Journal of Manufacturing Systems 90 (2027) 1–23

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

Information modeling for concurrent engineering of product, process, and production system domains in manufacturing

Edoardo Palmitessa [a,*,1], Sazvan Saeed [b,2], Julian Haller [b,3], Amon Göppert [b,4], Adalberto Polenghi [a,5], Elisa Negri [a,6], Robert H. Schmitt [b,c,7]

[a] Department of Management, Economics and Industrial Engineering, Politecnico di Milano, P.zza Leonardo Da Vinci, Milan 20133, Italy
[b] Laboratory for Machine Tools and Production Engineering WZL of RWTH Aachen University, Aachen, Germany
[c] Fraunhofer Institute for Production Technology IPT, Aachen, Germany

ARTICLE INFO

Keywords:
Product-Production-CoDesign
Concurrent engineering
Model-based systems engineering
SysML
Asset administration shell
Circular manufacturing

ABSTRACT

Product-production concurrent design is a key enabler for manufacturing firms to address mass customization and sustainability requirements. However, existing approaches still consider production system design as a downstream activity after product design, limiting integration across these domains. This paper proposes an information model to support concurrent engineering across product, process, and production system domains, addressing literature gaps related to the limited integration of the production system and the lack of explicit modeling of interdependencies among these three domains. The proposed model is operationalized and demonstrated in industrial assembly and disassembly contexts through a Concurrent Engineering Data Pipeline. The results show that this approach supports cross-domain data management and integrated decision-making in product-production co-design, fostering data lifecycle management approaches that integrate linear manufacturing and circular production from early design stages toward the deployment of circular factories.

1. Introduction

Mass customization is increasingly driving manufacturing companies to introduce a growing variety of product variants to the market [1]. This trend is accompanied by rising complexity in customer demand, characterized by timely modification requirements in product features such as dimensions, colors, or functionalities. This complexity not only affects product design but also impacts associated manufacturing processes and production systems, which need to become more flexible and adaptable to promptly address customer demand [2]. In parallel, the transition toward circular factories, where linear manufacturing and circular production co-exist to retain value across multiple product lifecycles, further amplifies this complexity [3]. In these contexts, production systems must handle both new and returned products, while ensuring production efficiency and scalability. This creates a strong need for integrated approaches capable of managing cross-functional engineering knowledge across product and production domains [4,5].

The concurrent engineering of products, processes, and production systems represents a well-established approach to effectively deal with this operating context [3,6]. This approach supports the co-evolution of product and production domains from the early lifecycle stages by relying on structured and interconnected data management [4]. Consequently, by enabling data and information consistency and continuity from beginning of life (BoL) stages, it enables more informed and

* Corresponding author.
E-mail addresses: edoardo.palmitessa@polimi.it (E. Palmitessa), sazvan.saeed@wzl-iqs.rwth-aachen.de (S. Saeed), julian.haller@wzl-iqs.rwth-aachen.de (J. Haller), amon.goeppert@wzl-iqs.rwth-aachen.de (A. Göppert), adalberto.polenghi@polimi.it (A. Polenghi), elisa.negri@polimi.it (E. Negri), robert.schmitt@wzl-iqs.rwth-aachen.de (R.H. Schmitt).
[1] 0009-0008-6021-228X
[2] 0009-0005-6462-2372
[3] 0000-0002-2469-3126
[4] 0000-0001-5976-5655
[5] 0000-0002-3112-1775
[6] 0000-0002-0006-7497
[7] 0000-0002-0011-5962

https://doi.org/10.1016/j.jmsy.2026.11.008
Received 27 April 2026; Received in revised form 4 November 2026; Accepted 13 November 2026
Available online 18 November 2026
0278-6125/© 2026 The Authors. Published by Elsevier Ltd on behalf of The Society of Manufacturing Engineers. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

circular manufacturing decision-making in subsequent lifecycle phases, while simultaneously integrating cross-functional knowledge to achieve flexibility and adaptability of manufacturing systems [7].

Achieving a concurrent engineering approach requires robust interoperability mechanisms and data management across multiple engineering domains, considering that data are stored in heterogeneous sources and models across organizations [8]. This results in unstructured data formats and fragmented knowledge across different business functions and organizations [9,10]. In this regard, formal information modeling approaches for manufacturing systems are recognized as promising to enhance interoperability and enable knowledge sharing across multiple engineering domains, namely product, process, and production system [11,12]. Among these approaches, the Systems Modeling Language (SysML) has gained increasing attention in systems engineering as an extension of the Unified Modeling Language (UML), specifically tailored for system-level modeling [13]. By providing dedicated modeling constructs, including diagrams, elements, relationships, and stereotypes, SysML further enables the representation of system structures and interdependencies [14], making it particularly suitable for modeling complex manufacturing systems involving multiple interacting engineering domains [15].

Although previous research has explored formal modeling approaches for manufacturing systems (e.g., [16,17]), current advancements still exhibit limitations in supporting an integrated consideration of product, process, and production system during their concurrent design, without holistically considering all these domains [18,19]. In particular, most contributions primarily focus on the concurrent design of products and processes, while the production system is often addressed only sequentially. Furthermore, the interdependencies among these three domains remain insufficiently explored during their concurrent design, with existing studies providing only rough modeling representations [8,20]. These gaps constrain a proper mapping of data and information flow between these three cross-functional engineering domains, limiting knowledge exchange in product-production concurrent design and therefore avoiding a smooth transition to circular factories [3].

Considering these limitations, this study addresses the following research question (RQ):

RQ: How can the product, process, and production system domains and their interdependencies be formally modeled in an integrated way to support concurrent engineering?

To address this RQ, the objective of this study is to propose an integrated information model that formally represents the product, process, and production system domains and their physical, functional, and operational interdependencies. The information model is implemented and demonstrated through a Concurrent Engineering Data Pipeline (CEDP) in two industrial scenarios: an automotive assembly company and a household appliance disassembly company. The proposed approach enables an integrated data management system across multidisciplinary engineering functions, fostering appropriate cross-functional data management across product, process, and production system domains to address mass customization and circular factory requirements.

The remainder of the paper is structured as follows. Section 2 presents the state-of-the-art of formal modeling approaches to concurrent engineering processes and related background. Section 3 presents the research methodology. Section 4 illustrates the proposed information model. Section 5 introduces the CEDP through which the information model is operationalized and demonstrated. Section 6 presents its application in the two industrial companies. Section 7 discusses theoretical contributions and industrial implications of this study. Section 8 draws conclusions and future research work.

2. State-of-the-art of formal information modeling for concurrent engineering of product, process, and production system domains

This section aims to provide the state-of-the-art on formal modeling approaches to support concurrent engineering and design of product, process, and production system domains. In doing so, Subsection 2.1 introduces the relevance of concurrent design approaches for circular factories. Subsection 2.2 provides the fundamental principles of concurrent engineering processes. Subsection 2.3 presents current advancements in formal information modeling in manufacturing, while Subsection 2.4 reviews the application of formal modeling approaches to concurrent design processes. Finally, Subsection 2.5 reports the identified research gaps and final considerations.

2.1. Product-Production concurrent design for circular factories

The manufacturing industry is increasingly required to adopt circular manufacturing strategies to meet sustainability targets and align with the Sustainable Development Goals [21]. This transition requires manufacturers to establish practices for circular product design and circular factories, where linear manufacturing and circular production co-exist [3]. In this context, product design and production engineering can no longer be addressed sequentially, but must be approached in an integrated and concurrent manner to effectively support circular objectives [5]. Well-established approaches such as Concurrent Engineering [5] enable early and simultaneous consideration of product and production aspects [3,4], therefore providing foundations for the holistic integration of linear and circular production within circular factory environments through Circular Product-Production-CoDesign (CPPCD).

2.2. Concurrent engineering of product, process, and production system domains

Concurrent engineering and related approaches [22,23], such as PPCD [8], Simultaneous Engineering [24], and the Concurrent Evolution paradigm [6], were developed from the late 20th century onwards to overcome the limitations of traditional waterfall-like design processes, in which products, manufacturing processes, and production systems are designed sequentially [2]. Concurrent engineering instead promotes their integrated and simultaneous consideration from the early design stages, thereby bringing critical decisions forward and enabling potential design inconsistencies to be identified before later and more costly development phases [6].

This integration involves cross-functional engineering domains characterized by heterogeneous data sources, disciplinary backgrounds, information systems, and organizational responsibilities [24,25]. To this end, formal modeling approaches can support the integration of multiple engineering domains by providing structured information representation and enabling their interoperability [26].

Before reviewing the application of formal modeling approaches to concurrent engineering, two concepts that are central to the subsequent state-of-the-art analysis and to concurrent engineering approaches are defined:

- *Product-Process-Resource* (PPR): the PPR paradigm provides an integrated structure for representing manufacturing information by organizing it into three interrelated domains: Product, Process, and Resource [27,28]. This representation is consistent with the ISO 15531 MANDATE (*MANufacturing management DATa Exchange*) standard [29], which provides a framework for modeling manufacturing management data and supporting its exchange between software applications. PPR-based representations can therefore facilitate information integration across engineering and enterprise systems, including Product Lifecycle Management (PLM) and Enterprise Resource Planning (ERP) systems [30]. In this study,

the Resource domain represents the resources and components through which the production system performs manufacturing, assembly, or disassembly activities. The broader production system domain may additionally include its structure, configuration, hierarchy, and operating characteristics;

- *Interdependency*: an interdependency is defined as a cross-domain relationship through which a property, constraint, state, or change associated with an entity in one engineering domain affects or constrains an entity, requirement, engineering decision, or performance variable in another domain [15,20,31]. An interdependency therefore differs from a generic association: it expresses an identifiable influence or constraint between the connected entities. Interdependencies may be directional or reciprocal and can provide the basis for analyzing the propagation of engineering changes across domains. The following interdependency types are considered:
  - o *Physical interdependencies*, concerning structural aspects such as geometry, material, mass, spatial arrangement, accessibility, and physical interfaces;
  - o *Functional interdependencies*, concerning functions, required capabilities, technical compatibility, requirements, and engineering constraints;
  - o *Operational interdependencies*, concerning process parameters, task sequences, task-resource assignments, processing times, and production system operating conditions.

2.3. Formal information modeling approaches in manufacturing

The following formal modeling approaches have been developed for manufacturing systems, each with specific characteristics and a different level of domain representation detail [32]:

- **Conceptual model**: refers to an informal or semi-formal high-level representation of the fundamental concepts and relationships of a domain, aiming to support shared understanding among different stakeholders. It describes *what* exists in the domain and *how* elements are related to each other, providing an abstract and conceptual view of the system through not pre-defined and structured modeling languages [33,34];
- **Information model**: formalizes the conceptual model by defining how concepts are structured and exchanged as data through dedicated modeling language, mainly based on Model-based Systems Engineering (MBSE) [14,35] – e.g., UML or SysML – providing its direct usage as basis for data population and exchange [10]. It represents a formally structured and implicitly semantics-enriched conceptual model, defining an unambiguous representation of domain concepts and their relationship, while remaining independent from specific data storage or implementation technologies;
- **Data model**: refers to a structured representation of data elements, their relationships, and constraints that is oriented toward practical use in information systems; it may include details relevant to data storage, access, and manipulation [32];
- **Ontology**: enriches structural representation with explicit semantics and automated reasoning capabilities, intended to be machine-processable and to support logical inference [34,36]. Ontologies are promising not only in sharing common understanding of the data/information structure, but also reusing knowledge domain and providing formal analysis of terms and on the domain itself [11]. Ontology modeling languages are described in detail by [34];
- **Asset Information Modeling Framework (IMF):** refers to a recent approach aimed at supporting the systematic design of interdependent engineering information models, enabling their effective use across multiple stakeholders with heterogeneous backgrounds and expertise in manufacturing and production environments [9]. The IMF builds upon existing industrial standards and – rather than defining a standalone modeling language – provides a structured framework and methodology to guide the development of consistent, interoperable, and reusable information models. It ensures both human interpretability and machine readability, supporting the coherent evolution of information models throughout the entire system lifecycle [9].

The selection of the most appropriate formal modeling approach depends on the required system design phase, levels of implementation, and semantic formalization. Accordingly, different modeling approaches are typically adopted at different stages to address specific modeling needs. In early design phases, conceptual and/or information models are often developed to identify domain entities and their relationships [12]. As the design progresses, data models and ontologies may be introduced depending on the required level of formalization and intended application (see relationships in Fig. 1). These modeling approaches can also be combined and interconnected by leveraging their complementary characteristics. For instance, ontologies can extend data models by providing explicit semantics and enabling reasoning capabilities, while data models can be structured based on the vocabulary and domain knowledge defined by ontologies [37].

The development of formal information models can benefit from alignment with reference enterprise modeling frameworks and established industrial standards to ensure consistency, interoperability, and a lifecycle perspective across different stakeholders, organizations, and engineering domains [38]. In this regard, enterprise modeling frameworks such as CIMOSA (Computer Integrated Manufacturing Open System Architecture) [39] and the Zachman Framework [40] provide complementary approaches for structuring enterprise knowledge and organizing information representation. On the one hand, CIMOSA supports the systematic modeling of enterprise knowledge by positioning the model with respect to three different axes, respectively identifying the enterprise model phase, the model view, and the model genericity [38,39]. On the other hand, the Zachman Framework provides a reference structure for organizing enterprise models according to multiple perspectives and viewpoints [40].

Established industrial standards should also be considered when developing formal information models for industrial systems, including ISO 19439 [41], which builds upon the CIMOSA framework by standardizing enterprise modeling concepts, ISO/IEC 81346:2022 [42] for reference designation systems, and ISO 15926–14:2020 [43] for industrial information integration and semantic representation.

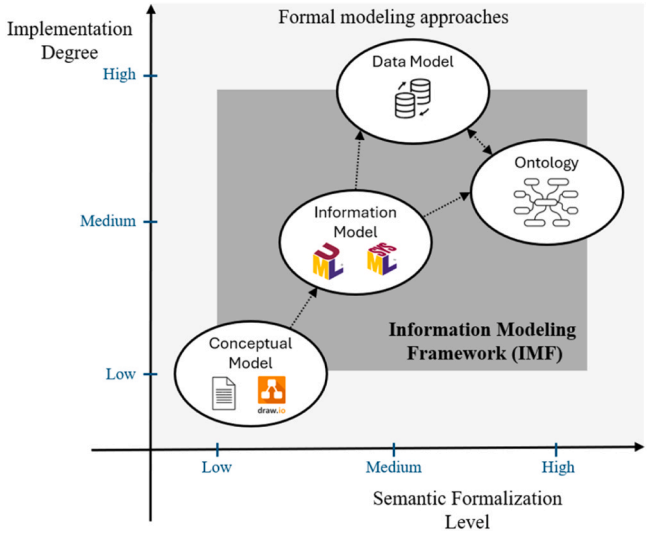

Fig. 1. Classification of Conceptual model, Information model, Data model, Ontology, and IMF according to Implementation Degree and Semantic Formalization levels.

2.4. Formal modeling approaches for concurrent engineering of product, process, and production system

This subsection reviews the state-of-the-art on current formal modeling approaches for the concurrent and integrated engineering of product, process, and production system domains. In particular, Subsection 2.4.1 presents the methodology adopted to identify the most relevant studies in this field, and Subsection 2.4.2 presents the analysis and its results.

2.4.1. Methodology for the literature review

The state-of-the-art analysis followed a structured literature review procedure combining the analysis of existing review studies, a systematic database search, and backward and forward snowballing (Fig. 2). This strategy was selected because the broader fields of concurrent engineering and integrated product and production development have already been examined in previous reviews. Therefore, rather than replicating a full systematic literature review of these broad research areas, the present review specifically aimed to identify and critically compare formal modeling approaches that support the integrated representation of product, manufacturing process, and production system domains and their interdependencies. The three main steps conducted were the following:

1) *Identification of existing review studies*: reviews were identified through a dedicated Scopus search and retained when they addressed integrated or concurrent product and production development, PPCD, or closely related research streams. The systematic and state-of-the-art review by Albers *et al.* [8], Disselkamp *et al.* [44], and Vizitiu *et al.* [45] satisfied these criteria. The included reviews were used to define the broader research streams, refine the search terminology, and identify relevant primary studies;
2) *Update of the literature review*: a systematic database search was conducted to identify primary studies published after the search periods covered by the included reviews. The update search included a temporal overlap with the most recent review and covered the period from January 2022 to July 2026. The search was conducted in Scopus using the following query: *"TITLE-ABS-KEY(("Concurrent Engineering" OR "Product-Production*Design" OR "Integrated Product Development" OR "Concurrent Evolution") AND ("Formal*" OR "Information model*" OR "Data model*" OR "ontolog*" OR "Model-based Systems Engineering" OR "MBSE"))"*. Of the 66 records retrieved, titles and abstracts were screened before full-text assessment. Studies were eligible if they proposed or applied a conceptual, information, or data model, or an ontology, covering at least two PPR domains, their explicit interdependencies, or both; five studies met these criteria;
3) *Snowballing approach*: backward and forward snowballing was conducted on the eligible review studies and primary studies. For backward snowballing, the publication year of the foundational contribution on Concurrent Engineering by Winner *et al.* [22] was adopted as the lower temporal boundary of the search.

As a result of this process, 18 papers were considered the most relevant primary studies to be embedded in the state-of-the-art analysis presented in Subsection 2.4.2.

2.4.2. State-of-the-art analysis

Table 1 analyzes the selected studies retrieved from the literature search strategy reported in Subsection 2.4.1. In the table, *Product*, *Process*, and *Production System* columns indicate the depth with which each domain is represented. The assessment considers three elements: the modeling of domain entities, their attributes or properties, and their internal relationships or hierarchies:

- An empty circle indicates that the domain is not explicitly represented;
- A half-filled circle indicates that only a subset of these elements is modeled or that the domain is treated at a general level;
- A filled circle indicates that all three elements are explicitly represented at a detailed level.

