

Article

AI4PD—Towards a Standardized Interconnection of Artificial Intelligence Methods with Product Development Processes

Benjamin Gerschütz ^{*}, Stefan Goetz  and Sandro Wartzack 

Engineering Design, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Martensstr. 9,
91058 Erlangen, Germany

* Correspondence: gerschuetz@mfk.fau.de

Featured Application: AI4PD presents an ontology that relates Digital Engineering methods to product development, with an intent to identify data-driven methods for use cases that enable the support or substitution of specific tasks in product development.

Abstract: The transformation of virtual product development to Digital Engineering (DE) requires the successful integration of Digital Engineering or data-driven methods into existing product development processes. Those methods allow for the analysis and usage of existing data. However, missing knowledge about these methods, as well as their performance or limitations, is a major burden for their application, especially in small and medium-sized enterprises. In order to close this gap, this paper proposes the AI4PD ontology, linking product development processes (PD) and Digital Engineering methods (AI). This knowledge representation gives companies an overview of the available methods to support them in selecting a suitable solution for their problems. The representation of AI4PD is performed in Protégé using the W3C standard OWL syntax. The opportunities of AI4PD are shown by a use case of identifying a DE-Method for predicting manufacturing possibilities based on test data and CAD files. Furthermore, after possible problems in existing product development processes are identified, AI4PD covers the necessary knowledge for a successful method of identification and integration to transform virtual product development to Digital Engineering.

Keywords: ontology; digital engineering; artificial intelligence; product development; formalisation; knowledge reuse; knowledge representation



Citation: Gerschütz, B.; Goetz, S.; Wartzack, S. AI4PD—Towards a Standardized Interconnection of Artificial Intelligence Methods with Product Development Processes. *Appl. Sci.* **2023**, *13*, 3002. <https://doi.org/10.3390/app13053002>

Academic Editor: Dimitris Mourtzi

Received: 10 February 2023

Revised: 21 February 2023

Accepted: 24 February 2023

Published: 26 February 2023



Copyright: © 2023 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

Digitalisation, artificial intelligence (AI) and cloud computing are changing the way we work, think and act in every aspect of our lives. The last disruptive technological leap of similar magnitude was the invention of the car or electricity. Of course, this development also affects products, transforming them into cyberphysical products or Product Service Systems (PSS) [1]. Furthermore, the established virtual product development process has to change and adapt to the new situation and development goals [1]. Therefore, existing data are part of the new critical assets for business success [2]. Most companies have a large amount of complex data at their disposal [3], representing the company's knowledge over a long time. Integrating methods and tools that enable the evaluation and usage of existing data is crucial to transforming virtual product development into the new paradigm of Digital Engineering (DE). Gerschütz et al. [4] define **Digital Engineering** as the consistent use and evaluation of data generated during design, testing and operation. The evaluation is performed using data-driven methods. Therefore, there is a growing number of techniques that use data to evaluate unknown relationships. With those methods, decision problems are supported. For example, multi-objective supply chain models can help to minimise transportation cost or delivery time [5]. They can also generate or preserve corporate knowledge by making it explicit. When small and medium-sized enterprises (SMEs) try to

integrate these Digital Engineering methods, they face the problem of missing knowledge about the methods due to the lack of specialised staff focused on these new methods. This missing knowledge results in a competitive disadvantage for these mostly locally operating companies compared with large, international enterprises.

The abovementioned DE methods support or make decisions based on data [6]. They are summarised by the phrases Data Science or AI. Well-known paradigms are Machine Learning (ML) [7] and Data Mining (DM) [8], as well as their subsequent methods and algorithms. In this paper, DE methods are mainly referred to as Data Mining and Machine Learning methods since those cover most use cases of product development. The following gives a short clarification and definition of the terms. Figure 1 shows a categorisation of the relevant terms.

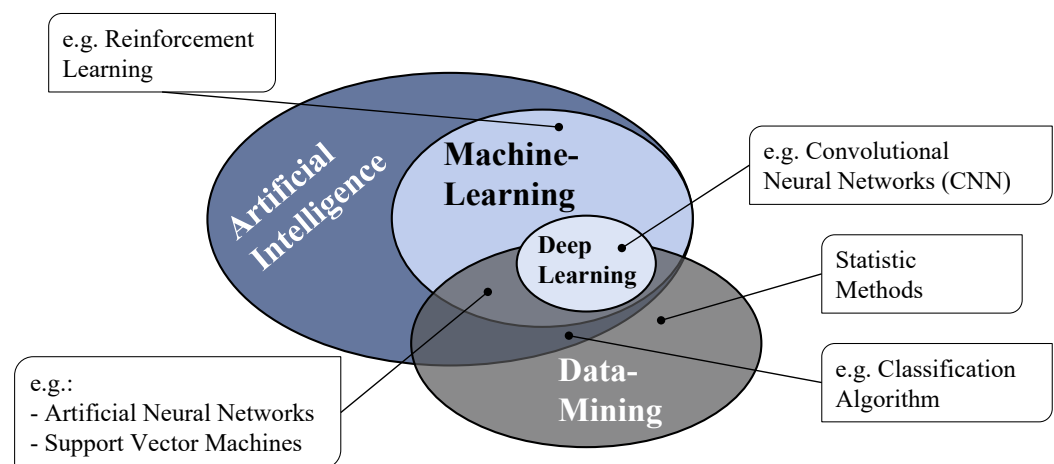


Figure 1. Categorisation of the terms Machine Learning and Data Mining according to [9].

Data Science is a research field dealing with topics and tasks that work with data, such as data collection, storage, analysis, preparation, visualisation and evaluation. Data Mining, Machine Learning and other methods, such as statistics and business research, are used. Data Science aims to extract knowledge from large amounts of data to develop management recommendations. With these recommendations, companies can optimise production or development processes and improve the quality of business decisions [10].

Artificial Intelligence (AI) aims to transfer human intellect capabilities to machines, especially computer programs. This intelligence includes, but is not limited to, the abilities of reasoning, learning, goal setting, problem-solving and adaptability. A key aspect is the provisioning for performing self-correction. Such solutions can use the facilities of human intelligence differing from the observations of living beings, for example, by a higher processing speed of large-scale computations [11,12].

Data Mining (DM) aims to identify and analyse new, previously unrecognised knowledge in data. The main use cases of DM are classification, regression, segmentation and clustering. Therefore, algorithms similar to those from Machine Learning are used [8,13].

Machine Learning (ML) is a subfield of AI that designs algorithms so that systems can learn from previous data sets through experience and optimise themselves. By ‘feeding’ the algorithms with data, the system can adapt its programming to become more efficient for specific activities and make its own decisions based on unknown data. Machine Learning applications are distinguished into supervised, unsupervised and reinforcement learning [10].

Both paradigms, DM and ML, support tasks with many different objectives. An in-depth analysis of all available Digital Engineering methods and tools, see Figure 2, would go beyond the scope of this article. Both share methods and algorithms but differ in scope—just evaluation (in the case of data mining) or trying to predict something based on known data and unknown inputs (in the case of Machine Learning).

Data Mining							
Classification/ Regression		Segmenting/ Clustering		Correlation Analysis	Outlier Detection	Prognosis	
K-nearest Neighbour	Principal component analysis	K-means	Linkage	Apriori- Algorithm	Local Outlier Factor	Time series analysis	
Support Vector Machine	Case-Based- Reasoning	Fuzzy-c-means Algorithmus	Ward-procedure	Frequent Pattern Growth Algorithmus	Isolation Forest	...	
Probabilistic Regression	Extrem Learning Machine	DBSCAN/ OPTICS	Extrem Learning Machine	...	Elliptic envelope	Visualization	...
Decision Trees	Gradient Boosting	Expectation maximization	Gradient Boosting		One-Class SVM	Self-Organizing Maps	Monte Carlo methods
Linear Regression	Similar Nearest Neighbours	Maximum Margin Clustering	Similar Nearest Neighbours			...	Temporal Difference Learning
Discriminant analysis	...		...				Policy Search
Supervised Learning		Unsupervised Learning					Reinforcement Learning
Machine Learning							

Figure 2. Overview of Data Mining and Machine Learning methods.

