

Article

A Systematic Procedure for Utilization of Product Usage Information in Product Development

Quan Deng^{1,2,*}  and Klaus-Dieter Thoben^{1,2} ¹ Faculty of Production Engineering, University of Bremen, 28359 Bremen, Germany; tho@biba.uni-bremen.de² Bremer Institut für Produktion und Logistik GmbH, 28359 Bremen, Germany

* Correspondence: dqu@biba.uni-bremen.de; Tel.: +49-(0)421-218-50116

Abstract: Product design is crucial for product success. Many approaches can improve product design quality, such as concurrent engineering and design for X. This study focuses on applying product usage information (PUI) during product development. As emerging technologies become widespread, an enormous amount of product-related information is available in the middle of a product's life, such as customer reviews, condition monitoring, and maintenance data. In recent years, the literature describes the application of data analytics technologies such as machine learning to promote the integration of PUI during product development. However, as of today, PUI is not efficiently exploited in product development. One of the critical issues to achieve this is identifying and integrating task-relevant PUI fit for purposes of different product development tasks. Nevertheless, preparing task-relevant PUI that fits different product development tasks is often ignored. This study addresses this research gap in preparing task-relevant PUI and rectifies the related shortcomings and challenges. By considering the context in which PUI is utilized, this paper presents a systematic procedure to help identify and specify developers' information needs and propose relevant PUI fitting the actual information needs of their current product development task. We capitalize on an application scenario to demonstrate the applicability of the proposed approach.

Keywords: product usage information; product development; data-driven design; PUI utilization context; data provision; data interpretation and integration; closed-loop product lifecycle management



Citation: Deng, Q.; Thoben, K.-D. A Systematic Procedure for Utilization of Product Usage Information in Product Development. *Information* **2022**, *13*, 267. <https://doi.org/10.3390/info13060267>

Academic Editor: Dimitris Apostolou

Received: 29 March 2022

Accepted: 23 May 2022

Published: 25 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Companies are forced to meet ever-changing customer expectations to achieve high acceptance of their products on the market. A considerable number of approaches can improve product design quality, such as quality function deployment, concurrent engineering, design for X (DfX), early customer involvement [1], and the utilization of product usage information (PUI) [2,3]. As emerging technologies such as IoT technologies become widespread, producers can collect large-scale PUI from different stakeholders, including retailers, customers, users, and the providers of product-related services such as maintenance, renting, and leasing. Its content, format, and quality are heterogeneous and cover, for example, service reports, review reports, and embedded systems' measurements. This information provides producers a better understanding of user expectations, usage situations, product behavior, and performance. Consequently, this understanding might revise product developers' assumptions and experiences about the current state of the world and might lead to changed expectations, goals, and hypotheses in product development. It helps substitute assumptions for evidence, increase simulation models' accuracy, create comprehensive test cases, and evaluate solution alternatives. For instance, the analysis of performance information can substantiate failure and degradation mechanisms that a producer can use to prioritize critical issues, analyze root causes, and eventually improve current and future product generations [4].

Despite its potential value, PUI has not been widely utilized in product development. Previous research focused on how this information improves product design conceptually

and with realized application cases. These studies are rather specific, focusing on specific product usage information or specific product development tasks. To exploit the value of PUI and facilitate producers to utilize PUI widely in different product development tasks, a general framework that guides the capture, sharing, integration, analysis, and use of different types of PUI in different product development tasks would be helpful. Many challenges have been identified, including, for example, data collection during product use [5]. This article is not intended to cover the complete general framework. It deals with part of this, about identifying and integrating task-relevant PUI that fits different product development tasks.

The literature describes the data-driven design and the application of various data analytics methodologies to get knowledge out of PUI for product development tasks. To apply data analytics, PUI needs to be prepared [6]. It is a very important but significant challenge to provide the appropriate information to users of information systems in a specific situation to support a work task [7]. For problem solving in a given product development task, it is important to discard useless data and have a set of PUI relevant to the needs of the given task. The set of PUI should be helpful for decision-making to answer questions or solve the problems in the given task. However, there are many different PUI distributed in departments or organizations along with product usage phases. Finding and integrating relevant PUI for work tasks is time-consuming and does not always bring the desired result.

Research studies about professionals' information needs and information-seeking behaviors have been carried out for decades [8,9]. Different approaches and systems have been developed to support engineers in finding required information for their tasks, such as Desktop Search, PLM Systems, and Enterprise Search. With these approaches, engineers often give their information needs in the form of keyword queries to retrieve engineering documents. To improve the level of information for engineers with appropriate information that supports their current product development task, context-based approaches taking into consideration domain-specific knowledge have been proposed [10,11]. However, these approaches have mainly focused on recommending and providing engineering documents and knowledge items. In addition, the context models of these approaches are relatively specific for their targeted areas.

Rather than supporting the provision of engineering documents and knowledge items, this article presents a systematic procedure for product developers to identify and integrate relevant PUI for their current product development tasks. The term "developers" in this article refers to the product development team that is involved in the product development process, including, for example, designers, data scientists, and product managers. We aim for an approach that (1) helps to identify and specify information needs of developers regarding PUI, and (2) helps to prepare relevant PUI fitting the needs of the current work situations of developers. To achieve both, we propose a systematic procedure, including a product development task model, to characterize and contextualize the situation in which a developer is utilizing PUI and support the provision of PUI.

The remainder of this paper is structured as follows. Section 2 describes the conceptual foundations of this article and the related work. Afterward, Section 3 presents the proposed approach, followed by the description of two application scenarios in Section 4. After that, a short discussion is given. Finally, Section 6 concludes and states current limitations and future works.

2. Background and Related Work

This section covers previous work on PUI and its usage in product development. The first two parts of this section present PUI and its benefits and use in product development. The following parts describe the seeking and context-based provision of PUI for product development tasks and their challenges, which motivates the proposal of a new systematic procedure in the next section.

2.1. Product Usage Information (PUI)

A product is something sold by an enterprise to its customers [12]. Companies manage product-related information through Product Lifecycle Management (PLM). This lifecycle often extends from the product’s early design stage to the manufactured items’ disposal. Figure 1 illustrates the product lifecycle from an information management perspective. The presented product lifecycle has three subsequent phases. Each phase’s name corresponds to its sequence position, i.e., beginning of life (BOL), middle of life (MOL), and end of life (EOL).

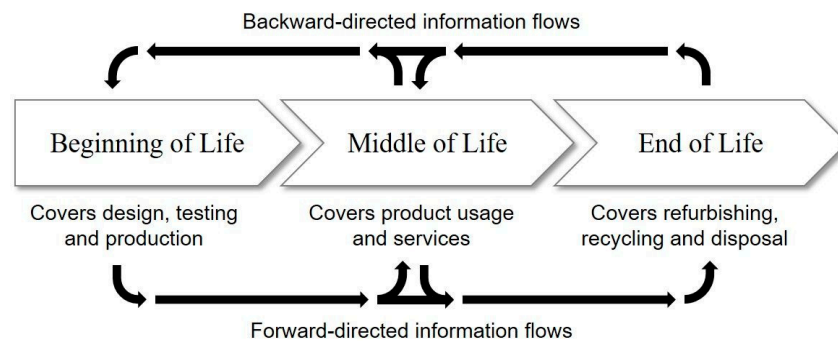


Figure 1. Phases and information flow of the product lifecycle [13].

This article focuses on information extracted from the MOL phase of a product or, in other words, the usage phase of a product. Wellsandt, Hribnik, and Thoben [13] used the term Product Usage Information (PUI) for this information. They defined it as ‘product-related information that is created after the product is sold to the end customer and before the product is no longer useful for a user’. The term PUI is similar to field data [14], in-service information [15], and field feedback [16], but it indicates where the information comes from in the product lifecycle without constraining it to a specific data source, format, or content. This distinction is subtle but necessary, in our view, to better connect it to the extensive conceptual world of PLM.

Companies can acquire PUI through various communication channels, such as call centers, help desks, measurement systems, software logging, and various website and service types on the Internet. The latter include, for instance, weblogs, social networking services, product review services, and online marketplaces. The broad range of conventional and newer channels facilitates the rapid collection of PUI. Table 1 provides an overview of relevant communication channels. This overview is not comprehensive, but it outlines the complexity of PUI in product development. The source types are according to the Big Data source classification schema proposed by the United Nations Economic Commission for Europe (UNECE) to structure the PUI sources and channels [17]. UNECE uses this classification in their official statistics, and thus we consider it an acknowledged and credible schema. The schema differentiates three data source classes: human-sourced information (HSI), process-mediated data (PMD), and machine-generated data (MGD).

Table 1. PUI channels usable in product development.

Channels	Typical Contents	Source Types
Sensor data repository	Environmental information; performance	MGD
Logfile of embedded control system	Product behavior	
Service report	Maintenance, repair, and failure information; product condition	PMD
Online discussion forum	Opinions; ratings; complaints; customer/user profiles;	
Online customer review platforms	constructive critique	HSI
Telephone	Opinions; complaints; customer profiles	
Email	Opinions; complaints; customer/user profiles; constructive critique	

HSI is mainly loosely structured or ungoverned data created by customers or users and includes, amongst others, opinions, complaints, and expectations. This feedback refers to product instances, but there is often no unique identifier to associate an instance to its user. Missing identifiers are sometimes a problem because the employees cannot combine the HSI with other information about an instance (e.g., measurements). Therefore, HSI can provide insights about many product aspects but often without being precise. HSI quality and amount rely much on the customers or users producing it and the available channels. Producers typically have more control over MGD and PGD.

MGD is mostly well-structured and typically originates from sensors and log files of product-embedded control systems. Producers can control MGD quality if employees take the product development's information needs into account during the product's sensor system design.

PMD integrates HSI and MGD. Service records, for instance, have text input from service personnel and data from measurements. They contain maintenance, repair, and failure information, which are helpful for producers to identify a product's weaknesses.

Within the lifetime of a product, the PUI of a product can range from over many years. It is mostly written information. Each PUI refers to a piece of usage information from, for instance, a sensor of a product instance. These PUI all give limited insight into product use and generally only reflect problems with the products, and they offer little insight into general patterns of product usage [18]. To have meaningful and holistic insight into product usage, companies usually require integrating and analyzing PUI from many product instances. A high number of pieces of PUI makes manual reading and analysis often hardly feasible.

2.2. Benefits and Usage of PUI in Product Development

Academic literature describes a variety of use cases that outline how product development tasks can benefit from PUI. PUI with information about reference products' real-world usage revises the producer's understanding of the product, its users, and usage context [3]. This revised understanding suggests changes in requirements and product designs. It supports producers in capturing new requirements for product improvement and developing the next generation of product design. In the task clarification phase, the support focuses on changing the requirements list by adding, removing, or editing entries, based on a deeper understanding of, for instance, product usage and customer expectations. For example, Joung et al. [19] contributed to identifying latent customer needs during the pre-execution and post-execution steps of product use by customers with an analysis of customer complaints. In the product conceptual, embodiment, and detailed design phases, it helps to analyze problems, search for solutions, and select and evaluate solutions. For example, Abramovici and Lindner [20] and Abramovici et al. [21] analyzed various types of PUI, including both unstructured data (e.g., MRO reports) and structured data (e.g., sensor data about temperature, workload, speed) to gain a comprehensive view of possible influencing factors and causes. Based on the gained insight on influence factors (e.g., temperature, speed), product developers can determine changes on respective design parameters (e.g., material, limitation of rotation speed) to reduce the failure probability. Alkahtani et al. [22] analyzed warranty data and manufacturing data to recommend optimized design parameters and tolerance for maximum reliability and minimum cost. van der Vegte et al. [23] explored time-stamped usage data for an engineering simulation to reveal how use-related phenomena influence product performance, and to evaluate and prioritize design variations. Stietenron et al. [24] introduced real-life PUI-supported simulations into the product design process to eradicate the need for repetitive and costly rounds of prototype development, production, and testing.