Interdependency modeling is assessed using the *Physical*, *Functional*, and *Operational* categories defined in Subsection 2.2. For each category, an empty circle denotes no identifiable cross-domain relationship, a half-filled circle denotes an implicit or partial representation, and a filled circle denotes explicit cross-domain relationships with defined semantics. Only elements explicitly presented in each study are considered. The assessment concerns the scope of the proposed model rather than the overall quality of the study or its suitability for its stated application.

The reviewed studies address two related modeling objectives. One is to develop structured and consistent representations of PPR information by defining domain entities, their attributes, and internal relationships. The other is to formalize cross-domain interdependencies to analyze how properties, constraints, or changes in one domain affect another. Despite being distinguishable, these objectives cannot be treated independently. Modeling interdependencies directly influences the way each

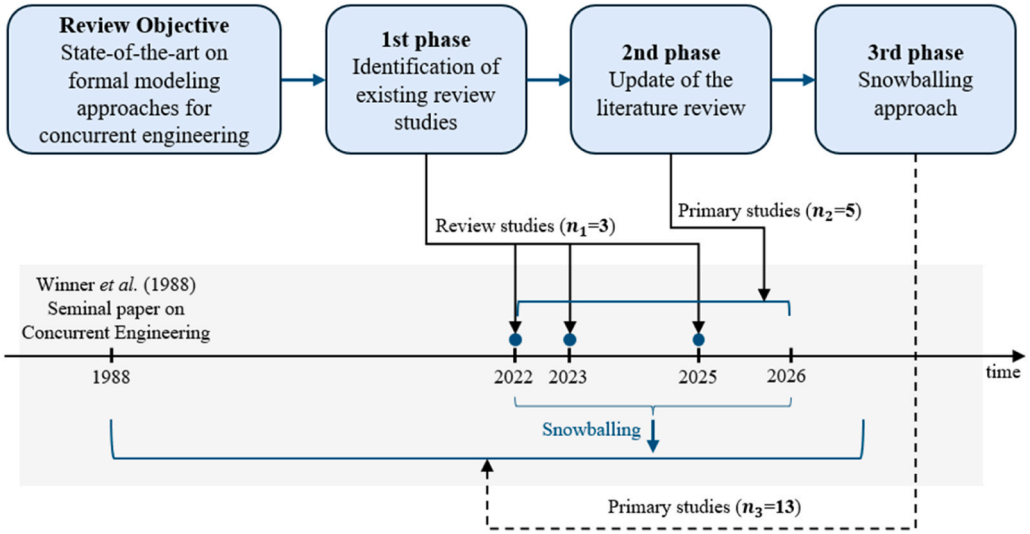

Fig. 2. Structured state-of-the-art review procedure combining existing review studies, a systematic database update, and a snowballing approach.

Table 1
Concurrent modeling and engineering of product, process, and production system domains and their interdependencies (○: not represented; ◐: partially represented or at a general representation level; ●: substantively represented or at a detailed representation level with explicit semantics).

Paper	Modeling formalism	Product	Process	Production System	Interdependency		
					Physical	Functional	Operational
[17]	Ontology	●	●	◐	○	◐	◐
[18]	Ontology	◐	●	◐	○	◐	●
[16]	Ontology	◐	◐	◐	○	◐	◐
[33]	Conceptual model	◐	●	◐	○	◐	◐
[46]	Ontology	●	●	◐	○	◐	◐
[10]	Information model	●	●	◐	◐	◐	●
[47]	Ontology	◐	◐	◐	○	◐	◐
[20]	Information model	◐	◐	◐	●	●	●
[15]	Information model	◐	◐	◐	◐	◐	◐
[48]	Information model	●	◐	◐	◐	○	○
[49]	Information model	◐	◐	◐	◐	◐	◐
[31]	Information model	◐	◐	◐	◐	●	◐
[50]	Information model	◐	◐	○	○	◐	◐
[51]	Ontology	○	◐	◐	○	◐	●
[19]	Ontology	◐	●	◐	○	◐	○
[52]	Information model	●	◐	○	◐	○	○
[53]	Conceptual model	◐	◐	◐	○	○	○
[54]	Information model	◐	◐	○	●	◐	○
Current paper objective	*Information model*	●	●	●	●	●	●

domain is represented because the entities and properties involved vary accordingly. Conversely, an interdependency model remains partial when the connected PPR domains are described only abstractly for the same reason.

The first modeling objective has been addressed by several previous studies. For instance, Lemaignan et al. [17] propose an ontology to represent entities, operations, and resources for cost estimation and multiagent manufacturing communication. Following this track, Borgo and Leitão [18] develop a core ontology to ensure consistent interpretation of manufacturing data exchanged during operations and scheduling phases. These relationships support classification, information exchange, and task execution, but do not characterize cross-domain influence as defined in Subsection 2.2. Colledani et al. [33] move closer to an integrated PPR representation through a UML reference framework; however, its general database and implementation are only partially validated and focused on manufacturing processes' capabilities. Lanz et al. [16] use feature-based rules to derive assembly processes and system requirements from product information, although the system domain is limited mainly to resources, areas, actors, and skills. Usman et al. [46] formalize core manufacturing concepts in common logic, but full interoperability, parametric design and production links, and industrial implementation remain unaddressed, limiting broader adoption. Garetti and Fumagalli [47] structure production system entities across physical, technological, and control layers, yet primarily support production system description and software portability rather than balanced PPR integration. These works establish shared manufacturing domain semantics, but incorporating interdependencies also requires representing the properties, constraints, and states through which cross-domain effects occur.

A set of application-oriented models partially bridges the two aforementioned modeling objectives by introducing more actionable links, but usually around one specific decision problem. Wu et al. [10] use SysML constraints and a three-layer mapping mechanism to coordinate product and process variety in production system reconfiguration. This supports cost analysis related to product variety, mostly focusing on operational interdependencies in cross-domain analysis. Mandel et al. [48] use SysML allocations to trace product functions and physical features to processes, modules, and machines, enabling an additional step forward to formalized change propagation analysis. Schäfer et al. [49] derive production sequences from features and joints in previously designed variants through an automatic approach. In this study, information flows from the product to the production process domain rather than reciprocally among these domains. The graph-based approach proposed by Hu et al. [50] reduces restructuring between engineering and manufacturing bill of materials views, but primarily reconciles alternative product structures rather than formalizing different types of PPR influence. These models improve multi-domain data traceability and advance interdependency modeling, while their semantics remain tied to the target decision.

Albers et al. [15], Hoang et al. [20], and Schäfer et al. [31] make interdependencies themselves the object of modeling. Albers et al. [15] relate product functions and features to production processes and machines and aggregate these links in an impact matrix. The approach supports change impact analysis, but the required information, supported analysis cases, and benefit evaluation remain subjects of ongoing work. Hoang et al. [20] encode constraints and positive or negative correlations among PPR parameters in an interdependency matrix. This qualitative representation reduces modeling effort and supports the identification of adaptation options; quantitative and indicator-based evaluation and automatic database generation are deferred to later work. Schäfer et al. [31] combine an ontology and a SysML-Profile-based modeling method to define product-production system interdependencies. Overall, these studies advance the explicit representation of interdependencies across PPR domains. However, this focus is not accompanied by equally comprehensive representations of the connected domains. The resulting domain abstraction limits the entities and properties through which cross-domain effects can be expressed, thereby constraining the scope and semantics of the interdependencies themselves.

Studies published after 2022 extend formal modeling toward ontology-based digital threads [19], rule-based capability matching [51], knowledge-graph synchronization [52], and Concurrent Design Facilities [53]. Nevertheless, their scope remains limited to selected knowledge patterns, manufacturing capabilities, and/or product information. These contributions therefore strengthen the computational use of engineering knowledge without resolving the joint need for comprehensive PPR representation and explicit interdependency formalization.

Comprehensive PPR representation and interdependency formalization are mutually dependent rather than features that can be combined afterwards. Physical, functional, and operational interdependencies determine which entities, attributes, constraints, and operating states must be represented. Conversely, formal relationships cannot support comprehensive cross-domain analysis when the connected domain models remain abstract or incomplete. No prior approach in Table 1 addresses both requirements jointly: richer domain

structures generally contain only selected or implicit dependencies, whereas explicit constraint, correlation, or inference mechanisms operate on partial domain representations or narrowly defined knowledge patterns. The remaining gap is therefore compositional rather than an absence of integrated models: a consistent formal representation is still needed that combines detailed coverage of all three PPR domains with explicit semantics for their physical, functional, and operational interdependencies.

2.5. Summary of the state-of-the-art analysis and identified research gaps

The state-of-the-art analysis of Subsection 2.4 leads to two closely related research gaps:

- *Research gap 1 (RG1) – Asymmetric representation of the PPR domains*: existing integrated models lack a holistic and comprehensive representation of Product, Process, and Production System domains with comparable detail. In particular, the production system is frequently limited to the resources, capabilities, or equipment required for a specific application, without comprehensively modeling its structure, configuration, attributes, and operating characteristics. This limits the concurrent analysis of how product and process decisions affect production system design, and vice versa.
- *Research gap 2 (RG2) – Lack of joint domain and interdependency formalization*: the reviewed studies lack combined and integrated representations of all three PPR domains with explicit semantics for physical, functional, and operational interdependencies. Consequently, the analysis of cross-domain constraints and engineering change propagation remains fragmented across partial domain models and application-specific relationships.

These gaps are mutually reinforcing. An incomplete representation of the production system restricts the entities and properties through which its interdependencies with product and process domains can be expressed. Conversely, without explicitly formalized interdependencies, even detailed PPR representations cannot fully support cross-domain analysis and concurrent decision-making.

To address these interrelated gaps, the present study proposes an information model that (i) represents the production system jointly with, and at a level of detail comparable to, the Product and Process domains, and (ii) explicitly formalizes physical, functional, and operational interdependencies across the three domains. The model is operationalized through the CEDP, and its applicability is demonstrated through two industrial cases presented in the following sections.

3. Research methodology

This paper follows the Design Science Research (DSR) methodology, widely adopted to target real industrial problems and solve them through the proposal of novel artifacts [55]. In particular, the structure of the DSR methodology proposed is aligned with Peffers et al. [56], where the main phases of this methodology refer to *Problem Identification and Motivation*, *Design and Development* of new artifacts, and their *Demonstration and Evaluation* (Fig. 3), described as follows:

1. *Problem Identification and Motivation*. This phase aims to motivate the research problem and justifies the value of a solution, from both academic and industrial perspectives [57]. To this end, according to the different potential entry points in the DSR [56], the current research adopts the industrial need for innovative approaches to perform concurrent and collaborative design processes aimed at reducing time-to-market (TTM). After a review of the scientific literature to identify the state-of-the-art and current advancements in this regard (see Section 2), two main research gaps were identified (RG1 and RG2), which pave the way for the formulation of the RQ of this study.
2. *Design and Development*. This phase aims to create a meaningful and novel artifact to answer the problem [57]. Therefore, to address the identified gaps RG1 and RG2, the research objective is to develop an integrated information model that formally represents the product, process, and production system domains and their physical, functional, and operational interdependencies, and to demonstrate its operationalization through the CEDP in two industrial contexts (see Section 4). According to the DSR methodology, this model represents the main research artifact, bridging the identified literature gaps and advancing the state-of-the-art while addressing the industrial problem that motivated the research. The proposed information model was initially developed by synthesizing recent advances reported in the literature and critically examining existing conceptual, information, and ontological models; the reference models considered during this preliminary phase are mentioned in Section 4 (and introduced beforehand in Section 2). This approach operationalizes knowledge reuse through the identification, assessment, and selective incorporation or adaptation of established knowledge structures,

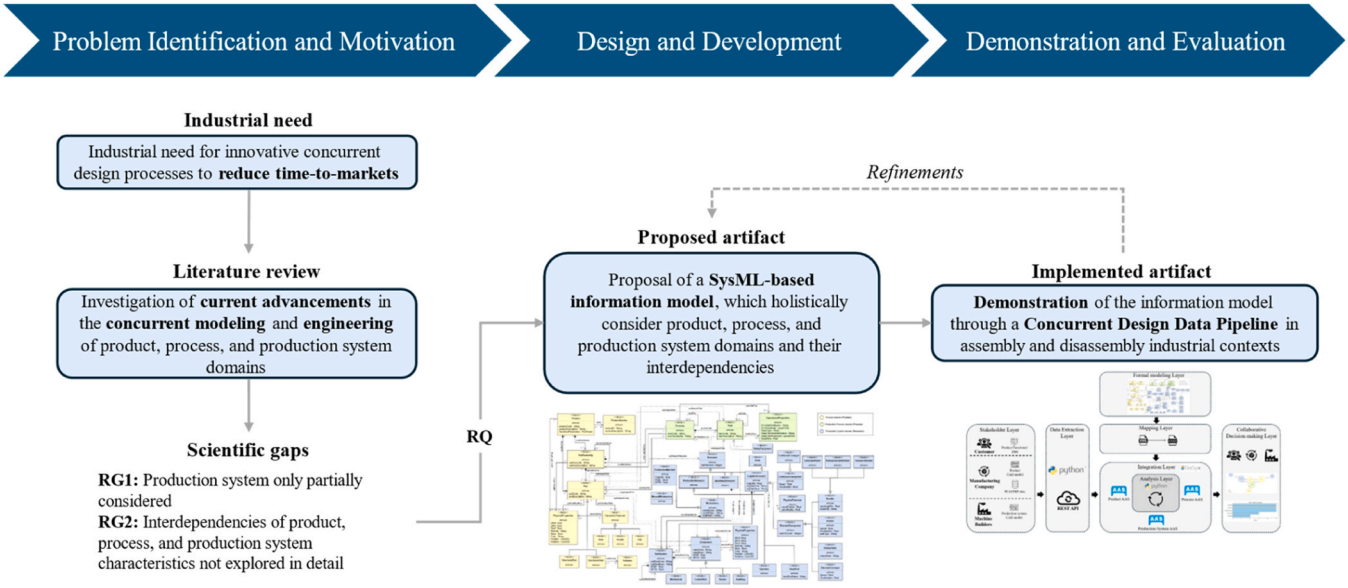

Fig. 3. DSR methodology adopted in the paper.

thereby avoiding the unnecessary redefinition of existing concepts and enhancing the consistency, quality, and interoperability of the resulting formal models [36]. Subsequently, the artifact was extended with additional elements missing from the current literature, mainly related to the representation of interdependencies among the product, process, and production system domains, enabling a more comprehensive and holistic information model. The artifact was first conceptually designed and then implemented in software, following iterative refinement cycles driven by the Demonstration and Evaluation phase of the DSR methodology. In particular, feedback from the assembly and disassembly industrial use cases led to successive refinements of the artifact, allowing the information model to consistently represent both industrial scenarios while progressively improving the initial draft.

3. *Demonstration and Evaluation.* This phase aims to demonstrate the artifact application in solving the targeted problem and evaluate its applicability through proofs, case studies, or other suitable activities [57]. In this research, the applicability of the developed information model was demonstrated through a CEDP (see Section 5) applied in two industrial contexts: an automotive assembly company and a household appliance disassembly company (see Section 6). Data were collected from Computer-Aided Design (CAD) models of products and production systems, as well as from spreadsheets containing production process information extracted and manipulated from the companies' information systems. When available data were insufficient or unclear for populating the proposed information model, informal discussions were conducted with product designers, process engineers, and production engineers. As further discussed in Section 6, the demonstration considered a selection of company domains in terms of product, process, and production system, mainly due to data availability, the research objective limited to the demonstration of the proposed artifact, and the need to avoid disclosing sensitive industrial information. Overall, the application in both industrial scenarios iteratively supported the refinement and improvement of the proposed information model by incorporating additional elements and details derived from both assembly and disassembly contexts. Rather than developing multiple candidate artifacts, the research followed an iterative refinement process aimed at consolidating the identified requirements into a single unified artifact. This allowed the final information model to consistently represent product, process, and production system aspects, together with their interdependencies, demonstrating its applicability across the two industrial contexts involved in the demonstration phase. The assessment of the two industrial implementations focused on technical feasibility and usability for cross-domain design impact analysis. Potential improvements in engineering workflows and empirical evidence on the effectiveness of the proposed approach were not systematically evaluated within the scope of this study.

4. Proposed information model for concurrent design of product, process and production system domains

Fig. 4 presents the proposed information model for the concurrent engineering of product, process, and production system domains and their interdependencies. In this study, the model is implemented using SysML block definition diagrams [14]. SysML was selected because its system-oriented block abstraction supports a consistent representation of heterogeneous entities across these engineering domains. With these constructs, typed properties describe entity characteristics, composition and generalization relationships represent decomposition hierarchies and specializations, while associations and dependencies connect related entities. Custom SysML-Profile stereotypes applied to dependency relationships further formalize attribute-level influences across product, process, and production system domains (further details are provided in Subsection 4.4). These constructs support the graphical definition and semantics of the information model, also providing the structured input for its transformation into Asset Administration Shell (AAS) representations through the CEDP (Section 5).

