

Patterns for Reuse in Production Systems Engineering

Kristof Meixner*[†], Arndt Lüder^{‡||}, Jan Herzog[§], Dietmar Winkler*[†], Stefan Biffi^{†||}

*Christian Doppler Laboratory SQL, [†]Inst. of Information Sys. Eng., TU Wien and ^{||}CDP, Austria

E-Mail: [first].[last]@tuwien.ac.at

[‡]Inst. of Ergonomics, Manufacturing Sys. and Automation, Otto-von-Guericke U., Germany

E-Mail: arndt.lueder@ovgu.de

[§]Department PPG-M/D, Volkswagen AG, Wolfsburg, Germany

E-Mail: [first].[last]@volkswagen.de

Abstract—In *Production Systems Engineering (PSE)*, domain experts aim at reusing production processes implemented as Industry 4.0 assets and software. However, the knowledge on reusable assets is often scattered on multi-disciplinary engineering artifacts and domain experts, making it hard to find suitable reusable assets and map them to requirements. In this paper, we (i) identify challenges and requirements for reuse in PSE based on a domain analysis; (ii) introduce the *Industry 4.0 Asset Network (I4AN)* that integrates multi-disciplinary dependencies between the assets and exposes recurring patterns; and (iii) present four patterns for reuse in PSE that aim at improving reuse efficiency and risk. We evaluate the I4AN with reuse scenarios in a feasibility study. The study results indicate that the I4AN model satisfies the elicited requirements and enables PSE domain experts to identify patterns for reuse in their contexts.

Keywords—Reuse, Production Systems Engineering, Industry 4.0 asset, Industry 4.0 component.

I. INTRODUCTION

The *Industry 4.0 (I4.0)* initiative¹ has led to an increased focus on research related to *Production Systems Engineering (PSE)* in various research fields [1]. The I4.0 initiative envisions flexible and highly customizable production systems that interconnect modern manufacturing with the latest information and communication technology, so-called *Cyber-Physical Production Systems (CPPSs)* [2] that can self-adapt to particular conditions. These CPPSs incorporate *I4.0 assets* representing objects of perceived or actual value, such as products, processes, or resources [3]. The *Asset Administration Shell (AAS)*, their standardized digital representation [3], can describe their *skills* [4] and adapt the I4.0 assets to changes in the production environment. The aim is to fulfill business demands for increased flexibility and distribution of production, i.e., *production as a service*, and to react to shorter product life-cycles with reduced PSE project duration and effort [5].

These demands require the (partial) reuse of process and resource solutions from previous projects or standardized catalogues [6], [7]. Examples in automotive manufacturing are *position and screw* tasks, like screwing a dashboard into a car. In such cases, product type variants and their parameters vary,

e.g., where and how tightly to screw which kind of screw to a dashboard. Yet, the processes and production resources executing these tasks, like robot arms, are quite similar. In addition, parts of the software controlling the resources and orchestrating the overall production system can be reused.

Reuse in PSE depends on efficiently identifying recurring patterns that can be integrated into a production system. These patterns need to follow reference architectures [8], [9] of (i) product types, e.g., car types, (ii) production processes, e.g., screwing processes, and (iii) production resource types and instances, e.g., screwing robots. Reuse also requires a pattern description on type and instance levels to facilitate referring to vendor catalogues or previous projects [6], [10].

The engineering of a production system is a collaborative effort of experts coming from many disciplines, like mechanical, electrical, and software engineering [11]. However, traditionally much of the engineering information is hidden in *scattered engineering artifacts* and much of the knowledge is *implicit domain knowledge* of engineering experts [5] (cf. Section IV). Furthermore, there is insufficient interdisciplinary exchange between the domains, leading to *hard to extract/collect/validate dependencies* from heterogeneous engineering artifacts and domain experts [12]. Hence, it is crucial in this multi-disciplinary environment to thoroughly model the (interdisciplinary) dependencies and boundaries in pattern analysis to reduce the risk of broken reusable assets.

Hence, we raise the main research question: *What approach can PSE experts use to efficiently identify patterns from existing engineering knowledge for reusing Industry 4.0 assets and related artifacts?*

In this paper, we (i) identify challenges and requirements for knowledge reuse in PSE, (ii) introduce the *Industry 4.0 Asset Network (I4AN)*, a model to integrate the scattered knowledge and enable engineers to identify patterns for reuse to improve the effectiveness and efficiency of the PSE life-cycle, and (iii) present four recurring high-level patterns in PSE as a basis for identifying applied solution patterns for similar problems. We evaluate these contributions with an instance of an I4AN. Therefore, we investigate to what extent typical reuse scenario questions can be answered as queries to the I4AN.

The remainder of this paper is structured as follows: Sec-

tion II summarizes related work. Section III presents the research questions and method. Section IV introduces an illustrative use case and identifies requirements for reuse in PSE. Section V introduces the *Industry 4.0 Asset Network (I4AN)* for reuse in PSE and four high-level reusable patterns. Section VI reports on a feasibility study to evaluate the I4AN capabilities and discusses the results and limitations of the research. Section VII concludes and outlines future work.

II. RELATED WORK

This section summarizes related work on *Production Systems Engineering (PSE)*, knowledge management, and reuse.

A. Production Systems Engineering

PSE is a multi-disciplinary process that involves various disciplines, like mechanical, electrical, and software engineering [11]. Engineering teams iteratively perform tasks, like mechanical design or implementation of the control software [13], to engineer the desired production system.