In terms of usage methodology, data-driven design or data-informed design, using data mining and machine learning on big data for decision-making support, have been increasingly discussed in the literature as product innovation enablers [5,25–27]. It has, on one side, chances to overcome the limitations of human analysis and guide developers

towards optimal design decisions, and on the other side, still many challenges. PUI is often analyzed and utilized to support data-driven design for product improvement. Regarding the usage and analysis methodology of PUI, the reviewed application cases often use data science methods and tools (e.g., machine learning, text and data mining, and statistic analysis) to derive useful knowledge or models for development tasks. In some cases, researchers use a set of integrated PUI directly for, for example, simulation analysis [28].

2.3. PUI Provision in Product Development

As mentioned in the previous section, various researches have been conducted on the usage of PUI in scenario-specific product development tasks. On the provision of PUI in product development tasks, these approaches are often under an implicit assumption that the required relevant PUI is already identified and prepared, for example, in the approach of using MRO reports for failure cause identification [21]. Abramovici and Lindner [2] present a framework, i.e., a PUI Feedback Design Assistant, for the acquisition, extraction, aggregation, and analysis of PUI for the generation and provision of knowledge to develop improved development product generations. Although the article describes the scope of considered PUI and the layers supporting the use of PUI in product development, it has little focus on providing the right PUI to meet the needs of different product development tasks. Voet et al. [29] developed a framework for capturing and analyzing product usage data for continuous product improvement. The framework is to determine which usage data, i.e., sensor data, should be captured in the next product generation, and then develops systems to capture and analyze these data so that the next generation of products can be optimized towards the given goal setting. To summarize, the research of Voet et al. is more from the viewpoint of capturing the correct usage data rather than the viewpoint of finding and utilizing the right existing usage data for a given product development task.

To find the correct information for product development tasks, information-seeking approaches are often discussed. Although many approaches support information seeking in product development, identifying task-relevant information from a rapidly growing number of structured and unstructured data is still challenging. Generic search tools, such as desktop search, database search, and web search, can help engineers use specific keywords to find experts and relevant documents, such as material information, technical reports, and patents. Enterprise Search supports the search of documents within the organization that contains relevant information and people with the right expertise [30]. PLM/PDM systems, e.g., Teamcenter, enable engineers to find different kinds of product information, for instance, CAD drawings, BOM data, parts information, and manufacturing instructions, across more domains and departments. These systems work primarily on seeking product information from the BOL phase, although some PLM systems and customer success and analytic tools provide functionalities to manage, search or even analyze PUI, for example, integrating sensor data for realizing digital twins. They have hardly focused on understanding the product development activities, in which PUI is utilized, and providing appropriate PUI to satisfy the respective information needs of tasks.

2.4. The Context and Context-Based PUI Provision in Product Development

It is worth noting that there are many different definitions for the term “context”. Bazire and Brézillon [31] and Liu et al. [32] have conducted literature reviews on context taxonomies. Edmonds [33] says that context is “the abstraction of those elements of circumstances in which a model is learnt . . .”. According to Merriam Webster [34], context is “the interrelated conditions in which something exists or occurs”. To achieve our objective of preparing task-relevant PUI, we focused on the representation of product development tasks as the interrelated conditions or circumstances in which task-relevant PUI is utilized, i.e., identified and integrated. Dey [35] use the term context to represent any information to characterize the situation of an entity. According to the definition, “if a piece of information can be used to characterize the situation of a participant in an interaction, then that information is context”. In this research, we mostly use information about product development to

characterize the current situation of a product developer during PUI utilization. The context is about, for example, for which product design object (e.g., medical imaging devices) and design activity (e.g., find the distribution and major sources of interoperability problems) the developer is looking for PUI. The context can significantly influence the identification, interpretation, and integration of the PUI. For different product design objects and design activities, the information needs of PUI could be different, and the PUI could be interpreted and utilized in a different way.

To improve the provision of appropriate information, context-based approaches have been proposed. They help provide knowledge and relevant documents applicable to the engineer's situation [11,36]. Redon et al. [11] presented a context-based search platform based on the identification of an engineering context and the subsequent pushing of applicable knowledge to that particular engineer. They introduced a context model to represent the engineer's context of an engineer working in a specific company, and then used it to index available knowledge. Liu et al. [37] have proposed a knowledge recommending approach for new product development based on workflow context matching. It can help engineers find similar outcomes, e.g., product drawing and work instructions, of former new product development tasks. With the consideration of context awareness, Mehrpoor [10] argues that existing systems either are too general or do not apply domain-specific tools and knowledge to improve information retrieval against stored data and information in file systems with the engineering focus and cannot satisfy engineering needs. Building on this assumption, Mehrpoor proposed a system to recommend relevant documents for engineers in specific work contexts. Therefore, he constructed an ontology-based context model as a knowledge domain to derive engineers' work contexts and then used the context model to support the identification of engineers' information needs. These approaches have built and applied context models to better satisfy the information needs of engineers but have their specific areas of focus. They have mainly focused on the recommendation and provision of documents and knowledge items in BOL phases, rather than PUI in MOL phases. Considering the characteristics of PUI, for instance, engineers often require integration and synthesis PUI from a large number of product instances to obtain meaningful insight for problem solving; approaches with the suggestion and recommendations of single documents and knowledge items are not optimally suitable for the provision of PUI during problem solving in product development activities. Furthermore, although there are many different context elements and context models around product development, for example, [38,39], they have targeted specific areas. Few of them focus on the purpose of this study, i.e., appropriately identifying and integrating PUI for product development tasks.

2.5. Summary

To summarize, as presented in the above sections, many research studies have investigated PUI, the benefits of PUI in product development, and the capture and analysis of PUI for data-driven product design. However, preparing task-relevant PUI that fits different product development tasks is often ignored. The literature describes several methods, systems, and tools to help provide appropriate information for product development tasks, such as enterprise search, PLM systems, and context-based approaches. Nevertheless, they mainly focus on documents and knowledge items from the BOL phases, rather than PUI from the MOL phases. In their research, data integration is often little relevant. They cannot optimally support the identification and integration of PUI, although this kind of integration is usually required during the analysis and use of PUI. Additionally, PUI is from different systems or stakeholders with different viewpoints, its relevance, meaning, and interpretation could depend on the current product development tasks. Developers could have different perspectives and interpret and integrate PUI differently in different product development tasks. This study addresses this research gap in preparing task-relevant PUI and rectifies the related shortcomings and challenges. It proposes a high-level product development task model as context model to characterize the work context of developers during the PUI utilization and highlight contextual factors directly relevant to the PUI

provision in product development tasks. Based on the context model, this study guides developers to identify the relevant PUI and interpret and integrate PUI appropriately according to their understanding and needs in the product development tasks. It aims for a systematic procedure that helps identify and integrate task-relevant PUI for different product development tasks.

3. Proposal of a Systematic Procedure

Figure 2 gives an overview of the proposed procedure for context-based PUI identification and integration. The procedure provides stepwise guidelines to support developers in understanding their current product development tasks, clarifying their information needs on PUI, and afterward identifying, interpreting, and integrating appropriate PUI for data-informed decision-making in the product development process. It is an iterative approach that allows going back and forth between steps until achieving a resilient result. The following paragraphs briefly describe the methodology. Following the short outlines, the sub-sections provide detailed descriptions for each step.

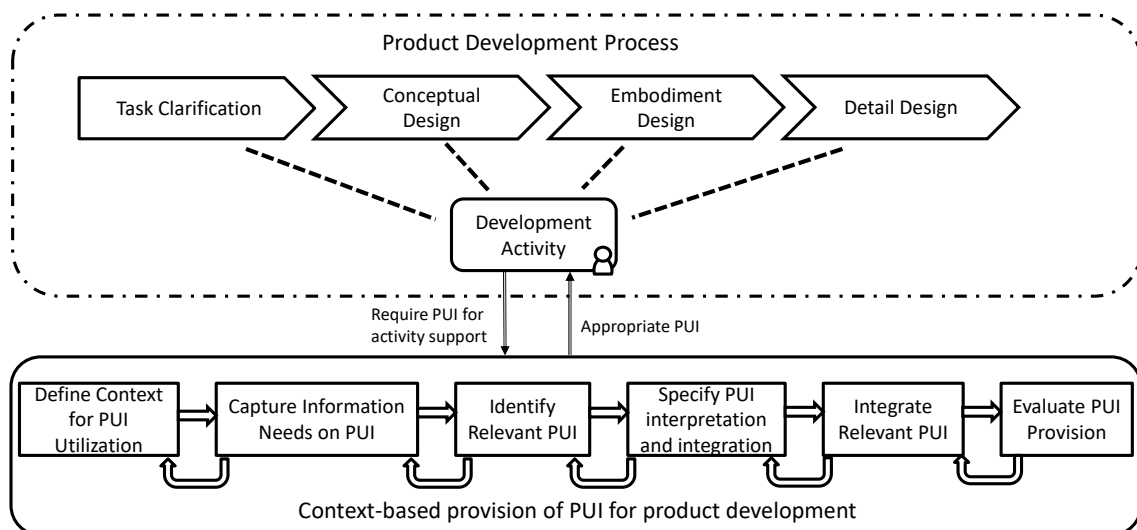


Figure 2. Methodology for context-based PUI provision.

The methodology consists of six steps:

1. Define context for PUI utilization. This step is to collect information about product development. It characterizes the situation in which a developer is utilizing PUI.
2. Capture information needs on PUI. This step describes the information needs of developers on PUI for product development tasks.
3. Identify relevant PUI. This step filters PUI relevant to the investigation product development's information needs and ignores irrelevance.
4. Specify PUI interpretation and integration. This step is to specify the interpretation and integration of the relevant PUI, which is identified in the previous step, from the perspective of the considered product development tasks.
5. Integrate relevant PUI. This step uses data integration tools to integrate the relevant PUI according to the specification described in the previous step.
6. Evaluate PUI provision. This step is to evaluate the satisfaction level of the provided PUI in product development tasks.

In a first step, a reference context model will be developed in the area for PUI provision. It will help product developers understand situations around their product development tasks and support them to capture the most important context factors relevant to PUI provision. After that, the methodology will guide developers to specify their information needs on PUI precisely and explicitly. Based on the awareness of product development tasks and information needs on PUI, the approach will support product developers to filter

PUI that is most relevant for problem solving in their product development tasks. In case relevant PUI is missing, companies could consider the necessity of new data acquisition processes, such as sensor installation or asking/purchasing PUI from partners. This relevant PUI will then be interpreted and integrated from the product developers' viewpoints according to the needs of the product development tasks. Most of the steps require involvement and support from domain experts of PUI sources, as they are more familiar with the datasets. PUI is generated in usage and is not fully controlled by developers. Despite the description of available PUI in the first step, it is not expected that developers understand all details of various PUI sources, and know exactly, for example, which PUI is relevant and how to interpret and integrate them for their current tasks. These steps can be performed manually, and are potentially further supported by assistance tools or context-based recommendation systems. We will discuss this shortly in the discussion section.

3.1. Step 1: Define Context for PUI Utilization

In this step, information around product development tasks is collected. To describe and characterize the situation in which PUI is utilized, it is necessary to develop a context model in this area. The starting point of the context model in this study is the upper-level ontology-based context model for enterprise applications [40,41] and the analysis of existing product development related context models, for example, [39]. This study focuses on elements that are directly relevant to the PUI provision in product development activities. The business context, information feature model, and user context in the upper-level context model for enterprise applications are specialized in this study, respectively, as task category, PUI category, and personnel category. Figure 3 illustrates the upper-level conceptual diagram for the product development task model. It represents the context model, which depicts the context of PUI utilization. It lists only generalized concepts, and it is possible to extend or add other concepts specific for companies or use-cases. Table 2 presents an overview and a short description of the product development task model.