In this setting, SysML represents one possible formalization language of the modeling proposed structure, rather than an essential methodological requirement. The approach requires explicit representations of domain entities, properties, hierarchies, relationships, and

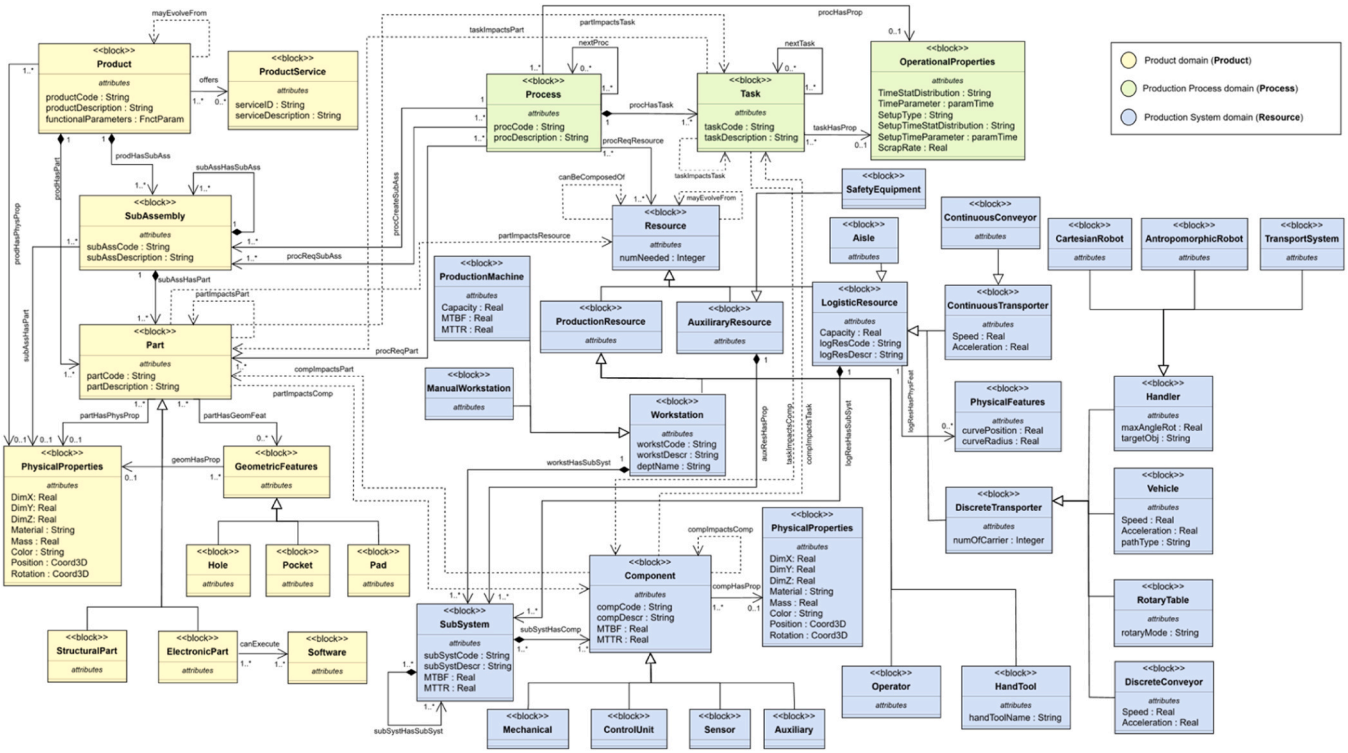

Fig. 4. Proposed information model for concurrent engineering of product, process, and production system domains and their interdependencies.

interdependencies, together with a machine-readable serialization that preserves the information and semantics required for transformation into AAS structures. Alternative modeling languages, including UML class diagrams extended with suitable profiles, could also be adopted if these requirements are met. Their use would require compatible parsing and mapping rules for the selected metamodel and serialization format, potentially involving adaptations to the implementation described in Subsection 5.4.

In line with the knowledge reuse principle introduced in Section 3 [36], the proposed model builds upon and extends the conceptual, information, and ontological models reviewed in Section 2, as described in the next subsections, which respectively dive into the specific domains of the proposed information model. From an enterprise modeling perspective [38–40], the proposed information model focuses on the engineering-domain level, providing a formal and integrated representation of product, process, and production system information, together with their physical, functional, and operational interdependencies, to support concurrent engineering activities. In this way, the model enables the integration of heterogeneous data originating from multiple engineering tools and information sources, such as CAD systems, spreadsheets, and information systems, fostering a consistent exchange of information across the domains involved in the concurrent engineering process and enabling more integrated and informed decision-making for circular products and factories.

4.1. Product domain

The product domain representation is illustrated in Fig. 5, which is built upon previous work [10,16,58]. From a physical and structural point of view, the hierarchy illustrates that *Product* is composed of one or more *Sub-Assembly* elements, which in turn can be composed of other *Sub-Assembly*. Accordingly, each *Sub-Assembly* is composed of multiple *Part* elements, which represent the lowest hierarchical level of the product domain. *Product*, *Sub-Assembly*, and *Part* can be described by *PhysicalProperties*, which report their main physical characteristics, including dimensions along the X, Y, and Z axes, material, mass, color, position, and in-plane rotation. The *PhysicalProperties* block and its attributes are also modeled based on previous research to describe the physical characteristics [48,59,60]. Given the focus of the proposed model on capturing the physical characteristics of the product domain required to establish connections with process and production system domains, *PhysicalProperties* is modeled separately, while remaining extensible to additional product characteristics, such as functional or logical properties [31]. This modeling choice allows maintaining a clear separation between product entities and their physical attributes, which are the main characteristics considered for establishing the targeted interdependencies in the proposed approach. Each *Part* can contain *GeometricFeature*, represented by e.g. *Hole*, *Pocket*, and *Pad* [16], which can be modeled as well through *PhysicalProperties*. Each *Product*, *Sub-Assembly*, and *Part* owns attributes with their code and description as string values. Considering that modern products increasingly consist of mechatronic systems [61], each *Part* can be further classified as either a *StructuralPart*, representing mechanical components, or *ElectronicPart*, representing electronic components capable of hosting and executing embedded *Software*.

A *Product* may also include a *functionalParameters* attribute, representing the physical, chemical, or other functional requirements that must be satisfied by its design (e.g., providing a specified pressure or withstanding a given force during operation). Additionally, a *Product* may offer *ProductService*. According to the product-service system perspective, a *Product* may offer zero or more services, such as maintenance and repair or remote condition monitoring. These services extend the product's value beyond its physical artifact and may support value creation throughout its lifecycle. Their explicit representation allows service-related requirements to be considered from the early design stages, as they may influence its characteristics and quality [62].

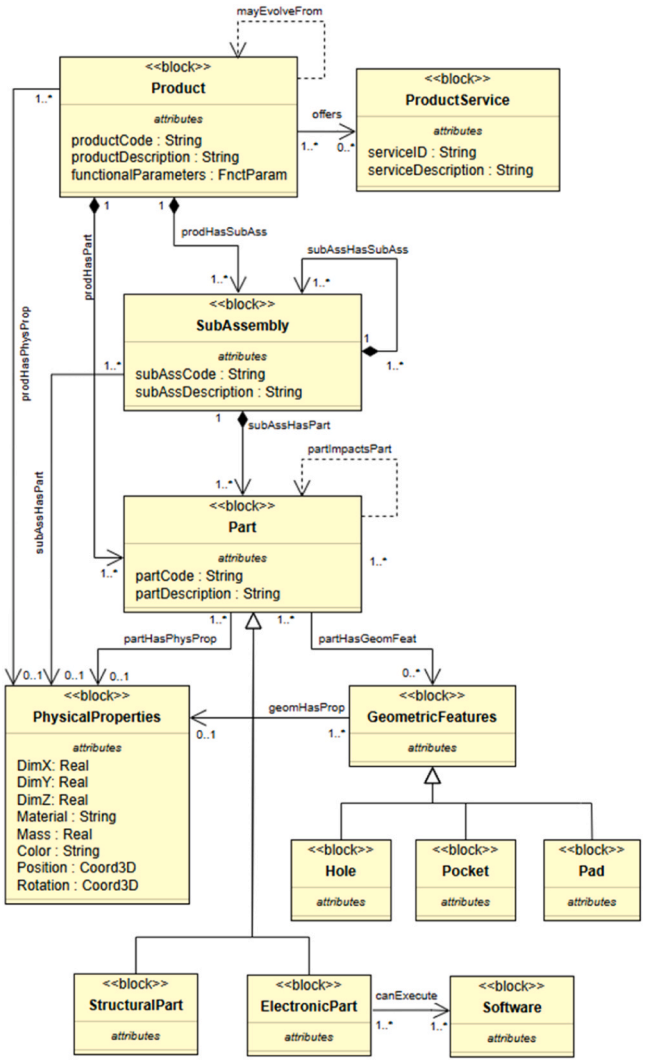

Fig. 5. Focus on Product domain modeling in the proposed information model.

In line with PPCD and its embedded principle of Product Generation Engineering (PGE) [8], the information model conceptually includes a *mayEvolveFrom* dependency between *Product* representations of different generations or variants. The relationship is directed from the considered product design to a previous reference design, identifying a potential basis for its development. It supports the reuse of engineering knowledge and provides a foundation for tracing changes across product generations.

The dependency relationship *partImpactsPart* is modeled through SysML-Profile as a stereotype of the dependency and is further explained in Subsection 4.4.

4.2. Process domain

Fig. 6 illustrates the formal representation of the process domain, built upon previous modeling [16]. In the proposed model, *Process* is composed of one or more *Task* elements (e.g., a welding process is composed of many tasks such as preparation of needed components, welding execution, etc.). Both blocks own specific attributes, including their unique code and description. Additionally, they can have *OperationalProperties*, including *timeStatDistribution* and *setupTimeDistribution* (e.g., Normal, Weibull), *timeParameter* and *setupTimeParameter*, needed to describe the statistical distributions of processing and setup times (i.e., *paramTime* represents a vector of three real values needed to describe the statistical distribution), *setupType* (e.g., based on product type or

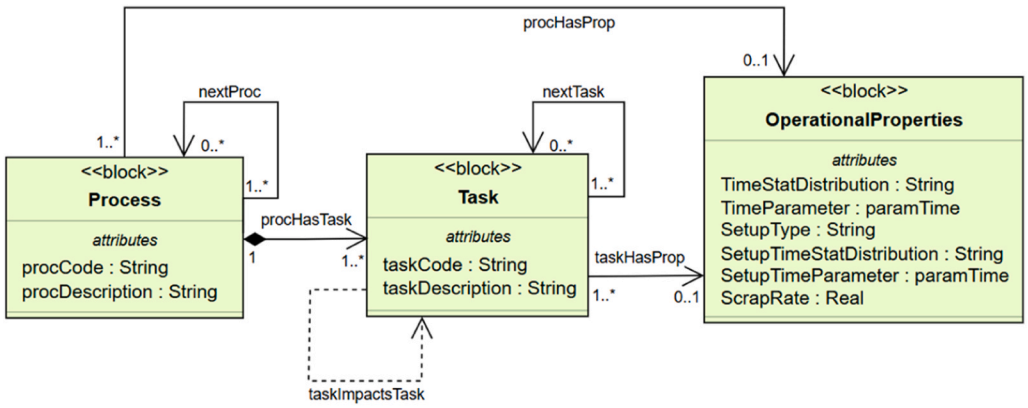

Fig. 6. Focus on Process domain modeling in the proposed information model.

quantity), and *scrapRate* (percentage of defective production). Each *Process* and *Task* points to its successor, if any. *taskImpactsTask* refers to a stereotype of dependency relationship, which will be further described in Subsection 4.4. Stereotypes are extension mechanisms used to customize existing modeling elements by enriching them with domain-specific semantics, constraints, and properties 4.4 [14].

4.3. Resource domain

The production system domain represents the least comprehensively modeled domain in the literature on concurrent design processes, as well as the most complex with respect to the other engineering domains. Fig. 7 illustrates the proposed formalization for modeling the resource domain according to the PPR paradigm, including both production and logistic perspectives [12,16,17,63].

The highest level is the *Resource*, which includes *ProductionResource*, *LogisticResource*, and *AuxiliaryResource*. Production resources are *Operator*, *HandTool*, and *Workstation* elements. The latter block includes both *ManualWorkstation* and *ProductionMachine*. Logistic resources are divided into *ContinuousTransporter* (e.g., *ContinuousConveyor*) and *DiscreteTransporter* (e.g., *Handler*, *Vehicles*, *RotaryTable*, *DiscreteConveyor*), as detailed in previous work [12]. *AuxiliaryResource* refers to any resource which does not belong to production or logistic systems but needs to be modeled as a *Resource* (e.g., *SafetyEquipment*).

Each *Workstation*, *AuxiliaryResource*, and *LogisticResource* is composed of *SubSystem* elements, which in turn may be composed of other *SubSystem* elements. *SubSystem* is then made up of *Component* elements (e.g., *Mechanical*, *ControlUnit*, *Sensor*, or *Auxiliary*), that represent the lowest hierarchical level of any *Resource*. Each *Component* has its own associated *PhysicalProperties* block, containing physical characteristics, as in the product domain for the *Part, SubAssembly* and *Product*. *ProductionMachine*, *SubSystem*, and *Component* are characterized by

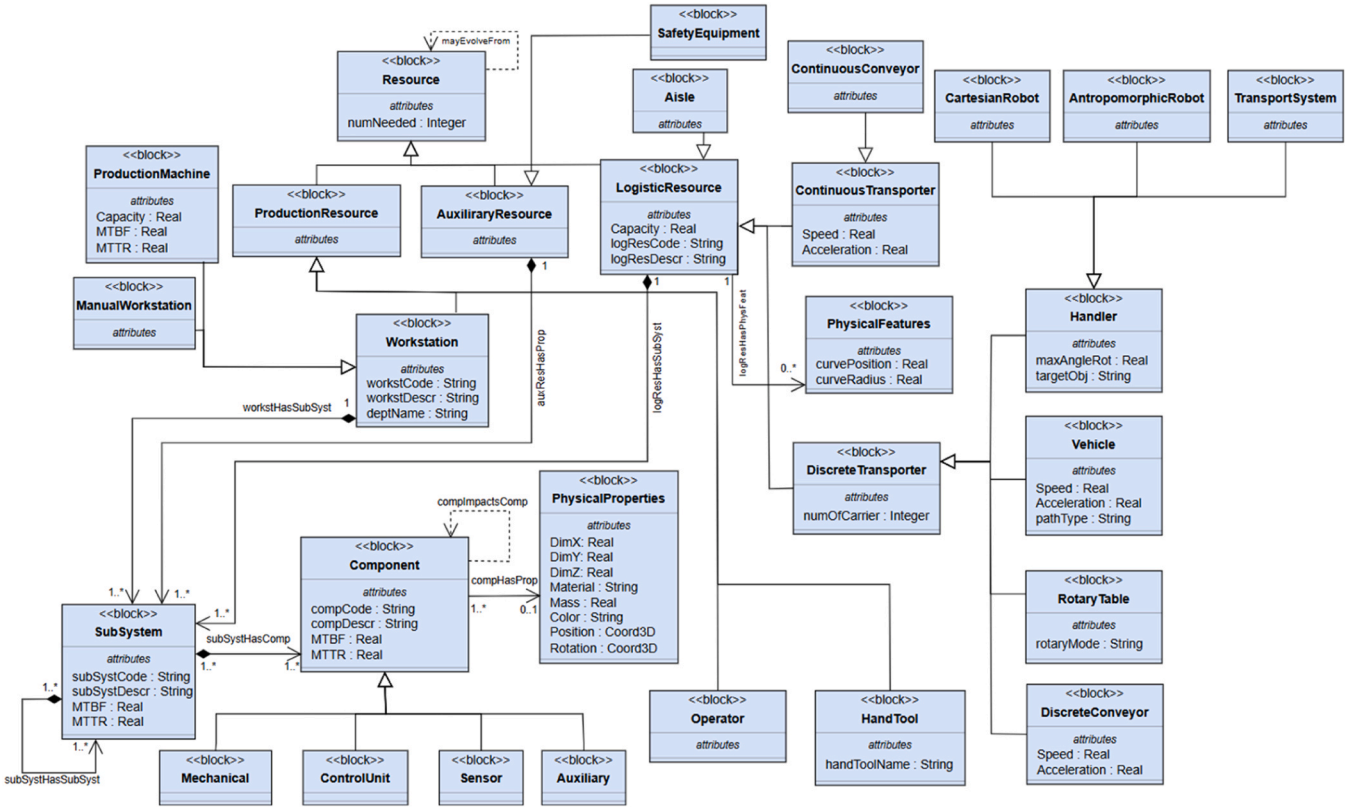

Fig. 7. Focus on Resource domain modeling in the proposed information model.

maintenance-related attributes, including Mean Time Between Failures (*MTBF*) and Mean Time To Repair (*MTTR*), enabling the modeling of their reliability, availability, and maintainability behavior. As well as *partImpactsPart*, each *Component* may influence other *Component* elements in case of design changes through *compImpactsComp* stereotype dependency relationship.

Similarly to the product domain, a *mayEvolveFrom* dependency between *Resource* representations conceptually links new or revised production and logistics resource configurations to their previous reference designs, providing a basis for reusing engineering knowledge and tracing changes across resource generations, according to the PPCD concept.

4.4. Interdependencies among Product, Process, and Resource domains

The proposed information model explicitly represents physical, functional, and operational interdependencies across the product, process, and production system domains. Physical interdependencies are captured through relationships between *Part* and *Component* entities, whereby predefined changes in dimensions or other physical characteristics require corresponding adjustments to their *PhysicalProperties*. Functional interdependencies are primarily represented by the *partImpactsResource* relationship, which expresses how changes in a *Part* structure or physical characteristics can affect the resources required for its manufacture. Operational interdependencies are captured through relationships involving the *Task* entity and their effects on processing and setup time parameters.

The proposed model supports the integrated design of product, production process and system to enable a comprehensive concurrent engineering approach. It goes beyond previous attempts to model interdependencies among these domains, which introduced interdependency matrices and mechanisms to handle interdependencies but without a clearly formal and semantic structure for holistic analysis [15,20]. To this end, Fig. 8 represents the proposed model only with a focus on interdependency relationships. All interdependency relationships of the proposed model are modeled through SysML-Profile by the introduction of the stereotypes *ImpactInteraction_A*, *ImpactInteraction_B*, *ImpactInteraction_C*, and *ImpactInteraction_D* for the dependency relationship [14]. This stereotype maps both inter- (between domains) (e.g., *partImpactsComp*) and intra-interdependencies (within a domain) (e.g., *compImpactsComp*) among elements.

The *ImpactInteraction* stereotypes consist of an extended dependency in which *N* properties are attached, where *N* is the result of the Cartesian product between all physical properties of *Part*, *Component*, *Resource*, and *Task* elements. While the first two blocks are characterized by eight physical properties each, *Resource* includes a single attribute representing the required number of resources, and *Task* is defined by six operational properties. To this end, Table 2 summarizes the number *N* of attributes associated with each stereotype dependency relationship, together with illustrative examples.

