processes

MDPI

Article

Towards Flexible Control of Production Processes: A Requirements Analysis for Adaptive Workflow Management and Evaluation of Suitable Process Modeling Languages

Alexander Schultheis [1,2,*,†], David Jilg [1,†], Lukas Malburg [1,2], Simon Bergweiler [3,4] and Ralph Bergmann [1,2]

1 German Research Center for Artificial Intelligence (DFKI), Experience-Based Learning Systems, Branch Trier University, 54296 Trier, Germany; david.jilg@dfki.de (D.J.); lukas.malburg@dfki.de or malburgl@uni-trier.de (L.M.); ralph.bergmann@dfki.de or bergmann@uni-trier.de (R.B.)
2 Artificial Intelligence and Intelligent Information Systems, Trier University, 54296 Trier, Germany
3 German Research Center for Artificial Intelligence (DFKI), Innovative Factory Systems, 67663 Kaiserslautern, Germany; simon.bergweiler@dfki.de
4 Technologie-Initiative SmartFactory^{KL} e.V., 67663 Kaiserslautern, Germany
* Correspondence: alexander.schultheis@dfki.de or schultheis@uni-trier.de
† These authors contributed equally to this work.

Abstract: In the context of Industry 4.0, Artificial Intelligence (AI) methods are used to maximize the efficiency and flexibility of production processes. The adaptive management of such semantic processes can optimize energy and resource efficiency while providing high reliability, but it depends on the representation type of these models. This paper provides a literature review of current Process Modeling Languages (PMLs). Based on a suitable PML, the flexibility of production processes can be increased. Currently, a common understanding of this process flexibility in the context of adaptive workflow management is missing. Therefore, requirements derived from the business environment are presented for process flexibility. To enable the identification of suitable PLMs, requirements regarding this are also raised. Based on these, the PMLs identified in the literature review are evaluated. Thereby, based on a preselection, a detailed examination of the seven most promising languages is performed, including an example from a real smart factory. As a result, a recommendation is made for the use of BPMN, for which it is presented how it can be enriched with separate semantic information that is suitable for the use of AI planning and, thus, enables flexible control.

Keywords: Industry 4.0; business process management; adaptive workflow management; imperative process modeling languages; flexible control processes

Citation: Schultheis, A.; Jilg, D.; Malburg, L.; Bergweiler, S.; Bergmann, R. Towards Flexible Control of Production Processes: A Requirements Analysis for Adaptive Workflow Management and Evaluation of Suitable Process Modeling Languages. *Processes* **2024**, *12*, 2714. https://doi.org/10.3390/pr12122714

Academic Editors: Minghai Yuan and Fengque Pei

Received: 28 October 2024
Revised: 18 November 2024
Accepted: 26 November 2024
Published: 1 December 2024

1. Introduction

In the context of *Industry 4.0* (I4.0) [1], *Artificial Intelligence* (AI) methods, data analysis techniques, the *Internet of Things* (IoT), and distributed systems are integrated into industrial processes. The analysis and control of such processes is addressed in the research area of *Business Process Management* (BPM) [2–4]. Current research investigates how these can be planned and executed dynamically [5], considering criteria such as energy and resource efficiency [6]. A particular focus is on *Adaptive Workflow Management* (AWM) [7–9], which adapts to changing conditions in real time through fast data processing and, thus, offers a high level of failure resilience. This enables scalable utilization of production and information technology resources so that capacities can be increased or reduced as required. For example, application of AWM for increasing the flexibility in production processes offers the potential to optimize the use of facilities as well as resources and to react quickly to problems in the supply chain or increasing demand. To make these processes that flexible, the use of suitable technologies such as AI Planning [10,11] is necessary. This enables analysis and planning processes to be created and optimized in such a way that resources are used efficiently and costs are reduced. This is carried out both initially, before the processes are executed, and during the execution in real time, in response to failures or other changing environmental conditions. To increase the performance of these methods, they can be combined with other AI techniques like *Case-Based Reasoning* (CBR) [12,13]. However, the use of such technologies requires a flexible, efficient, and resilient design on which the processes are implemented.

There are various models for representing manufacturing processes called *Process Modeling Language* (PML) (or also process representation language), such as *Business Process Model and Notation* (BPMN) [14]. Literature reviews on these are already conducted in previous research (e.g., [15–17]). However, none of these focused on PMLs for representation of industrial manufacturing processes and their suitability for AWM aiming at process flexibility. This process flexibility describes the ability to react automatically and dynamically to environmental changes [9] (pp. 215–217). The term flexible control, meaning AWM for increasing the flexibility in production processes, is defined in various ways in the literature [18], but has not yet been specified in a way that it can be measured in industrial practice. As a result, there is also a lack of practical requirements for the type of representation of such processes, based on which a suitable PML can be derived.

We address these problems by providing a current literature overview of suitable PMLs for the representation of production processes. To specify the definition of industrial control process flexibility on which this paper is based, we collect corresponding requirements from the industry. Analogous requirements for the representation formats are elicited, which we evaluate for the identified PMLs. On this basis, we present a preferred PML combined with an approach for semantic integration that enables the usage of AI methods for flexible production.

The paper is structured as follows: First, an overview of the necessary foundations and related work is given in Section 2. Then, in Section 3, the approach and the results of the literature research on PMLs for production are presented. In Section 4, the requirements for both the flexibility of processes and the PMLs are introduced. Based on the requirements, the suitability of the PMLs in Section 5 is assessed. Section 6 presents necessary extensions so that this modeling language fulfills the desired requirements. Finally, the paper is summarized in Section 7, and an outlook on future work is given.

2. Foundations and Related Work

In this section, the necessary background from the field of BPM is introduced in Section 2.1. Based on this, Section 2.2 presents the research field of AWM and the appropriate AI planning methodology. Section 2.3 introduced PMLs in the BPM context. In Section 2.4, related work on surveys of PMLs is presented.

2.1. Business Process Management

BPM [9] is a methodical approach to the continuous improvement in ways of efficiency, flexibility, and quality of business processes. It comprises the six phases of identification, discovery, redesign, implementation, monitoring, and controlling of the BPM lifecycle [9] (pp. 15–26), illustrated in Figure 1. At the beginning, the relevant business processes to be selected for analysis and improvement in the context of BPM are identified and prioritized. In the discovery step (also called Design), the identified business processes are examined in detail and documented to gain a clear understanding of how they currently work. This documentation involves the representation of processes in formal models, which can be visual or textual. This requires an accurate capture of the process logic, the interactions between process participants, and the data and resource flows. The selection of a suitable PML is crucial for the discovery step in BPM, as it ensures the precision and comprehensibility of the process models, enables interoperability with other systems, and forms the basis for automation and analysis. Based on the information gathered, the business processes are redesigned or optimized to achieve efficiency gains or other improvements (e.g., sustainability criteria as energy or resource optimization [19]). These adapted business processes are then put into practice by implementation. Continuous monitoring of the implemented processes measures their performance and ensures that they are working as expected. Finally, the monitored processes are checked in controlling and adjusted if necessary to correct deviations and ensure continuous improvements.

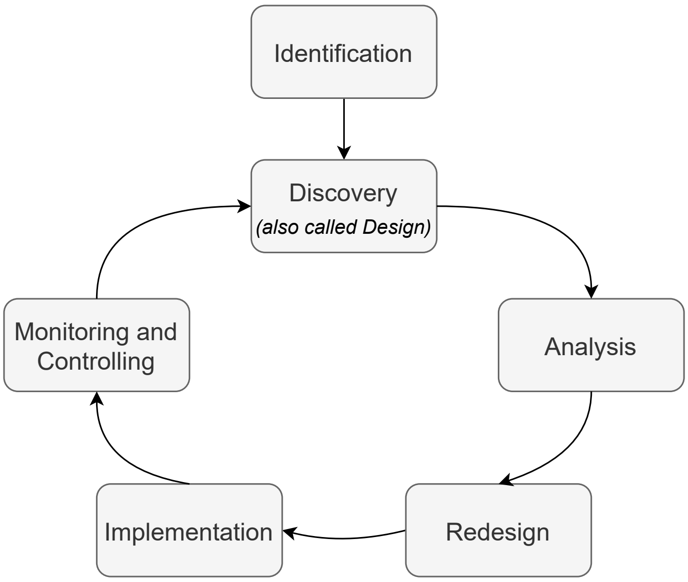

Figure 1. The BPM Lifecycle According to Dumas et al. [9] (p. 21).

2.2. *Adaptive Workflow Management and AI Planning*

The steps of process implementation and controlling can be addressed using AWM [7–9]. This is a field of research that concerns the flexible adaptation of workflows to changing environmental conditions. For example, a problem may be that the execution of a production workflow cannot be carried out as planned and therefore the production stops. This requires a flexible adaptation of the workflow to these external changes. An essential concept in this context is flexibility, which describes the ability to react to such changes [9] (pp. 215–217). In AWM, this reaction should be automatic and dynamic. There are various definitions of this term in the literature [18], but none from industrial practice.

One approach for addressing these flexible reactions in AWM is *AI planning* (also called automated planning) [20]. This technique aims to solve state transformation problems by identifying a sequence of steps to transition a discrete world model from an initial state to a desired goal state [10,11]. For example, in a production context, AI planning could be used to plan the production of certain products. The world model would thereby consist of the production capabilities (e.g., available machines and services, such as drilling or welding), while the initial state would model the availability of resources; the goal state would consist of the finished products. This method is already used in BPM to enhance automation and provide support [5]. It should be possible to take various metrics into account that, e.g., relate to the resilience or energy efficiency of the processes but also to its flexibility [21]. To plan control processes as well as other process types like analysis processes (e.g., for use in predictive maintenance [22,23] or the identification of data quality issues [24,25]), it is necessary for the information on which the planning is based to be available in a suitable form to be transferred to a planning problem [26]. This information must be returned appropriately to make these processes directly executable. As solving complex planning problems can be very computationally intensive, other AI methods can be used to reduce complexity. For example, CBR is used to reduce computational complexity and increase flexibility by reusing already solved problems [13,27].