Deep Learning is a subfield of ML and uses neural networks to analyse large amounts of data. The Deep Learning approach works analogously to the human brain. First, data are extracted and then analysed; finally, findings or predictions are made. The word ‘deep’ in this context refers to several hidden layers of neural networks [14]. Compared with classical Machine Learning, the main advantage of Deep Learning is that it can extract high-level features automatically from complex data without being developed by humans [10].

For the structuring of Data Mining and Machine Learning projects, procedure models such as the CRoss-Industry Standard Process for Data Mining (CRISP-DM) [15] or Knowledge Discovery in Databases (KDD) [8] can be used [16]. CRISP-DM focuses on the industrial use of data-driven methods and includes a domain-specific view. Figure 3 gives a process overview.

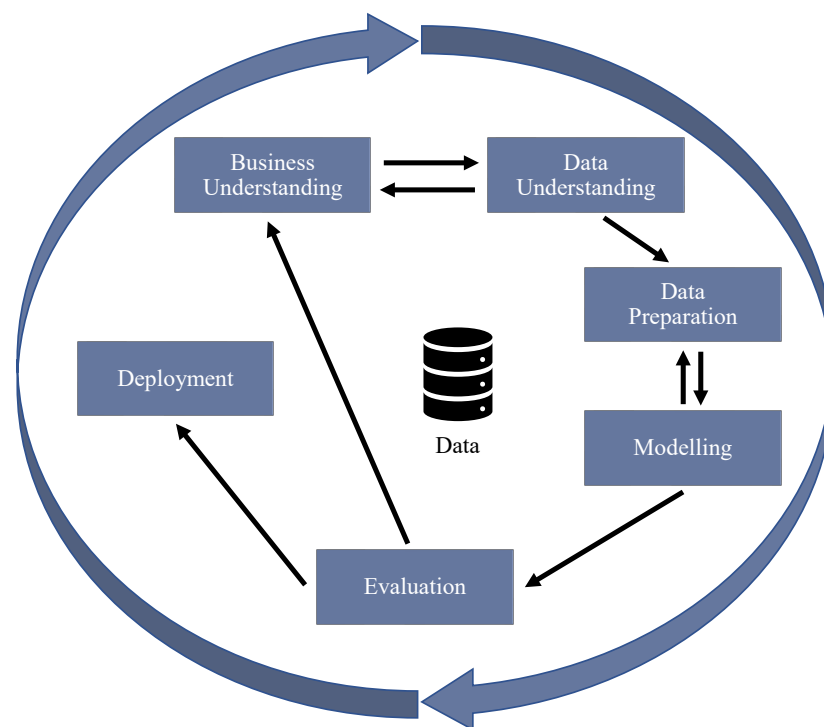


Figure 3. CRISP-DM process following [15].

Digital Engineering toolboxes are necessary for the application of Data Mining and Machine Learning methods. A toolbox describes a collection of different tools within the same context. A **tool** is an action, operation or object that supports or facilitates other actions, operations or objects. Such tools implement algorithms such as k-nearest neighbour, depicted in white boxes in Figure 2 [17].

An initial market overview of software for Data Mining and Machine Learning is given by Gartner, Inc. [18]. The authors classified 20 toolbox vendors into four categories based on their applicability and completeness of vision: market leaders, followers, visionaries and niche players. It is essential to keep in mind that the market overview mainly shows commercial toolboxes. In addition, there is also open-source software that is equally powerful. Data Mining and Machine Learning software include stand-alone products and a complementary selection of packages, components, libraries or frameworks. For the broad integration of data-driven methods into existing product development processes, there is a need for specialised employees. Especially, method knowledge—e.g., concerning potentials, weaknesses and prerequisites [19]—is necessary [20]. Moreover, given the continuously expanding number of available toolboxes for Data Mining and Machine Learning, choosing the right software is becoming increasingly challenging [21]. Therefore, introducing methods often fails because selecting methods is complicated, since few best

practices are publicly available. Furthermore, possible fields of application are unknown to the companies.

In engineering design, the use of methodological development processes is well known. Therefore, several different models are available [22,23]. These models focus on the mechanical product development process and do not consider electronics or software. Therefore, newer models focus on mechatronic product development processes [24,25]. In general, product development processes describe the procedure of continuous problem solving, starting with unsolved requirements and leading to the final product [1]. Most of the product development processes are milestone-driven, with an eminent focus on time-frames and deliverables [1]. Although there are several theoretical methods for product development, the industrial practice may differ from those. Nevertheless, the basic ideas remain the same. Furthermore, basic tasks and concepts can be found in all development processes, regardless of the individual company or developed product.

Besides the process itself, the available data and the used methods, the Digital Engineering team has a major influence on the subsequent success. Especially, method-specific knowledge in a team is identified as a crucial factor [26].

An ontology could help by linking the area of product development and the existing fields of application with Digital Engineering methods. An ontology is defined as an ‘explicit specification of a conceptualisation’ [27], providing domain-relevant concepts and other relevant objects and relations between them. The power of those ontologies has been shown earlier in different, mainly experiment-intensive use cases such as tribology [28] or gene-bases [29]. Although initial approaches towards a Digital Engineering ontology exist, they mainly focus on individual aspects or closed subdomains. At the beginning of the product development process, Chen et al. managed customer needs in product development by an ontology-based structure [30]. During the following part and assembly design, collaboration became more and more relevant in the changing work environments. Therefore, an ontology-supported framework can resolve assembly constraints and lead to a standardisation of design [31]. The next step in the development process would be the simulation-based validation of designs. Since simulation experts are rare, automation would be helpful. Therefore, the ontology presented by Kestel et al. [32] provides a structured vocabulary for the interconnection of simulation knowledge obtained through text mining and building simulations based on it. Since product knowledge can become obsolete, forgetting is a crucial part of knowledge management and can be supported by ontologies as well [33]. For a more general view of product development, besides individual use cases in the process, Yang et al. [34] presented an ontology representing and managing the product development process. Furthermore, for structuring data-driven methods in the context of product development, the first semantic web application exists [20]. Besides these engineering-based ontologies, descriptions of abstract processes are realised in the PROV-ontology (PROV-O) [35], and persons and teams can be made by the friends-of-a-friend (FOAF) ontology [36].

Several ontologies in the area of product development and data-driven methods exist. On the downside, they are not sufficiently connected to link the segregated areas. However, this link is needed to support the identification of suitable data-driven methods for a given task in the product development process that should be automated. Therefore, the innovation here is the establishment of this linkage, which was not available before. The central goal of this contribution is a new ontology that connects data-driven methods and tools with the product development process to support the implementation of Digital Engineering in SMEs. Therefore, the input and output data of the individual processes should be represented as well as the involved employees. Both aspects influence suitable methods and tools since they require different input data formats or enable several outputs. Furthermore, the individual knowledge of employees limits the tools that can be used, as they are needed to program or use command line tools. The presented problems and gaps in the existing studies lead to the following research question:

How can the domains of artificial intelligence and product development be linked to simplify the introduction of Digital Engineering in small and medium-sized enterprises by improving method knowledge availability?

To answer this, the remaining contribution is structured as follows: In Section 2, the functionality of AI4PD is shown after the presentation of an initial use case in Section 2.1. Afterwards, the performance and functionality of AI4PD are evaluated by two use cases in Section 3 and discussed concerning the degree of fulfilment of the given research question in Section 4. The article closes with the conclusion and outlook on future work in Section 5.