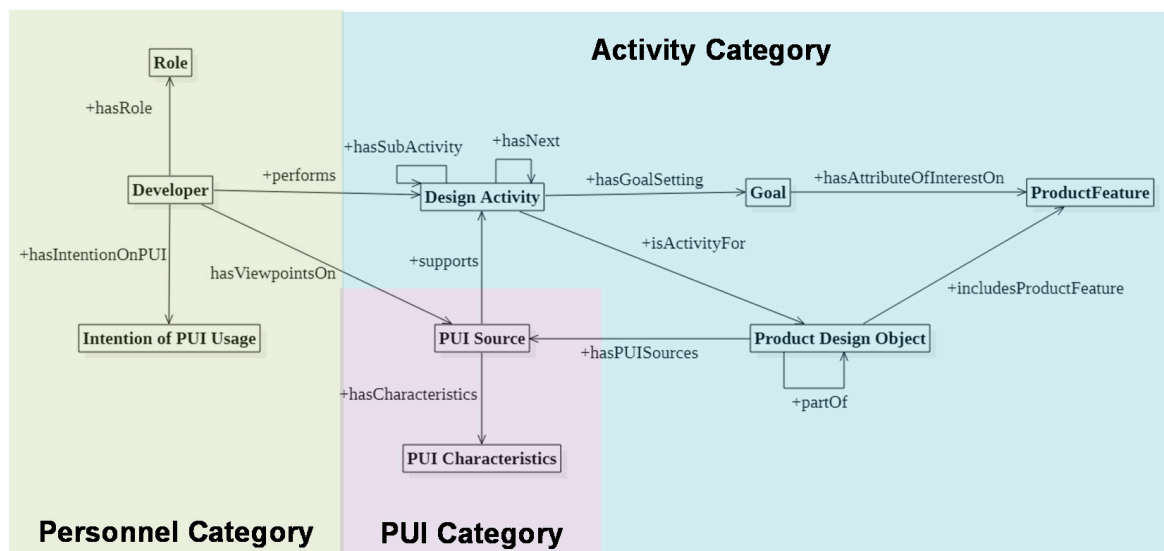


Figure 3. Upper-level conceptual diagram of the product development task model.

The activity category captures information about the current state of product development activity, representing developers' main concerns. It covers the information about the product design object, various ongoing, past, and future design activities around the product design object, and the goals and relevant product features of these activities. Information about the product design object can be derived from, for example, design documents. The product development plan and the development process in practice would help define the design activity, the goal, and focused product features in the design activity.

Table 2. Overview of key elements in the product development task model.

Elements	Short Description
Activity Category	
Product design object	It represents a product or product component that is under development.
Design activity	It represents the current product development activity.
Goal	This concept represents the goal-setting of the design activity.
Product feature	It represents features and aspects of a product design object, which the design activity focuses on and optimizes.
PUI Category	
PUI source	It represents available information sources of PUI
PUI characteristic	It describes the characteristics of the PUI within the PUI sources.
Personnel Category	
Developer	This concept is the individual, i.e., product developer, who performs product development activities and requires the support of PUI for problem solving or decision making.
Role	A developer's role represents his position, discipline, and responsibility in the product development activities, for example, product manager.
Intention of PUI Usage	It represents the rational and subjective reason for the usage of PUI, which motivates the needs of PUI.

The PUI category captures the information about available PUI sources and their related characteristics. It represents all PUI that developers can access for problem solving in product development activities. The PUI sources need to be registered and described by additional domain experts, for example, call center administrators for customer call logs and service personnel for the service reports from service activities. This is because developers often have no holistic view on available PUI sources scattered under different departments or organizations from various product lifecycle activities. The PUI characteristics describe the characteristics and features of the PUI within the PUI sources. Wellsandt, Hribernik, and Thoben [13,42,43] have provided an overview about the characteristics of PUI. It covers, for example, the lifecycle activity (e.g., maintenance) in which PUI is generated, data format, originator, and time frame. These characteristics would affect the provision and usage of PUI. For example, in the design activity of trend identification for textile products, developers may only require PUI from a certain time frame that originated from users. The captured information about PUI can be shared in different product development tasks, i.e., PUI utilization contexts, to give developers an overview of available PUI.

The personnel category captures the information about the developers' identity, their involvement in the product development activities, and preference on PUI usage in the problem solving of the development activities. Other information about developers, such as physical location, access devices, or mood, is less relevant in this research. We mainly target engineers working in offices with computers.

3.2. Step 2: Capture Information Needs on PUI

Weber [44] stated the necessity of defining information needs in terms of both content and characteristics. This article will specify information needs on PUI from two dimensions: information content (what do developers need information about?) and information characteristics (what is the quality and attributes the information must have?). The information content and information characteristic enable identifying relevant PUI with required characteristics.

The content of information needs can be expressed in different forms, such as keywords, information objects with detailed content descriptions, natural language questions, query terms, or ontologies. As the data science methods and tools used for PUI analysis often require the integration of PUI, probably on certain fields (e.g., failure description,

cause, and solution), the specification of information needs should be precise and explicit. Henniske [45] explicitly represents the information needs of archive users in the form of an ontology, which is possible to express information needs explicitly and precisely. However, as information needs evolve from an unexpressed need to an explicit formalized one [46], it is difficult to model the information needs explicitly and precisely in the form of ontologies at the beginning. Therefore, this study suggests formulating the information needs on PUI in terms of the content of the information in different detail levels: PUI profile, PUI view, PUI object, and PUI variable (Figure 4). It is from generic description through to more precise definition respectively. A PUI profile describes the overall content of PUI that is important in the context of PUI utilization. A PUI view represents an individual aspect of product usage that is important to address specific concerns in the design activity. PUI object type is a type of information entity about product usage representing a set of or a class of parameters. PUI variable is used to represent individual parameters and content of PUI object. The PUI object type and PUI variable can be further seen, respectively, as Classes and Properties in ontology for semantic integration.

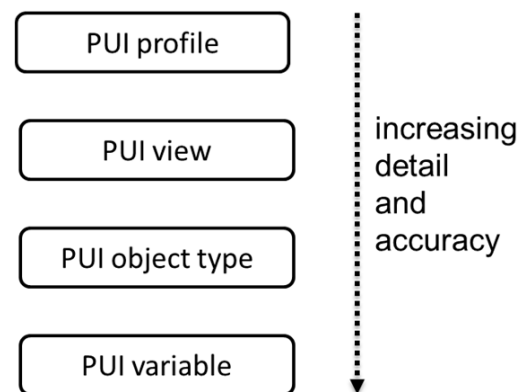


Figure 4. Information content specification of PUI in different detail levels.

Besides content, information needs need to be examined concerning characteristics that the PUI must have. We can take the characteristics provided by [13,42,43] as the first guideline to define the required information characteristics. It includes, for instance, lifecycle activity, timeliness, and data format.

3.3. Step 3: Identify Relevant PUI

This step is to identify pieces of PUI that are relevant to the product development tasks and can meet developers' information needs. On one side, we have the range of available PUI sources described in the PUI model. On the other side, we have the defined information needs that should be satisfied. This step will match the information needs with regards to content and characteristics against that of PUI sources to find the right pieces of PUI that satisfy developers' information needs for performing different product development tasks. It covers both the locating of relevant PUI sources and the locating of pieces of PUI within the sources.

3.4. Step 4: Specify PUI Interpretation and Integration

This step intends to specify the interpretation and integration of the relevant PUI identified in the previous step. Whereas the last step focuses on which PUI is relevant for information needs, this step is mainly about how to interpret and integrate the relevant PUI appropriately to satisfy the information needs. PUI is often captured and organized by different systems and stakeholders, e.g., consumers and customer services. They usually have different concerns and viewpoints from developers. In this step, we identify and capture the knowledge related to interpreting PUI from the developers' viewpoints. Following the captured knowledge, PUI from various sources will then be interpreted, transformed,

and integrated to meet the information needs. For example, when a design activity is about “Find the distribution of interoperability problems per country”, the value “Germany” in the “Country” field of the PUI source customer call logs would be interpreted as a country “Germany”. For a design activity about “Find the distribution of interoperability problems per region”, the same value “Germany” in the “Country” field of the PUI source customer call logs would be interpreted as a region “Europe”. Figure 5 illustrates the interpretation of the same set of PUI from different viewpoints to address the information needs in different product development tasks and PUI utilization contexts.

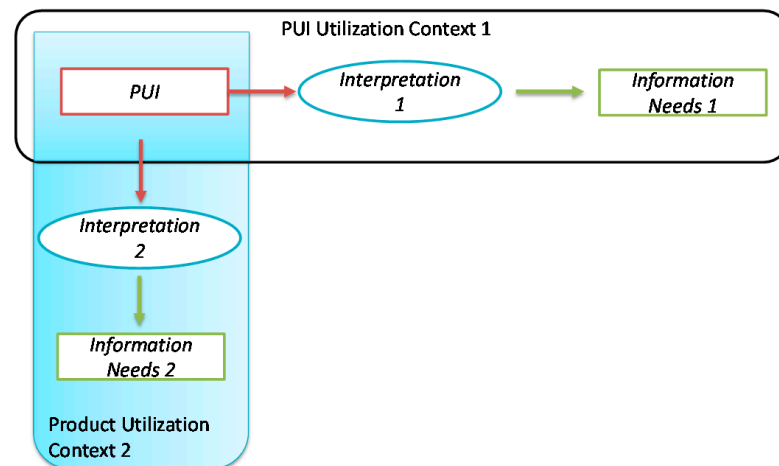


Figure 5. Interpretation of PUI for information needs in different contexts.

3.5. Step 5: Integrate Relevant PUI

This step is the implementation of the specified context-based interpretation and integration from the last step. The implementation is often supported by IT personnel or data scientists. This includes the access of disparate PUI sources and using various software tools to interpret, transform, and integrate PUI from different sources. The information about used software tools and their configurations can be managed for further reuse in similar cases. The main deliverable for this step is the appropriately integrated PUI that is relevant and helpful for product development tasks. Based on the context-based interpreted and integrated PUI, developers can apply various knowledge discovery models and methods to obtain decision-making support in product development.

3.6. Step 6: Evaluate PUI Provision

At this step, the integrated PUI is evaluated. According to the evaluation, developers can go back to previous steps to make necessary adjustments. This evaluation feedback will assist developers in balancing the benefit of PUI usage, selecting the best practice, and improving PUI usage for problem solving in future product development activities.

4. Application Scenarios

This section demonstrates the application of the proposed procedure on the basis of two application scenarios. These two scenarios are different in terms of, for example, product, product development stage, and types of data sources. The first scenario is about improving a product family of medical imaging devices, while the second is about developing a specific component of a boat that is still at the early stage of its development. In terms of data sources, while the first scenario includes weakly- or un-structured product log files and customer call logs, the second scenario mainly focuses on structured IoT data. For clarity and comprehensibility, not all details are described in the second scenario. Figure 6 gives an overview of the application scenarios, as well as the steps and the links between their outcomes.

