. Technical artifacts propose technical solutions to given problems (Farooq et al., 2008, p. 123). Although these artifacts contain technical descriptions, these artifacts can be formulated independent from a specific technological implementation (Lewis et al., 2002, p. 1). Organizational artifacts on the other hand consider organizational decisions and structures (Voit et al., 2020, p. 329). These artifacts can be multidisciplinary and can include, to a certain extent, technical aspects. A further characteristic is methodology. Methodological artifacts focus on guidelines for cross-domain processes. The fifth category methodology analyzes the design process of the artifact and the underlying knowledge that it builds on. The characteristic problem experts is applicable, when domain experts are directly involved in the design process of the artifact (Beck et al., 1996, p. 110). When such an artifact is developed together with the experts, the demand and requirements can be specified and the validity of the artifact discussed during the design process (Meurisch et al., 2017, p. 149). Contrary to this, the characteristic literature contains artifacts that only indirectly acknowledge experts knowledge by analyzing existing artifacts and their documentations as well as observations (Lewis et al., 2002, p. 1). The last characteristic of this category is theoretical foundation, which means that an artifact is mainly based on theories. Within the category language I analyze the documentation of the artifact. Artifact with the characteristic modeling language use a (standardized) modeling language to document the results (Fettke and Loos, 2009, p. 473). Other artifacts with the characteristic template use a self-designed template for documenting the results (Beck et al., 1996, p. 103). The process to derive the artifacts is analyzed in the category design process. In a recurring design, the artifact documents existing designs (Antonio et al., 1998, p. 2) that proofed to work in a certain context (Patil et al., 2018, p. 726). Hence, the process of identification and abstraction of the existing solutions is the main contribution of these artifacts (Beck et al., 1996, p. 112). Novel artifacts on the other hand do not build on existing designs. Rather, the designer creates a new artifact that was not necessarily known before (Meurisch et al., 2017, p. 149). The novel aspect can either be valid for (parts of the) artifact or the context of the parts (Romero et al., 2019, p. 229). Within the adaption category I analyze how the abstract artifact can be adapted to the context of the real-world instantiation. Artifacts with the characteristic single solution propose one solution (Noble, 1998, p. 3). Such a solution needs to be modified by the user to fit the context of the instantiation. Another way of dealing with context variety is followed by artifacts with the characteristic of alternative solutions (Lewis et al., 2002, p. 2). These artifacts propose multiple solutions for each part of the solution as implementation variants in different contexts (Wang et al., 2018, p. 379). Thereby, the adaption problem becomes to a selection problem (Noble, 1998, p. 2). With the category of adoption I analyze how the artifact and its parts can be adopted by practicioners. Some artifacts are split into independent parts. These parts of the artifact can be self-contained and adopted independently from each other (Gamma et al., 1994, p. 202). Other artifacts might also consist of self-contained parts, but the adoption requires the adoption of a part in the context of other parts (Alexander, 1977, p. 10). Last, artifacts with the characteristic of fixed system are complex systems with interwoven structures (Wang et al., 2018, p. 379). The main contribution of these artifacts is the context of the parts. These artifacts are intended to be adopted as a whole.

4 Paper Analysis

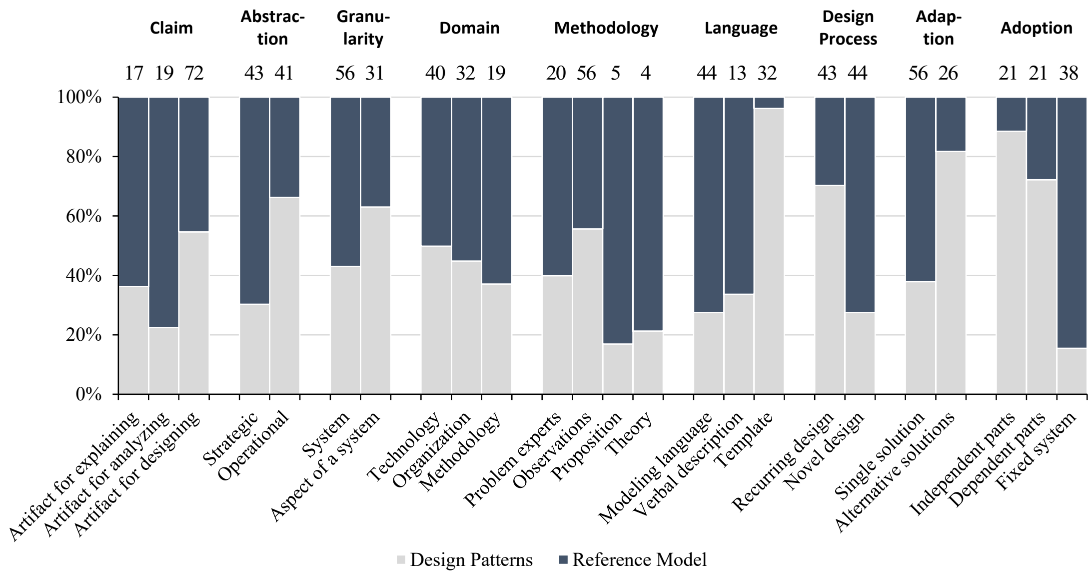


Figure 4-1 Distribution of the differently called artifacts to the characteristics of the classification scheme after normalization

Figure 4-1 shows the quantitative distribution of the papers to the characteristics of the classification scheme. Each paper of the literature pool might belong to zero, one or multiple characteristics per category. In the diagram I removed the influence of unequal number of papers per artifact type by normalizing the results. In the following analysis I investigate the differences between the artifacts that researchers call design patterns / pattern languages and reference models.