2.3. Process Modeling Languages

A process is a partially ordered set of tasks or steps aimed at reaching a specific goal [28]. In the context of BPM, this goal is specified by the corresponding business and the process embedded in its structure, defining roles and relationships [29]. The modeling of such a business process serves to describe, analyze, or enact a process [28]. There are various PMLs that focus on one of these aspects, usually harming the others [30]. Depending on the user, such modeling can be viewed differently. Curtis [28] identifies the functional, dynamic, informational, and organizational view of such processes. In the context of industrial manufacturing processes, the first three of these views are usually relevant.

PMLs can be divided into declarative and imperative languages [31]. A declarative PML models the underlying logic of a process and specifies how a result should look. No concrete production steps are specified. Instead, the focus is on the rules and conditions that must be fulfilled, leaving flexibility for execution. In contrast, an imperative PML describes how a sequential process is executed. Thereby, the exact steps and procedures required to carry out a process are presented. These languages specify the sequence of activities and the conditions under which they can be carried out.

2.4. Literature Reviews on Process Modeling Languages

Previous researchers have already published literature reviews on PMLs and associated requirements. In 1999, Conradi and Jaccheri [32] set out various requirements regarding PMLs, such as the modeling of process elements and the addressing of the BPM lifecycle, and compare ten languages against this background. The focus of this work is on software processes that involve the development and maintenance of software. In the same year, Lou and Alex Tung [33] design a framework for evaluating PMLs that focuses on the BPM modeling objectives. This is demonstrated by an example comparing two languages. In 2002, Söderströn et al. [34] present a more general metamodel, which they use to compare three PMLs. Several application areas from the BPM field are mentioned, such as workflow management or process reengineering. Two years later, Aguilar-Savén [35] propose a cross-application classification of PMLs also containing the users perspective and compare nine languages based on it. In 2005, Nysetvold and Krogstie [36] present a generic framework for assessing the quality of PMLs specialized to the needs of a company. They compare three different languages based on the use case of an insurance company. A year later, List and Korherr [37] provide another generic framework for comparing business PMLs. They use it to evaluate seven different languages. In 2008, Korherr [38] presents a domain-independent evaluation framework for business PMLs and compares six languages in detail. One year later, Recker et al. [39] consider twelve different PMLs in a general comparison. Moreover, in 2009, Fahland et al. [31] examine the theoretical differences between declarative and imperative PMLs regarding their understandability. Nine examples of PMLs are given to show that imperative or declarative property is not an absolute but a spectrum. In 2010, Mili et al. [30] create an overview of BPM languages and introduced a categorization for them. They present 17 different PMLs, including examples. In the same year, zur Muehlen and Indulska [40] compare the *Bunge-Wand-Weber* (BWW) format with four other PMLs, focusing on ontological completeness and overlap. In 2012, Scanavachi Moreira Campos and Teixeira de Almeida [41] propose a framework for selecting a PML based on modeling objectivex and demonstrate it based on an example. Two years later, García-Borgoñón et al. [42] conduct a systematic literature study to investigate which PMLs are defined to that date, as well as how they are related to and dependent on each other. In the publication, they present 46 languages and examine their areas of application. In 2015, Goedertier et al. [43] provide an overview only focusing on declarative PMLs and evaluate these. Two years later, Kožíšek and Vrana [16] consider *Unified Modeling Language* (UML), BPMN, and *Event-Driven Process Chains* (EPC) as PMLs and compare them based on various properties motivated by the application in the agri-food industry. In the same year, Hu et al. [15] introduce an evaluation framework that identifies the main requirements for supporting context-adaptive processes. Based on this framework, *Couloured Petri Nets* (CPN), BPMN, *Yet Another Event-Driven Process Chain* (YAWL), and *Unified Modeling Language Activity Diagram* (UML-AD) are evaluated using various real-life scenarios. Brouns et al. [17] 2018 analyze the state of the art for the modeling of IoT-aware business processes. PMLs are considered as well as IoT extensions for these. In 2024, Farshidi et al. [44] present a decision model to identify a suitable PML and consider 23 languages.

3. Process Modeling Languages for Manufacturing

As described in Section 2.4, literature reviews on PMLs already exist. However, none of these surveys focuses on languages that are suitable for representing flexible control processes in the context of AWM in industrial manufacturing. Therefore, by a systematic literature review [45], formats for the representation of sequences and concatenations have been examined, based on which a format for the representation of production processes is to be selected. In addition to the literature study, an online search has also been carried out to provide further results. The languages gathered have been examined in detail, whereby these were categorized into declarative and imperative languages. In the context of this publication, only imperative PMLs that make it possible to map an industrial process, e.g., a manufacturing control process, are considered (see Section 2.3). This means that control-flow elements, i.e., a connection of the tasks in the process, must be supported. Therefore, declarative PMLs such as ADEPTflex [46], *Business Process Constraint Network* (BPCN) [47], ConDec [48], *Process Entailment from the Elicitation of Obligations and Permissions* (PENELOPE) [49], or *Planning Domain Definition Language* (PDDL) [50] (also applicable in this context) are filtered out. In addition, hybrid languages are not considered, such as the hierarchical PMs considered by Slaats et al. [51], which also use imperative or declarative languages at the local level, or BXtendDSL [52], which combines languages of both types. Overall, 27 imperative PMLs are identified that basically support the representation of manufacturing processes. In Table 1, an overview of these is provided.

To evaluate which of the languages is most suited for representing flexible business processes, requirements need to be identified. For this purpose, we conducted a requirement analysis regarding the flexible control of processes and the underlying process representation. The results of this analysis are presented in the following section.

Table 1. Identified Imperative Languages for Process Modeling with Acronyms and the Respective References.

Language Name	Acronym	Reference
Architectural Modeling Box for Enterprise Redesign	AMBER	[53]
Business Process Execution Language	BPEL	[54]
Business Process Definition Metamodel	BPDM	[55]
Business Process Model and Notation	BPMN	[56]
Case Management Model and Notation	CMMN	[57]
Coloured Petri Nets	CPN	[58]
ConDec	ConDec	[48]
COSBI LAB Language	$\mathcal{L}$	[59]
Data Flow Diagrams	DFD	[60]
Data Petri Nets	DPN	[61]
extended/enhanced Event-Driven Process Chains	eEPC	[62]
Event-Driven Process Chains	EPC	[63]
Flow Charts	FC	[64]
NEST Graphs	NEST	[65]
Industry 4.0 Process Modeling Language	I4PML	[66]
Integrated Definition/ICAM Definition	IDEF3	[67]
Linear-Time Temporal Logik	LTL	[68]

Table 1. *Cont.*

Language Name	Acronym	Reference
Process Modeling Language	PML	[69]
Petri Nets	PN	[70]
Role Activity Diagrams	RAD	[71]
Resource Event Agent	REA	[72]
Software Process Engineering Metamodel Specification	SPEM	[73]
Temporal Logic of Actions	TLA	[74]
Unified Modeling Language Activity Diagram	UML AD	[75]
Unified Modeling Language Sequence Diagrams	UML SD	[76]
Business Process Executable Language for Web Services	WS-BPEL	[77]
Yet Another Event-Driven Process Chain	YAWL	[78]

4. Requirements for Flexible Control and the Representation of Corresponding Processes

In the context of this publication, two different sets of requirements are necessary: First, it is essential to define what constitutes the industrial requirements for process flexibility for manufacturing control. Secondly, the PMLs presented in Section 2 must be evaluated for their suitability for such flexible control processes. These requirements can also be used to derive necessary extensions for the existing languages. Initially, a literature research identifies established requirements. As only very general requirements could be derived in this way, further requirements were collected based on a focus group interview [79]. This is a qualitative study that is characterized by its interactive group nature. For this reason, the focus group interview involves selected participants discussing a specific topic, using group dynamics to produce more in-depth insights than individual interviews. We conducted this method of elicitation with the consortium of the *EASY* project (https://easy-edge-cloud.de/), which focuses on the analysis and control of dynamic production processes [6]. This group included various representatives of industrial companies, namely ArtiMinds Robotics GmbH (https://www.artiminds.com/), coboworx GmbH (https://www.coboworx.com/), Empolis Information Management GmbH (https://www.empolis.com/) and Robert Bosch GmbH (https://www.bosch.com/) as well as industry-related research collaborators such as SmartFactory[KL] (https://www.smartfactory.de/), SmartFactoryOWL (https://smartfactory-owl.de/) and IoT Lab Trier (https://iot.uni-trier.de/). Their representatives were able to contribute their respective expert knowledge from production. In the interview, the participants discussed their answers, but overall agreed on the desired requirements. Following the focus group interview, the requirements were elaborated and sent to the involved participants to ensure agreement with the results. The therefore derived requirements for process flexibility are presented in Section 4.1 and for process representation in Section 4.2.

4.1. Requirements for Process Flexibility

The flexibility of a business process (see Section 2.2) describes the extent to which it can adapt to a redesign measure, thus the ability to react to changes [9] (pp. 215–217). This definition must be specified in particular for the application area of flexible process control. Therefore, the requirements identified for *Process Flexibility* (PF) are presented in the following.

Req. 1 (PF): Supported Process Types. The flexible process control should support production processes with different levels of automation, e.g., from human workstations to completely automated processes. The information should be able to be transferred to the central planning system in a standardized form, e.g., via HTTPS, regardless of the degree of automation, to be considered in the flexible control. This communication should be possible both synchronously and asynchronously. Single-stage, linear, as well as modular production scenarios, such as matrix production, should be able to be represented in the process flexibility. Supporting processes, such as a transport process, are also integrated, but no external processes are considered due to safety precautions.

Req. 2 (PF): Robustness of the Processes. The flexible manufacturing processes should explicitly support redundant components and production paths to achieve process stability and a high level of robustness. This promises a high level of resilience to potential errors, as alternative paths are available in the event of failures, allowing the production processes to continue.

Req. 3 (PF): Variable Scalability of the Processes. The process flexibility should have variable scalability in its components. The batch size of the products to be manufactured should be freely variable, as should the number of process stations. Additionally, the stations should allow for flexible reconfiguration.

Req. 4 (PF): Process Data. Four sub-requirements are defined for the data generated as part of the flexible manufacturing processes. These are presented in the following.