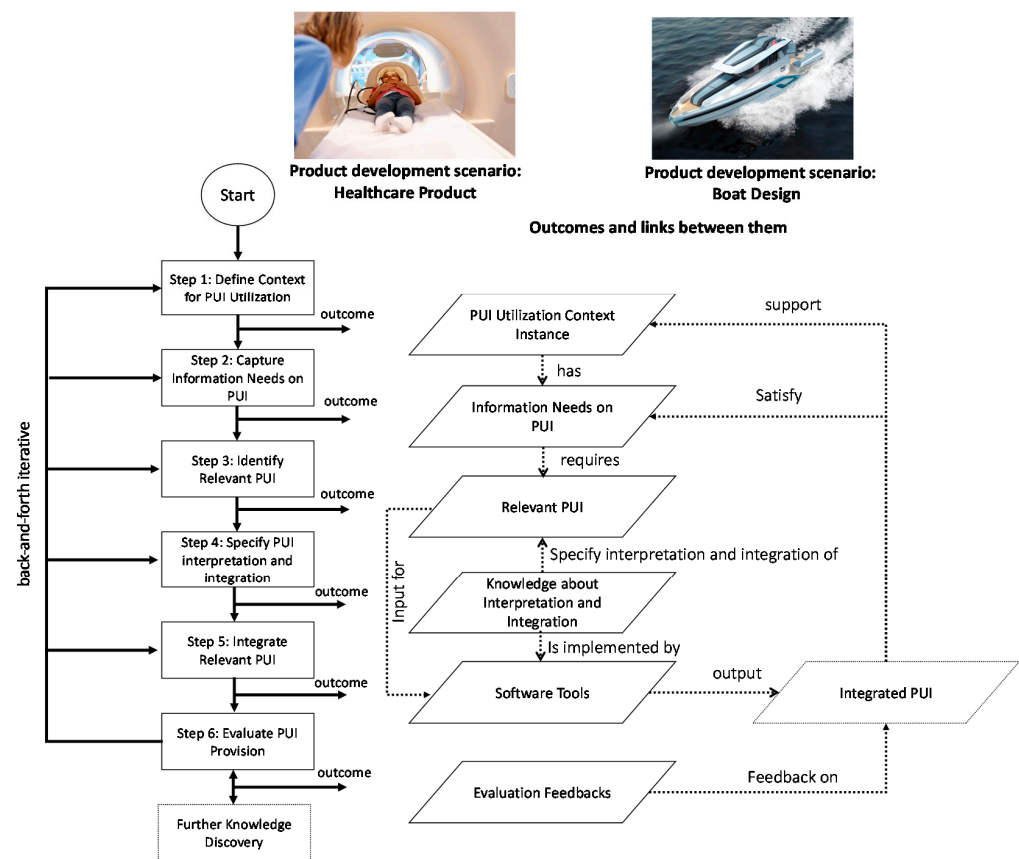


Figure 6. Overview of the steps in the application scenarios (healthcare products ¹ and boat ²). ¹ <http://www.falcon-h2020.eu/index.php/business-scenario-2-healthcare-products-lead-philips/>; (accessed on 29 March 2022) ² <http://www.lincolnproject.eu/business-cases/patrol-boat/> (accessed on 29 March 2022).

4.1. Product Development Scenario: Healthcare Product

The application scenario regards a company with the design and production of health-care products. The product is a family of medical imaging devices, e.g., magnetic resonance imaging (MRI) machine (see Figure 6). They are installed and operated in hospital facilities. With the products, clinical end-users perform scans on patients according to needs and selected protocols, i.e., configurations. The product works together with third-party systems to manage and analyze the images for diagnosis. From the producer side, they have collected a lot of product usage information through various channels, such as embedded sensors, logging systems, and helpdesks. However, the gathered usage information was not used systematically in the development department to improve the next generations of products.

In addition, according to the company, the product causes customer dissatisfaction concerning its usability and interoperability. Within this case study, the company intended to reduce customer dissatisfaction and improve the product based on bottlenecks identified by analyzing the usage information. This project has several business stories and tasks with different focused goals and product features, including, for instance, usability and interoperability. This study takes one of these tasks as an application scenario to demonstrate how the proposed systematic procedure can be applied in practice.

4.1.1. Step 1: Define Context for PUI Utilization

The task is mainly to obtain an improved overview of the interoperability issues in the field. Today, the interoperability issue is one of the most compliant issues in the field service. It emerges during the transfer of images and relevant patient data to other systems.

For example, certain data formats and special characters are not supported, or not properly accepted and interpreted by other systems. The producer intends to address this issue for a better user experience, improving customer satisfaction, and saving relevant costs in the long term. For that reason, the product manager wants first to have an improved overview of the interoperability issues, about where the majority of the interoperability-related customer calls are coming from. This helps to suggest and assign product improvements. Table 3 summarizes the key information in the activity category and personnel category.

Table 3. Healthcare scenario: summary of the elements the activity category and personnel category.

Elements	Values
Activity Category	
Product design object	Family of medical imaging devices
Design activity	Find the distribution and major sources of interoperability problems
Goal	Have an improved overview of the interoperability issues in the field
Product feature	Interoperability
Personnel Category	
Developer	Developer A
Role	Product Manager
Intention of PUI Usage	We plan to use PUI because PUI, such as customer call logs, with issues reported by customers, could probably support a better understanding of interoperability bottlenecks in product usage

Regarding the PUI category, the producer has mainly two types of data sources about this product from the usage phase: customer call logs and product logs. The customer call logs are from the channel helpdesk. When clinical end-users call the help desk for complaints and issues, the field service engineers use the customer call logs to keep the records. The records include details, for example, about the complained issue, seriousness of the issue, the work they have done, and whether the issue is solved. The customer call logs are currently managed in an Excel file. The file has more than 40 columns, for example, ‘Call ID’ (i.e., number to identify the call), ‘Call Open Date’ (i.e., date of the call), ‘Product’ (i.e., text describing the product/system), ‘Subsystem’ (i.e., text describing the part of the system that the call is about), and ‘Customer Complaint’ (i.e., text describing the part of the system that the call is about) (Figure 7). The product logs are from the embedded systems. They are text files and contain different information according to the configuration settings. The logs contain information about the product usage, such as the start/stop of the application, actions, and commands performed within an application.

MarketGroup	Market	Country	CallType	ProductGroup	CustomerComplaintSubject	SubSystem
NA	NA	USA	CM	CV Allura Xper FD10	table movement partly available/ free fl	AD7
APAC	Asean Pac	Thailand	PM	MoS Pulsera	Extended warranty_Silver agreement	Interoperability
Greater China	Greater China	China	CM	CV Integris Cardio mono-plane	01/Integris CV/the tube failure	Stand
APAC	Asean Pac	Korea	PM	CV Integris Vascular bi-plane	Preventive Maintenance	UI modules

Figure 7. Snippet from sample Customer Call Log file.

4.1.2. Step 2: Capture Information Needs on PUI

Concerning the current product development task and existing PUI sources, the product manager believes the customer calls could be helpful. The information needs of

PUI are identified in general as ‘we need interoperability-related customer calls that help determine from where the majority of the interoperability issues originate’. We captured this as a PUI profile. This task focuses on the interoperability issues in product usage, with a special concern on the sources and distribution of the interoperability issues. The product manager intends to check the interoperability issues per country, region, and product group. It helps to understand whether the issue is cultural-specific or related to specific product groups. According to the understanding, the product manager can afterward decide further product improvement actions, for example, training and product adaption concerning a specific region, or the improvement of specific product groups. As the product manager becomes clearer about what is required, the information needs are detailed in PUI view as ‘we need information about the distribution of interoperability related customer calls over countries/regions/product groups’. Finally, the information needs of this task are modeled as an ontology class, i.e., PUI object type, ‘Help_Desk’ (Figure 8). It has several data properties, i.e., PUI variables. PUI variables, ‘TypeOfProblem’, ‘appearedInCountry’, ‘appearedInRegion’, and ‘belongsToProductGroup’, are for analyzing the distribution of interoperability issues over country, region, and product group. The PUI variable ‘TypeOfProblem’ is to categorize the type of complaint problems, such as interoperability issues. This is to filter interoperability-related calls from the number of records with different issues during the analysis. The PUI variable ‘CallOpenDate’ is to filter customer calls according to time.

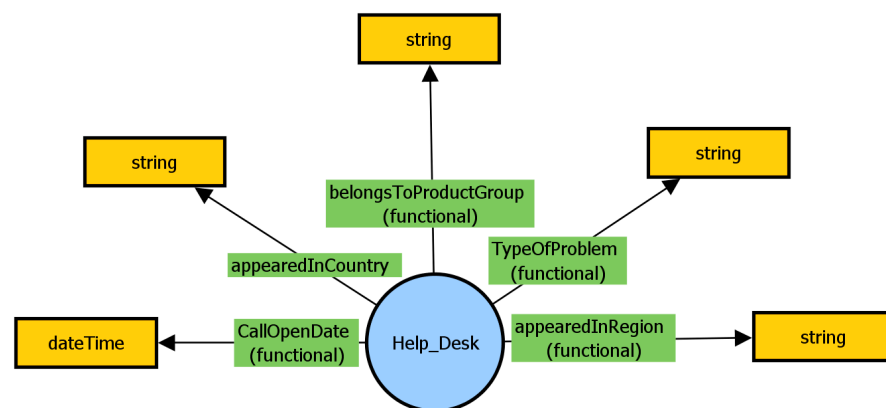


Figure 8. An ontology representing the information needs.

4.1.3. Step 3: Identify Relevant PUI

With the analysis of existing PUI sources in the product development task model, the product manager found the dataset, customer call logs, relevant. It is possible to extract the information content specified in the information needs from the customer call logs.

To locate pieces of relevant PUI within the customer call, we mapped for each detailed PUI variable in the information that needs the relevant customer call fields/columns. For example, the field ‘Country’ in the customer call record is relevant for the PUI variables ‘appearedInCountry’ and ‘appearedInRegion’ in the information needs. The field ‘Product-Group’ in the customer call record matches the PUI variable ‘belongsToProductGroup’ in the information needs. Finally, the PUI variable ‘TypeOfProblem’ in the information needs could be derived from the customer call fields ‘customer complaint subject’, ‘customer complaint’, ‘external engineer text’, ‘internal engineer text’, and ‘subsystem’. Figure 9 shows the match between the customer call fields and the PUI variable ‘TypeOfProblem’ in the information needs. This step is carried out with the support of domain experts, as they are more familiar with the dataset and can accelerate the process.

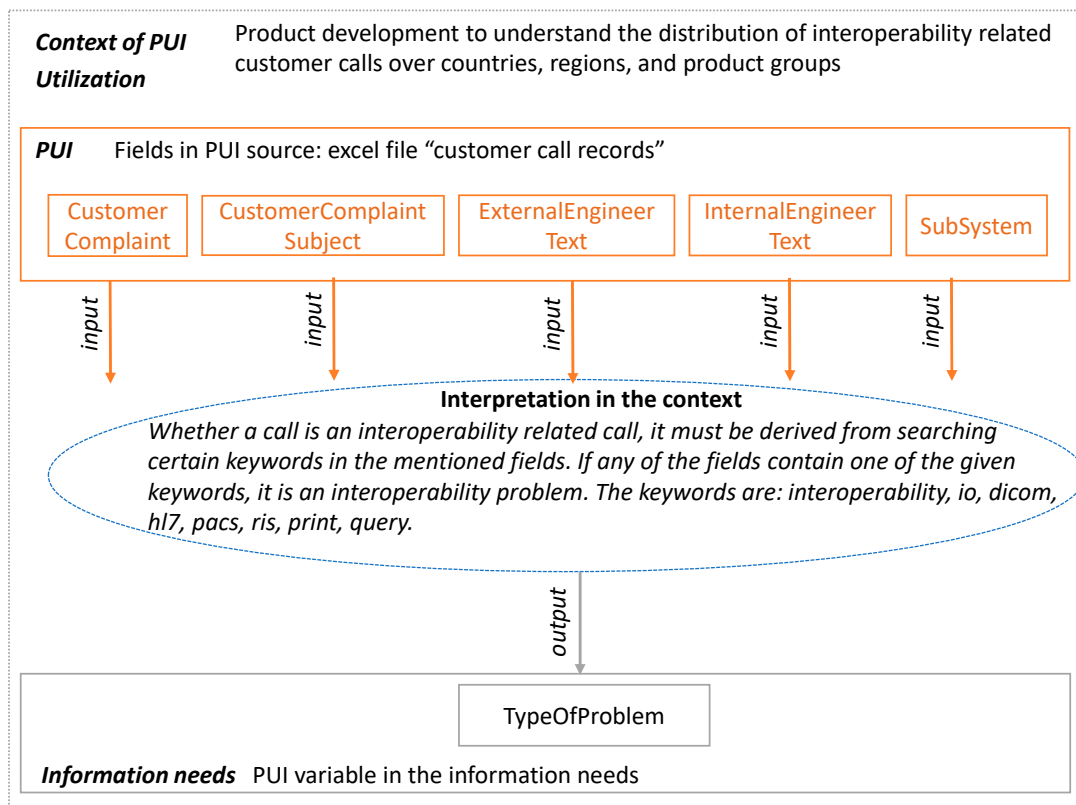


Figure 9. Context-based interpretation for the information needs ‘typeOfProblem’.