2. Materials and Methods

The aim of this section is to create a formal model of the Digital Engineering context with the relevant subconcepts. Therefore, the philosophy (Section 2.2), the purpose and scope (Section 2.3) are defined before the core concepts and terminology of the domains of product development and data-driven methods are analysed in Section 2.4. The analysis is based on previous work concerning process management in design [2], including the subsequent process analysis in cooperation with an industrial partner and the semantics of Digital Engineering methods [20] mentioned before. Combining this preliminary work, the achievement of the main objective of linking both domains is shown. Furthermore, regarding the concepts and terminology in Section 2.4, the formal definitions of the identified classes are given.

2.1. Use Case: Integrating Data-Driven Methods into Product Development Processes

In general, more attention should be paid to data in product development, for example, to gain insights into correlations that are not immediately apparent. Intensive Data Analysis has generally established itself as a new, strongly growing field of work and research, though it offers many new challenges. Powerful methods are available for this evaluation, as described above. It is essential to know the capabilities of these new methods and to identify a well-suited use case for initial implementation. Afterwards, companies can integrate these new methods into their product development process. For implementing data-driven methods, the availability of a sufficient amount of data must also be guaranteed, and it should be ensured that the industrial problem can be solved efficiently using data analysis methods. In everyday industrial practice, new data-driven techniques are rarely used. Either new methods need to be discovered or the acceptance level within the staff is low. Furthermore, employees need to be trained or the data situation needs to be improved. Additionally, companies must first identify projects with the potential for using Digital Engineering methods. Figure 4 summarises relevant aspects for identifying suitable methods.

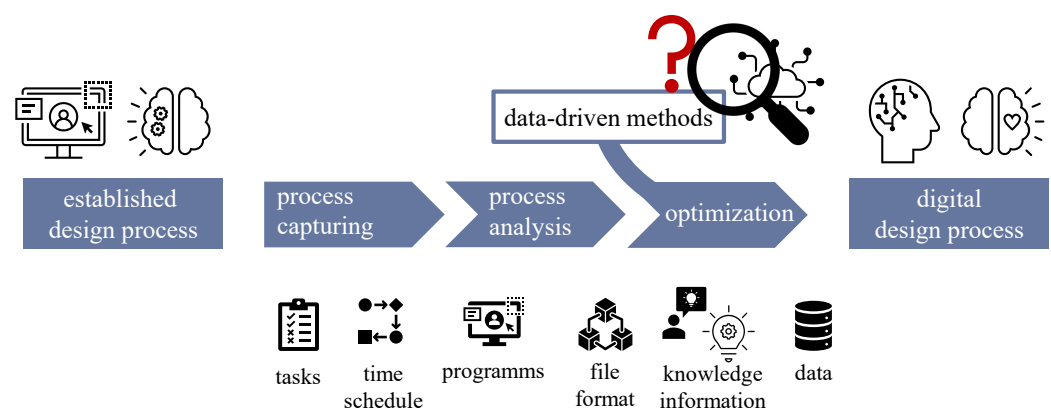


Figure 4. Process for integrating Digital Engineering methods into product development.

2.2. Philosophy of AI4PD

Companies need to know about applicable methods and use cases when integrating data-driven methods into established product development processes. At the moment, this

knowledge must be acquired with great effort from different sources. Furthermore, those sources only provide encapsulated knowledge, unlinked to use cases or best practices. This fact increases the hurdle for a successful transformation and should be reduced by the new ontology. The ontology gives companies with little or no experience in data-driven techniques, especially SMEs, an overview of these methods and tools. It is intended to facilitate and support companies in selecting a suitable technique for use in one of their product development processes. Thus, the ontology represents a collection of knowledge for data-driven techniques and toolboxes, tools and methods related to product development.

The central purpose of this ontology is the semantic and formal interconnection of the two domains of data-driven methods and product development processes. Following this purpose, the acronym AI4PD was created, which combines the acronyms for artificial intelligence (AI), a generic term for data-driven methods and product development (PD).

2.3. Purpose and Scope

According to Uschold and Gruninger [37], initial scenarios should motivate the development of an ontology. Therefore, scenarios for method identification, knowledge and use case optimisation are defined. Those scenarios have specific competency questions (CQs) for later evaluation. The following use cases and CQs define AI4PD:

1. *Use case coverage:* Avoid re-implementing since the ontology should answer queries about already implemented solutions for different problems and use cases. CQ: *Is there a solution for predicting manufacturing possibilities based on test data and CAD files?*
2. *Identification of suitable methods:* Available data-driven tools need different databases and IT infrastructure. Furthermore, they are only suitable for some tasks. The ontology should be able to answer queries about which method would fit the given prerequisites. CQ: *Which method and tool can support simulation engineers in estimating FEM results based on a CAD parameter configuration?*

2.4. Concepts and Terminology

AI4PD consists of four central concepts, namely, *People*, *Process*, *Method* and *Application*, which are specified and generalised following the middle-out approach [37]. Further, the *CamelCase* notation is used for labelling of AI4PD concepts, where *AI4PD* is used as a prefix for the Internationalized Resource Identifier (IRI). The core of AI4PD is a quartet, as shown in Figure 5, with the *TaskCluster* bridging the gap between *Process* and *Method* as well as *Data* linking *Process* and *Application*.

A *ProductDevelopmentProcess* has several phases and milestone goals, which are structured as *SubProcesses* and with a *Team* working on them. A formal definition of the *ProductDevelopmentProcess* can be given by

$$\begin{aligned} \text{ProductDevelopmentProcess} &\sqsubseteq \geq 1 \text{hasSubprocess.SubProcess} \\ &\sqsubseteq = 1 \text{doneBy.Team} \end{aligned}$$

A *SubProcess* can be decomposed down to two or more other *SubProcesses* or *Tasks*. Each *SubProcess* has defined input and output *Data* and a defined *Team*, or—in the case of a *Task*—a defined *Person* works on it. Furthermore, a *Subprocess* is part of one defined *TaskCluster*. The AI4PD concept *SubProcess* is formally defined as

$$\begin{aligned} \text{SubProcess} &\sqsubseteq = 1 \text{hasTask.Task} \vee \exists \text{hasSubprocess.SubProcess} \\ &\sqsubseteq \geq 1 \text{generatesData.Data} \\ &\sqsubseteq \geq 1 \text{UsesData.Data} \\ &\sqsubseteq = 1 \text{isPartOfProcess.ProductDevelopmentProcess} \\ &\sqsubseteq = 1 \text{isPartOf.TaskCluster} \\ &\sqsubseteq = 1 \text{doneBy.Team} \vee 1 \text{doneBy.Person} \end{aligned}$$

Subprocesses can be broken down into *Tasks*, which also require formal definitions, with a *Task* being the smallest possible indivisible unit. A *Task* is part of exactly one *SubProcess* and can use or generate the same *Data* as the *SubProcess*. Furthermore, every *Task* can be seen as an incremental *SubProcess*. Furthermore, to connect AI4PD to the generic PROV Ontology [35], the ai4pd:task class is set as equivalent to the prov:Activity class.