Req. 4a (PF): Data Capture. In the context of process flexibility, it should be possible to prioritize certain parts of the available data during the collection and processing to allow for lower latency by ignoring irrelevant data. Additionally, the volume of data to be captured and processed should also be variable. For example, it should be allowed to increase the volume captured above the normal level during a failure analysis to better understand the nature of the failure.

Req. 4b (PF): Near-Real-Time Data Processing. The data from flexible manufacturing processes should be provided with minimal latency to enable near real-time monitoring and dynamic integration. This allows for quick reactions and interventions to rectify failures or events that impair performance. Automated checking and logging of data should be variable, allowing users to select process-specific data and choose a logging level.

Req. 4c (PF): Data Representation. The data generated during the flexible manufacturing processes is transferred to a data model. This model should be based on established standards such as the *Asset Administration Shell* (AAS) [80] or the *Open Platform Communications Unified Architecture* (OPC-UA) [81]. So, the data can be processed automatically by as wide a range of systems as possible, or existing converters can be used. The model for data representation should support different types of data sources, such as process descriptions of individual tasks, shifts of different employees, lists of parts, machine status, images for quality inspection, measurement curves, or manually recorded data.

Req. 4d (PF): Data Access. It should be possible to grant access to data on different levels of abstraction. For example, sometimes it is beneficial to be able to access unprocessed data instead of the already processed abstracted data. The availability and origin of the data should be transparently traceable, with clear and transparent access authorizations ensuring data sovereignty. Data lifetimes should be planned according to user requirements, ranging from a few days to permanently.

Req. 5 (PF): Optimization Criteria Support for Intelligent Process Planning. When planning flexible manufacturing processes, it should be possible to consider various optimization criteria, both at a global and a local level. At the global level, e.g., this could be the shop floor, for which the production times are optimized, or at the local level, e.g., a machine, or product for which the costs incurred are to be minimized. It should also be possible to define more abstract target values for the optimization, such as a high error tolerance for the production process or sustainability criteria, such as minimizing the CO_2 footprint.

4.2. Requirements for Process Discovery

Process Discovery (PD) is a phase in the BPM lifecycle (see Section 2.1) investigating how the existing processes are to be documented in the form of PMLs [9] (p. 22). The requirements identified regarding this and addressing process flexibility for control as defined in Section 4.1 are presented in the following.

Req. 1 (PD): Representation of the Entire Control Flow. The PML must enable a representation of the control flow as it is present in business processes. This can only be an imperative language (see Section 2.3). However, the PML should further enables the representation of possible branches running in parallel and the associated gates within the control flows to be able to simulate real production processes. The data required for execution of the process must also be presentable.

Req. 2 (PD): Compatibility with ERP and MES Systems. The PML should be compatible with established *Enterprise Resource Planning* (ERP) and *Manufacturing Execution System* (MES) systems to minimize the hurdles for industrial use. This is achieved through generic connectors so that only minimal adaptations are necessary for the respective system. In addition, standardized data spaces should be used to further simplify the connection, e.g., by using established information models such as the AAS [80] or the *Gaia-X Federation Services* (GXFS) [82]. This should enable execution in distributed cloud-based data infrastructure ecosystems, for example, with an edge-cloud continuum [6].

Req. 3 (PD): Interfaces to Other Process Modeling Languages. Flexible process control requires suitable forms of representation of manufacturing processes that can be interpreted by computers and used as input for AI-based planning approaches. The aim is to achieve a broad compatibility, e.g., with processes that are already formally represented, but also with unstructured forms, such as natural language. Therefore, the systems implemented based on the PML should contain interfaces that are as generic as possible so that converters or adapters for other PMLs can be connected to them. To make this possible, the PML should be as established and as well documented as possible.

Req. 4 (PD): Included Data. In addition to the data that describes the actual processes (see Req. 1 (PD)), manufacturing companies also have a wide range of other information that can be useful for controlling processes. The PML used for the flexible control of production processes should, therefore, further include information about the process environment in addition to information about the actual production process. This information can include, e.g., information about employee shifts or the shop floor of the workshop.

Req. 5 (PD): Semantic Information. To enable the processes using AI-planning procedures, it is necessary that the processes are described semantically. The PML should allow the semantic modeling of the individual work steps. It must be possible to query the current status of the running process at any time. An established standard for this is the OWL-S standard [83], in which services contain the following descriptions:

(a) *Inputs*: information passed to the service,
(b) *Outputs*: information returned after execution,
(c) *Preconditions*: conditions that need to be met in order for the service to be executable,
(d) *Effects*: changes to the world state after the service has been executed.

These attributes or a suitable equivalent must be supported by the PML. The need for semantic enrichment of the process representation is already identified in the research area of Semantic Business Process Management [84].

5. Analysis of Process Modeling Languages

In this section, the PMLs collected in Section 3 are compared based on the requirements from Section 4.2. Due to the scope of the 27 languages, a pre-selection is first made based on the requirements, which is presented in Section 5.1. The remaining seven PMLs are then examined in detail in Section 5.2.

5.1. Pre-Selection

In a preliminary step, the 27 identified imperative PMLs are filtered. The aim is to be able to subsequently examine possible suitable languages in greater detail. The PMLs are therefore evaluated based on their popularity. Req. 2 (PD) specifies that the selected PML should be compatible with common ERP and MES systems. This depends on the popularity of the PML, as there tend to be more systems that support popular languages or converters that convert them into another popular and common language. Req. 3 (PD) also includes the last point for the interfaces to other PMLs. If there are no freely available converters, there is at least a high probability that published approaches exist for converting one language into another. The popularity of the PMLs can therefore be used as a further criterion for the preselection.

Therefore, we introduce a popularity score for relevant PMLs, which is composed of the following three dimensions:

1. The *Occurrence in Literature Reviews* (see Section 2.4) provides information about the relevance and acceptance of the language in the scientific community. Frequent occurrence indicates high popularity, while rare or no occurrence indicates low acceptance.
2. The number of *Citations of the Basic Paper* that introduces the PML or scientifically elaborates on it is used, whereby citations of basic papers are more meaningful than those of elaborating publications. For some languages, it should be noted that no basic paper is available or that this is not a scientifically established paper (e.g., for BPDM [55], BPMN [56], SPEM [73], or UML [85]).
3. The number of *Search Results on Google Scholar* (https://scholar.google.de) provides information on how frequently the PML is used and how established it is. Here, the languages are searched for in written form and not as abbreviations.

The popularity score is made up of the three dimensions, whereby restrictions such as the lack of availability of a basic paper are considered. The underlying queries were carried out in January 2023. Based on the results, 20 PMLs can be classified as unsuitable using the popularity score. The imperative languages that are considered in more detail are BPMN [56], EPC [63], and *Petri Nets* (PN) [70], with the subtypes of CPN [58], *Data Petri Nets* (DPN) [61], *Resource Event Agent* (REA) [72], and UML-AD [75].

5.2. Detailed Comparison

A simple example process is considered to examine the basic suitability of the PMLs for modeling production processes, which is demanded by Req. 1 (PD). This describes an exemplary process in the real-world print module located in the SmartFactory^{KL}. The modeling of this example process in the individual PMLs can be used to determine whether a language is suitable for modeling sequences of work steps with branches and data elements. The process represented in the several PMLs is shown in the following, where the languages are discussed.

To assess compatibility with existing ERP and MES systems (see Req. 2 (PD)), the data from the popularity score analysis from Section 5.1 is also considered. When modeling the example process, it is also investigated which tools are available for creating process models in the respective language. The availability of tools for modeling process models in the evaluated PMLs can also be used to assess the prevalence of the respective language. To assess how the PML can be connected to industrial systems by simple interfaces (see Req. 3 (PD)), it is investigated whether there is a standardized file format for which a parser or converter can be implemented. In addition, the possibilities offered by the PMLs for modeling data (see Req. 4 (PD)) and the semantics of the processes (see Req. 5 (PD)) are considered.

BPMN is very well established (Req. 2 (PD)) due to numerous scientific publications and widespread use in practice. In this language, control flows can be represented by various events, gateways, and directed edges between these elements (Req. 1 (PD), see Figure 2). It offers a standardized XML format for system exchange (Req. 3 (PD)), which facilitates integration into other systems. However, BPMN has limited possibilities for data integration and process semantics (Req. 4 (PD) and Req. 5 (PD)), as annotations of data have no influence on the control flow and effects of work steps cannot be represented.

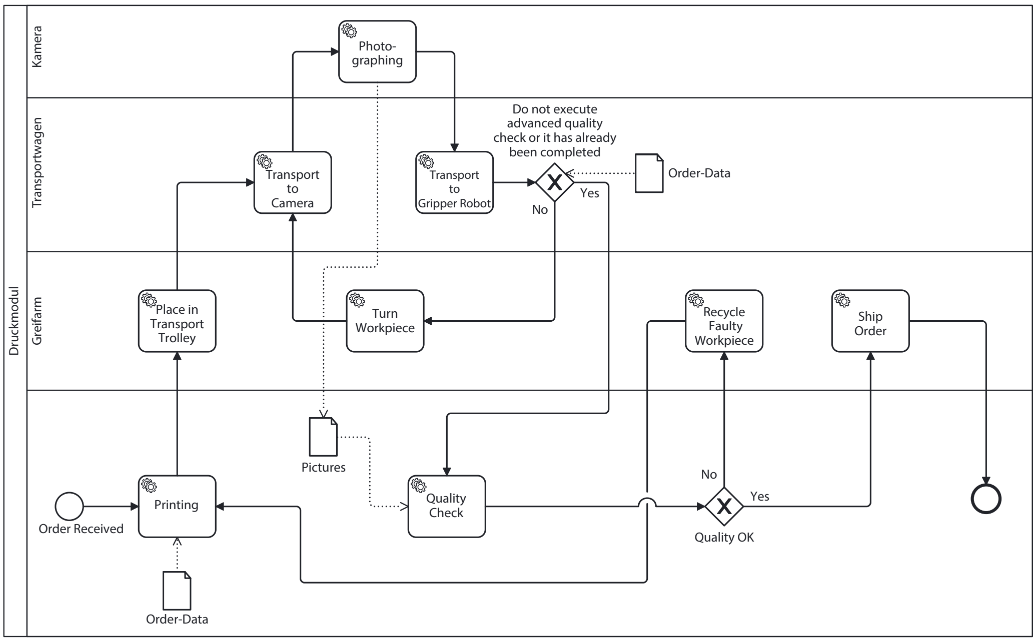

Figure 2. The Example Workflow Represented as BPMN.