4.1.4. Step 4: Specify PUI Interpretation and Integration

With the relevant PUI identified from the previous step, this step describes how to properly interpret and integrate the PUI to fulfill the information needs. For example, in the last step, we have described that the PUI variable ‘TypeOfProblem’ in the information needs could be derived from the five customer call fields. This step is to specify how to interpret and integrate the five identified customer call fields to the PUI variable ‘TypeOfProblem’ in the information needs. As this product development task focuses on interoperability issues, it is required to interpret the five fields from the interoperability viewpoint, i.e., to understand whether a customer call is interoperability-related. After discussion with the domain experts of the customer call logs, it is known that ‘Whether a call is an interoperability related call, it must be derived from searching certain keywords in the mentioned fields. If any of the fields contain one of the given keywords, it is an interoperability problem. The keywords are: interoperability, io, dicom, hl7, pacs, ris, print, query’. Figure 9 illustrates the interpretation of the five customer call fields within this product development task.

4.1.5. Step 5: Integrate Relevant PUI

For the integration, this application scenario used the Semantic Mediator, i.e., the social media wrapper [47]. With knowledge from the previous steps, the social media wrapper is configured to extract information from identified fields in the customer call logs for satisfying the defined information needs. The following texts illustrate how outcomes from the previous steps (shown in Figure 9) guide the interpretation and integration of customer call logs for the ‘TypeOfProblem’ in the information needs.

The system architecture of the social media wrapper approach is shown in Figure 10. It follows three main steps to transform the customer call logs into specific domain ontology: (1) data acquisition, (2) text pre-processing, and (3) semantic transformation. The customer call logs related to the design task are accessed in the data acquisition phase. Afterward, in the text pre-processing module, common Natural Language Processing (NLP) text

pre-processing methods, e.g., tokenizing, POS Tagging, are applied on the unstructured texts (e.g., text in the field ‘Customer Complaint’) to provide a basis for further text analysis. In the last step, different pieces of information are extracted based on the given ontology representing information needs (from step 2) and the captured knowledge for interpretation and integration (from steps 3 and 4). In this approach, the text processing and information extraction are based on the NLP tool: GATE (<https://gate.ac.uk/> accessed on 29 March 2022). The extracted pieces of information are further associated together to populate ontology ABoxes and finally integrated into the product design knowledge base, implemented as Triple-Store, to satisfy the defined information needs. The product design-related information extracted from the customer call logs can also be additionally integrated with that from other data sources with the help of the given ontology and the Mediator for Product Design Data Federation module. For example, in case sensor data for the elicitation of interoperability problems is available, the interoperability problems derived from sensor data can be integrated with the information extracted from the customer call logs through the Mediator for Product Design Data Federation module.

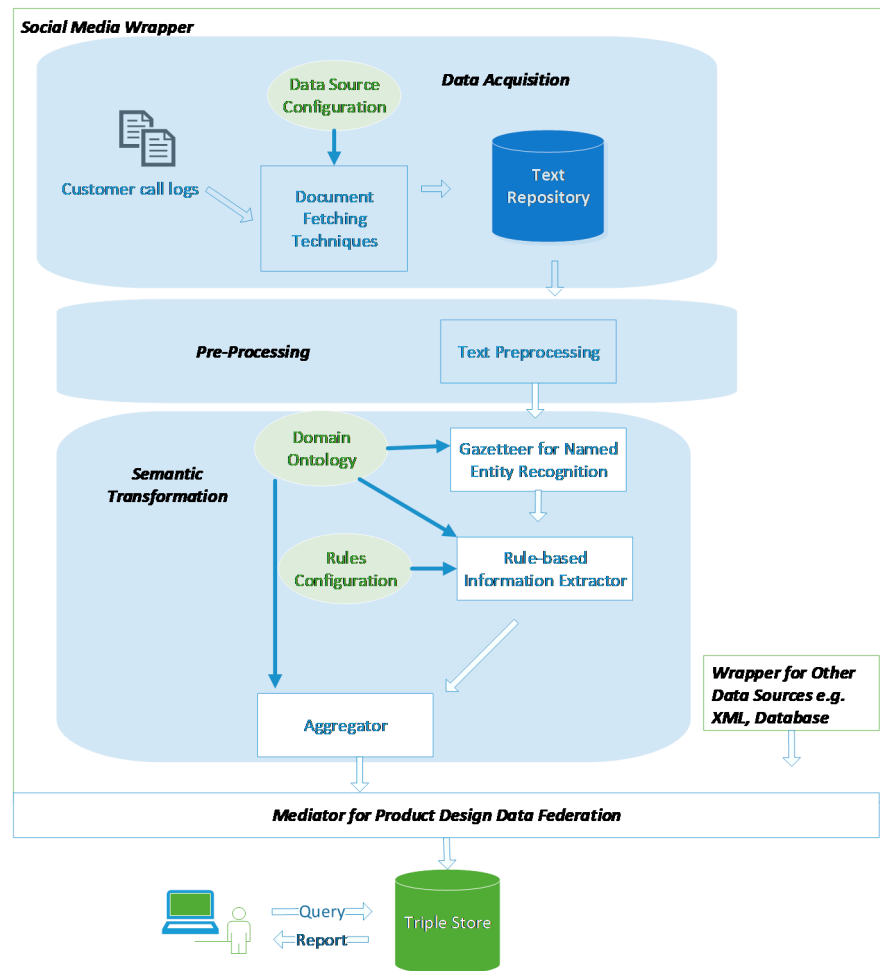


Figure 10. Social Media Wrapper Architecture.

The semantic transformation process can be flexibly guided by specifying an ontology and a configuration file that specifies how to extract semantic information from the data sources, e.g., the customer call logs, and mapping them to the given information needs ontology, according to the outcomes from previous steps. The configuration file lists the formal ontology representing the information needs and a semantic mapping file. The semantic mapping file specifies the information extraction techniques for establishing the

semantic mapping from the PUI sources, i.e., customer call logs to the information needs ontology, according to the knowledge for interpretation and integration from the last step.

A sample snippet of the semantic mapping file is shown in Figure 11. The sample semantic mapping has mainly applied the gazetteer-based named entity recognition and Java Annotation Patterns Engine (JAPE)-based linguistic rules. The gazetteer lists are used to find occurrences of entities in text. The linguistic rules technique concerns the extraction of specific information from text using linguistic rule matching technique. The rules are in general based on regular expression combining with the information from Tokenizer, POS tags, and Gazetteer, etc. The sample semantic mapping above gives rules to map and extract information for the ‘TypeOfProblem’ in the information needs, according to the outcomes shown in Figure 9. The gazetteer list, ‘issue.lst’, has listed all possible categories of issues and related keywords (Figure 12). For example, the keywords ‘io’, ‘dicom’, and ‘interoperability’ are related to the interoperability problem. With the help of linguistic rules specified in the JAPE file ‘Issue.jape’ (Figure 13), once the interoperability-related keywords are detected in the customer call fields (i.e., ‘customer complaint subject’, ‘customer complaint’, ‘external engineer text’, ‘internal engineer text’, and ‘subsystem’), the value ‘Interoperability’ is assigned to the ‘TypeOfProblem’ in the information needs. Similarly, the other required information such as ‘appearedInCountry’, ‘appearedInRegion’, and ‘belongsToProductGroup’ can be extracted from the customer call logs. At the end, we can obtain a knowledge base with a list of required information extracted from the customer call logs, as shown in Figure 14.

```
<!--data property of the class Help_Desk-->
<DatatypePropertyMapping propertyName="TypeOfProblem"
  Unit="unknown">
  <SubClass>Help_Desk</SubClass>
  <Detection>
    <Gazetteer major="Issue" minor="">issue.lst</Gazetteer>
    <JAPE>Issue.jape</JAPE>
  </Detection>
</DatatypePropertyMapping>
```

Figure 11. Example mapping for the information needs ‘typeOfProblem’.

List name	Major	Minor	Language	Annotation type	Value	Feature 1	Value 1
country_alias.lst	CountryAlias			Lookup	interoperability	category	Interoperability
issue.lst	Issue			Lookup	io	category	Interoperability
					dicom	category	Interoperability
					hl7	category	Interoperability
					pacs	category	Interoperability
					ris	category	Interoperability
					print	category	Interoperability
					query	category	Interoperability
					image quality	category	Image quality

Figure 12. A snippet of the gazetteer list ‘issue.lst’.

4.1.6. Step 6: Evaluate PUI Provision

After the integration, the product manager used the resulted dataset to analyze the distribution and major sources of interoperability problems. A straightforward statistical analysis can be carried out to find the distribution of interoperability-related customer calls over countries/regions/product groups. In the first step, the product manager selects

all customer call records related to the type of problem “Interoperability”; after that, the product manager groups the records by “appearedInRegion” to obtain an answer for their design activity with the “appearedInRegion” related statistic. For example, the product manager may obtain the result that the number of interoperability complaints from North America is much more than from the south Asia region in the last year (Figure 15). Figure 15 illustrates the example process to have a visualization of the analysis results from the customer call logs.

Phase: Issue
 Input: owlClassDomain Lookup Token CustomerComplaintSubject CustomerComplaint ExternalEngineerText InternalEngineerText SubSystem
 Options: control = appelt

```

Rule: Issue
(
  {Lookup.majorType == Issue, Lookup within CustomerComplaintSubject} |
  {Lookup.majorType == Issue, Lookup within CustomerComplaint} |
  {Lookup.majorType == Issue, Lookup within ExternalEngineerText} |
  {Lookup.majorType == Issue, Lookup within InternalEngineerText} |
  {Lookup.majorType == Issue, Lookup within SubSystem}
):issue
-->
{
  System.out.println("Issue property");

  AnnotationSet issueAs = bindings.get("issue");
  Annotation issueAnn = issueAs.iterator().next();

  // data Property: TypeOfProblem of Help_Desk
  FeatureMap helpdeskFeatures = Factory.newFeatureMap();
  helpdeskFeatures.put("classname", "Help_Desk");
  AnnotationSet helpdeskAs = inputAS.getFirstNode().getOffset(),
                          inputAS.getLastNode().getOffset().get("owlClassDomain", helpdeskFeatures);
  Annotation helpdeskAnn = helpdeskAs.iterator().next();

  ...

  // get issues that already added
  String currentIssueCategoryName = issueAnn.getFeatures().get("category").toString();
  if (null == helpdeskAnn.getFeatures().get("existingIssues"))
  {
    helpdeskAnn.getFeatures().put("existingIssues", "");
  }
  String existingIssueCategoryNames = helpdeskAnn.getFeatures().get("existingIssues").toString();

  // if the issue is new for this helpdeskAnn, then add the issue into outputAS as owlDataTypeProperty
  if (!existingIssueCategoryNames.contains(currentIssueCategoryName))
  {
    FeatureMap issueFeature = Factory.newFeatureMap();
    issueFeature.put("propertyName", "TypeOfProblem");
    issueFeature.put("propertyValue", currentIssueCategoryName);
    issueFeature.put("domainId", helpdeskAnn.getFeatures().get("representationId"));
    issueFeature.put("kind", "DatatypeProperty");
    outputAS.add(issueAs.getFirstNode(), issueAs.getLastNode(), "owlDataTypeProperty", issueFeature);

    // add new issue category to the helpdeskAnn
    helpdeskAnn.getFeatures().put("existingIssues", existingIssueCategoryNames + ";" + currentIssueCategoryName);
  }
}
    
```

Figure 13. A snippet of the JAPE file “Issue.jape”.

Help_Desk	belongsToProductGroup	appearedInCountry	appearedInRegion	TypeOfProblem
healthcare:GCS44032023	"CV Integris Cardio mono-plane"	"China"	"East Asia & Pacific"	"Interoperability"

Figure 14. Example information extracted from customer call logs.