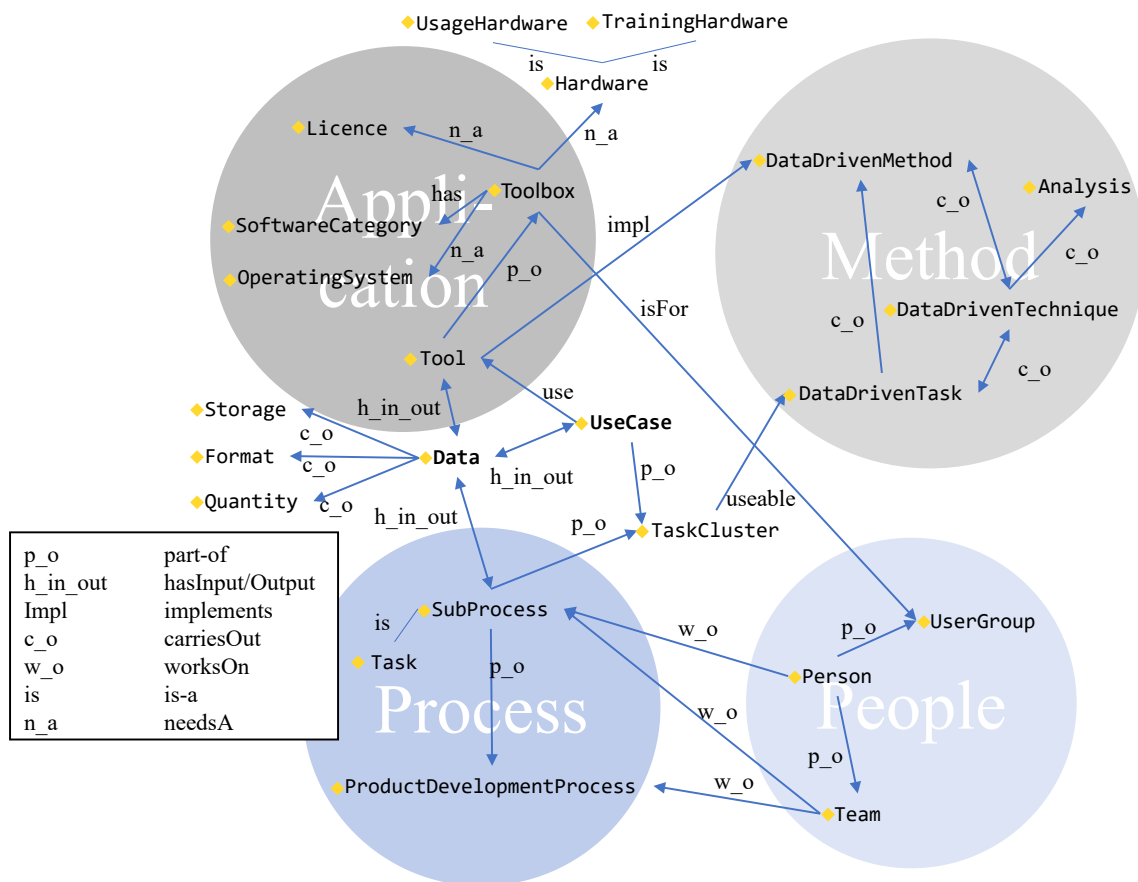
$$\begin{aligned} \text{Task} &\sqsubseteq \text{isPartOfProcess.SubProcess} \\ &\sqsubseteq \exists \text{generatesData.Data} \\ &\sqsubseteq \exists \text{UsesData.Data} \\ &\equiv \text{prov:Activity} \end{aligned}$$


Figure 5. Core concepts (yellow diamond) and relationships (blue arrows) of AI4PD.

Every company has individual process details, which makes it challenging to give general statements about applicable methods to specific tasks or subprocesses. Therefore, the concept of *TaskCluster* is introduced. Examples for these clusters are given in Section 3.1. These *TaskClusters* allow a generalisation of individual and company-specific process definitions regarding inputs, outputs and used systems. The cluster focuses on the overall goal (e.g., transforming an input, such as requirements, to an output, such as a CAD-Model, or using a CAD-Model to create a FEM result). The work result (e.g., the individual FEM result) is abstracted during the cluster formation. Since iterations and error corrections represent loops back to process steps that have already been performed, they are not recorded separately as clusters. *TaskClusters* have defined input and output data. They are used in abstract data types such as *ASCIIData* or *EngineeringData* rather than explicit data types. Furthermore, task clusters use a particular type of program. In distinction to Digital Engineering tools, the term program refers to CAD programs, office programs, etc.

Furthermore, a *TaskCluster* can use a specific *DataDrivenTask*. The following axiom describes a *TaskCluster* in consideration of the definitions above:

$$\begin{aligned} &\exists \text{hasInput.Data} \cap \exists \text{hasOutput.Data} \cap \exists \text{use.Tool} \\ &\cap \text{use.DataDrivenTask} \sqsubseteq \text{TaskCluster} \end{aligned}$$

A product development *Team* consists of several team members, each with expertise and qualifications. Additionally, a team works on one or more *ProductDevelopmentProcesses* at a time. The formal definition of a *Team* is given by

$$\begin{aligned} \text{Team} &\sqsubseteq > 1 \text{hasMember.Person} \\ &\sqsubseteq \geq 1 \text{workingOn.ProductDevelopmentProcess} \end{aligned}$$

For *Persons* and connections between them, the friends-of-a-friend (FOAF) [36] ontology provides a good insight. Therefore, the foaf:Person class is set as equivalent to the ai4pd:Person class. The FOAF properties are not listed to keep Figure 5 easily readable. Unfortunately, when trying to implement the given use case of identifying suitable data-driven methods, the FOAF information is insufficient and further details are necessary. The *Person* concept is extended by the definition of a *UserGroup* to assign a certain background knowledge (e.g., programming, simulation, management). Furthermore, a *Person* works on a *Task* and is part of a *Team*. Therefore, the following axioms describe a *Person*:

$$\begin{aligned} \text{Person} &\sqsubseteq \geq 1 \text{MemberOf.Team} \\ &\sqsubseteq \geq 1 \text{WorkingOn.Task} \\ &\sqsubseteq = 1 \text{IsUserType.UserGroup} \\ &\equiv \text{foaf : Person} \end{aligned}$$

Concerning *Data*, they are generated in a defined format and in a *SubProcess* or *Task*. Additionally, they have a certain quality or missing elements that must be considered during the data evaluation. The *Data* concept is further specified in the areas of *ASCIIData*, *Document* and *EngineeringData*, as shown in Figure 6. This is necessary since these general data types have a significant influence on the usable *DataDrivenMethod*. Overall, the *Data* concept is defined as follows:

$$\begin{aligned} \text{Data} &\sqsubseteq \geq 1 \text{generatedBy.Task} \vee 1 \text{generatedBy.SubProcess} \\ &\sqsubseteq \geq 1 \text{generatedBy.Tool} \\ &\sqsubseteq \geq 1 \text{usedBy.Task} \vee 1 \text{usedBy.SubProcess} \end{aligned}$$

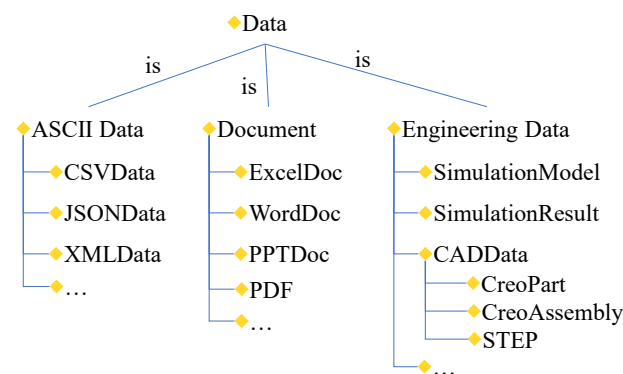


Figure 6. Specification of the data concept (yellow diamonds symbolise core concepts).