Each attribute defined within these stereotypes assumes a Boolean value, indicating whether the first property influences the second, while explicitly accounting for the directionality of the dependency as specified by the arrow of the relationship. This approach extends the interdependency matrices proposed in previous studies [15,20,59] for modeling relationships among product, process, and production system domains by providing them with semantic meaning and formal representation. Rather than being limited to Boolean values, the attributes defined in these stereotypes may also be modeled using Integer or Real types, enabling quantitative and numerical analyses across different blocks. The selection of the attribute type depends on the specific objectives of the use case under consideration.

5. Concurrent engineering data pipeline for operationalizing the information model

This section presents the implementation and demonstration of the applicability of the proposed information model in the field of concurrent engineering of product, production process and system domains. Fig. 9 illustrates the proposed data pipeline for this purpose. The pipeline is structured into multiple layers, which are described in detail in the following subsections.

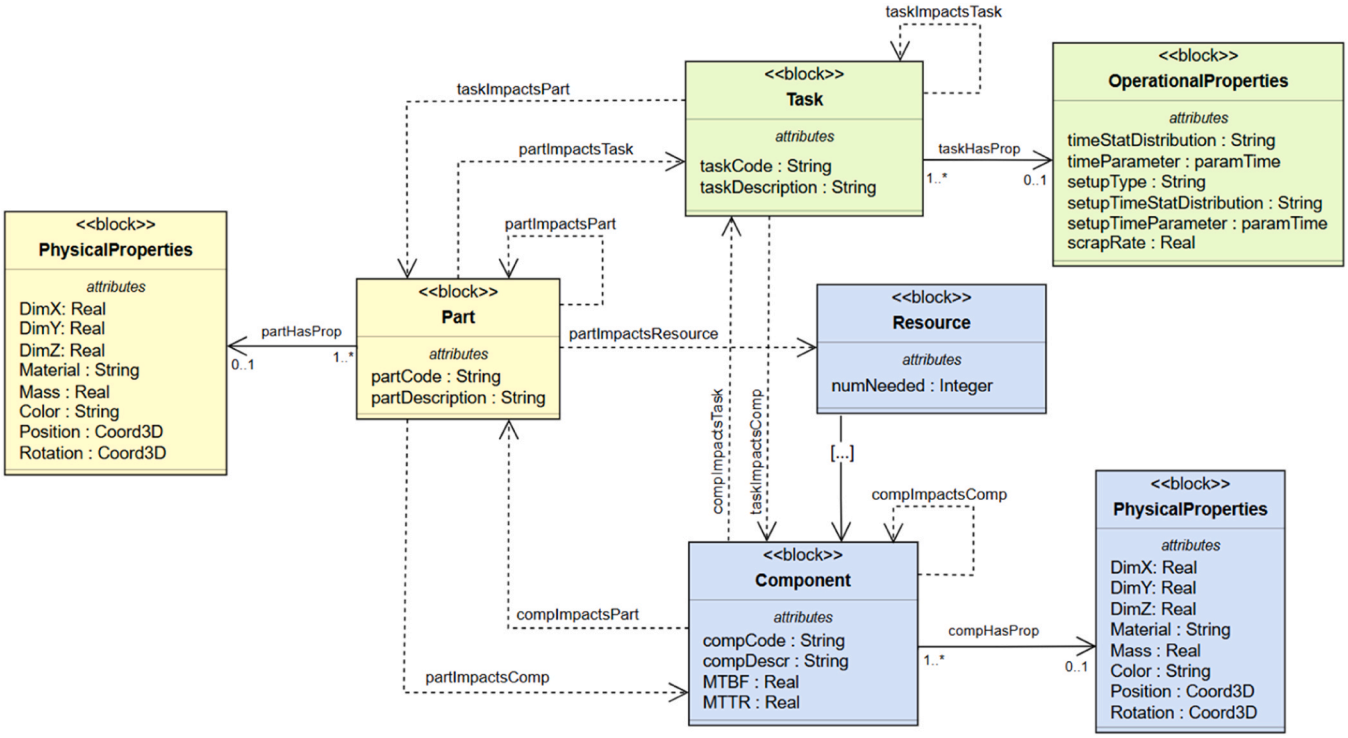

Fig. 8. Mapping of interdependencies among Product, Process, and Resource.

Table 2
Stereotypes of Interdependency relationships.

Stereotype	Interdependency Relationships	Number of Attributes (*N*)	Example of Attributes
ImpactInteraction_A	*partImpactsPart*, *compImpactsComp*, *partImpactsComp*, *compImpactsPart*	64 (8 ×8)	DimXImpactsDimY, MassImpactsMaterial
ImpactInteraction_B	*partImpactsTask*, *taskImpactsPart*, *compImpactsTask*, *taskImpactsComp*	48 (8 ×6)	MaterialImpactsScrapRate, SetupTimeImpactsMass
ImpactInteraction_C	*taskImpactsTask*	36 (6 ×6)	SetupTypeImpactsSetupTime, SetupTimeImpactsScrapRate
ImpactInteraction_D	*partImpactsResource*	8 (8 ×1)	DimXImpactsNumNeeded, MaterialImpactsNumNeeded

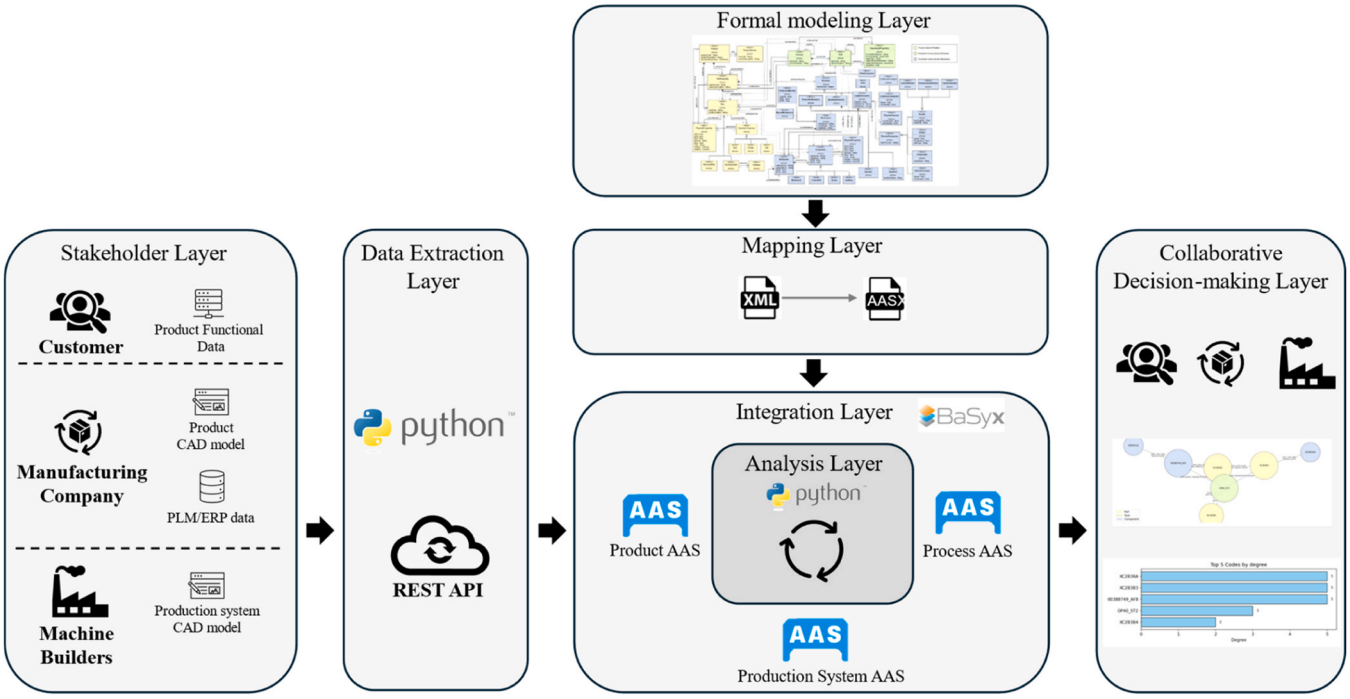

Fig. 9. Concurrent Engineering Data Pipeline through integrated modeling of product, process, and production system.

The developed CEDP is intended to support machine builders in the efficient design of production systems by enabling the systematic identification of priorities and critical parts, tasks, components, and resources relevant to the optimization and design of production systems. These engineering processes, which rely on shared information models, require a standardized and formalized mechanism for data exchange among stakeholders [16]. Furthermore, ensuring the usability and interpretability of engineering data across lifecycle phases requires an interoperable approach to managing asset-related information. In this context, the AAS provides a standardized framework for implementing interoperable digital twins of all assets in an Industry 4.0 environment [64].

The AAS metamodel structures the digital representation of an asset into submodels that represent different aspects of the asset, using defined submodel elements such as properties and relationships [65]. The abstract nature of the metamodel permits domain-specific structuring of asset information based on an underlying information model. Several studies have demonstrated the transformation of MBSE models into AAS models while ensuring semantic consistency and interoperability across stakeholders and life cycle phases [66–69]. Moreover, the potential of the AAS to support engineering change processes was explored in recent research [70–72].

The pipeline transforms the SysML model into an AAS domain metamodel and deploys it (Subsection 5.4). Use case data then instantiate this metamodel, taken from CAD models for the product domain and from tabular files for the process and production system domains (Subsection 5.2 and Subsection 5.5). Annotated relationships inside the AAS record the interdependencies among the instantiated elements (Subsection 5.5). Reading the repository back yields the knowledge graph on which the analyses run (Subsection 5.6). The user thereby learns which elements a proposed change would affect across the product, process, and production system domains, and which elements are most exposed to, or most influential on, changes elsewhere. Section 6 demonstrates both for two industrial cases.

5.1. Stakeholder layer

The Stakeholder Layer encompasses the key actors involved in concurrent design processes, who can leverage the proposed information model and data pipeline to support both new product development and production system design. These actors include Customers, Manufacturing companies, and Machine builders. Each stakeholder contributes specific data and knowledge essential for enabling an integrated design approach across product, process, and production system models. Manufacturing companies typically manage product-related information stored in PLM/ERP systems, including product CAD models, while Machine builders provide production system data, such as operational parameters and CAD models of production equipment required to manufacture the designed products. Customers drive the overall design process by specifying additional or modified product

features and functionalities, which in turn require manufacturing companies to develop new product variants capable of meeting these requirements and machine builders to design production systems able to manufacture such products. These stakeholders take part in the collaborative decision-making process of the proposed CEDP.

Given this multi-disciplinary ecosystem and the need for both inter- and intra-organizational collaboration, data and information from these heterogeneous stakeholders must be jointly structured, stored, and made accessible to support the analysis, synchronization, and optimization of product, process, and production system models within a concurrent and integrated design framework, not only in the design phase but along the overall product lifecycle.

5.2. Data Extraction Layer

The Data Extraction Layer turns CAD models into the instance data that the Integration Layer stores. Product geometry enters the pipeline here, whereas process and production system data enter through the tabular interfaces described in Subsection 5.5. The layer is implemented as a Python program built on PythonOCC, a wrapper around the OpenCASCADE geometry kernel.

Extraction expects a STEP file exported with its assembly structure and its component names retained. The program reconstructs the assembly tree from the product definition and assembly usage occurrence entities of the file, and treats as root the product definition that has children but no parent. For every solid it then computes the extents of the axis-aligned bounding box, the volume, the surface area, the coordinates of the center of mass, and the color, the last of these read from the XCAF presentation data of the file.

The pipeline writes the bounding box extents, the center of mass, and the color into the PhysicalProperties submodel of each Part and Component AAS, where each value carries a semantic identifier. Mass, material, and rotation remain empty, because STEP encodes shape rather than material data and rotation is not yet extracted; these values must be supplied from PLM or ERP systems. One AAS is instantiated per node of the assembly tree: leaf nodes become Part AAS, intermediate nodes become SubAssembly AAS, and the root becomes the Product AAS. A HierarchicalStructure submodel (IDTA 02011) preserves the parent-child relations, each of its nodes referencing the corresponding AAS through a *sameAs* relationship. Because every AAS additionally carries a use case scoped Uniform Resource Name (URN), the extracted product structure is directly addressable by the interdependency import of Subsection 5.5 and by the analyses of Subsection 5.6.

5.3. Formal modeling Layer

The Formal modeling Layer defines the underlying data structure that enables the storage, management, and access of information required for the concurrent modeling and design of product, process, and production system domains. Beyond providing a structured and standardized data representation, this layer establishes the formal relationships and interaction rules among these domains, ensuring a consistent and interoperable integration across engineering models. By capturing the interdependencies between product, process, and

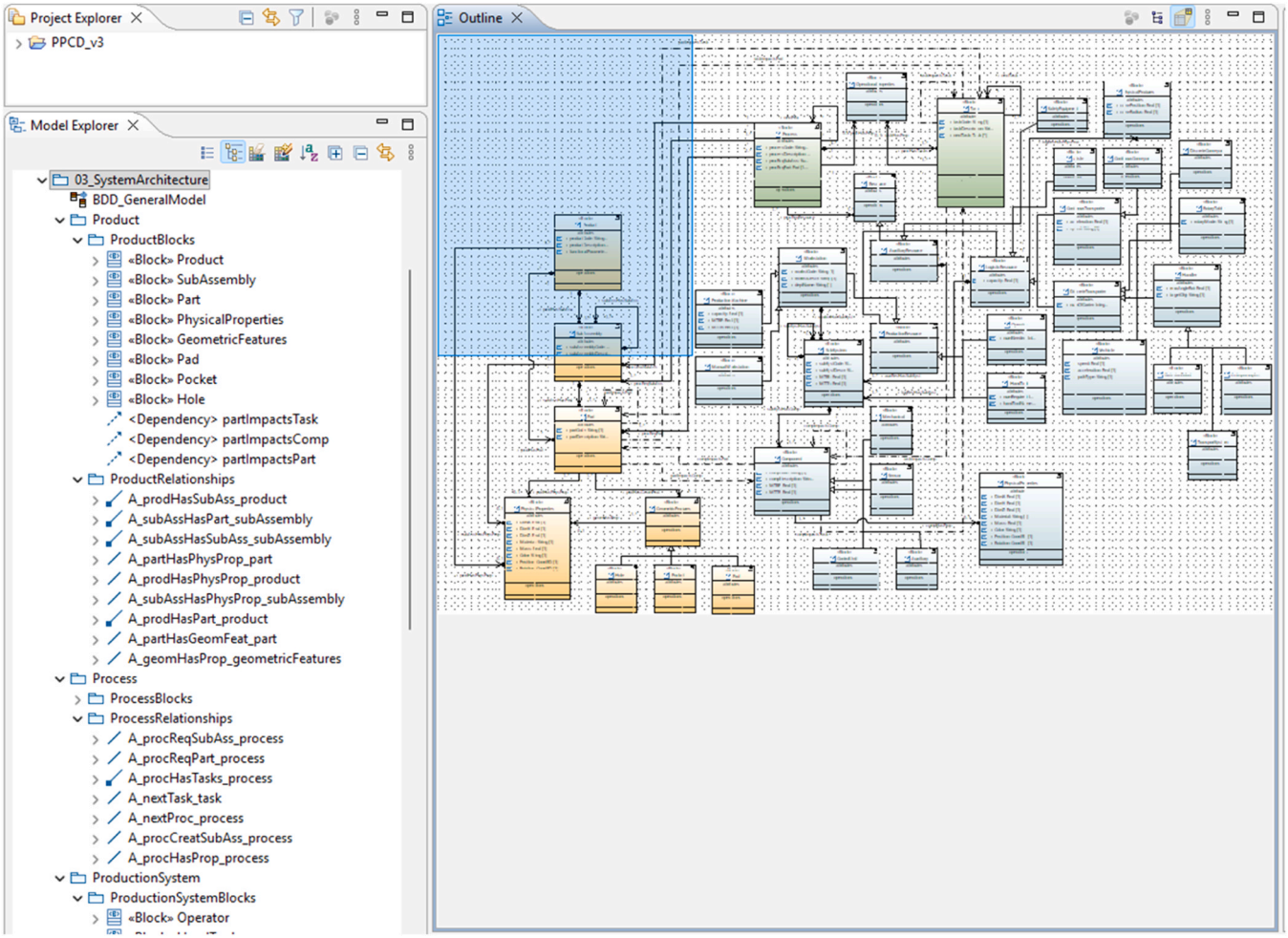

Fig. 10. Extract from Eclipse Papyrus SysML modeling software.

production system entities, it supports data-driven impact analysis during design activities.

For this specific work, the information model (Fig. 4) was developed in Eclipse Papyrus through SysML v1.6 [14]. The block definition diagram (Fig. 10) provides a graphical representation for defining and editing the model, while the underlying XML Metadata Interchange (XMI) serialization (Fig. 11) provides the machine-readable model elements and stereotype definitions processed by the Mapping Layer.

5.4. Mapping Layer

The Mapping Layer implements a model-driven transformation from the SysML representation adopted in this study to an AAS-based domain metamodel, realizing the automated deployment of an AAS-based domain metamodel (Fig. 12). The implemented parser and mapping routines are tailored to the source metamodel and its XML-based serialization. Applying the same transformation approach to an alternative modeling language would require corresponding parsing and mapping rules that preserve the domain entities, property types, hierarchies, associations, and interdependency semantics defined in Section 4. The automated transformation is realized using the BaSyx AAS environment for deployment and Python Software Development Kit (SDK) for creating and editing modeling elements [73].

The transformation is initiated by invoking an XML parser to extract a structured model summary comprising the four semantic categories: *blocks*, *compositions*, *associations*, and *impact interactions*. Each category is subsequently processed through dedicated mapping routines. The SysML blocks are mapped to either AAS elements or Submodel elements, as determined by a configurable mapping schema (Fig. 13). The mapping logic resolves UML class inheritance chains, identifying the root ancestor to select the appropriate AAS metamodel counterpart. Each AAS block is instantiated with a companion *Metadata* Submodel containing the block's structural properties, which are themselves mapped from UML typed attributes to either AAS Property elements (for primitive datatypes) or SubmodelElementCollection structures (for complex, user-defined types).