I investigate the intended use of the artifacts in the category *claim*. In total there are three purposes that artifacts claim to have, which are explaining or analyzing an existing solution or designing a new solution. I found that most artifacts that are called design patterns intend to guide developers to design a solution to a problem in a given context (Lewis et al., 2002, p. 4; Detweiler and Hindriks, 2012, p. 912). Patterns recommend specific solutions (Nadschläger and Küng, 2017, p. 1) and how to implement them (Stanoevska-Slabeva, 2003, p. 1). Moreover, they often intend to guide designers with no or rather little domain knowledge to design a good solution (Voit et al., 2020, p. 329; Bauer and Baumgartner, 2011, p. 2). Similar, most artifacts that authors call reference models act as guidelines to help designers creating good solutions to given problems (Farooq et al., 2008, p. 123; Hönigsberg et al., 2019, p. 214). Such artifacts build the foundation for designers to create a solution in their context (Jay Modi et al., 2006, p. 2). Unlike artifacts that are called design patterns, reference models do not explicitly emphasize the guidance of novel designers. Besides the design purpose, few artifacts that are called design patterns aim to explain or analyze existing solutions (Miguel Rosado da Cruz, 2014, p. 1; Moreira and Paiva, 2009, p. 2). I found that many of those artifacts that have a purpose to explain or analyze something have that purpose as a side-effect in addition to the purpose of guiding the design (Kartseva et al., 2006, p. 1; de Vreede et al., 2020, p. 685-686). Only a few design patterns in our literature pool have a focus to analyze or explain an existing solution (Bloom et al., 2018, p. 5; Miguel Rosado da Cruz, 2014, p. 1). Artifacts that the researchers call reference models on the other hand tend to have a stronger focus on the claim to explain or analyze something (Wang et al., 2003, p. 1; Meurisch et al., 2017, p. 149). Many of these artifacts also have the purpose to explain or analyze something as a side effect (Mears and Zhu, 2017, p. 1). Unlike design patterns, many artifacts that are called reference models also claim to explain or analyze an existing solution as their main and only contribution (Pousttchi and Hufenbach, 2011, p. 299; Han et al., 2016, p. 1).

In the second category *abstraction* I analyze the artifacts' degree of abstraction. Artifacts with a high level of abstraction and a claim to guide the design of a solution help designers to make strategic design decisions (Giebler et al., 2020, p. 62-64). Thereby, such artifacts do not guide the designers in their implementation decisions directly, but in the decision of the overall system (Zimmermann, 1980, p. 431). The distribution of the artifact types in this category shows that artifacts that are called reference models tend to be more abstract, although I also found many artifacts that use the notion of design patterns and have a high degree of abstraction. Nevertheless, researchers are more likely to choose the notion design patterns when they give operational design recommendations for direct implementations (Lewis et al., 2002, p. 4; Lindberg et al., 2014, p. 51-54). Artifacts that have a high level of abstraction and a claim to explain or analyze are mostly called reference models. Often, these artifacts analyze underlying concepts that are valid for multiple instances or even generally valid (Hall and Rapanotti, 2003, p. 1; Gamero et al., 2019, p. 60). Thereby, these artifacts aim to create scientific knowledge (Fettke and Loos, 2009, p. 473). There are only few abstract artifacts that use the notion design pattern and claim to explain or analyze something (Bloom et al., 2018, p. 5). Moreover, such artifacts for explaining or analyzing are more likely to explain a certain operational implementation of a solution rather than the underlying concepts (Moreira and Paiva, 2009, p. 4-7).

As a result from analyzing the *granularity* of the artifacts I identified that artifacts that use the notion reference model are more likely to address holistic systems than artifacts that use the notion design patterns. By addressing systems instead of certain aspects of it, the artifacts include the interaction of multiple different stakeholders (Tom and Sitte, 2006, p. 33) and different sub-systems (Cespedes, 2012, p. 3). A prominent measure of many artifacts that are called reference models to cope with their holistic nature and the variety of topics is the use of multiple layers or views (Zimmermann, 1980, p. 425; Otto and Ofner, 2010, p. 6). Such an approach to use layers and views is popular for artifacts that are called reference models but not for those that are called design patterns (Hughes and Duncan, 2012, p. 29; van Turnhout et al., 2014, p. 364). Instead, many of the artifacts that use the notion design patterns and which address holistic systems use a pre-defined perspectives and describe the system with patterns that are on the same level of abstraction (Cremonesi et al., 2015, p. 69; Isaku and Iba, 2009, p. 5). When it comes to operational artifacts, the differently called artifacts are rather similar. Both notions are used with a limited focus on a certain aspect of a system and describe the solution to the given problem on the same level of abstraction without considering sub-systems (Wu et al., 2016, p. 262; Dormann and Neuvians, 2012, p. 1).

The analysis of the artifacts' *domain* shows no big differences in the distributions of the differently called artifacts in the three characteristics. Due to the fact that some artifacts that are called reference models are more likely to address multiple aspects of a system, some of those artifacts also belong to multiple domains (Schrödl et al., 2011, p. 7). Thereby, they have a higher share in each characteristic of this category. In general, the high number of artifacts that deal with technological topics can´t be interpreted as general findings. Rather, this might be influenced by the choice of the databases.