Sufficient formalisation is more complicated with *DataDrivenTechnique* and *DataDrivenMethod*, as there are no general definitions or distinctions between these areas. The concept *DataDrivenTechnique* gives a first layer of abstraction. Here, the general principle purpose

of the underlying items is defined, which can be *DataHandling*, *ModelHandling* or *Modelling*. Furthermore, these *DataDrivenTechniques* can perform an *Analysis* and are used by a *DataDrivenTask*. The *DataDrivenTechnique* can only give a first distinction of all available possibilities. The next and more detailed layer in the AI4PD-Ontology is the *DataDrivenTask*. They specify the overall task with which the underlying *DataDrivenMethod* can work on. At this level, terms such as association, classification, clustering or supervised learning are settled. Following the given terminology, all entities in the knowledge base can be described by the following axiom for the *DataDrivenTechnique* concept:

$$\text{DataDrivenTechnique} \sqsubseteq \exists \text{DoesAnalysis}. \text{Analysis} \cap \exists \text{UsedBy}. \text{DataDrivenTask}$$

DataDrivenMethod analyses *Data*. Since the used *Tool* defines the usable data types, the connection to *Data* is made within the *Tool* concept. These *DataDrivenMethods* differ in aspects such as training time, necessary data and accuracy. A further distinction can be made if a technique predicts new data or evaluates only existing ones. The unique algorithms and methods are subsumed within the concept of a *DataDrivenMethod*. Examples of individuals of this concept are *k-nearest Neighbour* or *Convolutional Neural Networks*, algorithms that can be applied directly but without taking the explicit implementation into account. An overview of available *DataDrivenMethods* is given in Figure 2 by the white boxes.

Since a *DataDrivenMethod* can be considered as a (theoretical) algorithm or concept, it must be implemented in some *Tool* to make it usable. Additionally, a *Tool* implicitly deploys a *DataDrivenTask*. This *Tool* has input and output *Data*. In general, a *Tool* is not deployed alone but collected in a *Toolbox*. This can consist of multiple tools, each implementing one defined algorithm. The software platform SAS (<https://www.sas.com>, accessed on 9 February 2023), for example, can be seen as a *ToolBox*, collecting several different *Tools* realising *DataDrivenMethods* such as decision trees or support vector machines. A further example is the well-known machine learning package SciKit-Learn (<https://scikit-learn.org>, accessed on 9 February 2023). A *Toolbox* runs on different *OperatingSystems* and uses one *ProgrammingLanguage*. Some toolboxes have commercial licences while others are freeware or open source but every instance has some *Licence*. Additionally, they require defined *TrainingHardware* and *UsageHardware*, and aim at a certain *UserGroup*.

Finally, the *UseCase* concept enables the linking of already implemented solutions. A *UseCase* is part of a defined *TaskCluster* and needs specific input and output *Data*. Furthermore, it uses at least one *Tool*, *DataDrivenMethod* and *DataDrivenTask*. Additionally, a *UseCase* has a *Name* and an *Author* or a *company*. In the case of academic publications, a *DOI* exists too. Formally, the following axiom defines all *UseCase* entities:

$$\begin{aligned} \text{UseCase} \sqsubseteq & \geq 1 \text{hasInput}. \text{Data} \cap \geq 1 \text{hasOutput}. \text{Data} \cap \geq 1 \text{use}. \text{Tool} \\ & \cap \geq 1 \text{use}. \text{DataDrivenMethod} \cap \geq 1 \text{use}. \text{DataDrivenTask} \cap 1 \text{partOf}. \text{TaskCluster} \end{aligned}$$

The formal representation of AI4PD is in Web Ontology Language OWL 2 DL [38] and was modelled in the ontology editor Protégé [39].

3. Results

In the following section, two use cases show the power of the developed ontology. Furthermore, an example of the presented task clusters is given.

3.1. Task Cluster Examples

The *TaskClusters* were generated through a process analysis in a Bavarian automotive company and validated through an internal workshop at the institute and in collaboration with the industrial partners. With such clusters, the most important *Tasks* and *SubProcesses* can be placed into this cluster system. Therefore, using this cluster system, the whole *ProductDevelopmentProcess* can be described. A first overview of the cluster is given in Table 1. Afterwards, the clusters are described in more detail.

Table 1. Overview of the defined task clusters.

Cluster	Goal	Knowledge Work	Data Type Change	Input	Output
Creation of Reports	documentation	no	yes	engineering data	report
Approval	release of defined statuses	yes	yes	report	decision
Relationships	transform inputs to defined outputs	yes	yes	engineering data	different but engineering data
Data as basis	use old data for new product generation	yes	no	engineering data	same as input
Result evaluation	evaluation of measurements or simulations	yes	yes	engineering data	plausibility assessment
Knowledge management	maintenance of knowledge management systems	no	yes	knowledge	formalised knowledge
Assignment of other departments	transfer information to subcontractors	no	yes/no	task-dependent	task-dependent
Data procurement	obtain missing data or information	no	no	task-dependent	task-dependent

Cluster 1—Creation of reports and (development) documentation: The central goal is the documentation of the product development process. Here, an engineering data type (CAD part, FEM result, etc.) is converted into a report data type (PowerPoint or PDF). Although they are highly relevant in every product development process, the tasks in this cluster are monotonous, repetitive work without significant knowledge content.

Cluster 2—Approval: Based on reports (input), an employee evaluates results and makes decisions (output). The tasks in this cluster depend on the results produced in Cluster 1. The final decision is often made by a member of the management, either rule-based or based on individual experience. The share of knowledge work cannot be clearly assigned here.

Cluster 3—Formation of interrelationship: Based on knowledge, employees generate appropriate outputs from engineering inputs. For example, geometry is built depending on requirements. Alternatively, based on geometry and simulation instructions, deformations can be estimated. The tasks are highly knowledge-based and require a great amount of expertise. The results of this cluster are transformed into a report or documentation in Cluster 1.

Cluster 4—Using existing data as basis: In contrast to the previous cluster, the focus here is on using existing data of the same type (e.g., geometry) as a basis for the generation of new developments or the analysis of existing data to obtain information. In addition, data analysis that is not directly linked to product development (e.g., market data) for information acquisition is located here. This is no classical knowledge-based work; however, it includes the experience and knowledge of the employees.

Cluster 5—Analysis and evaluation of results: Here, the focus is on evaluating measurement reports and simulations. The input is a result format, often in graphical or numerical form, generated in cluster 3. The output is an abstract and knowledge-based plausibility or quality assessment. Finally, a conclusion is drawn concerning the component quality. The transfer of the findings into formal development reports is performed in cluster 1.

Cluster 6—Knowledge management system maintenance: This cluster focuses on maintenance of the knowledge management systems for the supply of collected knowledge in later development processes. Findings from all clusters are processed here. However, the focus is on the findings from cluster 2 (e.g., new approval rules) or cluster 3 (e.g., the experience of changing geometry based on changing requirements). This repetitive work requires knowledge and experience as input and converts it into formalised knowledge.

Cluster 7—Assignment of other departments or companies: The focus here is on the transfer of information to subcontractors or subordinate departments to reduce the department's workload through outsourcing. A well-known example for this cluster is the assignment of a simulation service provider. The subcontractor performs a task from another cluster.

Cluster 8—Data and information procurement or consultations: This task type is not a value-adding activity. Tasks from this cluster can provide the knowledge or data basis for clusters 3 and 4. In most practical processes, the required data and information are not directly available for all employees and must be collected first, especially when new projects start. In addition, these tasks result in missing information from previous development steps and are often associated with waiting times.

3.2. Use Case 1: Use Case Coverage

As mentioned above, one key challenge is to find already implemented solutions for integrating data-driven methods in product development. This is due to the fact that no standardised documentation of applied use cases can be found in the industry or research. AI4PD can link the domains and allows a description of relevant use cases by linking the available data sources. Relevant data are primarily the following:

- A description of the product development process;
- Used tools and toolboxes as well as the applied data-driven method;
- Required input and corresponding output data.