UML composition relationships are encoded into *HierarchicalStructures* Submodels [75], realizing Bill-of-Materials (BoM) representations. For each composing block, an entry node entity is constructed containing child node entities. Each node is cross-referenced to its corresponding AAS or Submodel instance via *sameAs* relationship elements, thereby preserving structural containment semantics within the AAS infrastructure.

UML associations are classified by their role labels: *Has*-typed associations trigger the attachment of target Submodels to source AAS instances, encoding capability or property ownership. *Next*-typed associations are reified as *AnnotatedRelationshipElement* instances referencing the source AAS reflexively, representing precedence or sequencing constraints, and are embedded in the corresponding Metadata Submodel. *Impact interactions*, encoding cross-domain influence

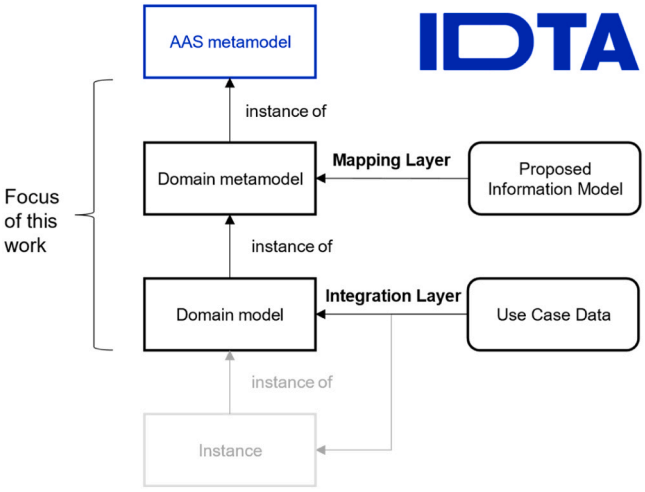

Fig. 12. Multi-layer metamodeling stack based on [74].

relationships between system elements, are similarly encoded as *AnnotatedRelationshipElement* with boolean-annotated properties, capturing the specific influence dimensions between client and supplier AAS instances.

All AAS models are first created using the BaSyx Python SDK and then serialized into JSON objects. These JSON objects are then uploaded into the BaSyx AAS environment where they are stored in the internal BaSyx database. The AAS models are then retrievable through the BaSyx REST API. As the AAS models are domain metamodels based on the SysML blocks, an identification by their names is needed in the Integration Layer. BaSyx only has a native UUID system for identification but does not enable identification by name. Therefore, a name-to-identifier mapping is serialized into a JSON file when the mapping process is finished.

5.5. Integration Layer

The Integration Layer enables the representation of use case-specific data in the structure of the mapped AAS models. As indicated in Subsection 5.2, different data sources are processed for different types of use case data. While product data are entirely extracted from CAD files and processed into a product and component AAS representation, production system and process data can be passed to the system in a tabular format using a predefined structure. Each piece of data is linked to the domain metamodel element it represents. Data belonging to the same use case is linked to each other through a URN ID structure following the schema: *{urn:ppcd:[domain-metamodel-element]:[use-case-id]:[name]}*.

Four sequential stages transform the input data into AAS elements deployed in the AAS environment. The first stage processes the input file

```
<?xml version="1.0" encoding="UTF-8"?>
<xmi:XMI xmi:version="20131001" xmlns:xmi="http://www.omg.org/spec/XMI/20131001" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:Blocks="http://www.eclipse.org/papyrus/sysml/1.6/SysML/Blocks"
xmlns:ConstraintBlocks="http://www.eclipse.org/papyrus/sysml/1.6/SysML/ConstraintBlocks" xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore"
xmlns:uml="http://www.eclipse.org/uml2/5.0.0/UML" xsi:schemaLocation="http://www.eclipse.org/papyrus/sysml/1.6/SysML/Blocks
http://www.eclipse.org/papyrus/sysml/1.6/SysML#//blocks http://www.eclipse.org/papyrus/sysml/1.6/SysML/ConstraintBlocks
http://www.eclipse.org/papyrus/sysml/1.6/SysML#//constraintblocks">
  <uml:Model xmi:id="_4OOIQFWoEfC744kl-GcSjg" name="PPCD_v3">
    <packagedElement xmi:type="uml:Package" xmi:id="_whvIQFWpEfC744kl-GcSjg" name="03_SystemArchitecture">
      <packagedElement xmi:type="uml:Package" xmi:id="_0dE8cFWpEfC744kl-GcSjg" name="Product">
        <packagedElement xmi:type="uml:Package" xmi:id="_4rHqMFWpEfC744kl-GcSjg" name="ProductBlocks">
          <packagedElement xmi:type="uml:Class" xmi:id="_-ms30FWpEfC744kl-GcSjg" name="Product">
            <ownedAttribute xmi:type="uml:Property" xmi:id="_UBRjAFWqEfC744kl-GcSjg" name="prodHasSubAss" type="_BXp8gFWqEfC744kl-GcSjg"
            aggregation="composite" association="_UBN4oFWqEfC744kl-GcSjg">

            ...
```

Fig. 11. Excerpt from XML file.

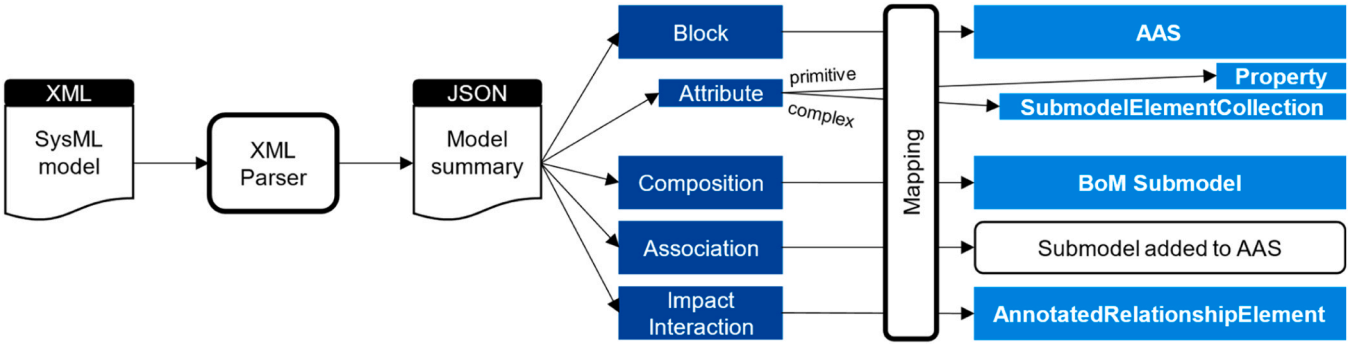

Fig. 13. Default mapping process and schema used in the Mapping Layer.

according to its format. For STEP files, the OpenCASCADE engine parses the CAD assembly into its parts and extracts the hierarchical assembly structure. For Excel and CSV files, the system parses the tabular data, identifies the parent-child relationships defined by the hierarchy columns, and locates the root entities. During this stage, the input documents are normalized into an in-memory hierarchy tree that represents the structural relationships regardless of the original format.

Before creating any AAS instances, the system queries the existing AAS environment to identify what instances already exist. The result of this single bulk query is stored in a local cache, which is critical for deduplication, preventing duplicate AAS creation if the process is rerun or if entities with the same name have already been instantiated. Entity names are normalized at this stage, removing special characters and converting them to valid AAS identifiers that conform to the naming restrictions.

The system traverses the hierarchy tree depth-first, creating AAS instances in a bottom-up manner. For each entity in the hierarchy, the process:

(1) queries the AAS environment for the appropriate domain metamodel element (e.g., Part, Product, ProductionMachine),
(2) deserializes the template using the BaSyx Python SDK,
(3) creates a deep copy of the template and modifies its metamodel attributes,
(4) instantiates all submodels referenced by the element (e.g., Metadata, PhysicalProperties), and
(5) posts the newly created domain models back to the AAS environment.

After the AAS domain models and their submodels are created, the final stage populates them with detailed properties. For STEP-sourced AAS, the OpenCASCADE engine's computed geometric data is inserted into the PhysicalProperties submodels. For Excel/CSV-sourced AAS, custom properties from the input files are parsed (including JSON-encoded complex properties) and inserted into the appropriate submodels.

The steps described so far instantiate the elements of the product, process, and production system domains. Additional input completes the repository: the interdependencies among those elements. They are supplied as a matrix whose rows name a changed element, an impacted element, the changed attribute, the impacted attribute, and a Boolean marker. Each marked row sets one annotation of the *AnnotatedRelationshipElement* that links the two elements in the Metadata submodel of the changed element, the annotation being named by concatenating the two attribute names. Every row thus instantiates one attribute of one *ImpactInteraction* stereotype of Table 2 and the stereotypes in turn define which rows the system accepts. Reading the relationships back for a given use case identifier yields a table of typed, directed, attribute-level dependencies. That table is the input of the Analysis Layer, and it is where the information model, the AAS repository, and the knowledge graph analyses of Section 6 meet.

5.6. Analysis Layer

The Analysis Layer analyzes the instantiated product, process, and production system of the case study under investigation by directly accessing the data held in the AAS. It draws on knowledge graph analysis: graph metrics (e.g., degree, centrality, connectivity) identify critical parts, tasks, and components; impact analysis determines what happens to other parts and components when a part or component feature changes; and change propagation analysis determines whether such a change requires a further change in a specific part or component feature. Section 6 further illustrates some of the output analysis from this layer based on the industrial case studies. The analysis can be performed not only for concurrent design purposes in BoL but across all lifecycle stages, as data are managed and gathered from design phases and kept synchronized along all phases to support data-driven End of Life (EoL) decisions of circular manufacturing strategies.

Three analyses operate on this graph:

- *Element-level reachability* returns what a changed element reaches directly and what it reaches through chains of intermediate steps;
- *Attribute-level reachability* narrows this to a named attribute of the changed element and a named attribute of the impacted one;
- *Graph characterization* computes degree, in-degree, out-degree, and betweenness centrality for every node and partitions the nodes into communities by modularity maximization. Each measure carries an engineering reading: in-degree marks what is most exposed to changes elsewhere, out-degree what propagates most widely, and betweenness where propagation paths concentrate. Fig. 17 and Fig. 19 report element-level reachability for Company A and Company B, and Fig. 18 and Fig. 20 report the graph characterization.

5.7. Collaborative Decision-making Layer

Based on the output of the Analysis Layer, this layer aims to support collaborative decision-making processes of the involved stakeholders taking part in the concurrent design process. Indeed, they can concurrently make decisions on the product, process, and production system design models, considering the most impacted or critical parts/tasks/components, enabling a concurrent consideration of both product and production domains.

6. Industrial implementations through the CEDP

The applicability of the proposed information model was demonstrated through the CEDP in two industrial cases (Table 3). A broad set of product, process, and production system entities was instantiated; however, the reported knowledge-graph analyses deliberately focus on the parts, tasks, components, and relationships most relevant to cross-

Table 3
Main characteristics of the two industrial companies involved in the artifact implementation.

Company name	Industry	Company Size (employees)	Design Process stage	Production Processes	Main industrial objective supported
Company A	Automotive	Large (>250)	Detailed Layout Design	Assembly	Mass customization
Company B	Household appliance	Large (>250)	Process Planning & Technical Layout Design	Disassembly	Circular manufacturing

domain change propagation and criticality assessment. Production system elements not exercised in these cases were derived from models previously applied in the literature, in line with knowledge reuse approaches [36], with the corresponding references reported in Section 4.

Company A is operating in the assembly of automotive components, while Company B is focused on the disassembly of household appliances. This dual-case approach provides a more comprehensive demonstration of the model, demonstrating its applicability to both assembly and disassembly scenarios, which are the forward and return loops in circular factories, respectively. Moreover, the two companies are at different stages of production system design (Fig. 14). Company A is engaged in the detailed layout design phase, developing CAD models of the production system and validating its control logic within the testing and debugging phase. On the other hand, Company B is in an earlier design stage where only product specifications and related CAD models are available, and only a small portion of production system information has been defined so far. Regarding the main objectives supported by the proposed approach, Company A applies it to address mass customization and the growing number of product variants requested by customers, whereas Company B uses integrated modeling to enable more informed circular manufacturing decisions by integrating product and production knowledge from BoL stages (Table 3).

This section presents the implementation of the proposed approach in Company A and Company B in Subsection 6.1 and 6.2, respectively, followed by the corresponding results in Subsection 6.3.

6.1. Company A implementation

Company A has defined the final design drawings of an innovative product and is currently involved in the detailed layout production system design stage (Fig. 14). This phase is primarily led by a machine builder, which collaborates with Company A to define the configuration and operational settings of the production system. However, the company expects that clients may request variations in product features, such as size, color, or material. To address this potential variability, Company A identified the need for a tool capable of assessing the implications of such changes on production processes and production system configurations. In particular, the company aims to evaluate how modifications in product characteristics may affect process parameters and machine configurations, supporting a more effective concurrent engineering approach with the machine builder.

Assessing these interdependencies during the design stage is essential to avoid costly redesign activities and late-stage modifications caused by inconsistencies between product and production system specifications. Early evaluation of these relationships enhances manufacturing system flexibility and reconfigurability, improving responsiveness to market requirements.

The proposed information model was recognized as a lever to analyze the cross-domain impacts between the product and production domains. The model enables the mapping and analysis of data flows and interdependencies between these engineering domains to support concurrent design of product, process, and production system.

The implementation focused on a subset of the company production system, including three manufacturing processes, three corresponding resources, and a selection of their subsystems and components (Fig. 15). Product data were automatically extracted from CAD models following the second step of the CEDP described in Subsection 5.2, and integrated into AAS within the BaSyx platform (Fig. 16). Due to confidentiality constraints, product details are not disclosed, while process and production system data were anonymized through randomization. Nevertheless, the structural relationships among system elements and the original process precedence diagram were preserved.

Based on the CEDP, impact and propagation analyses across product, process, and production system domains can be performed. For instance, users can input the unique identifier of a specific element to trace the chain of impacted entities (directly and indirectly as effect of chains) associated with that element (Fig. 17). In this way, users can study the impact of changing the size of a particular component or the color of a product part, identifying which are the elements impacted beyond product domain, analyzing effects on processes and production systems.

Alternatively, risk-oriented analyses can be conducted to identify critical parts, tasks, or components by evaluating the direct in-degree and out-degree of nodes within the network, which refer to the number of incoming and outgoing relationships of a specific node. This highlights entities that are either highly affected by or highly influential on other elements (Fig. 18).

These analyses are supported through a knowledge graph-based approach derived from the instantiation of product, process, and production system models based on the proposed information model. For the proposed case, node *Part_002*, corresponding to a product part,

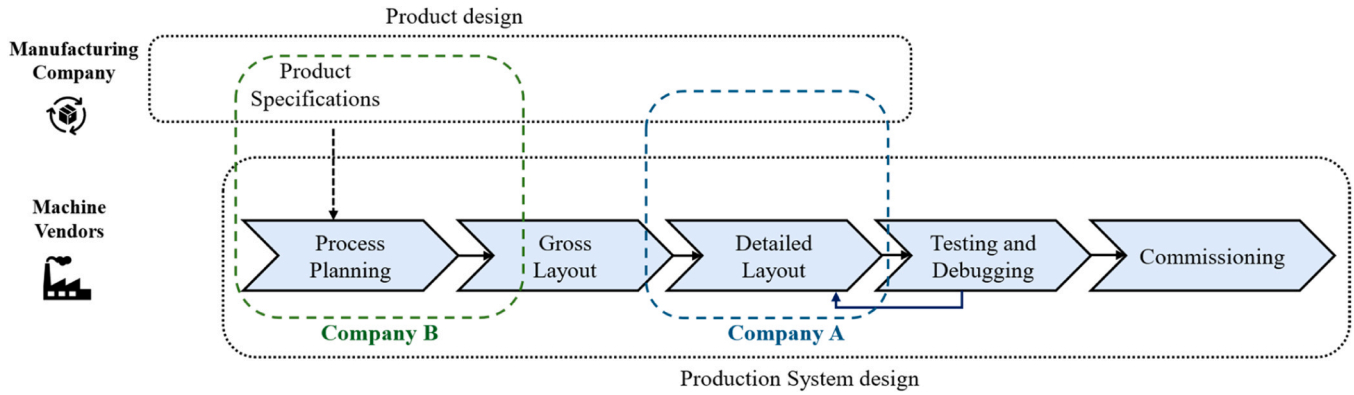

Fig. 14. Comparison of the design stages of the two industrial companies involved in the case study within the production system design process.