The far most common *methodology* to collect data for the two differently called artifacts is the observation of an existing solution. This observations includes literature analysis (Hönigsberg et al., 2019, p. 214; Moreira and Paiva, 2009, p. 3) and an analysis of existing implementations (Stanoevska-Slabeva, 2003, p. 2; Voit et al., 2020, p. 330). Besides, authors sometimes explicitly involve practitioners in the design process of the artifact. The designers of such artifacts often state that experts should be involved in order to ensure that the artifact is of help for the experts in the future (Arney et al., 2007, p. 24; Köppe et al., 2017, p. 3-4). Only few publications that use the notion design patterns or reference model are assigned to the characteristics proposition and theory. This clarifies that these methodologies do not play an important role for the DSR artifact development in general.

We analyze the language that is used to depict and document the artifact in the category *language*. Most artifacts that are called patterns use a template with standardized categories for documentation. The use of such a template is unique to patterns and only one artifact that is called reference model uses such a template (Lings et al., 2007, p. 4-8). Artifacts that are called reference models use a standardized or self-defined modeling languages instead, which is rare for artifacts that are called patterns. Independent from the artifact type, some artifacts are documented with a verbal description alone.

The differently called artifacts seem to differ in the *design process*. Multiple authors that use the term design patterns stress that patterns are recurring solutions that appear over and over again in existing solutions (Fortino et al., 2008, p. 416). An existing solution is considered as a pattern when it proofed to be a good solution to the problem (Sprenger and Mettler, 2016, p. 9). Hence, the artifact development process is more likely an identification process of good solutions. Thus, the only active design step is the abstraction of existing implementations. The process of the pattern identification ranges from manual discovery of existing solutions (Swacha and Muszyńska, 2016, p. 1) up to automated searches for patterns by applying machine learning (Voit et al., 2020, p. 330). Even some authors who develop design patterns from scratch as novel solutions instead of identifying existing solutions affirm in the documentation that design patterns are normally identified and not actively designed (Moreira and Paiva, 2009, p. 3). Besides, there are artifacts that are called design patterns that address unsolved problems, which could not be developed by identifying existing solutions (Detweiler and Hindriks, 2012, p. 911; Lindberg et al., 2014, p. 47). Some authors try to cope with missing solutions in the domain that they address by identifying patterns in other domains that have already solved the problem (O'Callaghan, 1999, p. 22; Kartseva et al., 2006, p. 2). As these authors do not identify recurring designs for the problem they aim to solve in their domain, I consider these solutions as novel designs. Similar to most design patterns, there are artifacts that are called as reference models that also follow the process of identification to derive the artifact. Nevertheless, only a few explicitly state that the analysis of existing solutions is their sole data source (Fischbach et al., 2013, p. 2; Wu et al., 2016, p. 250). Rather, it is more common that artifacts that use the notions reference model are actively designed. Therefore, authors gather recurring requirements instead of recurring solutions (Schmidt et al., 2011, p. 360; Reiter et al., 2013, p. 10). These requirements are used as a basis to identify the requirements and to actively design the artifact as a solution (Farooq et al., 2008, p. 126). Due to the active design process that is common in for artifacts that are called reference models, it is not unlikely that such an artifact solves an unsolved problem (Bretschneider and Hartman, 2018, p. 494; Jay Modi et al., 2006, p. 1-2).

As both notions are used for abstract solutions that solve multiple instances of a problem, the solution needs to be *adapted* to the varying context of each instantiations (Guerres and Nakagawa, 2015, p. 5; Zaki and Forbrig, 2012, p. 704). Measures to cope with varying context of the implementation is a high level of abstraction and the proposition of context-dependent and alternative solutions (Lings et al., 2007, p. 1). I found that half of the artifacts of our literature pool that use the notion design patterns propose alternative solutions (Bloom et al., 2018, p. 6; Liszio and Masuch, 2016, p. 3), while only few artifacts that use the notion reference model do so (Lings et al., 2007, p. 1). This is in line with our analysis in the category *abstraction*, where we found that most artifacts that use the term reference model use a higher level of abstraction than the artifacts that use the term design patterns (Giebler et al., 2020, p. 62-64). Alternative solutions for a problem might address the same part of the solution with different approaches or perspectives (German and Cowan, 2000, p. 4). The applicant of the artifact can then choose the solution that is best suited to solve the problem in the context of other predefined boundaries (Nguyn et al., 2015, p. 248) and thus, easily configure the solution (Stewart, 1997, p. 65; Bretschneider and Hartman, 2018, p. 8). As a result, the adaption process of artifacts with alternative solutions becomes an identification process of the solutions that are best suited. On the other hand, half of the artifacts that are called design patterns and almost all artifacts that are called reference models propose a single solution (Lewis et al., 2002, p. 4; Hall and Rapanotti, 2003, p. 1). Authors of such artifacts shift the task of adaption to the applicant (Jay Modi et al., 2006, p. 4). Thereby, they do not limit the creativity to a subset of possible instantiation variants. Nevertheless, good solutions that differ strongly can hardly be abstracted to a single solution.