The power of the use case coverage is shown in the example of the self-learning assistance system (SLASSY) [40]. SLASSY supports the design engineer during the development of sheet-bulk metal parts with added functional elements that are manufactured by sheet-bulk metal forming. A key aspect of this system is the self-learning component, which allows knowledge to be automatically extracted from existing data in the form of meta-models. This extraction is performed by a knowledge discovery in database (KDD) process based on simulation and experimental results. The process terminates in a regression-based prediction model. The system itself is divided into a synthesis and an analysis module. Since they aim at different goals, only the analysis module is further considered in the contribution. Figure 7 shows the formal representation of SLASSY.

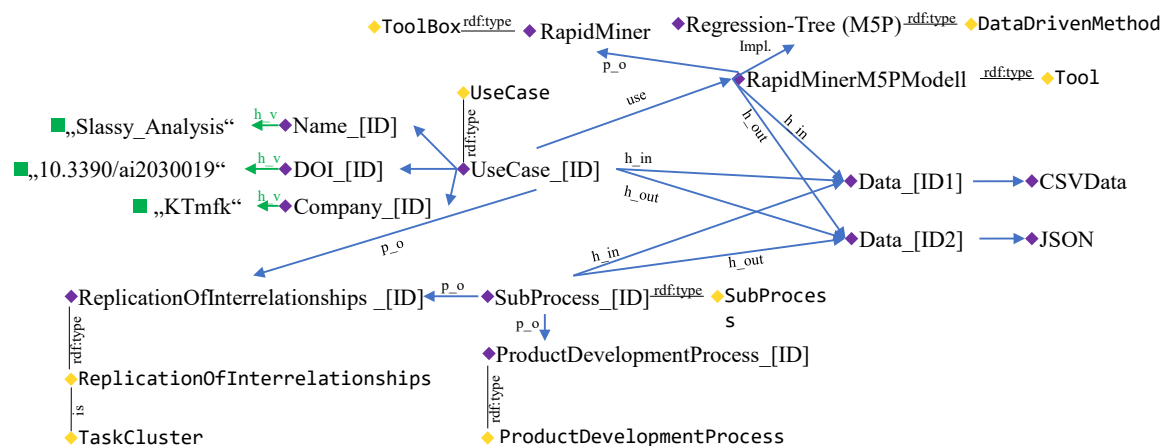


Figure 7. Formal setup of the SLASSY use case. (core concepts: yellow diamonds, relationships: blue arrows, individual instances of a concept: purple diamonds, properties: green squares)

The description of the use case encompasses the properties *Name*, *DOI* and *Company* with their defined values. SLASSY is assigned to the replication of interrelationship *TaskCluster*. The explicit *SubProcess* is modelled by a *SubProcess_[ID]* instance and not named since several tasks or *SubProcesses* are possible for the application of SLASSY. Figure 7 shows that the *Data* instances connect the *Tool*, *Process* and *UseCase* instances. Since two different types of *Data* entities exist, the input and output streams are modelled separately. The *Data_[ID1]* represents the input set, formatted as a comma-separated values (CSV) file. In this file, the test data results are stored. Consequently, *Data_[ID2]* represents the output data in JavaScript object notation (JSON)-format. The result predicts a manufacturing possibility for the given CAD file. Finally, SLASSY uses the *Tool* *RapidMinerRegressionModel*, which is part of the *RapidMiner Toolbox* and implements the *DataDrivenMethod* polynomial regression.

Therefore, all the necessary information is available so that the ontology can answer the competency question given in use case 1 in Section 2.3.

3.3. Use Case 2: Identification of Suitable Methods

The central aspect, and a further extension of use case 1, is the identification of suitable methods for problems found in individual product development processes. Therefore, the ontology is used to query the stored *Data* and *UseCase*. During a process optimisation process, the project leader analyses the given process to identify weak spots. Based on the relevant *SubProcess*, it is possible to identify the available *Data*—provided that they were collected. Combined with the *TaskCluster*, suitable *DataDrivenTasks*, *DataDrivenMethods* and *Tools* can be identified.

The following initial situation is assumed. A simulation department is identified as a bottleneck in the *ProductDevelopmentProcess*; in particular, iterations with the design department have led to a high workload and time delays. This is especially critical relatively early in the product development process since many different concepts are being evaluated quickly. The relevant *SubProcess* is to transfer the configuration of a design, represented in a parametric CAD file, into a simulation, which calculates and generates the results. The associated *Data* are stored in a simulation data management (SDM) system as simulation results. The results can be broken down into numerical displacement information. For training, a high-performance graphic processing unit (GPU)-based machine is available. The later usage should be possible with low computing power and through a command line solution. Table 2 gives a summary of the formal description.

Table 2. Formal problem description.

Concept	Value
Team	Simulation Department
SubProcess	Simulation
DataIn.Format	CAD-Data $\cap$ Numerical parameters
DataIn.Storage	PDM System
DataOut.Format	FEM Result $\cap$ Numerical value
DataOut.Storage	SDM System
TrainingHardware	GPU-based
UsageHardware	low computing $\cup$ command line

Transferring this description into a query based on the SPARQL 1.1 query language allows filtering the knowledge base for suitable methods and tools to support this process. The used query is shown in Listing 1.

Listing 1. SPARQL Select query.

```

SELECT ?DataDrivenTask , ?DataDrivenMethod , ?DataDrivenTool
WHERE {
    ?AI4PD:Team AI4PD:hasValue SimulationDepartment
    ?AI4PD:SubProcess AI4PD:hasValue Simulation
    ?AI4PD:DataIn ;
    AI4PD:hasFormat AI4PD:CADDData || AI4PD:CSVDData
    AI4PD:hasStorage AI4PD:PDM-System
    ?AI4PD:DataOut ;
    AI4PD:hasFormat AI4PD:FEMResult || AI4PD:CSVDData
    AI4PD:hasStorage AI4PD:SDM-System
    ?AI4PD:TrainingHardware AI4PD:hasValue GPUBased
    ?AI4PD:UsageHardware ;
    AI4PD:hasComputingPower AI4PD:lowComputing &&
    AI4PD:hasInterface AI4PD:commandLine
}

```

The SELECT expression queries three variables, which are filtered using the WHERE expression. The filter searches for the conditions in Table 2.

The output shows the used methods and tools from SLASSY since this use case aims at supporting the described problem. Additionally, this is the answer for the competency question from use case 2 in Section 2.3.

The output table in Table 3 shows the identified suitable method. It is possible to identify more than one possibility, since a *DataDrivenMethod* can be implemented in different *Tools*. In this case, further filtering should be performed based on licensing, for example, to only consider freeware. This should be carried out depending on the goals or internal requirements of the company.

Table 3. Example output for query in Listing 1.

DataDrivenTask	DataDrivenMethod	DataDrivenTool
Regression	Regression-Tree (M5P)	RapidMinerM5PModell

4. Discussion

In the previous section, the functionality of AI4PD was shown. Based on this, the following chapter discusses the results concerning the defined research question. The given research question can be categorised into three main objectives. The first one is linking the domains of artificial intelligence and product development processes. The second objective is simplifying the introduction of Digital Engineering in small and medium-sized enterprises. The third one is improving method knowledge availability. Each of these objectives is discussed and examined in depth.

4.1. Linking the Domains of AI and Product Development Processes

AI4PD can give a formal frame to link artificial intelligence and product development. A central challenge is an ambiguity in the terminology of data-driven methods and tools, making them hard to formalise and describe. Further problems are the variety of development processes and the used data types in the product development process. The first problem is tackled in AI4PD by the introduction of task clusters, which introduce a generalised level of abstraction, allowing integration of each company's individual boundary conditions. The ontology gives a first insight into available data types for the data type problem. In practical use, data type transformations are frequent. Most simulation results, for example, can be easily transferred to ASCII-Data types, such as JSON or CSV, making them accessible to more tools. At the moment, the *Data* concept allows for a transformation property, defining to which types a single data type can be transformed. Further analysis of

transformation options would be helpful but is not the focus of this contribution. Overall, AI4PD fulfils the goal of linking both domains by a formal model.