EPCs are relatively well established (Req. 2 (PD)) with some scientific publications but lack a standard format for exchange (Req. 3 (PD)), which makes integration more difficult. In this language, control flows can be represented by events, functions, logical link operators, and directed edges between these (Req. 1 (PD), see Figure 3). EPCs are unsuitable for data integration and semantics (Req. 4 (PD) and Req. 5 (PD)), as data artifacts remain unspecific and the semantics of links cannot be modeled.

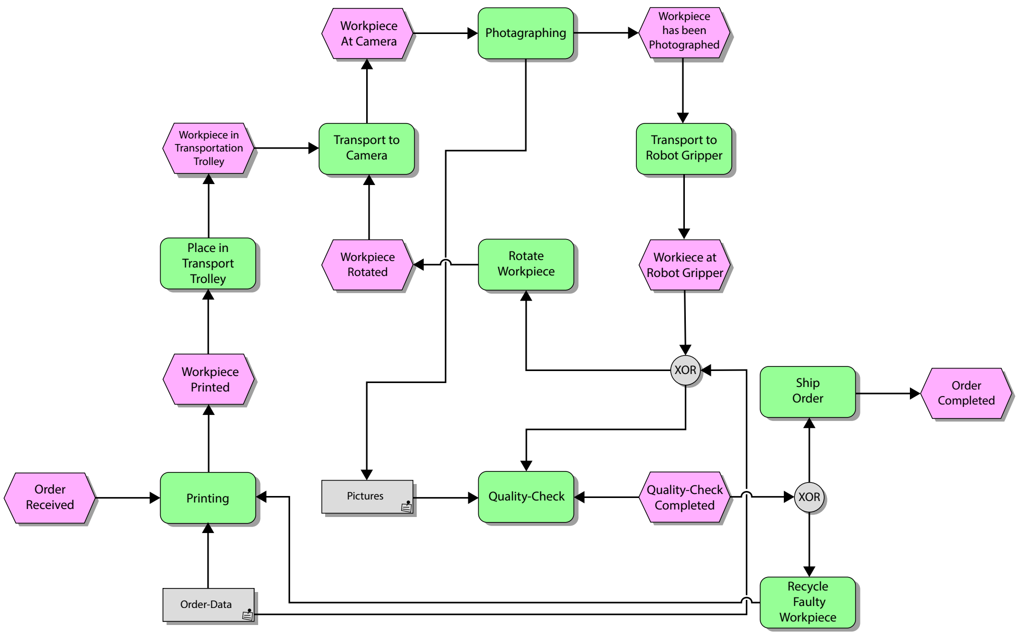

Figure 3. The Example Workflow Represented as EPC.

PNs are well established (Req. 2 (PD)) with the most scientific publications and widespread use. A standard exchange format (Req. 3 (PD)), the *Petri Net Markup Language*

(PNML) [86] (https://www.pnml.org/), is provided, which facilitates integration [87]. Control flows can be represented by places, transitions, and directed edges (Req. 1 (PD), see Figure 4). However, PNs do not meet the requirements for data integration and semantics (Req. 4 (PD) and Req. 5 (PD)), as only the control flow perspective can be modeled.

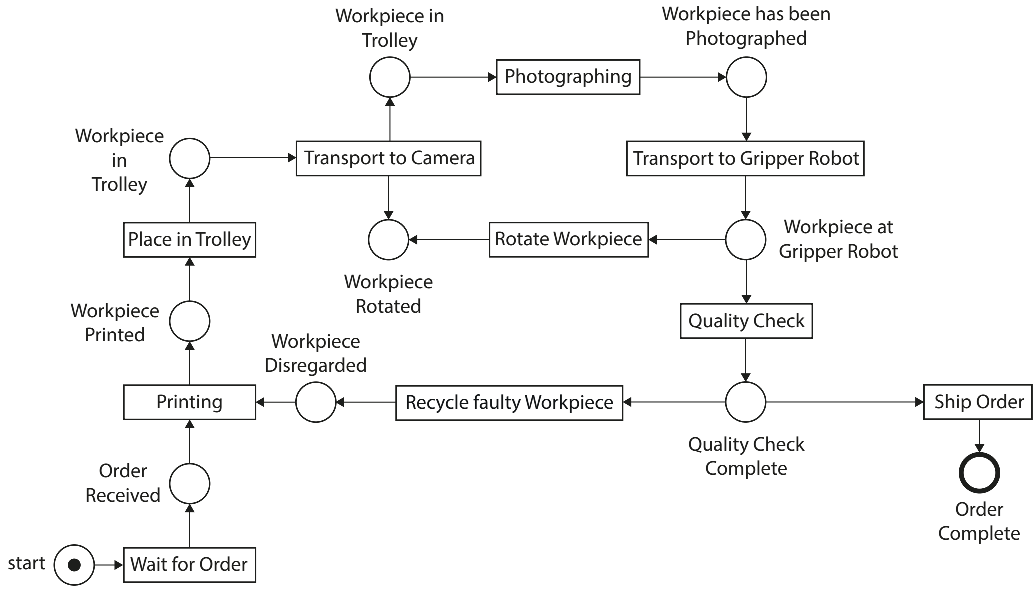

Figure 4. The Example Workflow Represented as PN.

DPNs are less established than regular PNs (Req. 2 (PD)), but tools such as ProM (https://promtools.org/) are available. As in PNs, control flows can be represented by places, transitions, and directed edges, which are additionally connected to data that can be used to represent further conditions and decision logics (Req. 1 (PD), see Figure 5). CPNs can be exchanged in a de facto standard format (Req. 3 (PD)), which enables integration. They are well suited for data integration (Req. 4 (PD)), as variables can be defined and partially fulfill the semantics (Req. 5 (PD)), as preconditions can be modeled, but effects of process steps cannot.

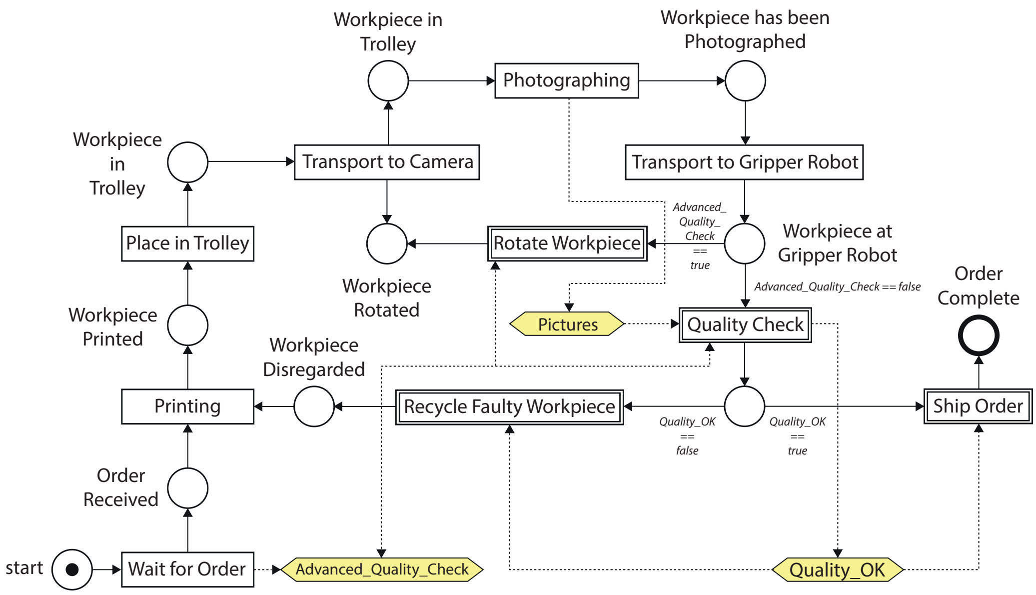

Figure 5. The Example Workflow Represented as DPN.

CPNs are relatively well established (Req. 2 (PD)) and allow the modeling of data on tokens. However, there are few tools and no established exchange standard (Req. 3 (PD)). As in PNs, control flows can be represented by places, transitions, and directed edges, whereby these can be used to provide additional information and conditions (Req. 1 (PD), see Figure 6). CPNs are well suited for data integration and semantics (Req. 4 (PD) and Req. 5 (PD)). Data can be attached to tokens, and these can influence the control flow.

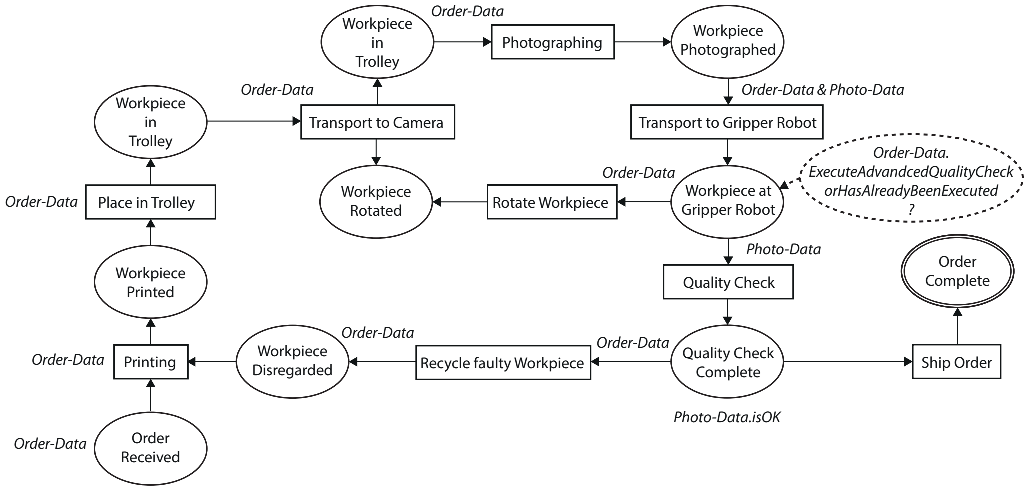

Figure 6. The Example Workflow Represented as CPN.