The *adoption* of an artifact requires its implementation/usage. Depending on the artifacts' characteristics and the problem in its' context, the user can either adopt the artifact as a whole or parts of it (independently). Our evaluation shows, that most artifacts that are called design patterns are designed in such a way that they can be adopted partially (Pongpradit and Prompoon, 2016, p. 5; Noble, 1998, p. 6), while the majority of the artifacts that are called reference models are fixed systems that should be adopted as a whole (Stewart, 1997, p. 65; Boillat and Legner, 2015, p. 114). Artifacts that consist of independent parts structure the overall problem into independent sub-problems (Gamma et al., 1997, p. 10; Lindberg and de Troyer, 2020, p. 4-5). Thus, such artifacts are collections of solutions for general recurring problems. Dependent solutions are similar, but contain a structure. Thereby, the parts of the artifact are independent from each other and can be adopted alone. Nevertheless, each part might have an influence on some other parts that need to be considered in the adoption (O'Callaghan, 1999, p. 23). Artifacts with (in-)dependent parts have the advantage that they can be integrated easier into existing contexts, where parts of the solution might already exist. Most artifacts that use the notion reference model on the other hand have the characteristic of a fixed system (Hannebauer et al., 2010, p. 2). These artifacts propose solutions with fixed relations between the parts that cannot be taken out of the context (Farooq et al., 2008, p. 128). Often, one of their main contributions are the interdependencies between the parts, rather than the individual parts themselves (Giebler et al., 2020, p. 57; Murturi et al., 2015, p. 65).

5 Discussion

In chapter 4 I analyzed existing artifacts that the designers call reference model or deign patterns in order to derive the artifact designers' understanding for the notions. By doing so, I observed similarities and differences between the artifacts that are called differently. It is worth to mention that there is no characteristic in which all artifacts that are called reference models differ from those that are called design patterns. This is clear, as no well acknowledged definitions and differentiations for both terms exist. By discussing possible differences and similarities between the artifacts in the following, I enable future artifact designers to purposefully choose the correct artifact notion for their intended use.

Most artifacts that are called reference model and design patterns somehow claim to guide practitioners in the design of a good solution to a recurring problem. Thus, both terms seem to be used for artifacts that have a similar purpose. Nevertheless, I observed that researchers are more likely to call their research artifact as a reference model in case that the artifact claims to substantially explain or analyze a phenomena and thereby create scientific knowledge (Fischbach et al., 2013, p. 1). This is especially valid, when the artifact claims to explain or analyze the phenomena as a sole purpose and not as a side effect of the design guidance (Wang et al., 2003, p. 1). Thereby, artifacts that claim to scientifically create knowledge are more likely be called reference models rather than design patterns. In addition I found that artifacts that are called reference model tend to be more abstract than those which are called design patterns. For instance, well acknowledged reference models often lead to the development of industry standards (Schmidt et al., 2011, p. 305) that are aligned with multiple stakeholders. Opposite, most design patterns seem to follow the design patterns approach from Alexander et al. (1977) and Gamma et al. (1997) by means of guiding the operational design and recommending practical solutions. Thereby, I did not find any design patterns that led to a standard. Furthermore, artifacts that are called reference model often include multiple perspectives or degrees of granularity. Therefore, the designers of such artifacts use the concept of different views (Schrödl et al., 2011, p. 5-8) or layers (Cespedes, 2012, p. 1), which is rare for artifacts that are called design patterns (Hughes and Duncan, 2012, p. 29). Rather, artifacts that are called design patterns tend to be on the same level of abstraction and view a problem from a single perspective (Noble, 1998, p. 4). A clear differentiation seems to be the presentation and documentation of the artifact. Almost all artifacts that use the notion reference model use a (standardized) modeling language (Martens and Teuteberg, 2011, p. 4). Only a few only use a verbal description (Hall and Rapanotti, 2003, p. 2-5) and only one uses a standardized template to document the elements. The presentation of artifacts that are named design patterns is different. Most of them use a template with standardized categories to systematically document the results (Froyd et al., 2007, p. 24). In addition to the templates, some of them use some kind of diagram to depict the context of the patterns to each other (O'Callaghan, 1999, p. 24). Nevertheless, some of the design pattern called artifacts only use a (standardized) modelling language (Nguyn et al., 2015, p. 249) or a verbal description (Katsali et al., 2016, p. 4-5), which is similar to artifacts that are called reference models. There seems to be another differentiation between the differently called artifacts regarding their design process. The designers of artifacts that are named reference models often use an active design process to design the artifact based on recurring requirements that they identify in practice (Tom and Sitte, 2006, p. 31). Therefore, researchers tend to start with an identification of recurring problems and recurring requirements (Schmidt et al., 2011, p. 305). Thereafter, the solution to the problem/requirement is actively designed (Meurisch et al., 2017, p. 150). On the other hand, most artifacts that are called design patterns are derived by identifying patterns in (good) existing solutions (Voit et al., 2020, p. 330; Alexander et al., 1977, p. 11). Therefore, the design process seems to be an identification and abstraction process rather than an active design process. Although there are artifacts that are called design patterns and which are actively designed, some designers of those artifacts state that patterns should be derived from existing solutions (Moreira and Paiva, 2009, p. 3), which clarifies that design patterns should actually be derived from existing solutions. As a result, artifacts that are called design patterns can only be designed for problems that are already solved by good solutions (Mancini et al., 2020, p. 1-2). In general, design patterns called artifacts tend to be designed for direct adaption and adoption to different implementation contexts in practice. Therefore, many artifacts that are called design patterns define multiple alternative patterns for each sub-problem (DeHon et al., 2004, p. 7). Then, each of the patterns acts as a solution in a specific context, which eases the adaption (German and Cowan, 2000, p. 4; Nguyn et al., 2015, p. 250). In order to integrate the adapted solution into practice, artifacts that are called design patterns often contain multiple patterns for different parts of the problems that are more or less independent from each other (Mears and Zhu, 2017, p. 3-4). Thereby, the individual patterns can be combined with other patterns and integrated in existing solutions, as not all patterns need to be adopted. Artifacts that are called reference model rarely define alternatives for different contexts (Fischbach et al., 2013, p. 10-11; Lings et al., 2007, p. 1). In addition, there is a clear trend that these artifacts are intended to be adopted as a whole, as these artifacts contain components that have fixed relations to each other. Often, the interdependencies between the components are the major contribution of reference models (Hall and Rapanotti, 2003, p. 1-2), which requires its complete adoption. This complicates the adoption, as existing solutions might be needed to be changed.