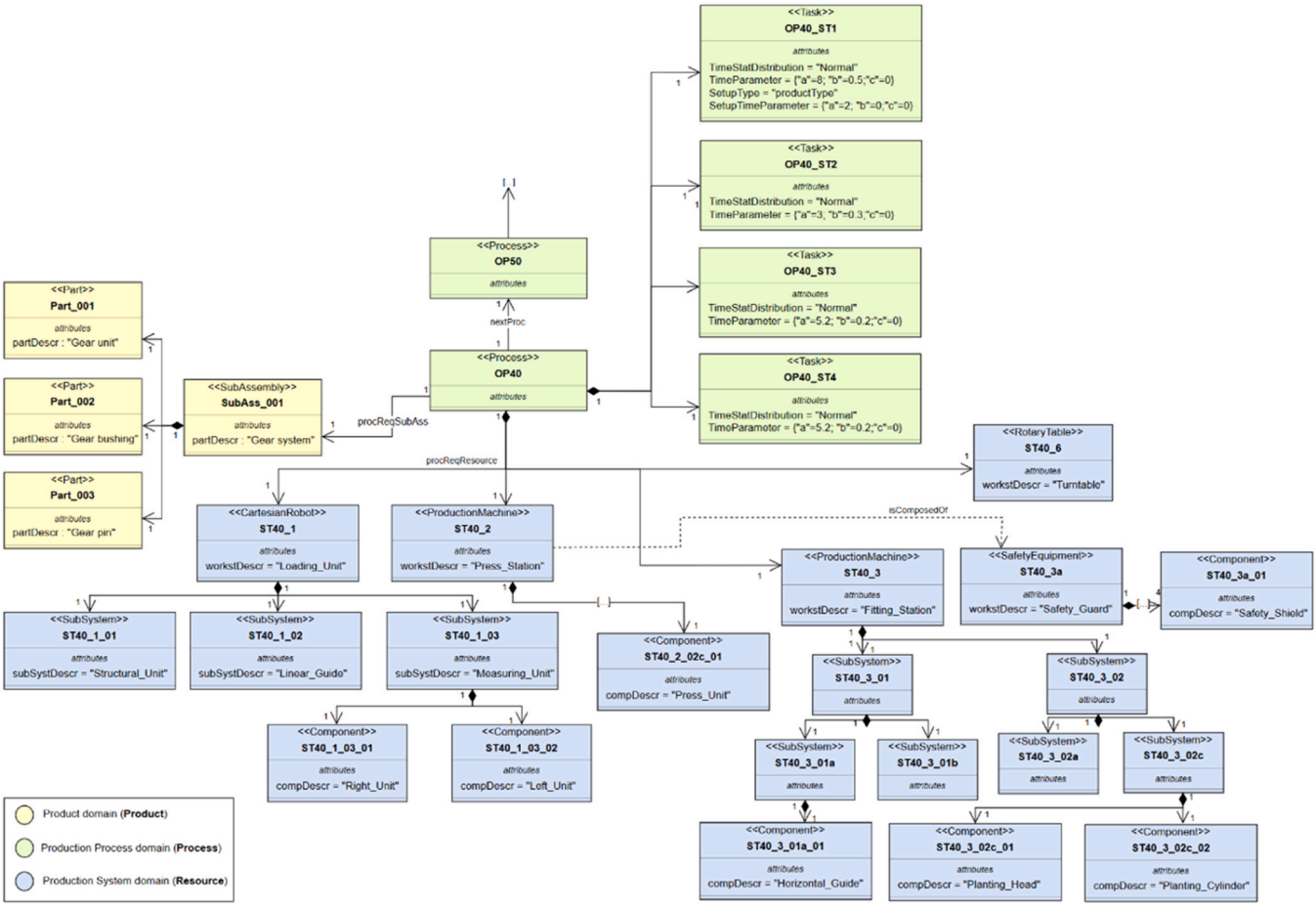

Fig. 15. Selection of the product, process, production system considered of Company A (the picture does not integrate all entities and relationships modeled, but only the ones needed to understand analysis related to Company A implementation. This picture aims to show how domain elements are instantiated based on the SysML model through AAS).

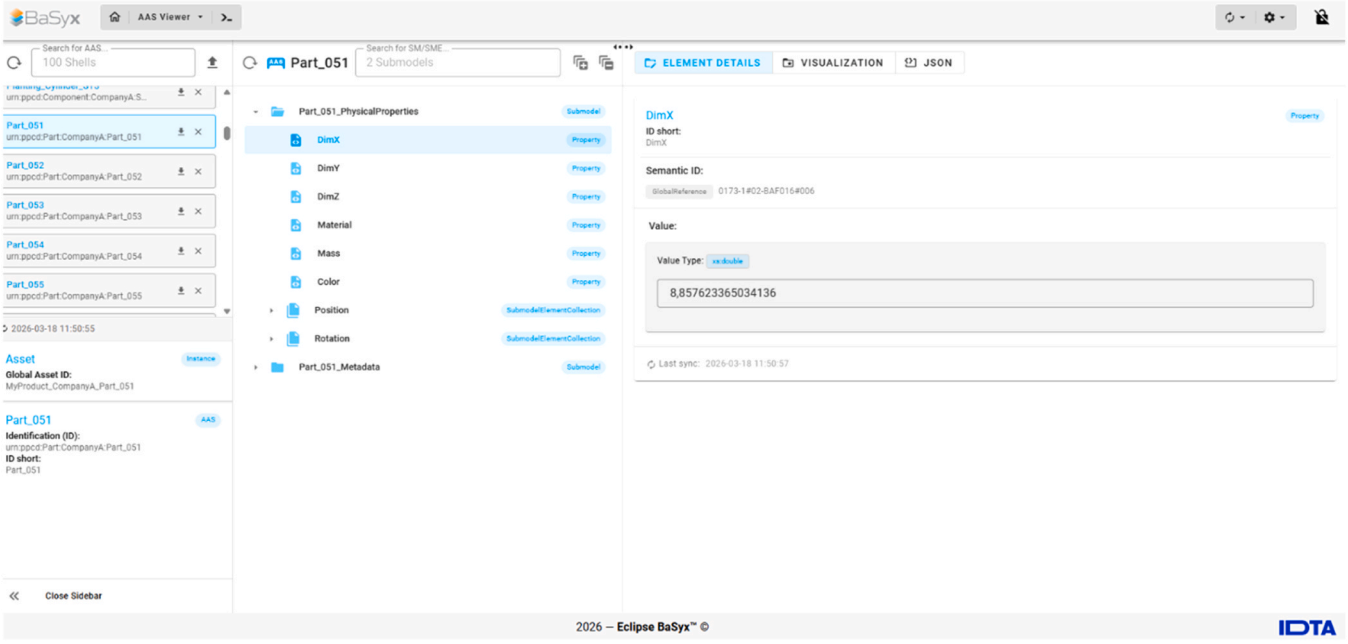

Fig. 16. Extract from AAS BaSyx platform with Company A entities and properties.

emerges as the most impacted element, while node *Part_001* represents the most impacting one (Fig. 18). Consequently, modifications to

```
ANALYSIS 4: List all codes impacted by a given code (transitive closure based on BoolValue=True).
Start code: Part_001
Codes impacted by Part_001 (6 total):
 - Part_002
 - Part_003
 - ST40_2
 - ST40_2_02c_01
 - ST40_3_02c_01
 - ST40_3_02c_02
```

Fig. 17. Direct and indirect impacts from input code by user.

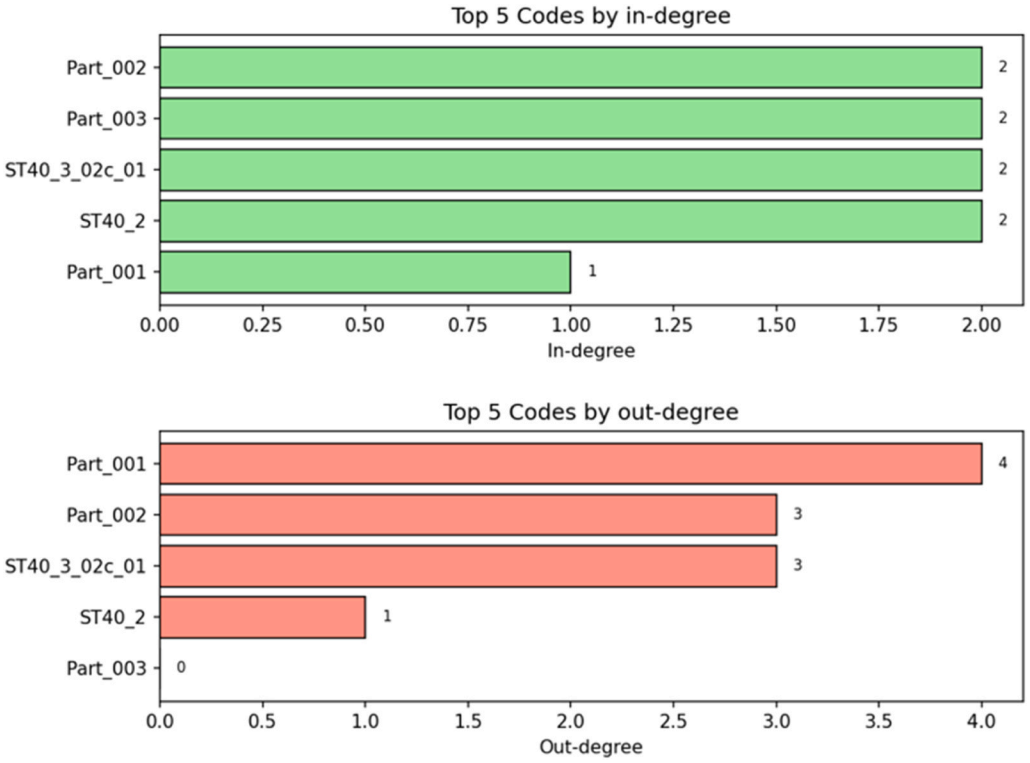

Fig. 18. Risk-oriented analysis of in- and out-degree for identification of most critical parts, tasks, and components of Company A.

Part_001 may propagate across several entities, potentially requiring multiple design adjustments. Once the most critical elements are identified, designers and engineers can search for their codes to study their precise directly or indirectly impacted nodes.

6.2. Company B implementation

Company B is in the development and planning phase for a circular value creation system in the white goods industry. The company already has linear assembly systems for its products and has developed a system-level concept for a hybrid production system integrating assembly, disassembly, and reconditioning of one specific product variant. This concept is based on a high-throughput scenario and consists of 65 workstations with dedicated process steps. Simultaneously, the product design is being reevaluated to enable efficient disassembly, reassembly, and linear assembly.

One explored redesign measure involves changing the connectors of three parts (referred to as *Part 1, Part 2,* and *Part 3*) from a connector that cannot be removed non-destructively to a connector that can be removed without affecting the components. This design change would enable a more targeted disassembly, as Parts 1–3 could be disassembled individually based on their conditions instead of disassembling the whole module. As the components are classified as critical for customer acceptance of refurbished and remanufactured products, there is a need for targeted disassembly. The design change is expected to affect the circular production layout, the specifications for component handling devices, or the dimensions of the manual workstations. To be able to evaluate the feasibility of such a design change for the production system design, decision support is needed for the concurrent reengineering with the aim of achieving more circularity.

A partial CAD model of the product was provided and used to first represent the parts and their dimensions in the AAS environment, based on the information model. Additionally, the description of the workstation and operation was available, where parts 1–3 are currently disassembled as one part. These descriptions included the composition of the manual workstation and dedicated operation as well as needed tools and expected processing times. This information was integrated into the CEDP using the defined interfaces for tabular data. Interdependencies for this subset of product, process, and production system information were added at the attribute level. An impact analysis was then conducted to mainly identify the impact of the design change on the production system.

The analysis results are visualized in Fig. 19 and Fig. 20 where both product designers and production system engineers can derive the workstation component that would be the most significantly impacted by a design change in *Part 1*, *Part 2,* or *Part 3*. Here, it becomes apparent that *Component 2* of the analyzed workstation *AP16* is the most impacted by the product design change. Potential implications include a redesign of *Component 2* to enable a wider variety of parts to be handled at workstation *AP16*. Within the product design change, *Part 2* has the

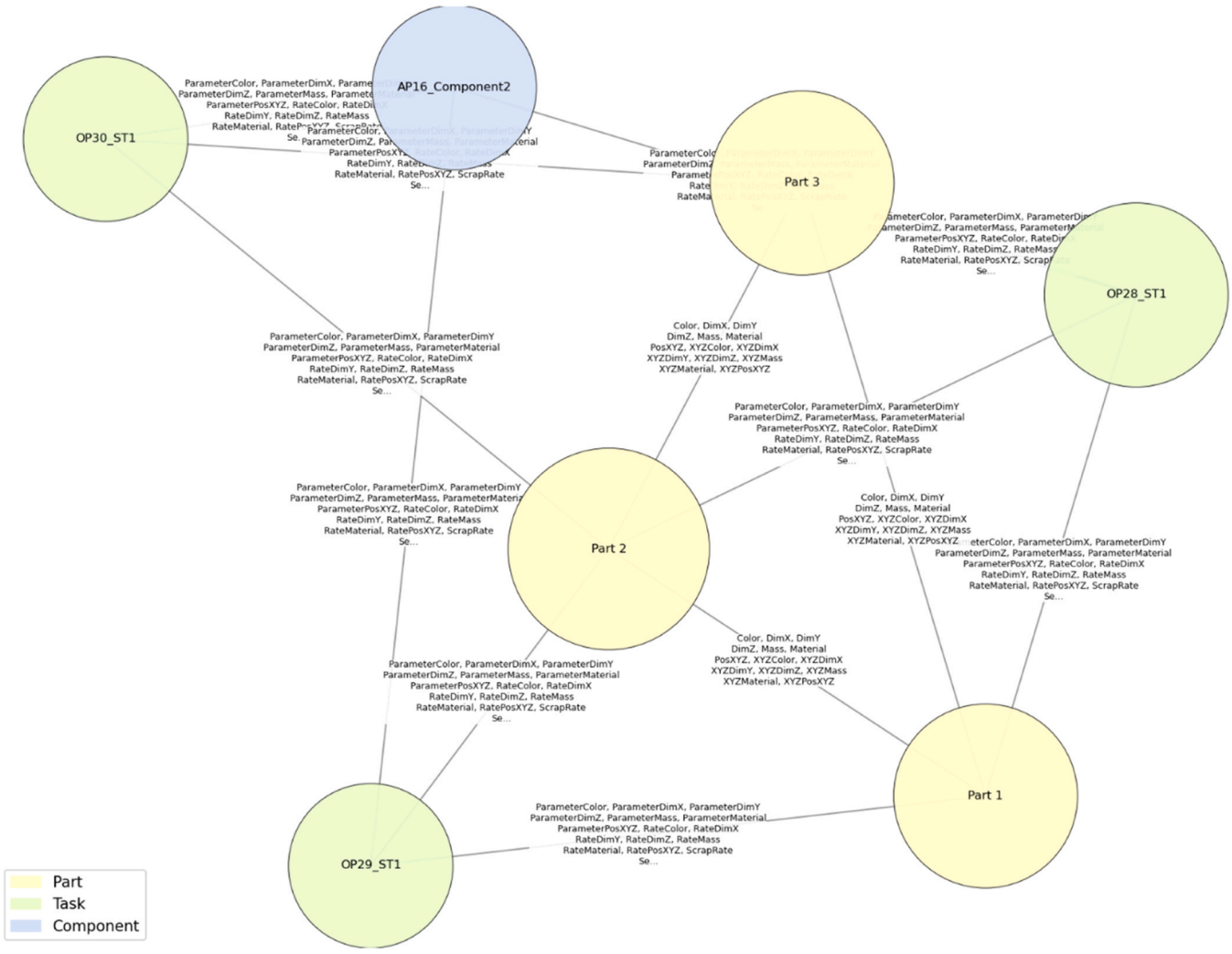

Fig. 19. System-level representation of chains of changes of instantiated parts, tasks, and components of Company B.

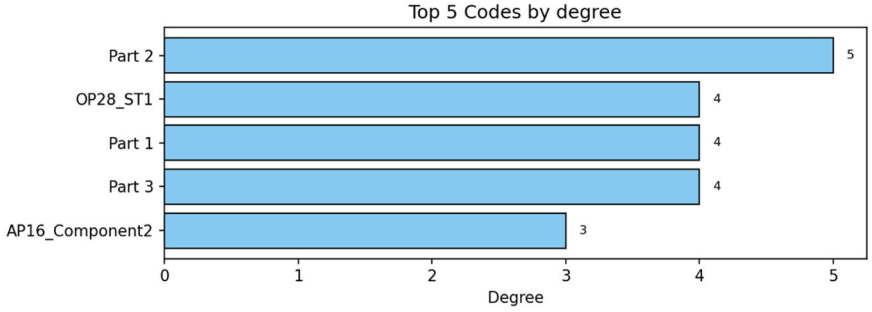

Fig. 20. Risk-oriented analysis of in- and out-degree for identification of most critical parts, tasks, and components of Company B.

biggest influence on the process and production system.

This case addresses the circular product creation challenge of coordinating design changes for selective and non-destructive disassembly with the corresponding adaptations to production processes and systems. By tracing these cross-domain impacts, the information model and CEDP support concurrent product and production system (re)design and help engineers assess alternatives aimed at component recovery. Although the approach does not select or optimize circular strategies, it provides the integrated PPR information required for more informed circular decision-making. If maintained throughout the lifecycle through the AAS, information established at the BoL can also support the evaluation of EoL strategies across all three domains.

6.3. Results

This section presents the results from the application of the proposed approach in the two industrial cases. Table 4 Reports the AAS instances created for the product, process, and production system domains, while Table 5 summarizes the nodes and relationships derived from the modeled interdependencies. The CAD models provided by Company A and Company B contained 66 and 2700 solids, respectively, reflecting different levels of product complexity.

Table 4
Instantiated AAS in Company A and Company B implementations.

PPR domain	AAS in Company A	AAS in Company B
Product	• 1 Product • 3 subAssembly • 62 Part	• 1 Product • 2699 StructuralPart • 1 ElectronicPart
Process	• 3 Process • 8 Task	• 30 Process • 30 Task
Production System	• 2 ProductionMachine • 1 CartesianRobot • 1 RotaryTable • 1 SafetyEquipment • 7 subSystem • 14 Component	• 1 ProductionMachine • 1 Operator • 1 Vehicle • 1 HandTool • 1 subSystem • 2 Component

Table 5
Nodes and Relationships in the knowledge graph implementation of Company A and Company B.

Graph element	Company A	Company B
Nodes	89	2768
Arcs	9	64

The CEDP enabled automatic extraction of product data from STEP files, including bounding-box dimensions, center-of-mass position, and color, and their integration with process and production system data from structured data frames. This generated interconnected AAS instances across the three domains. The resulting knowledge graphs comprised 89 nodes and 9 relationships for Company A and 2768 nodes and 64 relationships for Company B. Grounded in the information model, these structures supported change-propagation and risk-oriented analyses of cross-domain interdependencies.