REA is unsuitable as it does not offer the basic functionality of process representation (Req. 1 (PD)). Although it can model individual actions and their inputs and outputs, it cannot represent chains of activities or gates. It was therefore not possible to model the example process with this PML, which is why an illustration of the REA process is missing. The other requirements (Req. 2 (PD)–Req. 5 (PD)) are not assessable due to the specifications of the REA language.

UML-ADs are a specification of UML [88] that focuses on the representation of workflows. There, control flows are represented by activities, decision nodes, gates, and directed control flow edges (Req. 1 (PD), see Figure 7). The AD extension is common (Req. 2 (PD)) due to the widespread use of UML, but unsuitable for exchanging data (Req. 3 (PD)) as there are no known tools that support automated conversion to another format. It is also unsuitable for data integration (Req. 4 (PD)) and process semantics (Req. 5 (PD)), as there are no standard methods for modeling data and effects.

Table 2 shows an overview of the suitability of the PMLs. The comparison shows that none of the PMLs considered for process modeling in their original form is suitable for meeting all requirements. The best suitability for the requirements can be determined for BPMN and CPN. Both allow the entire control flow to be represented (see Req. 1 (PD)). BPMN stands out because it is a well-established standard (see Req. 2 (PD)) and has a high level of compatibility for connecting to various systems (see Req. 3 (PD)). CPN, on the other hand, enables the best integration of data into the process (see Req. 4 (PD)). In contrast to BPMN, CPN offers some options for modeling the process semantically (see Req. 5 (PD)). However, even with CPN, this is not sufficient to describe the processes according to the desired scope in the requirement. This is therefore not enough to be able to carry out the desired process control. Therefore, an extension of the CPN as well as the BPMN approach or finding another solution for semantic modeling is necessary in any case. Thus, the possibilities for semantic modeling provided by CPN do not represent an advantage over BPMN. On this basis, BPMN can be chosen as the PML for the production processes, for which a suitable extension must be developed additionally.

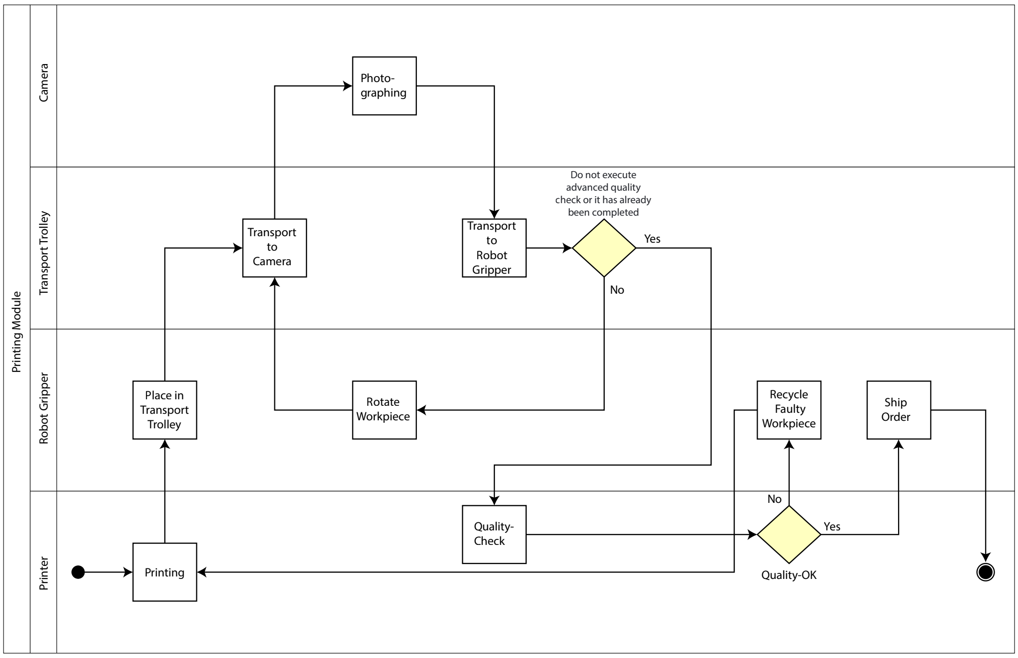

Figure 7. The Example Workflow Represented as UML-AD.

Table 2. Comparison of the Addressed Requirements Regarding Process Discovery (PD) of the Seven Pre-Filtered PMLs. (✓ = Suitable, ✗ = Unsuitable, (✓) = Integration Possible (Extension/Adaptation Necessary), ⁄ = Not Assessable.)

PML	Req. 1	Req. 2	Req. 3	Req. 4	Req. 5
BPMN	✓	✓	✓	(✓)	✗
EPC	✓	(✓)	✗	(✓)	✗
PN	✓	✓	✓	✗	✗
CPN	✓	(✓)	(✓)	✓	(✓)
DPN	✓	✗	(✓)	✓	(✓)
REA	✗	⁄	⁄	⁄	⁄
UML AD	✓	✓	(✓)	✗	✗

6. Integration of Semantics

As determined in Section 5, none of the existing PMLs completely fulfills the identified requirements. BPMN is perceived as the most suitable language, in which the possibility of adding the semantics required for flexible control can be integrated. Previous research [84] also identifies BPMN as the most promising PML in BPM without the context of process flexibility for control but also emphasized the need to integrate semantic information. In the following, the possibility of a BPMN extension (see Section 6.1) as well as the outsourcing of the semantics (see Section 6.2) are presented and discussed. Afterward, the preferred solution is justified based on the discussion of the two approaches in Section 6.3.

6.1. Extension of BPMN

The possibility of extending BPMN modeling can be divided into two variants:

(a) *Use of Existing Extensions*: There are already some extensions of *BPMN* that enable a semantic description of the processes [89–91]. Ardito et al. [92] as well as Braun and Esswein [93] provide an overview of existing approaches that focus on semantics.

(b) *Design of New Custom Extension*: If the existing extension approaches turn out to be unsuitable for flexible control, this variant would be an alternative. This would make it possible to extend the *BPMN* standard with precisely those elements that process planning requires.

The advantage of the BPMN extension option in both sub-variants is that the complete description of the services is encapsulated in a single structure. This means that it is not necessary to reference between different structures, such as *BPMN* and an AAS, to obtain all the information for planning and executing processes. A significant disadvantage of this approach, however, is the loss of compatibility that would result from deviations from the established BPMN 2.0 standard. On the one hand, the workflow management systems that are typically used to execute the processes generally do not support a specially designed extension or even an existing extension. This will result in the systems either no longer accepting the BPMN models and therefore throwing an error message or not being able to process the additional information. Furthermore, established tools for modeling BPMN will generally not support the extended models. In the case of a proprietary extension, it is impossible that suitable tools exist. It would therefore be necessary to design custom tools for modeling, which in turn means a significant amount of additional work. Another disadvantage of this approach is the redundancies that would result from integrating the semantics in the individual process instances. Many of the processes within an industrial company share work steps, e.g., a drilling task in a manufacturing workflow. If the semantics are integrated into the BPMN models, each of these processes would contain the complete semantic description of this work step.

6.2. Outsourcing of Semantics

Alternatively, the semantics can be separated from the actual *BPMN* model, and references to this external structure can be saved in the model instead of mapping the semantics. For example, the semantics can be stored in a classic ontology so that only the *URI*s (*Uniform Resource Identifier*) are referenced in the process model. Malburg et al. [94] use such an approach, in which the processes are modeled in *BPMN* due to the necessary compatibility with a workflow management system, and the semantics are modeled in a separate ontology. The advantage of outsourcing the semantics to a separate structure is that the design of this structure can be based on established standards, such as the AAS [95] or an ontology [96]. This simplifies modeling and makes it easier to use the semantic information for other purposes, as a structure based on established standards is supported by many systems. Another advantage that simplifies modeling is the ability to use existing ontologies or AAS submodels. For example, ontologies with building blocks for process descriptions already exist (e.g., OWL-S [97]). The use of existing structures reduces the effort required for modeling and at the same time has a positive influence on the compatibility of the model, which is then not only limited to BPMN but also holds for other representation languages. In addition, knowledge is outsourced to a single source of truth (cf. [98]), so that all knowledge for a company is collected in one place and is based on a shared semantic model. An obvious disadvantage of this approach is the need for a separate structure to fully describe processes semantically. This requires referencing between different structures to collect all the process information required for planning. This approach therefore increases the complexity of information access.

6.3. Preferred Solution

As discussed, the semantic integration into the BPMN language has significant disadvantages. Separating the semantic information from the actual process model, on the other hand, has clear benefits and only the drawback of increased complexity of information access. For this reason, the possibility of outsourcing the semantics can be preferred. While this should be based on established standards, existing structures such as ontologies or AAS submodels are to be reused wherever possible.

7. Conclusions and Future Work

In this paper, the flexible control of industrial production processes in the context of AWM and I4.0 is examined. Thereby, it is investigated what constitutes flexibility considering industrial process control and what criteria are most important for facilitating the use of AWM in an industrial setting. For this purpose, experts from several companies operating in industrial manufacturing, as well as practice-oriented and renowned research institutes, were consulted to establish requirements for allowing the flexible control of these processes. One of the findings is that semantic process models are needed for this goal. Therefore, further requirements regarding the PML used to create those models are established. A comprehensive literature review was conducted following the requirement analysis to identify promising PMLs that have not yet been evaluated for AWM applications and could enhance production process flexibility. The identified PMLs are systematically evaluated and compared against the established requirements. As a result, both BPMN and CPNs emerged as the most suitable candidates, despite their shared limitation in semantic expressiveness for modeling AI planning outputs. Since the BPMN standard is much more established through, it is already compatible with many ERP and MES systems in use. Therefore, BPMN is chosen as the best suited PML. Multiple options are explored that could be used to solve the problem of defining detailed semantics for BPMN process models. Therefore, using already existing BPMN extensions or creating a new extension are considered. However, due to this, resulting in losing most of the compatibility with existing ERP and MES systems, the option of outsourcing the detailed semantic information to another data structure and referencing it in the BPMN is investigated and preferred.