As a result, there is no characteristic that differentiates all reference models from all design patterns in our literature pool. Rather, there are trends that characterize the differently called artifacts. Due to these trends between the majority of differently called artifacts, artifact designers should carefully name their research artifacts in the future by acknowledging the different understanding for the artifact types in existing research. Based on this discussion, some of the artifacts that are called reference models should better be called design patterns (Fischbach et al., 2013, p. 2) and vice versa (Schmidt, 1996, p. 126). Based on the analysis of existing artifacts I see the chance position reference models and design patterns as two different types of research artifacts. By doing so, future researchers might take the opportunity and develop new and different artifacts and wisely choose an artifact type that matches their intend best.

6 Conclusion & Limitations

With this systematic literature review I investigated the differences and similarities between research artifacts that the artifact designers call reference models and design patterns. My contribution is twofold. First, I developed a classification scheme that future artifact designers can use to develop, analyze or explain the characteristics of their design science artifact. The classification scheme guides researchers to make active decisions about the characteristics of the artifact that they develop. Therefore, the classification scheme itself is a type 1 theory (Gregor, 2006, p. 622-623). The second contribution is the conceptual clarification of the terms reference model and design patterns and how the terms are used by artifact designers. By conceptualizing the differences and similarities on how the terms are used I give future researchers the opportunity to purposefully name their artifact.

This article has some limitations. First, I only considered papers that describe the development of reference models and design patterns. I did so as most of the characteristics of an artifact that are important for our analysis can only be answered, when the development process is well documented. Nevertheless, I excluded papers that document or analyze existing artifacts. The second limitation is that I analyzed existing artifacts that are called reference model or design patterns. Thereby I rely on the understanding of previous researchers for the different terms. Furthermore, I implicitly assume that the artifact developers made an active decision to name the artifact as a reference model or design patterns. This requires that the researchers know these terms and considered the differences, which might not always be the case.

7 References

Alexander, C. (1977). A Pattern Language: Towns, Buildings, Construction. *Oxford University Press.*

Antonio, G., Fiutem, R. and Cristoforetti, L. (1998). Using Metrics to Identify Design Patterns in Object-Oriented Software. *Proceedings Fifth International Software Metrics Symposium. Metrics*, 1-12.

Arney, D., Jetley, R., Jones, P., Lee, I. and Sokolsky, O. (2007). Formal Methods Based Development of a PCA Infusion Pump Reference Model: Generic Infusion Pump (GIP) Project. *2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperabilit*y, 23-33.

Baudry, B., Sunye, G., Yves, T. and Jezequel, J. (2001). Towards a 'Safe' Use of Design Patterns to Improve Software Testability. *12th International Symposium on Software Reliability Engineering*, 1-6.

Bauer, R. and Baumgartner, P. (2011). Showcase of Learning: Towards a Pattern Language for Working with Electronic Portfolios in Higher Education. *Proceedings of the 16th European Conference on Pattern Languages of Program*s, 1-30.

Beck, K. and Cunningham, W. (1987). Using Pattern Languages for Object-Oriented Programs. *87th Workshop on the Specification and Design for Object-Oriented Programming*, 1-2.

Beck, K., Crocker, R., Meszaros, G., Coplien, J., Dominick, L., Paulisch, F. and Vlissides, J. (1996). Industrial Experience with Design Patterns. *Proceedings of IEEE 18th International Conference on Software Engineering*, 103-114.

Bloom, G., Alsulami, B., Nwafor, E. and Bertolotti, I. (2018). Design Patterns for the Industrial Internet of Things. *14 th IEEE International Workshop on Factory Communication System*s, 1-10.

Boillat T. and Legner C. (2015). From Paper-Based to Mobile Checklists - A Reference Model. *12th International Conference on Wirtschaftsinformatik*, 106-120.

Bretschneider, U., Hartmann, M. and Leimeister, J. (2018). Keep them alive! Design and Evaluation of the ‘‘Community Fostering Reference Model’’. *Business & Information Systems Engineering* volume 60, 493–511.

Cespedes, R. (2012). A Reference Model for the Electrical Energy System Based On Smart Grids. Sixth IEEE/PES Transmission and Distribution: *Latin America Conference and Exposition*, 1-6.

Cremonesi, P., Elahi, M. and Garzotto, F. (2015). Interaction Design Patterns in Recommender Systems. *Proceedings of the 11th Biannual Conference on Italian SIGCHI Chapte*r, 66-73.

de Vreede, T., Steele, L., de Vreede, G. and Briggs, R. (2020). LeadLets: Towards a Pattern Language for Leadership Development of Human and AI Agents. *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 683-693.