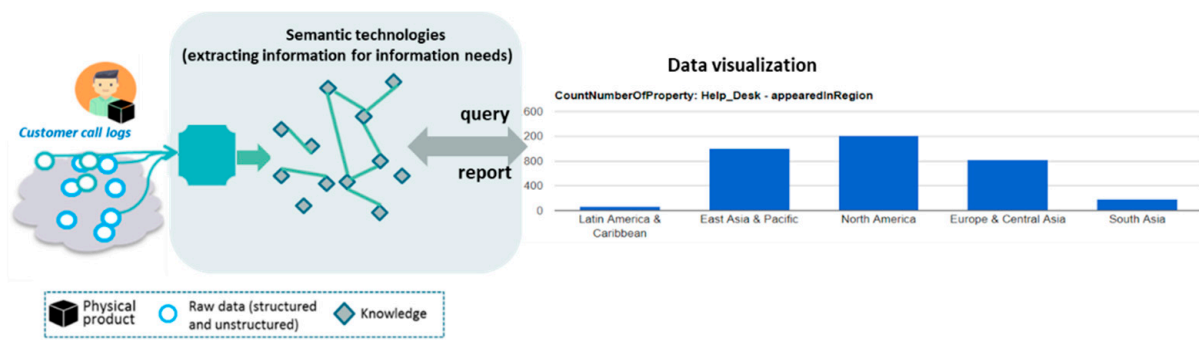


Figure 15. Example process from data to knowledge for the design activity.

The prepared PUI dataset can support the product development tasks as expected. Accordingly, the product manager can provide evaluations and comments on the PUI provision process.

4.2. Product Development Scenario: Boat Design

The second application scenario is about boat development. The company is a boat manufacturer. On the development of a new boat, the real operational conditions are often not clear with detail. Many decisions are based on experiences, but product design based on experiences and subjective judgements from key persons often results in non-optimized decisions, such as under- or over-engineering. In the EU funded research project, the company plans to capture and take advantage of product usage information in the product development process. In the project, the focused product is a high-speed patrol boat (as shown in Figure 6). It is designed for high-speed operation with a high level of safety and functionality. The product was still at the early stage of its development. The company had a prototype and collected sensor data from test runs of the prototype. For product developers, one of the main focuses of this project was to improve the boat design through exploring real-life product usage information and hydrodynamic simulations. This study takes one of these product development scenarios to demonstrate how the proposed procedure can be used in practice.

4.2.1. Step 1: Define context for PUI utilization

The use case is about the development of a high-speed patrol boat. The boat consists of many subsystems and components, such as ship equipment, drive systems and hull. One goal of this project is to reduce the hull cost with the use of materials based on the optimization of laminates, but without loss of stability. This product development task focuses on the hull of the vessel. It is to understand the response of the vessel’s hull, i.e., hull deflection, during high-speed maneuvers in various usage conditions. Meanwhile, an additional concern of this task is about understanding the impact of strong waves on the vessel’s hull. For clarity, the following steps will focus on the first main concern. The usage information will also be used in later product development stages to improve the simulations about the response of the vessel’s hull during high-speed maneuvers. Table 4 summarizes the key information in the activity category and personnel category.

Table 4. Boat scenario: summary of the elements in the activity category and personnel category.

Elements	Values
Activity Category	
Product design object	hull body
Design activity	understand the response of hull body, i.e., hull deflection, during high-speed maneuvers in various usage conditions
Goal	Short-term: (1) understand hull deflection in use conditions; (2) understand the impact of waves on the hull
Product feature	Long term: reduce the hull cost with the use of materials based on the optimization of laminates, but without loss of stability cost; stability; materials

Table 4. Cont.

Elements	Values
Personnel Category	
Developer Role	Developer A Product Developer
Intention of PUI Usage	Various sensor data from test runs of the boat prototype could be helpful in understanding (1) the hull deflection during high-speed maneuvers in various usage conditions and (2) the wave forces and impacts on the hull. This could improve the simulation supported virtual sea trials to save product development costs.

Usage information about this product is mainly from the following originators:

- Boat's on-board systems. Systems such as NMEA (National Marine Electronics Association) networks provide various vessel data. They include, for example, geographical location of the vessel, engine speed, engine temperature, fuel rate, engine boost pressure, oil pressure, and more. The vessel data provides a good overview of the boat's information.
- Dedicated sensors. There are various sensors installed in the boat prototype. It is possible to collect data from these dedicated sensors, including, for example, accelerometer sensors and laser deflection sensors. The magnetometer, accelerometer, and gyroscope sensors installed in different locations of the boat hull are for advanced acceleration and orientation data from the vessel, and the laser deflection sensor is to measure the displacement of the vessel floor, i.e., the deflection of the hull panel, with the water waves during high-speed patrolling operations. This is to provide dedicated information about specific parts of the vessel, such as the hull. This kind of information is important from the perspective of design improvement on specific parts, for instance, understanding how the hull responds during vessel sea trials/voyages.
- On-board weather station. A weather station is installed in the boat prototype to capture meteorological data. The meteorological data cover: true wind speed, true wind direction, apparent wind speed, apparent wind direction, air temperature, barometric pressure, relative humidity, global positioning system (GPS), and timestamp from GPS.
- Third-party services. Third-party services can provide additional Meteorological data. They include additional wave information in the sea trials area, such as wave height and wave direction.

The company used a sensor gateway to pre-process the IoT data from various information originators and devices, e.g., on-board systems, dedicated sensors, and on-board weather stations. After pre-processing, the gateway stored the information in a company-owned time-series database, "HydroliftLIN-COLN.autogen". The dataset "Wavemodel" with Meteorological data about, for example, wave height and direction, is managed by third-party services.

4.2.2. Step 2: Capture Information Needs on PUI

In this product development task, the developer needs two types of usage information. The first type is relevant to usage context, about the real usage conditions of the boat, and the second type is about the respective product behaviors, for example, speed and hull deflection. As a result, the PUI profile can be summarized as "we need usage information of the patrol boat about usage conditions and respective product behaviors including hull deflection and speed etc.". The main concern of this task is to understand the response of the vessel's hull, i.e., hull deflection, during high-speed maneuvers in various usage conditions. To address this concern, it requires information about usage conditions and product behaviors to support analysis and future simulations. With the existing PUI data sources, the usage condition considers mostly the weather conditions, i.e., the wave and wind information; and the product behaviors consider mostly the vessel's heading, acceleration, speed, and hull deflection, etc. Regarding this concern, we can detail the information needs in a PUI view as "simulation view about hull behavior on usage conditions", with

the description “We need following information for analyzing hull response: (1) weather conditions during high-speed operations, such as wave and wind information; (2) boat behaviours including heading, acceleration, speed, and hull deflection”. For simulation and data analysis, it was decided to integrate the data sources into a dataset in tabular format with the following required fields: timestamp, true wind speed, true wind direction, apparent wind speed, apparent wind direction, significant wave height, mean wave direction, mean wave period, vessel speed over water, vessel speed over ground, hull deflection, vessel heading, and accelerations and orientation measured on the hull, i.e., Accelerometer X, Accelerometer Y, Accelerometer Z, Roll, Pitch, Yaw (Figure 16).

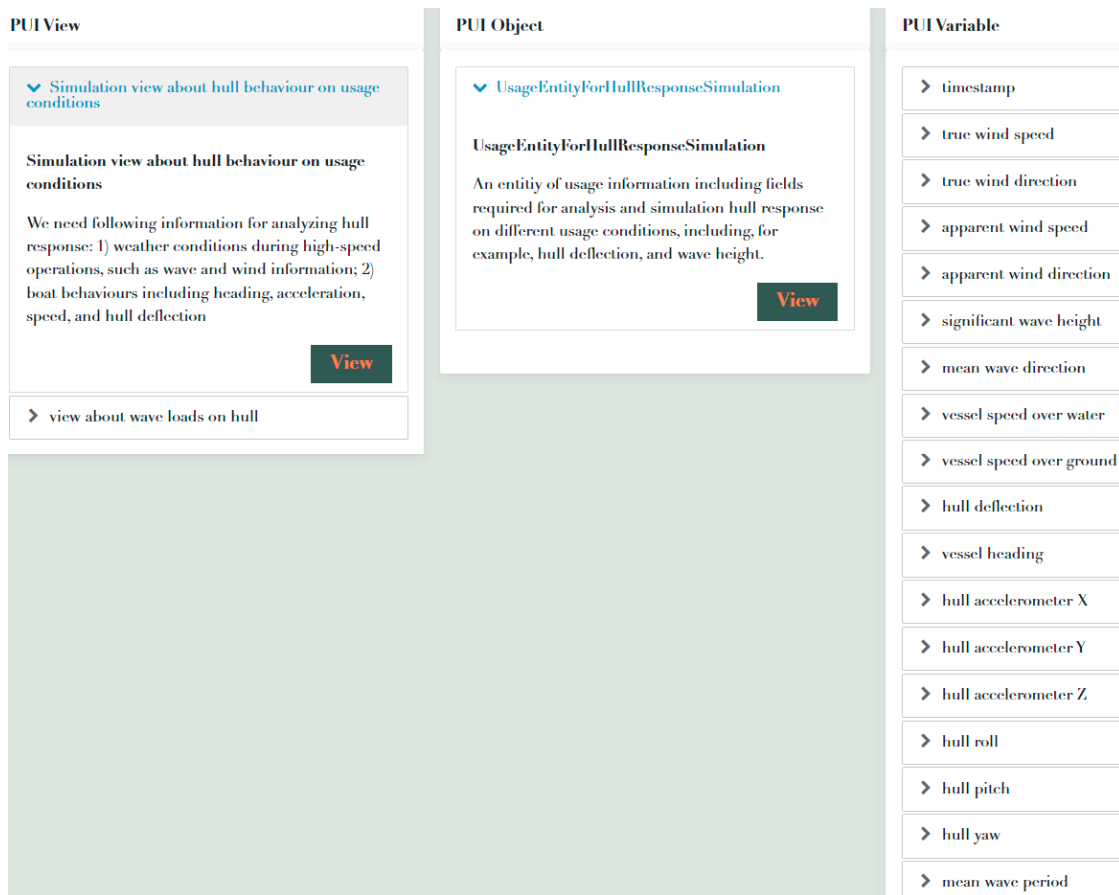


Figure 16. Information needs addressing the main concern of the task.

4.2.3. Step 3: Identify Relevant PUI

This step is to identify the PUI relevant to the specified information needs from existing information sources. It was determined that both the third-party PUI dataset “Wavemodel” and the company-owned dataset “HydroliftLINCOLN.autogen” are relevant. The dataset, “Wavemodel”, with meteorological data, can partially satisfy the information needs with wave relevant information, such as “mean wave direction”, and the dataset, “Hydro-liftLINCOLN.autogen”, can satisfy the information needs partially with vessel related information, such as “speed over ground” and “hull deflection”.

4.2.4. Step 4: Specify PUI Interpretation and Integration

This step helps specify how to interpret and integrate the relevant PUI to meet the defined information needs. In this scenario, the IoT data from different measurements need to be filtered, aligned, and integrated according to various fields, such as the timestamp. For example, the timestamp and GPS information of the boat is required to filter and integrate “mean wave direction”, which is from the dataset “Wavemodel”.

4.2.5. Step 5: Integrate Relevant PUI

Following the last steps, an individual approach was developed. It synchronized multiple measurements according to the timestamp and integrated the identified relevant PUI into a single dataset in tabular format with the required fields.

4.2.6. Step 6: Evaluate PUI Provision

Finally, the prepared dataset is analyzed to address the concerns of the current task. Based on preliminary results, the prepared dataset with relevant PUI is able to cover the information needs and support the problem solving of the current task.

5. Discussion

5.1. Challenges and Findings

To support the use of PUI in product development, it is important to isolate useless data and have a set of PUI relevant to the needs of given tasks. Without filtering and preparing task-relevant PUI, methodologies for analyzing and utilizing PUI in product development are ineffective. This article stated that current approaches supporting the identification and integration of different types of PUI for various product development tasks are insufficient. To overcome this, a systematic procedure with a context-based methodology to identify and integrate task-relevant PUI is proposed.