Future research will explore how these outsourced semantics can be organized, e.g., in an ontology, and referenced in BPMN models. Additionally, it might be examined how generic semantic models for planning domain knowledge can be integrated with BPMN models. A proof of concept that enables the automatic generation of planning domains based on domain ontologies is already presented in previous research [26]. This concept should be further investigated and combined with semantic BPMN models to facilitate a full AI planning pipeline for flexible control of production facilities using generic models applicable to arbitrary domains. Further research can address how planning system outputs can automatically generate BPMN models representing the production plan. The time series data generated during plan execution shall be explored as input for analysis and optimization of control processes [6,99]. Moreover, the modularity identified in the context of process flexibility (see Req. 1 (PD)) requires further examination. For orchestration in a shared environment, such as an edge-cloud continuum [6], this modularity enables distributed planning. This can be implemented, for example, through hierarchical planning [100] by creating an abstract plan at the distributed level and concretizing it at the individual executioners.

Author Contributions: Conceptualization, A.S. and D.J.; Methodology, A.S. and D.J.; Validation, A.S. and D.J.; Investigation, A.S. and D.J.; Writing—Original Draft Preperation, A.S. and D.J.; Writing—Review and Editing, L.M., S.B. and R.B.; Visualization, A.S. and D.J.; Supervision, L.M., S.B. and R.B.; Project Administration, S.B. and R.B.; Funding Aquisition, S.B. and R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the Federal Ministry for Economic Affairs and Climate Action under grant No. 01MD22002C *EASY*.

Data Availability Statement: The original contributions presented in this study are included in the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

AAS	*Asset Administration Shell*
AI	*Artificial Intelligence*
AWM	*Adaptive Workflow Management*
BPMN	*Business Process Model and Notation*
BPM	*Business Process Management*
BWW	*Bunge-Wand-Weber*

CBR	*Case-Based Reasoning*
CPN	*Couloured Petri Nets*
DPN	*Data Petri Nets*
EPC	*Event-Driven Process Chains*
ERP	*Enterprise Resource Planning*
I4.0	*Industry 4.0*
IoT	*Internet of Things*
MES	*Manufacturing Execution System*
PML	*Process Modeling Language*
PN	*Petri Nets*
REA	*Resource Event Agent*
UML-AD	*Unified Modeling Language Activity Diagram*
UML	*Unified Modeling Language*
YAWL	*Yet Another Event-Driven Process Chain*

References

1. Gilchrist, A. *Industry 4.0: The Industrial Internet of Things*; Apress: Berkeley, CA, 2016. https://doi.org/10.1007/978-1-4842-2047-4.
2. Janiesch, C.; Koschmider, A.; Mecella, M.; Weber, B.; Burattin, A.; Ciccio, C.D.; Gal, A.; Kannengiesser, U.; Mannhardt, F.; Mendling, J.; et al. The Internet-of-Things Meets Business Process Management. A Manifesto. *IEEE Syst. Man Cybern. Mag.* **2020**, *6*, 34–44. https://doi.org/10.1109/MSMC.2020.3003135.
3. Malburg, L.; Hoffmann, M.; Bergmann, R. Applying MAPE-K control loops for adaptive workflow management in smart factories. *J. Intell. Inf. Syst.* **2023**, *61*, 83–111. https://doi.org/10.1007/s10844-022-00766-w.
4. De Luzi, F.; Leotta, F.; Marrella, A.; Mecella, M. On the Interplay Between Business Process Management and Internet-Of-Things: A Systematic Literature Review. *Bus. Inf. Syst. Eng.* **2024**. https://doi.org/10.1007/s12599-024-00859-6.
5. Marrella, A. What Automated Planning Can Do for Business Process Management. In *Proceedings of the BPM 2017 Workshops*; LNBIP; Springer: Berlin/Heidelberg, Germany, 2018; Volume 308, pp. 7–19. https://doi.org/10.1007/978-3-319-74030-0_1.
6. Schultheis, A.; Alt, B.; Bast, S.; Guldner, A.; Jilg, D.; Katic, D.; Mundorf, J.; Schlagenhauf, T.; Weber, S.; Bergmann, R.; et al. EASY: Energy-Efficient Analysis and Control Processes in the Dynamic Edge-Cloud Continuum for Industrial Manufacturing. *Künstliche Intell.* **2024**. https://doi.org/10.1007/s13218-024-00868-3.
7. Schonenberg, H.; Mans, R.; Russell, N.; Mulyar, N.; van der Aalst, W.M.P. Process Flexibility: A Survey of Contemporary Approaches. In *Advances in Enterprise Engineering I. 4th International Workshop CIAO! and 4th International Workshop EOMAS, Held at CAiSE 2008, Montpellier, France, June 16–17, 2008, Proceedings*; LNBIP; Springer: Berlin/Heidelberg, Germany, 2008; Volume 10, pp. 16–30. https://doi.org/10.1007/978-3-540-68644-6_2.
8. Reichert, M.; Weber, B. *Enabling Flexibility in Process-Aware Information Systems—Challenges, Methods, Technologies*; Springer: Berlin/Heidelberg, Germany, 2012. https://doi.org/10.1007/978-3-642-30409-5.
9. Dumas, M.; Rosa, M.L.; Mendling, J.; Reijers, H.A. *Fundamentals of Business Process Management*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2018. https://doi.org/10.1007/978-3-662-56509-4.
10. Green, C. Application of Theorem Proving to Problem Solving. In Proceedings of the 1st IJCAI, IJCAI'69, Washington, DC, USA, 7–9 May 1969; pp. 219–240.
11. Haslum, P. Admissible Heuristics for Automated Planning. Ph.D. Thesis, Institutionen för Datavetenskap—Umeå Universitet, Umeå, Sweden, 2006.
12. Aamodt, A.; Plaza, E. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Commun.* **1994**, *7*, 39–59. https://doi.org/10.3233/AIC-1994-7104.
13. Malburg, L.; Brand, F.; Bergmann, R. *Adaptive Management of Cyber-Physical Workflows by Means of Case-Based Reasoning and Automated Planning*; LNBIP; Springer: Cham, Switzerland, 2022; Volume 466, pp. 79–95. https://doi.org/10.1007/978-3-031-26886-1_5.
14. Chinosi, M.; Trombetta, A. BPMN: An introduction to the standard. *Comput. Stand. Interfaces* **2012**, *34*, 124–134. https://doi.org/10.1016/j.csi.2011.06.002.
15. Hu, J.; Aghakhani, G.; Hasic, F.; Serral, E. An Evaluation Framework for Design-Time Context-Adaptation of Process Modelling Languages. In *The Practice of Enterprise Modeling 10th IFIP WG 8.1. Working Conference, PoEM 2017, Leuven, Belgium, November 22–24, 2017, Proceedings* ; LNBIP; Springer: Cham, Switzerland, 2017; Volume 305, pp. 112–125. https://doi.org/10.1007/978-3-319-70241-4_8.
16. Kožíšek, F.; Vrana, I. Business Process Modelling Languages. *AGRIS On-Line Pap. Econ. Inform.* **2017**, *9*, 39–49. https://doi.org/10.22004/ag.econ.263955.
17. Brouns, N.; Tata, S.; Ludwig, H.; Asensio, E.S.; Grefen, P. Modeling IoT-aware Business Processes—A State of the Art Report. *arXiv* **2018**, arXiv.1811.00652. https://doi.org/10.48550/arXiv.1811.00652.