DeHon, A., Adams, J., DeLorimier, M., Kapre, N., Matsuda, Y., Naeimi, H., Vanier M. and Wrighton M. (2004). Design Patterns for Reconfigurable Computing. *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 13-23.

Detweiler, C., and Hindriks, K. (2012). Value-Sensitive Design Patterns for Pervasive Health Care. *IEEE International Conference on Pervasive Computing and Communications Workshops*, 908-913.

Farooq, M., Shamail, S. and Mian, A. (2008). Reference Model for Devolution in e-Governance. *2nd International Conference on Theory and Practice of Electronic Governance*, 123-129.

Fettke, P. and Loos P. (2003). Classification of reference models: a methodology and its application. In: *Information Systems and e-Business Management.* Springer-Verlag, 35-53.

Fettke, P. and Loos P. (2009). Morning Star, Evening Star, and Venus – On the Use of the Words “Reference Model” and “Pattern”. In: *Patterns in Business and Information Systems Engineering.* Volume 6, 473-474.

Fischbach, M., Puschmann, T. and Alt, R. (2013). Enhancing Soa With Service Lifecycle Management - Towards A Functional Reference Model. *Proceedings of the 21st European Conference on Information Systems*, 1-13.

Fortino, A. (2008). A Pattern Language for Innovation Management. *Management of Engineering & Technology*, 2008, 1-9.

Frank, U. (2006). Evaluation of reference models. In: *Reference Modeling for Business Systems Analysis*, 118-140.

Froyd, J., Layne, J., Fowler, D. and Simpson, N. (2007). Design Patterns for Faculty Development. *37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, 1-5.

Gamero, A., Garcia, J. and Raymundo, C. (2019). Reference Model with a Lean Approach of Master Data Management in the Peruvian Microfinance Sector. *8th International Conference on Industrial Technology and Management*, 56-60.

Gamma, E., Helm, R., Ralph, E. and Vlissides, J. (1997). Design Patterns Elements of Reusable Object-Oriented Software. *Prentice Hall*; 1st ed.

German, D. and Cowan D. (2000). Towards a unified catalog of hypermedia design patterns. *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 1-8.

Giebler, C., Gröger, E., Hoos, E., Schwarz, H. and Mitschang, B.. (2020). A Zone Reference Model for Enterprise-Grade Data Lake Management. *24th International Enterprise Distributed Object Computing Conference*, 57-66.

Gregor, S. and Hevner, A. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS Quarterly* Vol. 37 No. 2, 337-356.

Gregor, S. (2006). The Nature of Theory in Information Systems. *MIS Quarterly* Vol. 30 No. 3, 611-642.

Guerra, E. and Nakagawa, E. (2015). Relating Patterns and Reference Architectures. *Proceedings of the 22nd Conference on Pattern Languages of Programs*, 1-9.

Hall, J. and Rapanotti, L. (2003). A Reference Model for Requirements Engineering. *Proceedings of the 11th IEEE International Conference on Requirements Engineering,* 181-187.

Han, E., Suh, B. and Shin, S. (2016). Developing a Reference Model for Analyzing Mobile Platform Business: From an Ecosystem View. *Proceedings of the AMCIS 2016*, 1-15.

Hannebauer, C., Wolff-Marting, V. and Gruhn, V. (2010). Towards a Pattern Language for FLOSS Development. *Proceedings of the 17th Conference on Pattern Languages of Programs*, 1-10.

Heer, J. and Agrawala, M. (2006). Software Design Patterns for Information Visualization. *IEEE transactions on visualization and computer graphics*, vol. 12, no. 5, 853-860.

Hevner, A., March, S. and Park, J. (2004). Design science in information Systems research. *MIS Quarterly Vol. 28 No*. 1, 75-105.

Hohpe, G. and Woolf, B. (2002). Enterprise Integration Patterns. *Addison-Wesley Professional*.

Hönigsberg, S., Kollwitz, C. and Dinter, B. (2019). Designing a Reference Model for Digital Product Configurators. *14th International Conference on Wirtschaftsinformatik,* 214-228.

Hughes, J. and Duncan, I. (2012). Thinking Towards a Pattern Language for Predicate Based Encryption Crypto-Systems. I*EEE Sixth International Conference on Software Security and Reliability Companion,* 27-32.

ISACA (2019). Cobit 2019 Framework: Introduction & Methodology. 1-64.

Isaku, T. and Iba, T. (2009). Towards a Pattern Language for Cooking. *Proceedings of the 19th European Conference on Pattern Languages of Programs*, 1-12.

ISO/IEC (1994). Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. *International Standard*, 1-68.

Jay Modi, P., Regil, W. And Mayk, I. (2006). The Case for a Reference Model for Agent-Based Systems. *IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 321-325.

Junginger, S. and Moser, C. (2009). Patterns in Business and Information Systems Engineering In: *Patterns in Business and Information Systems Engineering*. 471-473.

Kartseva, V., Hulstjin, J., Tan, Y. and Gordjin, J. (2006). Towards Value-based Design Patterns for Inter- Organizational Control. *Bled 2006 Proceedings,* 1-17.

Katsali, K., Nikaein, N., Schiller, E., Favraud, R. and Braun, T. (2016). 5G Architectural Design Patterns. *IEEE International Conference on Communications Workshops*, 32-37.