For Company A, the approach provided an initial means of addressing mass customization by tracing how potential product-variant changes, such as modifications in size, color, or material, could propagate to process tasks, parameters, and production system configurations. The analysis identified *Part_001* as the most influential element and *Part_002* as the most affected, helping engineers prioritize further assessments and coordinate potential adaptations with the machine builder before finalizing the production system layout and control logic.

For Company B, the approach supported circular manufacturing by linking product changes for selective, non-destructive disassembly to their effects on processes and production resources. *Component 2* of workstation *AP16* was identified as the most affected production system element, while *Part 2* showed the greatest cross-domain influence. By integrating product and production knowledge generated at the BoL, the approach provided a structured basis for assessing adaptations required for component recovery and informing EoL decisions. However, it supports the assessment of circular alternatives without selecting or optimizing a specific strategy.

Finally, the results indicate that integrating the information model and AAS through the CEDP provides a structured basis for managing PPR information and interdependencies, supporting cross-functional information exchange and lifecycle-oriented decision-making.

7. Discussion

This section presents the theoretical implications of the current study in Subsection 7.1, the industrial implications in Subsection 7.2, and finally provides discussions on the industrial integration and scalability of the proposed information model and CEDP in Subsection 7.3.

7.1. Theoretical contributions

From a theoretical perspective, this work contributes to the literature on PPCD and Concurrent Engineering by addressing two interrelated gaps identified through the state-of-the-art analysis. First, existing integrated approaches represent the PPR domains asymmetrically, with the production system generally modeled less comprehensively than the product and process domains (*RG1*) [8,76]. Second, cross-domain interdependencies are formalized only in a limited way, especially under an integrated and comprehensive representation of PPR domains (*RG2*). In answering the stated RQ in the light of RG1 and RG2, the study provides the following contributions:

1) The proposed information model advances formal modeling of manufacturing systems during concurrent engineering processes by providing an integrated and comprehensive modeling of the production system domain under an integrated perspective with product and process domains. In particular, it represents production system entities, attributes, and internal hierarchies at a level of detail comparable to the other two domains. The model is aligned with the PPR paradigm and extends existing formal models [10,12,15–17,20,47,58,63] to support more integrated and holistic PPCD.
2) The information model provides a formal mechanism for representing physical, functional, and operational interdependencies across PPR domains. Four SysML-Profile stereotypes are introduced to formalize these cross-domain relationships. These stereotypes make the direction and semantics of cross-domain influences explicit and embed them within the corresponding PPR domain representations. Compared with current available approaches based mainly on matrices or partial domain representations [20], this approach not only provides semantics of physical, functional, and operational interdependencies, but also enables their formalization and operationalization under a comprehensive representation of the PPR domains. This provides a structured basis for cross-domain traceability and change propagation analysis, while mitigating knowledge fragmentation across engineering functions and organizations [12,27].
3) The operationalization of the proposed information model through AAS provides a systematic connection between formal system modeling and interoperable digital asset representations. The Mapping Layer implemented within the CEDP translates SysML constructs into corresponding AAS submodels, enabling the instantiation and integration of product, process, and production system information originating from heterogeneous engineering sources [66]. The two industrial applications demonstrate that this mapping can generate risk-based and system-level analysis among the three modeled domains through their knowledge graph representations. This evidence demonstrates the applicability of the integrated artifact rather than constituting an exhaustive validation of its effectiveness or industrial benefits.
4) The resulting information structure provides a foundation for lifecycle data continuity in circular manufacturing systems. Product, process, and production system information established during design can remain available for subsequent analyses, including disassembly and end-of-life planning [4]. The approach provides structured and traceable cross-domain information to inform circular decisions, without prescribing or optimizing circular strategies. Its integrated PPR representation supports the information management requirements of CPPCD and the consideration of linear and circular production within circular factories [4,77]. Furthermore, the proposed information model and CEDP provide a conceptual basis for reusing engineering knowledge and tracing changes across product and production system generations, consistent with PPCD and PGE [8], by linking various product and resource entities across different generations.

Overall, the contribution lies in the integrated representation of the PPR domains, alongside the explicit formalization of their interdependencies, and their overarching operationalization through AAS. SysML provides the modeling implementation demonstrated in this

study, while the proposed modeling structure could be implemented using alternative languages that preserve the same information and relationship semantics by adapting the proposed parsing and mapping rules based on the targeted modeling language.

7.2. Industrial implications

From an industrial perspective, the main implications can be reported as follows:

a) The proposed approach enables the structured organization and management of data originating from heterogeneous and cross-functional sources. By providing a shared information model, it allows different engineering functions and organizations to integrate design data, avoiding sequential information transfers from product to production design. This fosters more efficient coordination across multidisciplinary stakeholders, who can access data and related analysis from a shared and persistent model. At the same time, its industrial adoption requires organizations to redesign and adapt existing product and production engineering workflows to support more integrated and concurrent engineering practices. This transformation may also require dedicated change management initiatives involving manufacturers, suppliers, and machine builders to facilitate the transition toward these new workflows, while assessing the need for new competencies and organizational resources to effectively support integrated engineering processes;
b) The proposed information model enables risk-informed and system-level decision-making during PPCD iterations, in line with the principles of Asset Management (ISO 55000) [78]. By providing a holistic representation of product, process, and production system domains, stakeholders can identify critical parts, components, and engineering activities from the early design stages. Furthermore, the explicit modeling and analysis of the interdependencies among these domains enables the assessment of change propagation across cross-functional engineering domains and lifecycle stages, supporting more informed and coordinated design decisions.
c) The proposed approach enables different organizations involved in concurrent engineering, such as product-driven manufacturing companies and machine builders, to perform integrated analyses across product, process, and production system domains. Machine builders can assess the impact of product features and requirements on production system configuration from the early design stages, while product-driven companies can evaluate the feasibility of new product variants considering production constraints. This integration could help identify cross-domain inconsistencies earlier and support parallel product and production system design. Despite being technically feasible, potential reductions in redesign iterations, development time, and TTM remain to be established through comparative and longitudinal evaluation.
d) This work lays the foundation for lifecycle data continuity and more informed circular manufacturing decisions. Although demonstrated primarily during BoL design and planning, the AAS-based implementation allows integrated product, process, and production system information established at the BoL to remain available and be enriched during middle of life and EoL. When applied from BoL stages and connected with physical systems during operations, this approach supports the development of digital twins, enabling real-time data acquisition, interoperability, and continuous cross-domain analysis throughout the system lifecycle. For EoL planning, companies can therefore use traceable PPR information to assess how circular strategies may affect all three domains, as illustrated by the cross-domain analysis conducted in the Company B case. This approach therefore enables more informed circular manufacturing decisions, without prescribing or optimizing the strategies themselves.

Overall, the integration of SysML modeling and AAS-based data management fosters concurrent engineering practices, providing a further step toward flexible, interoperable, and circular manufacturing systems (Fig. 21).

7.3. Discussion on industrial integration and scalability

The two industrial applications presented in Section 6 demonstrated the feasibility of the proposed information model and CEDP within the selected configurations. Nevertheless, broader industrial adoption requires addressing challenges related to larger and more complex product and production systems, uncertainty in circular manufacturing, and the management and maintenance of the information model and CEDP across lifecycle stages. The following points discuss some of these relevant challenges and outlines possible extensions to address them.

- **Model maintenance and scalability.** Additional physical properties may be needed to characterize *Part* or *Component* entities for specific use cases. This expands both the entities' property sets and the *ImpactInteraction* stereotypes, whose Cartesian product structure may increase scalability challenges. For example, increasing the *Part* properties from eight to nine expands the *Part-Part* stereotype from 64 to 81 influence attributes. These changes require revising the affected stereotypes and corresponding AAS templates. A possible strategy would combine version identifiers for model definitions, templates, and instances with migration procedures that preserve existing values and references and assess only newly introduced influences to limit computational loads, avoiding scalability issues. Selective instantiation of entities and relationships relevant to the engineering objective could reduce modeling and maintenance effort throughout the lifecycle, while retaining intermediate links needed to trace indirect effects.

Furthermore, scalability challenges may arise from larger datasets or more complex structures of products and production machines. In this context, a persistent representation of AAS data in Resource Description Framework (RDF) could allow dependency queries to be executed directly in a database through SPARQL. The CEDP could then retrieve the entities or properties connected through direct and indirect influence links without loading the complete graph into the Python analysis process. This could support the reachability analyses described in Subsection 5.6, potentially reducing data transfer and application memory requirements and enabling reuse of the stored graph across queries.

- **Uncertainty of returned products in circular manufacturing.** Returned products can differ in structure and condition, including missing parts and uncertain wear [79]. However, the proposed approach does not model stochasticity and uncertainty regarding the condition of returned products, which represents a relevant aspect in circular manufacturing [79]. To this end, a possible extension would represent alternative part presence and condition states, with associated probabilities and conditional task and resource requirements to estimate the downstream impact probabilities for disassembly tasks. These representations would need to be defined in the information model, mapped into AAS, and populated using inspection data, historical records, or knowledge from experts.
- **Semantic formalization and migration to SysML v2.** In the present SysML v1.6 implementation, custom stereotypes encode property-level influences interpreted by the CEDP, without directly defining executable propagation behavior. SysML v2 extends the Kernel Modeling Language (KerML) metamodel, providing a formally grounded semantic foundation [80]. A possible transition would first map existing PPR blocks and properties to appropriate SysML v2 definitions, usages, and typed attributes. The custom influence relationships would then need to be reformulated, preserving

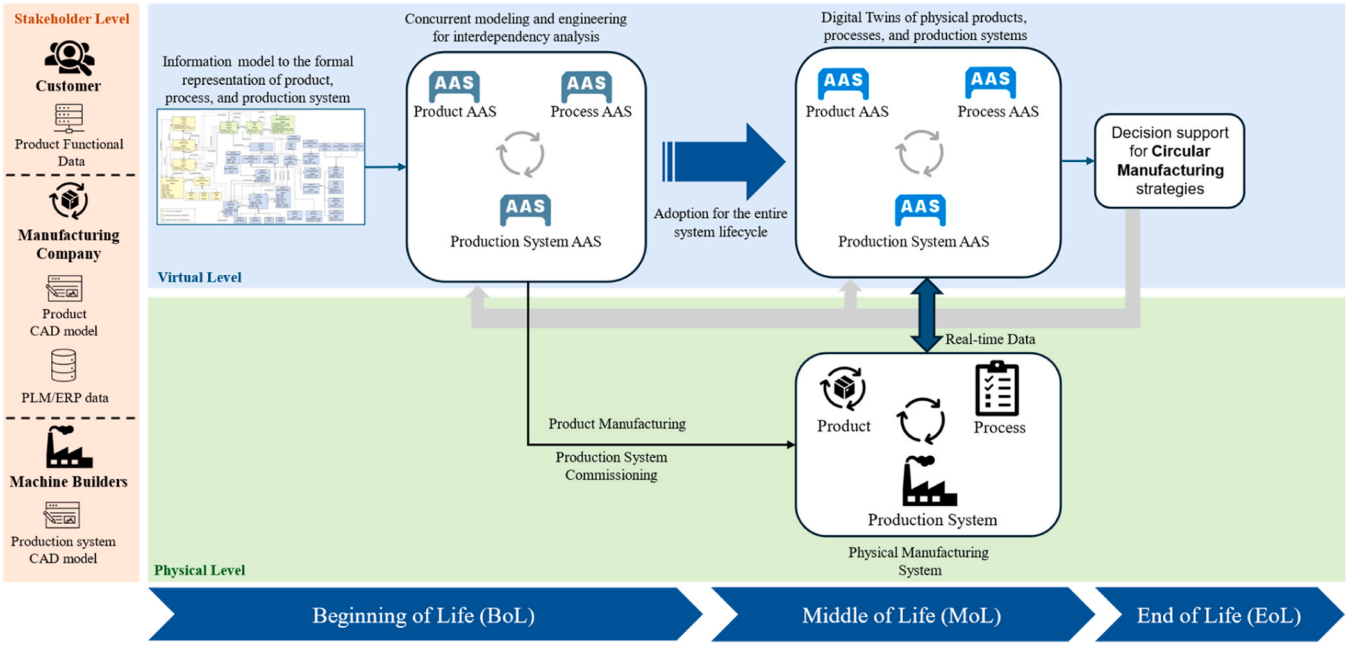

Fig. 21. Implications and usage of the proposed information model and Concurrent Engineering Data Pipeline for the manufacturing system lifecycle.

their direction, source and target properties, and influence values. Subsequently, the CEDP parser and AAS mapping rules would need to be adapted to the new metamodel and serialization. A pilot migration on a representative configuration could compare the generated AAS structures and reachability results with those obtained from SysML v1.6, checking that the intended information and influence semantics are preserved.

8. Conclusion

This study proposes an information model that enables the formal and comprehensive modeling of product, process, and production system domains, along with their interdependencies, within concurrent engineering processes. The proposed information model is operationalized through a CEDP and implemented in industrial assembly and disassembly contexts. The results demonstrate the capability of the approach to effectively manage data and information flows across domains, while supporting risk-based and cross-functional interdependency analysis. This enables concurrent evaluation of product, process, and production system configurations, potentially supporting greater flexibility and adaptability in response to mass customization requirements and increasing need for product-production integration toward circular factories. In this way, the proposed approach not only fosters concurrent engineering processes by means of integrated representations and analysis of cross-domain engineering domains but also paves the way for a lifecycle-oriented perspective through deployment of AAS, facilitating circular manufacturing decisions by means of efficient data lifecycle management from early design stages.

Despite its contribution, the following limitations remain. Interdependencies modeling still rely on manual data preparation, which increases modeling efforts. Moreover, the current implementation does not allow automatic instantiation of new child entities in response to changes to the properties of product, process, or production system entities. Also, the two industrial cases provide an initial demonstration of applicability, without evaluating measurable improvements in engineering and design workflows.

Building on these limitations and the industrial integration and scalability challenges outlined in Subsection 7.3, future research could investigate PSUM (Precise Semantics for Uncertainty Modeling)-SysMLv2 [81] to explicitly characterize uncertainty in returned product conditions and its effects on disassembly tasks and resource requirements. Integrating these representations with AAS and suitable analysis algorithms could support probabilistic change propagation analyses for circular manufacturing. Furthermore, future research should compare equivalent change assessment tasks performed using existing engineering workflows and the proposed information model and CEDP. Evaluation should measure completion time and the completeness and accuracy of impact identification against an independently reviewed engineering reference, while accounting for model preparation and maintenance effort.

Further research could also pursue the following research avenues. A key direction is the development of an ontology layer on top of the proposed information model to enable semantic enrichment and automated reasoning. Furthermore, the integration of Artificial Intelligence techniques within the analysis layer to support massive and integrated data analysis across the entire lifecycle is another relevant future direction. In parallel, extending and validating the approach beyond the BoL phase toward a full lifecycle perspective would further enhance its impact. This advancement could foster more advanced circular manufacturing strategies, supporting sustainable and circular objectives, particularly in EoL decision-making through improved cross-domain data management.

CRediT authorship contribution statement

Edoardo Palmitessa: Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sazvan Saeed:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Julian Haller:** Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation. **Amon Göppert:** Writing – review & editing, Supervision. **Adalberto Polenghi:** Writing – review & editing, Supervision. **Elisa Negri:** Writing – review & editing, Supervision. **Robert H. Schmitt:** Funding acquisition.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used OpenAI ChatGPT to enhance the English language of selected paragraphs. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was produced while the corresponding author attended the PhD program in Management Engineering at Politecnico di Milano, Cycle XXXIX, with the support of a scholarship co-financed by the Ministerial Decree no. 117 of 02/03/2023, based on the PNRR (Piano Nazionale di Ripresa e Resilienza) - funded by the European Union - NextGenerationEU - Mission 4 "Education and Research", Component 2 "From Research to Business", Investment 3.3.

This work was funded by the Federal Ministry for Economic Affairs and Energy (BMWE) in the GreenTech funding framework 'Development of digital technologies' as the joint research project DiCES (Digital Transformation of Circular Economy for Industrial Sustainability) with the funding code 01MN23022E.

This was work funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612 and Project Human Decision – 543081196 of Priority Programme SPP 2443: Hybrid decision support in product creation.

Disclosure statement

The authors report that there are no competing interests to declare.

Data Availability

Industrial use case data is confidential and could thus only be reported in anonymized form in this contribution. The publication's code is publicly available at https://git-ce.rwth-aachen.de/wzl-mq-public/digital-assembly/aas-mbse.

References

[1] Andersen R, Napoleone A, Andersen A-L, Brunoe TD, Nielsen K. A systematic methodology for changeable and reconfigurable manufacturing systems development. J Manuf Syst 2024;74:449–62. https://doi.org/10.1016/j.jmsy.2024.04.008.

[2] Wu B, Liu K, Zhang Z, Wu Z. Collaborative Design and Virtual Commissioning of Automated Pharmacy Based on Digital Twin. Proceedings of 2023 IEEE 16th International Conference on Electronic Measurement and Instruments. ICEMI; 2023. https://doi.org/10.1109/ICEMI59194.2023.10270467. 2023.

[3] May M.C., Schäfer L., Lachnit T., Lanza G. Product-Production-CoDesign Thinking for Sustainable Manufacturing, 2025, p. 185–193. https://doi.org/10.1007/978-3-031-77429-4_21.

[4] Acerbi F, Sassanelli C, Terzi S, Taisch M. A Systematic Literature Review on Data and Information Required for Circular Manufacturing Strategies Adoption. Sustainability 2021;13:2047. https://doi.org/10.3390/su13042047.

[5] Sassanelli C, Urbinati A, Rosa P, Chiaroni D, Terzi S. Addressing circular economy through design for X approaches: A systematic literature review. Comput Ind 2020;120:103245. https://doi.org/10.1016/j.compind.2020.103245.

[6] Yildiz E, Møller C, Bilberg A. Demonstration and evaluation of a digital twin-based virtual factory. Int J Adv Manuf Technol 2021;114:185–203. https://doi.org/10.1007/s00170-021-06825-w.