18. von Garrel, J.; Schenk, M.; Seidel, H. Flexibilisierung der Produktion—Maßnahmen und Status-Quo. *Flex. Produktionskapazität Innov. Manag.* **2014**, 81–126. https://doi.org/10.1007/978-3-642-39896-4_2.
19. Houy, C.; Reiter, M.; Fettke, P.; Loos, P. Towards Green BPM—Sustainability and Resource Efficiency through Business Process Management. In *Proceedings of the BPM Workshops*; LNBIP; Springer: Berlin/Heidelberg, Germany, 2010; Volume 66, pp. 501–510. https://doi.org/10.1007/978-3-642-20511-8_46.
20. Marrella, A. Automated Planning for Business Process Management. *J. Data Semant.* **2019**, *8*, 79–98. https://doi.org/10.1007/s13740-018-0096-0.
21. Zhang, Y.; Ge, L. Method for process planning optimization with energy efficiency consideration. *J. Adv. Manuf. Technol.* **2015**, *77*, 2197–2207. https://doi.org/10.1007/s00170-014-6631-8.
22. Zonta, T.; Da Costa, C.A.; da Rosa Righi, R.; de Lima, M.J.; da Trindade, E.S.; Li, G.P. Predictive maintenance in the Industry 4.0: A systematic literature review. *Comput. Ind. Eng.* **2020**, *150*, 106889. https://doi.org/10.1016/j.cie.2020.106889.
23. Schultheis, A. Exploring a Hybrid Case-Based Reasoning Approach for Time Series Adaptation in Predictive Maintenance. In Proceedings of the 32nd ICCBR Workshops, CEUR Workshop Proceedings, Mérida, Mexico, 1 July 2024; Volume 3708, pp. 230–235.
24. Bertrand, Y.; Belle, R.V.; Weerdt, J.D.; Serral, E. Defining Data Quality Issues in Process Mining with IoT Data. In *Process Mining Workshops ICPM 2022 International Workshops, Bozen-Bolzano, Italy, October 23–28, 2022, Revised Selected Papers*; LNBIP; Springer: Cham, Switzerland, 2022; Volume 468, pp. 422–434. https://doi.org/10.1007/978-3-031-27815-0_31.
25. Schultheis, A.; Malburg, L.; Grüger, J.; Weich, J.; Bertrand, Y.; Bergmann, R.; Serral Asensio, E. Identifying Missing Sensor Values in IoT Time Series Data: A Weight-Based Extension of Similarity Measures for Smart Manufacturing. In *Case-Based Reasoning Research and Development 32nd International Conference, ICCBR 2024, Merida, Mexico, July 1–4, 2024, Proceedings*; LNCS; Springer: Cham, Switzerland, 2024; Volume 14775, pp. 240–257. https://doi.org/10.1007/978-3-031-63646-2_16.
26. Malburg, L.; Klein, P.; Bergmann, R. Converting semantic web services into formal planning domain descriptions to enable manufacturing process planning and scheduling in industry 4.0. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106727. https://doi.org/10.1016/j.engappai.2023.106727.
27. Hammond, K.J. Case-Based Planning: A Framework for Planning from Experience. *Cogn. Sci.* **1990**, *14*, 385–443. https://doi.org/10.1207/s15516709cog1403_3.
28. Curtis, B.; Kellner, M.I.; Over, J. Process Modeling. *Commun. ACM* **1992**, *35*, 75–90. https://doi.org/10.1145/130994.130998.
29. Workflow Management Coalition. *Terminol. Gloss.* **1999** .
30. Mili, H.; Tremblay, G.; Jaoude, G.B.; Lefebvre, E.; Elabed, L.; El-Boussaidi, G. Business process modeling languages: Sorting through the alphabet soup. *ACM Comput. Surv.* **2010**, *43*, 1–56. https://doi.org/10.1145/1824795.1824799.
31. Fahland, D.; Lübke, D.; Mendling, J.; Reijers, H.; Weber, B.; Weidlich, M.; Zugal, S. Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In *Enterprise, Business-Process and Information Systems Modeling 10th International Workshop, BPMDS 2009, and 14th International Conference, EMMSAD 2009, held at CAiSE 2009, Amsterdam, The Netherlands, June 8–9, 2009, Proceedings*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 353–366. https://doi.org/10.1007/978-3-642-01862-6_29.
32. Conradi, R.; Jaccheri, M.L. Process Modelling Languages. In *Software Process: Principles, Methodology, Technology*; LNCS; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1500, pp. 27–52. https://doi.org/10.1007/3-540-49205-4_3.
33. Luo, W.; Alex Tung, Y. A framework for selecting business process modeling methods. *Ind. Manag. Data Syst.* **1999**, *99*, 312–319. https://doi.org/10.1108/02635579910262535.
34. Söderström, E.; Andersson, B.; Johannesson, P.; Perjons, E.; Wangler, B. Towards a Framework for Comparing Process Modelling Languages. In *Advanced Information Systems Engineering 14th International Conference, CAiSE 2002 Toronto, Canada, May 27–31, 2002 Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2348, pp. 600–611. https://doi.org/10.1007/3-540-47961-9_41.
35. Aguilar-Savén, R.S. Business process modelling: Review and framework. *Int. J. Prod. Econ.* **2004**, *90*, 129–149. https://doi.org/10.1016/S0925-5273(03)00102-6.
36. Nysetvold, A.G.; Krogstie, J. Assessing Business Processing Modeling Languages Using a Generic Quality Framework. In Proceedings of the 10th EMMSAD, CEUR Workshop Proceedings, Porto, Portugal, 13–14 June 2005; Volume 363, pp. 159–170.
37. List, B.; Korherr, B. An Evaluation of Conceptual Business Process Modelling Languages. In Proceedings of the SAC 2006, Dijon France, 23–27 April 2006; ACM: New York, NY, USA, 2006; pp. 1532–1539. https://doi.org/10.1145/1141277.1141633.
38. Korherr, B. Business Process Modelling—Languages, Goals, and Variabilities. Ph.D. Thesis, Technische Universität Wien, Vienna, Austria, 2008.
39. Recker, J.; Rosemann, M.; Indulska, M.; Green, P.F. Business Process Modeling—A Comparative Analysis. *J. Assoc. Inf. Syst.* **2009**, *10*, 1. https://doi.org/10.17705/1jais.00193.
40. zur Muehlen, M.; Indulska, M. Modeling languages for business processes and business rules: A representational analysis. *Inf. Syst.* **2010**, *35*, 379–390. https://doi.org/10.1016/j.is.2009.02.006.
41. Campos, A.C.S.M.; de Almeida, A.T. Multicriteria framework for selecting a process modelling language. *Enterp. Inf. Syst.* **2016**, *10*, 17–32. https://doi.org/10.1080/17517575.2014.906047.
42. García-Borgoñón, L.; Barcelona, M.A.; García-García, J.A.; Alba, M.; Escalona, M.J. Software process modeling languages: A systematic literature review. *Inf. Softw. Technol.* **2014**, *56*, 103–116. https://doi.org/10.1016/j.infsof.2013.10.001.
43. Goedertier, S.; Vanthienen, J.; Caron, F. Declarative business process modelling: Principles and modelling languages. *Enterp. Inf. Syst.* **2015**, *9*, 161–185. https://doi.org/10.1080/17517575.2013.830340.

44. Farshidi, S.; Kwantes, I.B.; Jansen, S. Business process modeling language selection for research modelers. *Softw. Syst. Model.* **2024**, *23*, 137–162. https://doi.org/10.1007/s10270-023-01110-8.
45. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; EBSE 2007-001; Keele University: Keele, UK; Durham University: Durham, UK, 2007.
46. Reichert, M.; Dadam, P. ADEPT$_{flex}$-Supporting Dynamic Changes of Workflows Without Losing Control. *J. Intell. Inf. Syst.* **1998**, *10*, 93–129. https://doi.org/10.1023/A:1008604709862.
47. Lu, R.; Sadiq, S.W.; Governatori, G. On managing business processes variants. *Data Knowl. Eng.* **2009**, *68*, 642–664. https://doi.org/10.1016/j.datak.2009.02.009.
48. Pesic, M.; van der Aalst, W.M.P. A Declarative Approach for Flexible Business Processes Management. In *Business Process Management Workshops BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4–7, 2006, Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4103, pp. 169–180. https://doi.org/10.1007/11837862_18.
49. Goedertier, S.; Vanthienen, J. Designing Compliant Business Processes with Obligations and Permissions. In *Business Process Management Workshops BPM 2006 International Workshops, BPD, BPI, ENEI, GPWW, DPM, semantics4ws, Vienna, Austria, September 4–7, 2006, Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 2006, Volume 4103, pp. 5–14. https://doi.org/10.1007/11837862_2.
50. Ghallab, M.; Knoblock, C.; Wilkins, D.; Barrett, A.; Christianson, D.; Friedman, M.; Kwok, C.; Golden, K.; Penberthy, S.; Smith, D.; et al. *PDDL—The Planning Domain Definition Language*; Technical Report; ISAE: Toulouse, France, 1998.
51. Slaats, T.; Schunselaar, D.M.M.; Maggi, F.M.; Reijers, H.A. The Semantics of Hybrid Process Models. In *On the Move to Meaningful Internet Systems: OTM 2016 Conferences Confederated International Conferences: CoopIS, C&TC, and ODBASE 2016, Rhodes, Greece, October 24–28, 2016, Proceedings*; LNCS; Springer: Cham, Switzerland, 2016; Volume 10033, pp. 531–551. https://doi.org/10.1007/978-3-319-48472-3_32.
52. Buchmann, T.; Bank, M.; Westfechtel, B. BXtendDSL: A layered framework for bidirectional model transformations combining a declarative and an imperative language. *J. Syst. Softw.* **2022**, *189*, 111288. https://doi.org/10.1016/J.JSS.2022.111288.
53. Eertink, H.; Janssen, W.; Luttighuis, P.O.; Teeuw, W.B.; Vissers, C.A. A Business Process Design Language. In *FM'99—Formal Methods World Congress on Formal Methods in the Developement of Computing Systems, Toulouse, France, September 20–24, 1999, Proceedings, Volume I*; LNCS; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1708, pp. 76–95. https://doi.org/10.1007/3-540-48119-2_7.
54. Khalaf, R.; Mukhi, N.; Curbera, F.; Weerawarana, S. The Business Process Execution Language for Web Services. In *Process-Aware Information Systems: Bridging People and Software Through Process Technology*; Wiley: Hoboken, NJ, USA, 2005; pp. 317–342. https://doi.org/10.1002/0471741442.ch13.
55. Object Management Group (OMG). *Business Process Definition MetaModel Volume II: Process Definitions*; Object Management Group: Needham, MA, USA, 2008.
56. Object Management Group (OMG). *Business Process Model and Notation (BPMN)*; Version 2.0; Object Management Group: Needham, MA, USA, 2011.
57. Cognini, R.; Hinkelmann, K.; Martin, A. A Case Modelling Language for Process Variant Management in Case-Based Reasoning. In *Business Process Management Workshops BPM 2015, 13th International Workshops, Innsbruck, Austria, August 31–September 3, 2015, Revised Papers*; LNBIP; Springer: Cham, Switzerland, 2015; Volume 256, pp. 30–42. https://doi.org/10.1007/978-3-319-42887-1_3.
58. Schiffers, M.; Wedde, H.F. Analyzing Program Solutions of Coordination problems by CP-Nets. In *Mathematical Foundations of Computer Science 1978 7th Symposium Zakopane, Poland, September 4–8, 1978. Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 1978; Volume 64, pp. 462–473. https://doi.org/10.1007/3-540-08921-7_94.
59. Nikolic, D.; Priami, C.; Zunino, R. A Rule-Based and Imperative Language for Biochemical Modeling and Simulation. In *Software Engineering and Formal Methods 10th International Conference, SEFM 2012, Thessaloniki, Greece, October 1–5, 2012. Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7504, pp. 16–32. https://doi.org/10.1007/978-3-642-33826-7_2.
60. Liu, L.; Özsu, M.T. Data Flow Diagrams. In *Encyclopedia of Database Systems*; Springer: New York, NY, USA, 2009; p. 581. https://doi.org/10.1007/978-0-387-39940-9_2353.
61. Lazic, R.; Newcomb, T.C.; Ouaknine, J.; Roscoe, A.W.; Worrell, J. In Nets with Tokens which Carry Data. *Fundam. Informaticae* **2008**, *88*, 251–274.
62. van Hee, K.M.; Oanea, O.; Sidorova, N. Colored Petri Nets to Verify Extended Event-Driven Process Chains. In *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2005, Agia Napa, Cyprus, October 31–November 4, 2005, Proceedings Part I*; LNCS; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3760, pp. 183–201. https://doi.org/10.1007/11575771_14.
63. Keller, G.; Nüttgens, M.; Scheer, A.W. *Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (epk)*; Institut für Wirtschaftsinformatik—Universität des Saarlandes: Saarbrücken, Germany, 1992.
64. Chapin, N. Flowchart. In *Encyclopedia of Computer Science*; John Wiley and Sons Ltd.: Hoboken, NJ, USA, 2003; pp. 714–716.
65. Bergmann, R.; Gil, Y. Similarity assessment and efficient retrieval of semantic workflows. *Inf. Syst.* **2014**, *40*, 115–127. https://doi.org/10.1016/j.is.2012.07.005.