Köppe, C., Norgard, R. and Pedersen, A. (2017). Towards a Pattern Language for Hybrid Education. *2017 Conference on Pattern Languages of Program*, 1-17.

Kuechler, W. and Vaishnavi, V. (2012). A Framework for Theory Development in Design Science Research: Multiple Perspectives. *Journal of the Association for Information Systems* 13(6), 395-423.

Legner, C., Pentek, T. and Orro, B. (2020). Accumulating Design Knowledge with Reference Models: Insights from 12 Years' Research into Data Management. *Journal of the Association for Information Systems*, 735-770.

Lewis, T., Rosson, M., Carroll, J. and Seals, C. (2002). A Community Learns Design: Towards a Pattern Language for Novice Visual Programmers. *Proceedings IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, 168-176.

Lindberg, R. and de Troyer, O. (2020). Towards a Reference Model of Guidelines for the Elderly Based on Technology Adoption Factors. *Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good*, 30-35.

Lindberg, S., Wärnestal, P., Nygren, J. and Svedberg, P. (2014). Designing Digital Peer Support for Children: Design Patterns for Social Interaction. *Proceedings of the 2014 conference on Interaction design and children*, 47-56.

Lings, B., Lundell, B., Agerfalk, J. and Fitzgerald, B. (2007). A reference model for successful Distributed Development of Software Systems. *International Conference on Global Software Engineering*, 130-139.

Liszio, S. and Masuch, M. (2016). Lost in Open Worlds: Design Patterns for Player Navigation in Virtual Reality Games. *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology*, 1-7.

Mancini, F., Bruvoll, S., Melrose, J., Leve, F., Mailloux, L., Ernst, R. and Rein, K. (2020). A Security Reference Model for Autonomous Vehicles in Military Operations. *Workshop on Cyber-Physical Systems Security*, 1-8.

March, S. and Smith, G. (1995). Design and natural science research on information technology. *Decision Support Systems* 15, 251-266.

March, S. and Storey, V. (2008). Design science in the information systems Discipline: an introduction to the special issue on design science research. *MIS Quarterly* Vol. 32 No. 4, 725-730.

Martens, B. and Teuteberg, F. (2011). Risk and Compliance Management for Cloud Computing Services: Designing a Reference Model. *Proceedings of the Seventeenth Americas Conference on Information Systems*, 1-10.

Matthes, F. and Ernst, A. (2009). Capturing, Structuring, and Passing Knowledge for the Design of Complex Systems as Patterns In: *Patterns in Business and Information Systems Engineering*. 469-470.

McDonald, C. (2012). A Design Pattern for Responsible Information Systems Education. *23rd Australasian Conference on Information Systems*, 1-8.

Mears, B. and Zhu, J. (2017). Design Patterns for Silent Player Characters in Narrative-Driven Games. *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 1-4.

Meurisch, C., Ionescu, M., Schmidt, B. and Mühlhäuser, M (2017). Reference Model of Next-Generation Digital Personal Assistant: Integrating Proactive Behavior. *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, 149-152.

Miguel Rosado da Cruz, A. (2014). A Pattern Language for Use Case Modeling. *Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development,* 408-414.

Moreira, R. and Paiva, A. (2009). Towards a Pattern Language for Model-Based GUI Testing. *Proceedings of the 19th European Conference on Pattern Languages of Programs,* 1-8.

Murturi, A., Kantarci , B. and Oktug, S. (2015). A Reference Model for Crowdsourcing as a Service. *IEEE 4th International Conference on Cloud Networking*, 64-66.

Nadschläger and Küng (2017). Towards a Pattern Language for Knowledge Processing Systems: Expert Systems. *IEEE 4th International Conference on Cloud Networking*, 64-66.

Nguye, P., Yskout, K., Heyman, T., Klein, J., Scandariato, R. and Les Traon, Y. (2015). SoSPa: A System of Security Design Patterns for Systematically Engineering Secure Systems. *Proceedings of the 18th International Conference on Model Driven Engineering Languages and System*s, 246-255.

Noble, J. (1998). Towards a Pattern Language for Object Oriented Design. *Proceedings Technology of Object-Oriented Languages*, 2-13.

O'Callaghan, A. (1999). Focus Issue on Legacy Information Systems and Business Process Change: Migrating Large-Scale Legacy Systems to Component-Based and Object Technology: The Evolution of a Pattern Language. *Communications of the Association for Information Systems* 2(3), 1-40.

Otto, B. and Ofner, M. (2010). Towards a Process Reference Model for Information Supply Chain Management. *18th European Conference on Information Systems*, 1-12.

Patil, S., Drozdov, D. and Vyatkin, V. (2018). Adapting Software Design Patterns to Develop Reusable IEC 61499 Function Block Applications. *16th International Conference on Industrial Informatics*, 725-732.

Pongpradit, P. and Prompoon, P. (2016). Constructing initial Design Patterns for Online-Social-Network based applications. *15th International Conference on Computer and Information Science*, 1-7.

Pousttchi, K. and Hufenbach, Y. (2011). Value Creation in the MobileMarket - A Reference Model for the Role(s) of the FutureMobile Network Operator. *Business & Information Systems Engineering* 5, 299-311.

Reiter, M., Fettke, P. and Loos, P. (2013). Towards a reference model for ecological it service management. *Thirty Fourth International Conference on Information Systems*, 1-20.