4.2. Simplify the Introduction of Digital Engineering in SMEs

Most companies identify Digital Engineering as a significant competitive advantage of the future. However, it requires adequate knowledge of the given possibilities and support in introducing those methods. SMEs typically lack experts with this knowledge. Therefore, simplifying the introduction of Digital Engineering in SMEs is the second objective of the research question. For this reason, a transformation process of the design process to Digital Engineering—including the steps of process capturing, process analysis and optimisation—by integrating data-driven methods is proposed, see Figure 4. The ontology especially supports the third aspect. For the first two, companies need other process management methods, such as the one presented by Gerschütz et al. [2]. Using this ontology, the identification of suitable data-driven methods for given product development problems is possible. To obtain this, the identified problem must be described formally, as shown in Table 2 and Listing 1. This description and transformation to a SPARQL query can be challenging for inexperienced process engineers. Therefore, further support, such as a GUI, would be helpful. After identifying suitable data-driven methods for a given product development process, it can be challenging to integrate those methods into a given process. The available tools have to be adapted to individual companies or process circumstances. This often requires significant customisation, an implementation effort as well as process redesigns. AI4PD supports this final step by providing best practices and implemented use cases. Furthermore, expanding the whole ontology to integrate customised approaches of methods would be possible.

4.3. Improve Method Knowledge Availability

Knowledge of existing and suitable methods is essential for successfully transforming the product development process from virtual to Digital Engineering. As already mentioned above, SMEs lack this knowledge. Therefore, the third aspect of the presented research question is improved knowledge availability. The process of method integration requires great expert knowledge, primarily available in an unstructured manner. Possible sources are scientific publications, best practice reports or web-based tool(box) documentation. Therefore, the initial aim was to consider a formal model of the Digital Engineering context with its subconcepts, especially methods and applications. This goal is achievable, as shown in Section 2, where the interconnections are described. For further usage of AI4PD, a (semi)automated integration and annotation of knowledge, available in the already mentioned unstructured sources, will be crucial. The manual workload for integrating method knowledge in this state of the ontology is high, making efficient usage difficult. One possible solution could be using text-mining techniques, especially for analysing and integrating use cases, as presented by Kestel et al. [32]. Another challenge is to keep the knowledge up to date, as the area of methods and toolboxes is currently undergoing very rapid development. In particular, expansion, revision and contraction methods are needed to ensure that only correct and up-to-date information is presented [41]. As mentioned by Kügler et al. [28], OWL ontologies enable flexible extensions due to their open world assumption (OWA), especially when compared to common databases with the closed world assumption (CWA). The essence of OWA is to allow all assumptions until explicitly disproved. In contrast, CWA prohibits everything until it is proven true. Therefore, OWA gives an excellent representation of incomplete or rapidly changing information, as is the case in the domain of Digital Engineering. Removing or including concepts is crucial for AI4PD, especially since AI4PD does not claim completeness, particularly for data types, methods and toolboxes. After deciding on a particular method or tool, further knowledge is required since the usage of a certain method can have further prerequisites. Possible implications are insufficient computing power or inappropriate data for a given method. The ontology allows the evaluation of the methods according to the mentioned aspects

since simple yes or no questions are possible as well. For example, the question can be asked whether the existing hardware is sufficient to use a particular tool.

To summarise, AI4PD provides sufficient linking and knowledge management regarding data-driven methods for product development. Furthermore, support for integrating data-driven methods in SMEs is given. Therefore, the given research question has been successfully answered.

5. Conclusions and Outlook

The contribution provides a formal specification to support a more unified understanding of Digital Engineering. Therefore, the AI4PD ontology bridges the gap between the domains of artificial intelligence and product development processes. The ontology implements four main concepts: Application, Method, Process and People. This allows the knowledge management of *DataDrivenMethods*, *Tools* and *ToolBoxes* implementing *DataDrivenMethods* and aspects of *ProductDevelopmentProcesses*. The latter users of integrated methods are considered in the *Person* context. Moreover, existing use cases, such as SLASSY, can be integrated into the ontology, which allow for building a database of already realised solutions. By querying the ontology, simplified knowledge reuse is possible too. These aspects support small and medium-sized enterprises in integrating data-driven methods into their product development and transforming it into a new Digital Engineering process. Future work will focus on integrating the ontology into an overall Digital Engineering process optimisation method. Furthermore, the already mentioned automated knowledge integration is a central goal of future research to further enhance the applicability of the proposed approach.

Author Contributions: Conceptualization, B.G.; methodology B.G.; writing—original draft preparation, B.G.; writing—review and editing, B.G., S.G. and S.W.; visualization, B.G.; supervision, S.G. and S.W.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Bayerische Forschungsstiftung (BFS) grant number AZ-1392-19.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: This research work is part of “FORCuDE@BEV—Bavarian research association for customized Digital Engineering for bavarian SME’s” and funded by the “Bayerische Forschungsstiftung (BFS)”. The authors are responsible for the content of this publication. A special thank goes to the Bayerische Forschungsstiftung (BFS) for financial support of the whole research project. We acknowledge financial support by Deutsche Forschungsgemeinschaft and Friedrich-Alexander-Universität Erlangen-Nürnberg within the funding programme “Open Access Publication Funding”. Furthermore, L. Stöhr, M.Sc. is thanked for supporting this research with her masters thesis.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
CRISP-DM	Cross-industry standard process for Data Mining
CAD	Computer-Aided Design
CSV	Comma-separated values
CWA	Closed world assumption
DM	Data Mining

FEM	Finite Element Method
GPU	Graphics Processing Unit
IRI	Internationalised Resource Identifier
JSON	JavaScript object notation
KDD	Knowledge discovery in databases
ML	Machine Learning
OWA	Open world assumption
PD	Product development
PDM	Product data management
PDP	Product development process
SDM	Simulation data management
SME	Small and medium-sized enterprise