66. Petrasch, R.; Hentschke, R. Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method. In Proceedings of the 13th JCSSE, Khon Kaen, Thailand, 13–15 July 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–5. https://doi.org/10.1109/JCSSE.2016.7748885.
67. Mayer, R.J.; Menzel, C.P.; Painter, M.K.; Dewitte, P.S.; Blinn, T.; Perakath, B. *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*; Knowledge Based Systems Inc. (KBSI): College Station, TX, USA, 1995.
68. Mukund, M. *Linear-Time Temporal Logic and Büchi Automata*; Tutorial Talk; Winter School on Logic and Computer Science, Indian Statistical Institute: Calcutta, India, 1997.
69. Anderl, R.; Raßler, J. PML, an Object Oriented Process Modeling Language. In *Computer-Aided Innovation (CAI) IFIP 20th World Computer Congress, Proceedings of the Second Topical Session on Computer-Aided Innovation, WG 5.4/TC 5 Computer-Aided Innovation, September 7–10, 2008, Milano, Italy*, 2nd ed.; IFIP Advances in Information and Communication Technology; Cascini, G., Ed.; Springer: Boston, MA, USA, 2008; Volume 277, pp. 145–156. https://doi.org/10.1007/978-0-387-09697-1_12.
70. Petri, C.A. Kommunikation Mit Automaten. Ph.D. Thesis, Universität Hamburg, Hamburg, Germany, 1962.
71. Wilson, J.M. Business Processes: Modelling and Analysis for Re-engineering and Improvement. *J. Oper. Res. Soc.* **1996**, *47*, 595–596. https://doi.org/10.1057/jors.1996.68.
72. McCarthy, W.E. The REA Accounting Model: A Generalized Framework for Accounting Systems Shared Data Environment. *Account. Rev.* **1982**, 554–578.
73. Object Management Group (OMG). *Software Process Engineering Metamodell SPEM 2.0 Revised Submission*; Object Management Group: Needham, MA, USA, 2006.
74. Lamport, L. The Temporal Logic of Actions. *ACM Trans. Program. Lang. Syst.* **1994**, *16*, 872–923. https://doi.org/10.1145/177492.177726.
75. Dumas, M.; ter Hofstede, A.H.M. UML Activity Diagrams as a Workflow Specification Language. In *UML 2001—The Unified Modeling Language. Modeling Languages, Concepts, and Tools 4th International Conference, Toronto, Canada, October 1–5, 2001. Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2185, pp. 76–90. https://doi.org/10.1007/3-540-45441-1_7.
76. Object Management Group. *OMG Unified Modeling Language, Specification*; Version 1.3; Object Management Group: Needham, MA, USA, 2000.
77. Andrews, T.; Curbera, F.; Dholakia, H.; Goland, Y.; Klein, J.; Leymann, F.; Liu, K.; Roller, D.; Smith, D.; Thatte, S.; et al. *Business Process Execution Language for Web Services*; Version 1.1; Standards Proposal by BEA Systems: San Jose, CA, USA; International Business Machines Corporation: Armonk, NY, USA; Microsoft Corporation: Redmond, WA, USA, 2003.
78. Mendling, J.; Neumann, G.; Nüttgens, M. Yet Another Event-Driven Process Chain. In *Business Process Management 3rd International Conference, BPM 2005, Nancy, France, September 5–8, 2005, Proceedings*; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3649, pp. 428–433. https://doi.org/10.1007/11538394_35.
79. Rabiee, F. Focus-Group Interview and Data Analysis. *Proc. Nutr. Soc.* **2004**, *63*, 655–660. https://doi.org/10.1079/PNS2004399.
80. Wei, K.; Sun, J.; Liu, R. A Review of Asset Administration Shell. In Proceedings of the 2019 IEEM, Macao, China, 15–18 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1460–1465. https://doi.org/10.1109/IEEM44572.2019.8978536.
81. Leitner, S.; Mahnke, W. OPC UA—Service-oriented Architecture for Industrial Applications. *Softwaretechnik-Trends* **2006**, *26*.
82. Kraemer, P.; Niebel, C.; Reiberg, A. *Gaia-X and Business Models: Types and Examples*; White Paper 1/2023; Gaia-X Hub: München, Germany, 2023.
83. Martin, D.; Burstein, M.; Hobbs, J.; Lassila, O.; Mcdermott, D.; Mcilraith, S.; Narayanan, S.; Paolucci, M.; Parsia, B.; Payne, T.; et al. OWL-S: Semantic markup for Web services. *W3C Memb. Submiss.* **2004**, *22*.
84. Hepp, M.; Leymann, F.; Domingue, J.; Wahler, A.; Fensel, D. Semantic Business Process Management: A Vision Towards Using Semantic Web Services for Business Process Management. In Proceedings of the 2005 ICEBE, Beijing, China, 12–18 October 2005; IEEE Computer Society: Piscataway, NJ, USA, 2005; pp. 535–540. https://doi.org/10.1109/ICEBE.2005.110.
85. Object Management Group (OMG). *Unified Modeling Language*; Object Management Group: Needham, MA, USA, 2008.
86. Billington, J.; Christensen, S.; van Hee, K.M.; Kindler, E.; Kummer, O.; Petrucci, L.; Post, R.; Stehno, C.; Weber, M. The Petri Net Markup Language: Concepts, Technology, and Tools. In *Applications and Theory of Petri Nets 2003 24th International Conference, ICATPN 2003, Eindhoven, The Netherlands, June 23–27, 2003, Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2679, pp. 483–505. https://doi.org/10.1007/3-540-44919-1_31.
87. Kindler, E. Using the Petri Net Markup Language for exchanging business processes? Potential and limitations. *XML4BPM* **2004**, 43–60.
88. Booch, G.; Jacobson, I.; Rumbaugh, J. The Unified Modeling Language for Object-Oriented Development. *UNIX Rev.* **1996**, *14*, 5.
89. Marrella, A.; Mecella, M.; Sardiña, S. Intelligent Process Adaptation in the SmartPM System. *ACM Trans. Intell. Syst. Technol.* **2017**, *8*, 1–43. https://doi.org/10.1145/2948071.
90. Wong, P.Y.H.; Gibbons, J. A Process Semantics for BPMN. In *Formal Methods and Software Engineering 10th International Conference on Formal Engineering Methods ICFEM 2008, Kitakyushu-City, Japan, October 27–31, 2008, Proceedings*; LNCS; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5256, pp. 355–374. https://doi.org/10.1007/978-3-540-88194-0_22.
91. Intrigila, B.; Penna, G.D.; D'Ambrogio, A. A Lightweight BPMN Extension for Business Process-Oriented Requirements Engineering. *Computers* **2021**, *10*, 171. https://doi.org/10.3390/computers10120171.

92. Ardito, C.; Caivano, D.; Colizzi, L.; Verardi, L. BPMN Extensions and Semantic Annotation in Public Administration Service Design. In *Human-Centered Software Engineering 8th IFIP WG 13.2 International Working Conference, HCSE 2020, Eindhoven, The Netherlands, November 30–December 2, 2020, Proceedings*; LNCS; Springer: Cham, Switzerland, 2020; Volume 12481, pp. 118–129. https://doi.org/10.1007/978-3-030-64266-2_7.
93. Braun, R.; Esswein, W. Semantics in the Context of BPMN Extensions—State of Affairs and Research Challenges. In Proceedings of the MKWI 2016, Ilmenau, Germany, 8–11 March 2016; Universitätsverlag Ilmenau: Ilmenau, Germany, 2016; Volume II, pp. 1119–1130.
94. Malburg, L.; Klein, P.; Bergmann, R. Semantic Web Services for AI-Research with Physical Factory Simulation Models in Industry 4.0. In *Proceedings of the IN4PL 2020*; SciTePress: Setúbal, Portugal, 2020; pp. 32–43. https://doi.org/10.5220/0010135900320043.
95. Diedrich, C.; Belyaev, A.; Blumenfeld, R.; Bock, J.; Grimm, S.; Hermann, J.; Klausmann, T.; Köcher, A.; Maurmaier, M.; Meixner, K.; et al. *Information Model for Capabilities, Skills & Services*; Plattform Industrie 4.0; Plattform Industrie: Berlin, Germany, 2022. https://doi.org/10.24406/publica-477.
96. Uschold, M.; Gruninger, M. Ontologies: principles, methods and applications *Knowl. Eng. Rev.* **1996**, *11*, 96–136. https://doi.org/10.1017/S0269888900007797.
97. Martin, D. L.; Burstein, M. H.; McIlraith D. V.; McIlraith, S. A.; Paolucci, M.; Sycara, K. P.; McGuinness, D. L.; Sirin, E.; Srinivasan, N. Bringing Semantics to Web Services with OWL-S. *World Wide Web* **2007**, *3*, 243–277. https://doi.org/10.1007/s11280-007-0033-x.
98. Sesboüé, M.; Delestre, N.; Kotowicz, J.; Khudiyev, A.; Zanni-Merk, C. An Operational Architecture for Knowledge Graph-Based Systems. *Procedia Comput. Sci.* **2022**, *207*, 1667–1676. https://doi.org/10.1016/j.procs.2022.09.224.
99. Malburg, L.; Schultheis, A.; Bergmann, R. Modeling and Using Complex IoT Time Series Data in Case-Based Reasoning: From Application Scenarios to Implementations. In Proceedings of the 31st ICCBR Workshops, CEUR Workshop Proceedings, Aberdeen, Scotland, 17–20 July 2023; Volume 3438, pp. 81–96.
100. Sacerdoti, E.D. Planning in a Hierarchy of Abstraction Spaces. *Artif. Intell.* **1974**, *5*, 115–135. https://doi.org/10.1016/0004-3702(74)90026-5.