Romero, R., Gonzales, A. and Raymundi, C (2019). Data Governance Reference Model under the Lean Methodology for the Implementation of Successful Initiatives in the Peruvian Microfinance Sector. *Proceedings of the 2019 8th International Conference on Software and Information Engineering*, 227-231.

Schlieter, H., Juhrisch, M. and Niggemann, S. (2010). The Challenge of Energy Management – Status- Quo and Perspectives for Reference Models. *PACIS 2010 Proceedings*, p. 411-422.

Schmidt, D. (1996). A Family of Design Patterns For Flexibly Configuring Network Services in Distributed Systems. *Proceedings of International Conference on Configurable Distributed Systems*, 124-135.

Schmidt, A., Otto, B. and Österle, H. (2011). A Functional Reference Model for Manufacturing Execution Systems in the Automotive Industry. *Wirtschaftsinformatik Proceedings 2011*, 1-24.

Schrödl, H., Gugel, P. and Turowski, K. (2011). Towards a Reference Model for the Identification of Strategic Supply Chains for Value Bundles. *Proceedings of the 2011 44th Hawaii International Conference on System Sciences*, 1-10.

Simon, H. (1996). The Sciences of the Artificial. *MIT Press*, Cambridge, 1-231.

Sprenger, M. and Mettler, T. (2016). On the utility of e-health business model design patterns. *Twenty-Fourth European Conference on Information Systems*, 1-16.

Stanoevska-Slabeva, K. (2003). Towards a Reference Model for M-Commerce Applications. *European Conference on Information Systems*, 1-14.

Stewart, G. (1997). Supply-Chain operations reference model - the first cross-industry framework for integrated supply-chain management. *Logistics Information Management*, Vol. 10 No. 2, 62-67.

Swacha, J. and Muszyńska, K. (2016). Design Patterns for Gamification of Work. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 763-769.

Thomas, O. (2006). Understanding the term reference model in information systems research: history, literature analysis and explanation. *Business Process Management Workshops*, 484-496.

Tom, M. and Sitte, J. (2006). Family System: A Reference Model for Developing Home Automation Applications. *IEEE International Conference on Systems, Man and Cybernetics*, 32-37.

Torraco, R. (2005). Writing Integrative Literature Reviews: Guidelines and Examples. *Human Resource Development Review 4(3*), 356-367.

Van Turnhout, K., Bennis, A., Craenmehr, S., Holwerda, R., Jacobs, M., Niels, R., Zaad, L., Hoppenbouwers, S., Lenior, D. and Bakker, R. (2014). Design Patterns for Mixed-Method Research in HCI. *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, 361-370.

Voit, T., Schneider, A. and Kriegbaum, M. (2020). Towards an Empirically Based Gamification Pattern Language Using Machine Learning Techniques. *IEEE 32nd Conference on Software Engineering Education and Training*, 1-4.

Vom Brocke, J., Fettke,, P., Loos, P., Junginger, S., Moser, C., Keller, W., Matthes, F. and Ernst, A. (2009). Interdisciplinary Design Theory In: Patterns in Business and Information Systems Engineering. *Business & Information Systems Engineering volume* 1, 470-471.

Walls, J., Widmeyer, G. and El Sawy, O. (1992). Building an Information System Design Theory for Vigliant EIS. *The Institute of Management Sciences*, 36-29.

Wang, Y., Patel, S., Patel, D. and Wand, Y. (2003). A layered reference model of the brain. *Proceedings of the Second IEEE International Conference on Cognitive Informatics*, 1-11.

Wang, Y., Zhang, C. and Wang, F. (2018). What Do I Know about the Tools of Detecting Design Patterns? *IEEE International Conference on Progress in Informatics and Computing,* 379-387.

Webster, J. and Watson, R. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2), 13-23.

Winter (2009). Patterns in Business and Information Systems Engineering. *Business & Information Systems Engineering volume 1*, 468.

Wu, H., Ren, S., Garzoglio, G., Timm, S., Bernabeu, G. and Noh, S. (2016). A Reference Model for Virtual Machine Launching Overhead. *Proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 374-383.

Zaki, M. and Forbrig, P. (2012). Towards a Pattern Language for Modeling Interactive Applications in Smart Meeting Rooms. *IEEE International Conference on Pervasive Computing and Communications Workshops*, 703-708.

Zimmermann, H. (1980). OS1 Reference Model-The IS0 Model of Architecture for Open Systems Interconnection. *IEEE transactions on communications*, 28(4), 425-432.

Eidesstattliche Erklärung zu § 14 Abs. 1 Nr. 6

Ich gebe folgende eidesstattliche Erklärung ab:

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig ohne unzulässige Hilfe Dritter verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und alle wörtlich oder inhaltlich übernommenen Stellen unter der Angabe der Quelle als solche gekennzeichnet habe.

Die Grundsätze für die Sicherung guter wissenschaftlicher Praxis an der Universität Duisburg-Essen sind beachtet worden.

Ich habe die Arbeit keiner anderen Stelle zu Prüfungszwecken vorgelegt. Eine Ausnahme besteht einzig bei dem Essay im Anhang, welcher bereits in einem Seminar vorgelegt und bewertet wurde. In der Dissertation wird auf diese vorherige Bewertung ausdrücklich hingewiesen.

Braunschweig, 07.10.2021,

Ort, Datum, Unterschrift