[7] Hörger M, Simon M, Hermann Y, Lachnit T, Benfer M, Lanza G. Leveraging Digital Product Passports to Derive Circular Strategies. Procedia CIRP 2026;139:411–7. https://doi.org/10.1016/j.procir.2025.09.033.

[8] Albers A, Lanza G, Klippert M, Schäfer L, Frey A, Hellweg F, et al. Product-Production-CoDesign: An Approach on Integrated Product and Production Engineering Across Generations and Life Cycles. Procedia CIRP 2022;109:167–72. https://doi.org/10.1016/j.procir.2022.05.231.

[9] Waaler A, Kiritsis D. Information Modelling Framework for Digital Engineering. Digit Eng 2025;4:100042. https://doi.org/10.1016/j.dte.2025.100042.

[10] Wu D, Zhang LL, Jiao RJ, Lu RF. SysML-based design chain information modeling for variety management in production reconfiguration. J Intell Manuf 2013;24:575–96. https://doi.org/10.1007/s10845-011-0585-6.

[11] Matsokis A, Kiritsis D. An ontology-based approach for Product Lifecycle Management. Comput Ind 2010;61:787–97. https://doi.org/10.1016/j.compind.2010.05.007.

[12] Negri E, Perotti S, Fumagalli L, Marchet G, Garetti M. Modelling internal logistics systems through ontologies. Comput Ind 2017;88:19–34. https://doi.org/10.1016/j.compind.2017.03.004.

[13] Hause M. SysML Model Lang 2006;9:1–12. https://www.omg.org/sysml/The_SysML_Modelling_Language.pdf.

[14] Friedenthal S, Moore A, Steiner R. A practical guide to SysML: the systems modeling language. Morgan Kaufmann; 2014. https://www.sciencedirect.com/book/monograph/9780128002025/a-practical-guide-to-sysml.

[15] Albers A, Stürmlinger T, Mandel C, Wang J, Frutos MBD, Behrendt M. Identification of potentials in the context of Design for Industry 4.0 and modelling of interdependencies between product and production processes. Procedia CIRP 2019;84:100–5. https://doi.org/10.1016/j.procir.2019.04.298.

[16] Lanz M, Garcia F, Kallela T, Tuokko R. Product-process ontology for managing assembly specific knowledge between product design and assembly system simulation. Springer; 2008. p. 99–108. https://doi.org/10.1007/978-0-387-77405-3_9.

[17] Lemaignan S, Siadat A, Dantan J-Y, Semenenko A. MASON: A Proposal For An Ontology Of Manufacturing Domain. IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), Prague, Czech Republic. IEEE; 2006. p. 195–200. https://doi.org/10.1109/DIS.2006.48.

[18] Borgo S, Leitão P. Foundations for a Core Ontology of Manufacturing. In: Sharman R, Kishore R, Ramesh R, editors. Ontologies, 14. Boston, MA: Springer US; 2007. p. 751–75. https://doi.org/10.1007/978-0-387-37022-4_27.

[19] Duverger E, Aubry A, Arista R, Levrat E. An ontology-based digital thread framework to support early concurrent engineering in the aerospace industry. J Ind Inf Integr 2025;48:100984. https://doi.org/10.1016/j.jii.2025.100984.

[20] Hoang X-L, Marks P, Weyrich M, Fay A. Modeling of interdependencies between products, processes and resources to support the evolution of mechatronic systems. IFAC-Pap 2017;50:4348–53. https://doi.org/10.1016/j.ifacol.2017.08.873.

[21] United Nations. About the Sustainable Development Goals- United Nations Sustainable Development [WWW Document]. 2019.

[22] Winner R.I., Pennell J.P., Bertrand H.E., Slusarczuk M.M.G. The Role of Concurrent Engineering in Weapons System Acquisition 1988.

[23] Meller RD, Gau K-Y. The facility layout problem: Recent and emerging trends and perspectives. J Manuf Syst 1996;15:351–66. https://doi.org/10.1016/0278-6125(96)84198-7.

[24] Putnik GD, Putnik Z. Defining Sequential Engineering (SeqE), Simultaneous Engineering (SE), Concurrent Engineering (CE) and Collaborative Engineering (ColE): On similarities and differences. Procedia CIRP 2019;84:68–75. https://doi.org/10.1016/j.procir.2019.07.005.

[25] Cimino C, Ferretti G, Leva A. Harmonising and integrating the Digital Twins multiverse: A paradigm and a toolset proposal. Comput Ind 2021;132:103501. https://doi.org/10.1016/j.compind.2021.103501.

[26] El Kadiri S, Kiritsis D. Ontologies in the context of product lifecycle management: state of the art literature review. Int J Prod Res 2015;53:5657–68. https://doi.org/10.1080/00207543.2015.1052155.

[27] Göppert A, Grahn L, Rachner J, Grunert D, Hort S, Schmitt RH. Pipeline for ontology-based modeling and automated deployment of digital twins for planning and control of manufacturing systems. J Intell Manuf 2023;34:2133–52. https://doi.org/10.1007/s10845-021-01860-6.

[28] Martin P. D'Acunto A. Design of a production system: An application of integration product-process. Int J Comput Integr Manuf 2003;16:509–16. https://doi.org/10.1080/0951192031000115831.

[29] Cutting-Decelle AF, Young RIM, Michel JJ, Grangel R, Le Cardinal J, Bourey JP. ISO 15531 MANDATE: A Product-process-resource based Approach for Managing Modularity in Production Management. Concurr Eng 2007;15:217–35. https://doi.org/10.1177/1063293X07079329.

[30] Panetto H, Dassisti M, Tursi A. ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. Adv Eng Inform 2012;26:334–48. https://doi.org/10.1016/j.aei.2011.12.002.

[31] Schäfer L, Günther M, Martin A, Lüpfert M, Mandel C, Rapp S, et al. Systematics for an Integrative Modelling of Product and Production System. Procedia CIRP 2023;118:104–9. https://doi.org/10.1016/j.procir.2023.06.019.

[32] Elmasri R, Navathe SB. Fundamentals of Database Systems. 7th ed. Pearson; 2016.

[33] Colledani M, Terkaj W, Tolio T, Tomasella M. Development of a Conceptual Reference Framework to Manage Manufacturing Knowledge Related to Products, Processes and Production Systems. In: Bernard A, Tichkiewitch S, editors. Methods and Tools for Effective Knowledge Life-Cycle-Management. Berlin, Heidelberg: Springer Berlin Heidelberg; 2008. p. 259–84. https://doi.org/10.1007/978-3-540-78431-9_15.

[34] Negri E, Fumagalli L, Garetti M, Tanca L. Requirements and languages for the semantic representation of manufacturing systems. Comput Ind 2016;81:55–66. https://doi.org/10.1016/j.compind.2015.10.009.

[35] Wu S, Wang G, Lu J, Huang J, Qiao J, Yan Y, et al. Cognitive digital thread toolchain for model versioning in model-based systems engineering. Adv Eng Inform 2025;67:103490. https://doi.org/10.1016/j.aei.2025.103490.
[36] Polenghi A, Roda I, Macchi M, Pozzetti A, Panetto H. Knowledge reuse for ontology modelling in Maintenance and Industrial Asset Management. J Ind Inf Integr 2022; 27:100298. https://doi.org/10.1016/j.jii.2021.100298.
[37] Chen J, Mashkova O, Zhapa-Camacho F, Hoehndorf R, He Y, Horrocks I. Ontology Embedding: A Survey of Methods, Applications and Resources. IEEE Trans Knowl Data Eng 2025;37:4193–212. https://doi.org/10.1109/TKDE.2025.3559023.
[38] Vernadat F. Enterprise modelling: Research review and outlook. Comput Ind 2020; 122:103265. https://doi.org/10.1016/j.compind.2020.103265.
[39] Kosanke K, Vernadat F, Zelm M. CIMOSA: enterprise engineering and integration. Comput Ind 1999;40:83–97. https://doi.org/10.1016/S0166-3615(99)00016-0.
[40] Sowa JF, Zachman JA. Extending and formalizing the framework for information systems architecture. IBM Syst J 1992;31:590–616. https://doi.org/10.1147/sj.313.0590.
[41] ISO 19439, 2006. Enterprise integration — Framework for enterprise modelling 2006.
[42] ISO/IEC 81346 2022. Industrial systems, installations and equipment and industrial products- Structuring principles and reference designations- Part 1: Basic rules 2022.
[43] ISO 15926-14 2020. Industrial automation systems and integration — Integration of life-cycle data for process plants including oil an 2020.
[44] Disselkamp J-P, Cieply J, Dyck F, Grothe R, Anacker H, Dumitrescu R. Integrated product and production development - a systematic literature review. Procedia CIRP 2023;119:716–21. https://doi.org/10.1016/j.procir.2023.06.198.
[45] Vizitiu C, Dinculescu AC, Marin MG, Nistorescu AI, Dominey KA, Kristály DM. Optimizing early-stage efficiency in complex system development via model-based systems engineering (MBSE) and concurrent engineering (CE) integration. Res Eng Des 2025;36:12. https://doi.org/10.1007/s00163-025-00452-2.
[46] Usman Z, Young RIM, Chungoora N, Palmer C, Case K, Harding J. A Manufacturing Core Concepts Ontology for Product Lifecycle Interoperability. In: Van Sinderen M, Johnson P, editors. Enterprise Interoperability, 76. Berlin, Heidelberg: Springer Berlin Heidelberg; 2011. p. 5–18. https://doi.org/10.1007/978-3-642-19680-5_3.
[47] Garetti M, Fumagalli L. P-PSO ontology for manufacturing systems. IFAC Proc Vol 2012;45:449–56. https://doi.org/10.3182/20120523-3-RO-2023.00222.
[48] Mandel C, Sturmlinger T, Yue C, Behrendt M, Albers A. Model-Based Systems Engineering Approaches for the integrated development of product and production systems in the context of Industry 4.0. 2020 IEEE International Systems Conference (SysCon). Montreal, QC, Canada: IEEE; 2020. p. 1–7. https://doi.org/10.1109/SysCon47679.2020.9275902.
[49] Schäfer L, Frank A, May MC, Lanza G. Automated Derivation of Optimal Production Sequences from Product Data. Procedia CIRP 2022;107:469–74. https://doi.org/10.1016/j.procir.2022.05.010.
[50] Hu X, Barwasser A, Werner A, Schuseil F, Lentes J, Hertwig M, et al. Graph Model Based Bill of Material Structure for Coupling Product Development and Production Planning. In: Huang C-Y, Dekkers R, Chiu SF, Popescu D, Quezada L, editors. Intelligent and Transformative Production in Pandemic Times. Cham: Springer International Publishing; 2023. p. 593–605. https://doi.org/10.1007/978-3-031-18641-7_55.
[51] Wawer M., Herrmann K., Plappert S., Wurst J., Gembarski P., Lachmayer R. Structuring and Provision of Manufacturing knowledge through the Manufacturing Resource Ontology. 2023. https://doi.org/10.35199/dfx2023.08.
[52] Gebhard J, Emrich A, Wang Y, Loos P. AI-powered knowledge graphs: Accelerating simultaneous engineering in the automotive digital era. Digit Eng 2025;7:100070. https://doi.org/10.1016/j.dte.2025.100070.
[53] Vizitiu C, Turcin I, Dominey K, Dinculescu A, Nistorescu A, Marin M, et al. Concurrent Design Facility Definition for Advancing the Integration of Smart Automation in Human Space Applications. IFIPAICT 2026;766:104–17. https://doi.org/10.1007/978-3-032-03538-7_8.
[54] Duverger E, Arista R, Aubry A, Levrat E. A Hybrid Knowledge Extraction Method to Support Early Concurrent Engineering in the Aerospace Industry. Aerospace 2026; 13:337. https://doi.org/10.3390/aerospace13040337.
[55] Simon HA. The Sciences of the Artificial, reissue of the third edition with a new introduction by John Laird. MIT press; 2019. https://doi.org/10.7551/mitpress/12107.001.0001.
[56] Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S. A design science research methodology for information systems research. J Manag Inf Syst 2007;24:45–77.
[57] Vom Brocke J, Hevner A, Maedche A. Introduction to Design Science Research. In: Vom Brocke J, Hevner A, Maedche A, editors. Design Science Research. Cases. Cham: Springer International Publishing; 2020. p. 1–13. https://doi.org/10.1007/978-3-030-46781-4_1.
[58] Ahmad M, Ahmad B, Harrison R, Ferrer BR, Lastra JLM, Meredith J, et al. A knowledge-based approach for the selection of assembly equipment based on fuel cell component characteristics. IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society. Yokohama: IEEE; 2015. p. 001002–7. https://doi.org/10.1109/IECON.2015.7392230.
[59] Aldanondo M, Rouge S, Ve´ron M. Expert configurator for concurrent engineering: Came´le´on software and model. J Intell Manuf 2000;11:127–34. https://doi.org/10.1023/A:1008982531278.
[60] Jacob A, Windhuber K, Ranke D, Lanza G. Planning, Evaluation and Optimization of Product Design and Manufacturing Technology Chains for New Product and Production Technologies on the Example of Additive Manufacturing. Procedia CIRP 2018;70:108–13. https://doi.org/10.1016/j.procir.2018.02.049.
[61] Meixner K, Feichtinger K, Fadhlillah HS, Greiner S, Marcher H, Rabiser R, et al. Variability modeling of products, processes, and resources in cyber–physical production systems engineering. J Syst Softw 2024;211:112007. https://doi.org/10.1016/j.jss.2024.112007.
[62] Geum Y, Park Y. Designing the sustainable product-service integration: a product-service blueprint approach. J Clean Prod 2011;19:1601–14. https://doi.org/10.1016/j.jclepro.2011.05.017.
[63] Negri E, Fumagalli L, Macchi M, Garetti M. Ontology for Service-Based Control of Production Systems. In: Umeda S, Nakano M, Mizuyama H, Hibino H, Kiritsis D, Von Cieminski G, editors. Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth, 460. Cham: Springer International Publishing; 2015. p. 484–92. https://doi.org/10.1007/978-3-319-22759-7_56.
[64] International Electrotechnical Commission. IEC 63278-1:2024 Asset administration shell for industrial applications – Part 1: Asset administration shell structure. Geneva: IEC; 2024.
[65] Industrial Digital Twin Association (IDTA). Details of the Asset Administration Shell – Part 1: The exchange of information between partners in the value chain of Industrie 4.0. 2022.
[66] Ferko E, Berardinelli L, Bucaioni A, Behnam M, Wimmer M. Towards Interoperable Digital Twins: Integrating SysML into AAS with Higher-Order Transformations. 2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C). 2024. p. 342–9. https://doi.org/10.1109/ICSA-C63560.2024.00063.
[67] Barbie P, Pollom A, Fischer R-P, Becker M. Automated Generation of Standardised Digital Twins Based on MBSE Models. Proceedings of the 13th International Conference on Model-Based Software and Systems Engineering. Porto, Portugal: SCITEPRESS - Science and Technology Publications; 2025. p. 75–84. https://doi.org/10.5220/0013190300003896.
[68] Geibel F. Digital Twin in Industrial Applications – How Model-Based Systems Engineering (MBSE) and Asset Administration Shell (AAS) complement each other. 2023. https://doi.org/10.22032/dbt.58900.
[69] Dhouib S, Huang Y, Smaoui A, Bhanja T, Gezer V. Papyrus4Manufacturing: A Model-Based Systems Engineering approach to AAS Digital Twins. 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). Sinaia, Romania: IEEE; 2023. p. 1–8. https://doi.org/10.1109/ETFA54631.2023.10275523.
[70] Grüner S, Eskandani N. Towards an Engineering Workflow Management System for Asset Administration Shells using BPMN. IFAC-Pap 2025;59:83–8. https://doi.org/10.1016/j.ifacol.2025.11.929.
[71] Ochoa W, Larrinaga F, Pérez A. Architecture for managing AAS-based business processes. Procedia Comput Sci 2023;217:217–26. https://doi.org/10.1016/j.procs.2022.12.217.
[72] Mediavilla MA, Lagnese M, Pomp A, Meisen T. Asset administration shell-based engineering change management process: Challenges and ways forward. Procedia CIRP 2023;120:1010–5. https://doi.org/10.1016/j.procir.2023.09.116.
[73] Eclipse Foundation. Eclipse BaSyx 2024.
[74] Döller V. Formalizing the Four-layer Metamodeling Stack – Potential and Benefits 2021. https://doi.org/10.48550/ARXIV.2105.01038.
[75] Industrial Digital Twin Association (IDTA). Submodel Template: Hierarchical Structures Enabling Bills of Material (BoM). Frankfurt Am Main: 2024.
[76] Cao Q, Zanni-Merk C, Reich C. Ontologies for Manufacturing Process Modeling: A Survey. In: Dao D, Howlett RJ, Setchi R, Vlacic L, editors. Sustainable Design and Manufacturing 2018, 130. Cham: Springer International Publishing; 2019. p. 61–70. https://doi.org/10.1007/978-3-030-04290-5_7.
[77] Cappelletti F, Rossi M, Germani M. How de-manufacturing supports circular economy linking design and EoL - a literature review. J Manuf Syst 2022;63: 118–33. https://doi.org/10.1016/j.jmsy.2022.03.007.
[78] Roda I, Macchi M. A framework to embed Asset Management in production companies. Proc Inst Mech Eng Part O J Risk Reliab 2018;232:368–78. https://doi.org/10.1177/1748006X17753501.
[79] Gao Y, Lou S, Zheng H, Tan J. A data-driven method of selective disassembly planning at end-of-life under uncertainty. J Intell Manuf 2023;34:565–85. https://doi.org/10.1007/s10845-021-01812-0.
[80] Object Management Group. OMG Systems Modeling Language (SysML), Version 2.0, Part 1: Language Specification. Object Management Group; 2026.
[81] Zhang M, Li Y, Yue T. Uncertain Model SysML v2 2026. https://doi.org/10.48550/arXiv.2602.21641.