The proposed approach helps to address many challenges in the process of identifying and integrating PUI for product development tasks. The first challenge is to understand different product improvement tasks, i.e., the contexts of PUI utilization. Among many context models and context factors around product development, this article proposed a high-level product development task model to highlight factors directly relevant to the PUI provision in product development tasks. It helps developers be aware of their current situation, such as available PUI sources and their problems and goals. The second challenge relates to having an explicit awareness of the information needs of PUI. Considering the complexity and dynamic of product development, the information needs of PUI in a specific product development task are often not well-known or not well-understood. Engineers eventually fail to ask the right questions and demand irrelevant information for the product development [48]. Even if the information needs of PUI are in the mind of developers, it is still a challenge to express and formulate them. The presented procedure guides product developers to consider their current product development task and the context for PUI utilization, for example, about their intentions of PUI usage, and then formulate their information needs for their tasks from general to very detailed and precise. It can be expressed in detail as 'PUI object type' and 'PUI variables', similar to Classes and Properties in the world of semantic ontology. This allows developers to extract the precisely specified information and semantic integration of relevant PUI for supporting data analytics or data-driven design. Another challenge is related to the interpretation and integration of PUI. PUI is provided by different systems and stakeholders, for instance, consumers and technicians, with different viewpoints. Its relevance and meaning for different tasks are different. Accordingly, data interpretation is subjective, and the real meaning of the information and semantic interoperability is highly context- and task-dependent [7]. It is impractical to interpret and integrate PUI in the same way for all different product development tasks. The proposed approach allows developers to work together with domain experts of PUI sources to interpret PUI from perspectives corresponding to the needs of their current tasks. This kind of knowledge for integration and interpretation could be helpful for future tasks.

5.2. Alignment with Existing Studies

Wynn and Maier [49] argue that feedback influences product development in many ways and is essential in the design and development process. They develop a conceptual framework and offer a system-theoretic perspective on the design and development process in terms of feedback. Rather than from a theoretical perspective, this study is more from a

practical perspective. It intends to help developers utilizing PUI as feedback in product development. In this regard, many recent works have discussed data-driven design [5,26]. Regarding usage data, a set of works have presented the utilization of PUI for product development, for example, as shown in [50–52]. These works are quite impressive, but they are rather specific for their tasks.

To support the application of different types of PUI widely in various product development tasks, the PUI relevant to the tasks has to be acquired and prepared. Regarding data acquisition, producers can collect PUI from different stakeholders with emerging technologies, such as IoT. The framework from Voet et al. [29] also helps to capture the correct usage data, i.e., sensor data, for product improvement. Bertoni [25] argued that the priority and relevance of different data could be context-dependent and biased based on personal understanding. To address a defined problem, developers could apply different approaches and require a different set of PUI. This article focuses on identifying and integrating PUI for different product development tasks. Although many works support information seeking and recommendations for product development tasks, such as enterprise search, PLM systems, and context-based approaches, e.g., [10,11], they are mostly focused on engineering documents and knowledge items and have little relation to PUI. The contexts about product development from Gericke et al. [39] and Wilmsen et al. [38] are useful for the adaptation of design methodologies, but less relevant for the adaptation of PUI utilization, i.e., the identification and integration of task-relevant PUI. The general methodology and workflow have specified steps to prepare data for analytics. For example, the Cross-Industry Standard Process for Data Mining (CRISP-DM), which could be used for standardization of the process of usage data analysis in product development, has the steps of business understanding, data understanding, and data preparation [53,54]. Additionally, the general workflow for analytics based on PUI has a dedicated step for data preparation [6]. However, they are too generic and do not apply domain-specific understanding to detail and tailor the process for identifying and integrating various types of PUI for different product development tasks.

5.3. Implications for Practice

Our aim was to help prepare task-relevant PUI that fits different product development tasks, so that large-scale product-related information acquired during the product usage phase can be efficiently exploited in product development. The proposed procedure guides product developers to understand their current situation and context for PUI utilization, clarify their information needs of PUI, determine relevant PUI for their information needs, and then interpret and integrate the relevant PUI from their perspectives for problem solving. In practice, it can potentially pave the way to utilize different PUI widely and systematically in various product development tasks.

Concerning applying the proposed procedure in practice, it is highly recommended to have an assistance tool with a good usability level. The proposed methodology involves managing an amount of information in different steps, such as various PUI sources, product development tasks, and respective information needs. Meanwhile, applying the methodology could involve interdisciplinary collaboration between stakeholders, such as product developers, domain experts for the PUI data sources, and data scientists. For example, domain experts for PUI data sources are valuable for understanding the existing PUI sources. Data scientists are helpful in data integration and knowledge discovery for task support. Although this kind of collaboration is required in the process of using PUI for product development, the entailed considerable efforts can possibly reduce the willingness to apply the proposed methodology and the readiness of PUI usage in product development tasks. As described in the next section, future research can be conducted in the recommendations of information needs, etc., based on the context in which a developer is utilizing PUI. This will potentially save manual efforts and avoid too much collaboration with domain experts for PUI etc.

5.4. Limitations and Future Research

Several potential weaknesses and limitations have been identified during the development and evaluation of the proposed procedure. A first limitation concerns the relations between tasks in the product development process. This study supports capturing context for PUI utilization and identifying and integrating relevant PUI for each product development task. Relations between these product development tasks are, however, not well-considered. Product development tasks are dependent on each other. The tasks in different sprints or iterations for a product design object connect with each other, from problem identification and task clarification, through problem analysis, solution finding, and evaluation. Their contexts, information needs, and relevant PUI could be similar. Although this study allows the specification of related tasks such as previous, next, and sub-activities in the product development, these relations are not well formulated and used to shorten the efforts required in individual tasks. The second limitation relates to the cost–benefit analysis of PUI usage. PUI is valuable for product developers in many development tasks, for example, suggesting changes in requirements and designs with a revised understanding of the product, its users, and usage context. It requires, however, skills and efforts for, for instance, data acquisition and data analysis. In some situations, other approaches, for example, co-creation with lead users, may allow the sharing of implicit knowledge and lead to better results. The proposed approach has little support to enable developers to quickly decide whether PUI usage is plausible for their current tasks.

Although this article proposes a model to characterize the work context of developers during PUI utilization, it takes only several elements that are directly relevant to the PUI provision into focus. The PUI characteristic, which helps describe the context of PUI, is based on existing research but not explored and detailed. Future research in this area is recommended to explore the context of product development activities and the context of PUI. The ISO 9241-11 [55], with definition and detailed information about the “context of use”, could help handle the context model in this research in a more structured way. Meanwhile, the information quality of PUI is a very important topic that needs to be explored in future. Defining quality characteristics for PUI from a product development perspective is important to ensure that decisions ground on accurate, timely, and complete data.

Currently, steps in the proposed procedure are carried out manually by developers with the support of domain experts of PUI sources and data scientists etc. To save manual efforts and address the challenges described more properly, it is recommended to manage a repository of historical application scenarios. As shown in Figure 6, it is possible to manage and link the outcomes from each step in the methodology and manage the historical application scenarios about identifying and integrating PUI for product development tasks. In addition, developers can add further annotations for the historical application scenarios, for example, about their development activities. In this way, future development activities can be supported by a sort of repository of development activities with PUI utilization. Learning from historical application scenarios would improve efficiency and eventually support the development of recommendation systems that enable the prediction and automatic provision of appropriate integrated PUI for various product development tasks. It could be helpful to propose context-based recommendations for developers. This could be based on, for example, machine learning, context similarity calculation, or matching with historical application scenarios. It would allow a developer who is utilizing PUI in a product development task or context similar to the one in historical application scenarios to have hints or suggestions, for instance, while defining the information needs of PUI. For example, when a new developer works on a healthcare product to find the distribution of *hardware* problems, results from the first application scenarios could give the developer some hints on the information needs of PUI, related PUI, knowledge for PUI interpretation and integration, etc.

Another limitation is about the selected application scenarios. This study takes two application scenarios derived from research projects for demonstration and evaluation.

Although they are product development tasks from manufacturing companies, the environment and developers involved in the tasks could be quite different from the ones in daily development tasks of the companies. For example, in research projects, the companies can work together with research partners to capture required PUI with new channels, have skills from partners for knowledge discovery, and so on. Meanwhile, in the daily work of companies, the situation could be different and much more complex. For example, companies may lack the skills related to the use of big data, have more PUI sources or limited availability of PUI for development tasks, and have restrictions on data access. It would be worthwhile to evaluate the proposed approach with more generic cases or daily product development tasks from companies.

6. Conclusions

Although increasing product-related information is available after the launch of a product, the value of product usage information is not yet widely exploited systematically in product development. Whereas the literature describes the application of various data analytics methodologies to get knowledge out of PUI for product development, this study aimed to develop a systematic procedure that helps identify and integrate different types of PUI for different product development tasks. This article presents a procedure with the consideration of the context of PUI utilization to help identify developers' actual information needs and prepare relevant PUI fitting the needs of their current product development tasks. Two application scenarios were selected to demonstrate the applicability of the approach and the achieved benefits. Overall, this article contributes to a very tiny area in exploiting PUI in product development or in the paradigm of the data-driven design: to help prepare task-relevant PUI that fits different product development tasks. Meanwhile, we have identified several potential weaknesses and limitations, as described in the last section. Besides the limitations and implications for future research discussed in the last section, additional work can be conducted to integrate this procedure into established frameworks, processes, architectures, and tools that make use of PUI in product development. A future effort may also focus on complementary research areas to support companies applying PUI in product development successfully and on a large scale. It can be, for example, the planning and acquisition of required PUI, sharing PUI between partners within data markets, handling information quality and data privacy issues of PUI, and supporting companies and developers with little big data skills to discover knowledge and use PUI in product development tasks.

Author Contributions: Conceptualization, Q.D. and K.-D.T.; writing—original draft preparation, Q.D.; writing—review and editing, K.-D.T.; supervision, K.-D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the European Union's Horizon 2020 research and innovation program under grant agreements No. 869991 (LEVEL-UP), No. 727982 (LINCOLN) and No. 636868 (FALCON). The APC was funded the Open Access Publishing Fund of the University of Bremen.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Glossary/Nomenclature/Abbreviations

BOL	Beginning of Life	MRI	Magnetic Resonance Imaging
CRISPDM	Cross-Industry Standard Process for Data Mining	MRO	Maintenance, Repair, and Operations
DfX	Design for X	NLP	Natural Language Processing
EOL	End of Life	NMEA	National Marine Electronics Association
GPS	Global Positioning System	PLM	Product Lifecycle Management
HSI	Human-Sourced Information	PMD	Process Mediated Data
IoT	Internet of things	POS	Part-of-Speech
JAPE	Java Annotation Patterns Engine	PUI	Product Usage Information
MGD	Machine-Generated Data	UNECE	United Nations Economic Commission for Europe
MOL	Middle of Life		