References

- Lunnemann, P.; Stark, R.; Wang, W.M.; Stark, R.; Manteca, P.I. Engineering Activities—Considering Value Creation from a Holistic Perspective. In Proceedings of the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira, Portugal, 27–29 June 2017; pp. 315–323. [\[CrossRef\]](#)
- Gerschütz, B.; Spieß, B.V.M.; Schleich, B.; Wartzack, S. An Adapted Method for Design Process Capturing to Meet the Challenges of Digital Product Development. In Proceedings of the International Conference on Engineering Design (ICED21), Gothenburg, Sweden, 16–19 August 2021; Volume 1, pp. 365–374. [\[CrossRef\]](#)
- Stark, R.; Brandenburg, E.; Lindow, K. Characterization and Application of Assistance Systems in Digital Engineering. *CIRP Ann.* **2021**, *70*, 131–134. [\[CrossRef\]](#)
- Gerschütz, B.; Bickel, S.; Schleich, B.; Wartzack, S. Enabling Initial Design-Checks of Parametric Designs Using Digital Engineering Methods. *Proc. Des. Soc.* **2022**, *2*, 405–414. [\[CrossRef\]](#)
- El Sayed, M.A.; Farahat, F.A.; Elsisy, M.A. A novel interactive approach for solving uncertain bi-level multi-objective supply chain model. *Comput. Ind. Eng.* **2022**, *169*, 108225. [\[CrossRef\]](#)
- Montáns, F.J.; Chinesta, F.; Gomez-Bombarelli, R.; Kutz, J.N. Data-Driven Modeling and Learning in Science and Engineering. *Comptes Rendus Mécanique* **2019**, *347*, 845–855. [\[CrossRef\]](#)
- Samuel, A.L. Some Studies in Machine Learning Using the Game of Checkers. *IBM J. Res. Dev.* **2000**, *44*, 206–226. [\[CrossRef\]](#)
- Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P. From Data Mining to Knowledge Discovery in Databases. *AI Mag.* **1996**, *17*, 37. [\[CrossRef\]](#)
- StackExchange. Distinction between AI, ML, Neural Networks, Deep Learning and Data Mining. Available online: <https://softwareengineering.stackexchange.com/q/366996> (accessed on 24 January 2023).
- Kulin, M.; Kazaz, T.; De Poorter, E.; Moerman, I. A Survey on Machine Learning-Based Performance Improvement of Wireless Networks: PHY, MAC and Network Layer. *Electronics* **2021**, *10*, 318. [\[CrossRef\]](#)
- Cambridge University Press. Artificial Intelligence | Meaning in the Cambridge English Dictionary. Available online: <https://dictionary.cambridge.org/de/worterbuch/englisch/artificial-intelligence> (accessed on 24 January 2023).
- Monostori, L. Artificial Intelligence. In *CIRP Encyclopedia of Production Engineering*; Laperrière, L., Reinhart, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 47–50. [\[CrossRef\]](#)
- Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed.; Elsevier: Amsterdam, The Netherlands, 2017. [\[CrossRef\]](#)
- Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: <http://www.deeplearningbook.org> (accessed on 25 January 2023).
- Chapman, P.; Clinton, J.; Kerber, R.; Khabaza, T.; Reinartz, T.; Shearer, C.; Wirth, R. *CRISP-DM 1.0 Step-by-Step Data Mining Guide*; The CRISP-DM Consortium: Chicago, IL, USA, 2000.
- Azevedo, A.; Santos, M.F. KDD, SEMMA and CRISP-DM: A Parallel Overview. In Proceedings of the IADIS European Conference on Data Mining 2008, Amsterdam, The Netherlands, 24–26 July 2008; Abraham, A., Ed.; IADIS: Jambi, Indonesia, 2008; pp. 182–185.
- Erlhoff, M.; Marshall, T.; Bruce, L.M.J.; Lindberg, S. *Design Dictionary: Perspectives on Design Terminology*; Birkhäuser: Berlin, Germany; Boston, UK, 2007. [\[CrossRef\]](#)
- Krensky, P.; Idoine, C.; Brethenoux, E.; den Hamer, P.; Choudhary, F.; Jaffri, A.; Vashisth, S. *Magic Quadrant for Data Science and Machine Learning Platforms*; 2022. Available online: <https://www.gartner.com/en/documents/3998753> (accessed on 30 May 2022)
- Wuest, T.; Weimer, D.; Irgens, C.; Thoben, K.D. Machine Learning in Manufacturing: Advantages, Challenges, and Applications. *Prod. Manuf. Res.* **2016**, *4*, 23–45. [\[CrossRef\]](#)
- Gerschütz, B.; Schleich, B.; Wartzack, S. A Semantic Web Approach for Structuring Data-Driven Methods in the Product Development Process. In Proceedings of the DS 111: 32nd Symposium Design for X, The Design Society, Tutzing, Germany 27–28 September 2021. [\[CrossRef\]](#)
- Mikut, R.; Reischl, M. Data Mining Tools. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 431–443. [\[CrossRef\]](#)
- Pahl, G.; Wallace, K.; Blessing, L.; Pahl, G. (Eds.) *Engineering Design: A Systematic Approach*, 3rd ed.; Springer: London, UK, 2007.
- VDI 2221 Blatt 1 *Design of Technical Products and Systems—Model of Product Design*; Beuth: Berlin, Germany, 2019.

24. VDI/VDE 2206:2021-11—*Development of Mechatronic and Cyber-Physical Systems*; Beuth: Berlin, Germany, 2021.
25. Nattermann, R.; Anderl, R. The W-model – Using Systems Engineering for Adaptronics. *Procedia Comput. Sci.* **2013**, *16*, 937–946. [CrossRef]
26. Avnet, M.S.; Weigel, A.L. The Structural Approach to Shared Knowledge: An Application to Engineering Design Teams. *Hum. Factors: J. Hum. Factors Ergon.* **2013**, *55*, 581–594. [CrossRef]
27. Gruber, T.R. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis.* **1993**, *5*, 199–220. [CrossRef]
28. Kügler, P.; Marian, M.; Schleich, B.; Tremmel, S.; Wartzack, S. tribAIn—towards an Explicit Specification of Shared Tribological Understanding. *Appl. Sci.* **2020**, *10*, 4421. [CrossRef]
29. Gene Ontology Consortium. The Gene Ontology (GO) Database and Informatics Resource. *Nucleic Acids Res.* **2004**, *32*, 258–261. [CrossRef]
30. Chen, X.; Chen, C.H.; Leong, K.F.; Jiang, X. An Ontology Learning System for Customer Needs Representation in Product Development. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 441–453. [CrossRef]
31. Kim, K.Y.; Manley, D.G.; Yang, H. Ontology-Based Assembly Design and Information Sharing for Collaborative Product Development. *Comput.-Aided Des.* **2006**, *38*, 1233–1250. [CrossRef]
32. Kestel, P.; Kügler, P.; Zirngibl, C.; Schleich, B.; Wartzack, S. Ontology-Based Approach for the Provision of Simulation Knowledge Acquired by Data and Text Mining Processes. *Adv. Eng. Inform.* **2019**, *39*, 292–305. [CrossRef]
33. Kügler, P.; Kestel, P.; Schon, C.; Marian, M.; Schleich, B.; Staab, S.; Wartzack, S. Ontology-based approach for the use of intentional forgetting in product development. In Proceedings of the 15th International Design Conference, Dubrovnik, Croatia, 21–24 May 2018; pp. 1595–1606. [CrossRef]
34. Yang, Q.; Miao, C.; Zhang, Y.; Gay, R. Ontology Modelling and Engineering for Product Development Process Description and Integration. In Proceedings of the 2006 IEEE International Conference on Industrial Informatics, Singapore, 16–18 August 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 85–90. [CrossRef]
35. Lebo, T.; Sahoo, S.; McGuinness, D. PROV-O The PROV Ontology. Available online: <https://www.w3.org/TR/prov-o/> (accessed on 24 January 2023).
36. Brickley, D.; Miller, L. FOAF Vocabulary Specification. Available online: <http://xmlns.com/foaf/0.1/> (accessed on 24 January 2023).
37. Uschold, M.; Gruninger, M. Ontologies: Principles, Methods and Applications. *Knowl. Eng. Rev.* **1996**, *11*, 93–136. [CrossRef]
38. Hitzler, P.; Krötzsch, M.; Parsia, B.; Patel-Schneider, P.F.; Rudolph, S. OWL 2 Web Ontology Language Primer (Second Edition). Available online: <https://www.w3.org/TR/owl2-primer/> (accessed on 24 January 2023).
39. Musen, M.A. The Protégé Project: A Look Back and a Look Forward. *AI Matters* **2015**, *1*, 4–12. [CrossRef]
40. Sauer, C.; Breitsprecher, T.; Küstner, C.; Schleich, B.; Wartzack, S. SLASSY—An Assistance System for Performing Design for Manufacturing in Sheet-Bulk Metal Forming: Architecture and Self-Learning Aspects. *AI* **2021**, *2*, 307–329. [CrossRef]
41. Alchourrón, C.E.; Gärdenfors, P.; Makinson, D. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *J. Symb. Log.* **1985**, *50*, 510–530. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.