References

- Katicic, J.; Häfner, P.; Ovtcharova, J. Methodology for Emotional Assessment of Product Design by Customers in Virtual Reality. *Presence Teleoperators Virtual Environ.* **2015**, *24*, 62–73. [CrossRef]
- Abramovici, M.; Lindner, A. Providing product use knowledge for the design of improved product generations. *CIRP Ann.* **2011**, *60*, 211–214. [CrossRef]
- Deng, Q.; Wellsandt, S.; Hribernik, K.; Thoben, K.-D. Understanding Users and Products in Product Development: The Application of Product Usage Information and its Challenges. *Proc. Des. Soc.* **2021**, *1*, 3299–3308. [CrossRef]
- Shin, J.-H.; Kiritsis, D.; Xirouchakis, P. Design modification supporting method based on product usage data in closed-loop PLM. *Int. J. Comput. Integr. Manuf.* **2014**, *28*, 551–568. [CrossRef]
- Briard, T.; Jean, C.; Aoussat, A.; Véron, P.; Le Cardinal, J.; Wartzack, S. Data-Driven Design Challenges in the Early Stages of the Product Development Process. *Proc. Des. Soc.* **2021**, *1*, 851–860. [CrossRef]
- Klein, P.; van der Vegte, W.F.; Hribernik, K.; Klaus-Dieter, T. Towards an Approach Integrating Various Levels of Data Analytics to Exploit Product-Usage Information in Product Development. *Proc. Des. Soc. Int. Conf. Eng. Des.* **2019**, *1*, 2627–2636. [CrossRef]
- El Kadiri, S.; Grabot, B.; Thoben, K.-D.; Hribernik, K.; Emmanouilidis, C.; von Cieminski, G.; Kiritsis, D. Current trends on ICT technologies for enterprise information systems. *Comput. Ind.* **2016**, *79*, 14–33. [CrossRef]
- Qin, H.; Wang, H.; Johnson, A. Understanding the information needs and information-seeking behaviours of new-generation engineering designers for effective knowledge management. *Aslib J. Inf. Manag.* **2020**, *72*, 853–868. [CrossRef]
- Leckie, G.J.; Pettigrew, K.E.; Sylvain, C. Modeling the Information Seeking of Professionals: A General Model Derived from Research on Engineers, Health Care Professionals, and Lawyers. *Libr. Q.* **1996**, *66*, 161–193. [CrossRef]
- Mehrpoor, M. An Ontology-Driven Recommender System for Engineering Projects, Doctoral Thesis, Norwegian University of Science and Technology, Norwegian. 2018. Available online: <https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/11250/2560714/5/Mahsa%20Mehrpoor.pdf> (accessed on 29 March 2022).
- Redon, R.; Larsson, A.; Leblond, R.; Longueville, B. VIVACE Context Based Search Platform. In *International and Interdisciplinary Conference on Modeling and Using Context*; Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 397–410. [CrossRef]
- Ulrich, K.T.; Eppinger, S.D. *Product Design and Development*, 5th ed.; McGraw-Hill Irwin: New York, NY, USA, 2012.
- Wellsandt, S.; Hribernik, K.; Thoben, K.-D. Sources and Characteristics of Information about Product Use. *Procedia CIRP* **2015**, *36*, 242–247. [CrossRef]
- Edler, A. Nutzung von Felddaten in der qualitätsgetriebenen Produktentwicklung und im Service. Doctoral Thesis, Technische Universität Berlin, Berlin, Germany, 2001.
- Jagtap, S.; Johnson, A. In-service information required by engineering designers. *Res. Eng. Des.* **2011**, *22*, 207–221. [CrossRef]
- Petkova, V. An Analysis of Field Feedback in Consumer Electronics Industry. Doctoral Thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 2003.
- United Nations Economic Commission for Europe (UNECE), Classification of Types of Big Data Last Modified. Available online: <https://statswiki.unece.org/display/bigdata/Classification+of+Types+of+Big+Data> (accessed on 29 March 2022).
- Hribernik, K.; Franke, M.; Klein, P.; Thoben, K.-D.; Coscia, E. Towards a platform for integrating product usage information into innovative product-service design. In Proceedings of the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira, Portugal, 27–29 June 2017; pp. 1407–1413.
- Joung, J.; Jung, K.; Ko, S.; Kim, K. Customer Complaints Analysis Using Text Mining and Outcome-Driven Innovation Method for Market-Oriented Product Development. *Sustainability* **2018**, *11*, 40. [CrossRef]
- Abramovici, M.; Lindner, A. Knowledge-based decision support for the improvement of standard products. *CIRP Ann.* **2013**, *62*, 159–162. [CrossRef]
- Abramovici, M.; Gebus, P.; Göbel, J.C.; Savarino, P. Utilizing unstructured feedback data from MRO reports for the continuous improvement of standard products. In Proceedings of the 21st International Conference on Engineering Design (ICED17), Design Information and Knowledge, Vancouver, WA, Canada, 21–25 August 2017; Volume 6.
- Alkahtani, M.; Choudhary, A.; De, A.; Harding, J.A. A decision support system based on ontology and data mining to improve design using warranty data. *Comput. Ind. Eng.* **2019**, *128*, 1027–1039. [CrossRef]

23. Van der Vegte, W.F.; Kurt, F.; Şengöz, O.K. Simulations Based on Product-Usage Information from Connected Products to Support Redesign for Improved Performance: Exploration of Practical Application to Domestic Fridge-Freezers. *J. Comput. Inf. Sci. Eng.* **2019**, *19*. [[CrossRef](#)]
24. Stietencron, M.; von Hribernik, K.A.; Røstad, C.C.; Henriksen, B.; Thoben, K.-D. Applying closed-loop product lifecycle management to enable fact based design of boats. In Proceedings of the 14th IFIP International Conference on Product Lifecycle Man-agement (PLM), Seville, Spain, 10–12 July 2017; pp. 522–531.
25. Bertoni, A. Role and Challenges of Data-Driven Design in the Product Innovation Process. *IFAC-PapersOnLine* **2018**, *51*, 1107–1112. [[CrossRef](#)]
26. Cantamessa, M.; Montagna, F.; Altavilla, S.; Casagrande-Seretti, A. Data-driven design: The new challenges of digitalization on product design and development. *Des. Sci.* **2020**, *6*, e27. [[CrossRef](#)]
27. Verganti, R.; Vendraminelli, L.; Iansiti, M. Innovation and Design in the Age of Artificial Intelligence. *J. Prod. Innov. Manag.* **2020**, *37*, 212–227. [[CrossRef](#)]
28. Van der Vegte, W.F.; Kurt, F.; Şengöz, O.K. Simulation of Product Performance Based on Real Product-Usage In-formation: First Results of Practical Application to Domestic Refrigerators. In Proceedings of the ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec, QC, Canada, 26–29 August 2018.
29. Voet, H.; Altenhof, M.; Ellerich, M.; Schmitt, R.H.; Linke, B. A Framework for the Capture and Analysis of Product Usage Data for Continuous Product Improvement. *J. Manuf. Sci. Eng.* **2018**, *141*. [[CrossRef](#)]
30. Kruschwitz, U.; Hull, C. Searching the Enterprise. *Found. Trends Inf. Retr.* **2017**, *11*, 1–142. [[CrossRef](#)]
31. Bazire, M.; Brézillon, P. Understanding Context before Using It. In *International and Interdisciplinary Conference on Modeling and Using Context*; Dey, A., Kokinov, B., Leake, D., Turner, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3554, pp. 29–40. [[CrossRef](#)]
32. Liu, W.; Li, X.; Huang, D. A survey on context awareness. In Proceedings of the 2011 International Conference on Computer Science and Service System, (CSSS 2011), Nanjing, China, 27–29 June 2011; pp. 144–147.
33. Edmonds, B. The Pragmatic Roots of Context. In Proceedings of the Modeling and Using Context: Second International and Interdisciplinary Conference, CONTEXT 1999, Trento, Italy, 9–11 September 1999.
34. Merriam, W. Definition of CONTEXT. 2022. Available online: <https://www.merriam-webster.com/dictionary/context> (accessed on 6 May 2022).
35. Dey, A.K. Understanding and Using Context. *Pers. Ubiquitous Comput.* **2001**, *5*, 4–7. [[CrossRef](#)]
36. Mehrpoor, M.; Ahlers, D.; Gulla, J.A.; Kristensen, K.; Sivertsen, O.I. Investigating contextual ontologies and document corpus characteristics for information access in engineering settings. *J. Inf. Technol. Case Appl. Res.* **2017**, *32*, 1–24. [[CrossRef](#)]
37. Liu, T.; Wang, H.; He, Y. Intelligent knowledge recommending approach for new product development based on workflow context matching. *Concurr. Eng.* **2016**, *24*, 318–329. [[CrossRef](#)]
38. Wilmsen, M.; Dühr, K.; Albers, A. A context-model for adapting design processes and methods. *Procedia CIRP* **2019**, *84*, 428–433. [[CrossRef](#)]
39. Gericke, K.; Meißner, M.; Paetzold, K. Understanding the Context of Product Development. In Proceedings of the 19th International Conference on Engineering Design (ICED13): Design for harmonies, Seoul, Korea, 19–22 August 2013; pp. 191–200.
40. Nadoveza, D.; Kiritsis, D. Ontology-based approach for context modeling in enterprise applications. *Comput. Ind.* **2014**, *65*, 1218–1231. [[CrossRef](#)]
41. Nadoveza, D.; Kiritsis, D. Concept for Context-Aware Manufacturing Dashboard Applications. *IFAC Proc. Vol.* **2013**, *46*, 204–209. [[CrossRef](#)]
42. Wellsandt, S.; Wuest, T.; Hribernik, K.; Thoben, K.-D. Information Quality in PLM: A Product Design Perspective. In Proceedings of the IFIP International Conference on Advances in Production Management Systems (APMS), Tokyo, Japan, 5–9 September 2015; pp. 515–523. [[CrossRef](#)]
43. Wellsandt, S.; Hribernik, K.; Thoben, K.-D. Content analysis of product usage information from embedded sensors and web 2.0 sources: A first analysis of practical examples. In Proceedings of the 2015 IEEE International Conference on Engineering, Technology and Innovation/International Technology Management Conference (ICE/ITMC), Belfast, UK, 22–24 June 2015; pp. 1–9. [[CrossRef](#)]
44. Weber, F. Formale Interaktionsanalyse: Ein Beitrag zur systematischen Gestaltung von Informations- und Kommunikationsstrukturen im Concurrent Enterprise durch die Berücksichtigung von Informationseigenschaften. Doctoral Thesis, University of Bremen, Bremen, Germany, 2005.
45. Henicke, S. What is the Real Question? An Empirical-Ontological Approach to the Interpretative Analysis of Ar-chival Reference Questions. Doctoral Thesis, Humboldt-Universität Zu Berlin, Berlin, Germany, 2017. [[CrossRef](#)]
46. Taylor, R.S. Question-Negotiation and Information Seeking in Libraries. *Coll. Res. Libr.* **1968**, *29*, 178–194. [[CrossRef](#)]
47. Deng, Q.; Franke, M.; Hribernik, K.; Thoben, K.-D. Exploring the integration of social media feedback for us-er-oriented product development. In Proceedings of the 21st International Conference on Engineering Design (ICED17), Design Methods and Tools, Vancouver, WA, Canada, 21–25 August 2017; Volume 4.
48. Höhn, M.; Hollauer, C.; Wilberg, J.; Kammerl, D.; Mörtl, M. Investigating usage data support in develop-ment processes-A case study. In Proceedings of the 21st International Conference on Engineering Design (ICED17), Design Theory and Research Methodology, Vancouver, WA, Canada, 21–25 August 2017; Volume 7.

49. Wynn, D.C.; Maier, A.M. Feedback systems in the design and development process. *Res. Eng. Des.* **2022**, *33*. [[CrossRef](#)]
50. Kpiebaareh, M.Y.; Wu, W.-P.; Agyemang, B.; Haruna, C.R.; Lawrence, T. A Generic Graph-Based Method for Flexible Aspect-Opinion Analysis of Complex Product Customer Feedback. *Information* **2022**, *13*, 118. [[CrossRef](#)]
51. Jin, J.; Liu, Y.; Ji, P.; Kwong, C.K. Review on Recent Advances in Information Mining from Big Consumer Opinion Data for Product Design. *J. Comput. Inf. Sci. Eng.* **2018**, *19*, 010801. [[CrossRef](#)]
52. Wang, Z.; Chen, C.-H.; Zheng, P.; Li, X.; Khoo, L.P. A novel data-driven graph-based requirement elicitation framework in the smart product-service system context. *Adv. Eng. Informatics* **2019**, *42*, 100983. [[CrossRef](#)]
53. Wirth, R.; Hipp, J. Crisp-dm: Towards a standard process model for data mining. In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, Manchester, UK, 11–13 April 2000; pp. 29–39.
54. Gholamzadeh, N.E.; Thoben, K.-D. On Applicability of Big Data Analytics in the Closed-Loop Product Lifecycle: Integration of CRISP-DM Standard. In *IFIP International Conference on Product Lifecycle Management*; Harik, R., Rivest, L., Bernard, A., Eynard, B., Bouras, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 457–467.
55. ISO/IEC, ISO 9241-11:2018: Ergonomics of Human-System Interaction—Part 11: Usability: Definitions and Concepts. Available online: <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en> (accessed on 20 May 2022).