In PSE, engineers create various types of engineering artifacts and models [5], [14]. However, the used formats and tools have traditionally been optimized for a single discipline, and while engineers are well connected within their domains, there is often an insufficient interdisciplinary exchange. Further, the engineering artifacts and information are scattered throughout the engineering landscape [5]. Much of the engineering knowledge is implicit knowledge of the domain experts. These issues pose an increasing challenge related to information management and reuse within PSE projects [15].

Yet, for the suitable and correct production system design, it is crucial to exchange information and knowledge between the disciplines effectively and efficiently [16]. In addition, the reuse of designs and specifications for recurring problems, e.g., using a robot type for similar tasks, improves the quality and helps to reduce PSE project duration, effort, and risk [8].

In this paper, we introduce a network, based on explicitly linked assets and artifacts, to provide the foundation for domain experts to link their specific knowledge representations.

Industry 4.0 (I4.0) addresses the overall digitalization and networking of production system elements, i.e., I4.0 assets, towards *Cyber-Physical Production Systems (CPPS)*. I4.0 assets are physical or immaterial objects of perceived or actual value [3]. An increasing focus can be recognized on product, process, and resource-related I4.0 assets, which we mainly refer to. An *Asset Administration Shell (AAS)* provides a digital representation [3] of I4.0 assets with their property views and skills. These descriptions should include the information and knowledge for an automated orchestration, which requires explicit knowledge on the I4.0 assets and their dependencies.

Pfrommer *et al.* [17] define a skill as the ability of a resource to perform a process, while a production skill gives the requirements [4]. Candidio *et al.* [18] understand a skill as ability to perform actions that are needed to support the production process. Meixner *et al.* [10] described how to abstract skills of resources from process requirements. Hence, models must represent the *required skills* of processes and the

provided skills of resources [17], [19]. However, the identification of reusable I4.0 Asset candidates requires representing skills as I4.0 Assets to provide the abstraction for the digital representation of boundaries between reusable patterns.

In this paper, we represent *skills* for the first time as I4.0 Asset, as an abstraction between processes and resources to foster the identification of reusable I4.0 Asset candidates.

B. Modeling Engineering Knowledge in PSE

Sabou *et al.* [20] introduced a knowledge graph for reuse in the software engineering domain, but without multi-model links that facilitate reuse in PSE.

For modeling Cyber-Physical Production Systems (CPPSs), two main research and development directions have been pursued. First, IT systems engineering uses CASE tools based on UML [21]. As a result, systems engineering methodologies have been created utilizing domain-crossing modeling standards like SysML² [22]. Second, engineering data exchange preserves the multi-model nature of PSE knowledge and builds on standardized data formats like *AutomationML*³ [16] to make data integration more efficient. In this paper, we build on cross-linking assets and engineering artifacts as a basis for an improved reuse considering dependencies.

Both directions require for explicitly modeling PSE knowledge to reflect the specifics of this domain, including (i) modeling *part-whole* relations, (ii) *connections between components* [23], and (iii) *technical dependencies* of the various involved technical disciplines [24]. The authors observed in PSE containment hierarchies to be well-established and frequently used to organize assets in PSE models. Furthermore, discipline-specific dependencies are often represented in discipline-specific models as interfaces.

Feldmann *et al.* [25] introduced an approach for managing inconsistencies in a multi-disciplinary multi-model environment using links between objects in PSE. However, the approach by Feldmann *et al.* [25] does not consider I4.0 assets and skills as first-class citizens in PSE. However, this integration is a foundation for better identifying reusable assets based on the digital representation of their skills. In this paper, we build on their meta-model [25] to integrate links between I4.0 assets coming from several engineering disciplines.

C. Reuse in Production Systems Engineering

Main approaches to reuse are (i) *clone and own* [26] and (ii) *reuse of components*, such as software libraries. However, these general reuse approaches do not sufficiently cover requirements in multi-disciplinary environments, like PSE.

In PSE, several reference frameworks address the reuse of assets. The guideline VDI 2206 [27] describes the V-Model as a procedure for structured PSE. It encourages to reuse requirements and partial implementations in later phases, like the test phase, without mentioning how. Jazdi *et al.* [8] provided first methodologies related to the systematic identification of

²SysML: <https://www.sysml.org>

³AutomationML: <https://www.automationml.org>

reusable system components. Most of them are based on the idea of mechatronic systems [28] following the VDI 2206 [27].

The guideline VDI 3695 [29] understands reuse as a method for engineering optimization and defines five types of reuse, i.e., *Reuse Levels (RL)*: Reuse (i) by employees on their own accord (RL-A), (ii) controlled within the project (RL-B), (iii) controlled from a central point across all projects (RL-C), (iv) based on a reference model (RL-D), and (v) based on internal and external standards (RL-E). The effectiveness and efficiency of reuse of assets depends on the level of reuse maturity [29] and the relations between assets [9]. Yet, this information is insufficiently available in PSE due to scattered artifacts and information.

In software engineering, one specific domain in the PSE engineering process, *design patterns* [30] are a widely adopted standard for reuse. Design patterns aim at developing software faster and in better quality while reducing risks and cost [30]. Therefore, design patterns provide adaptable design solution templates to general problems that software developers face. A design pattern consists of (i) a pattern name, (ii) a description of a problem that should be solved, (iii) a solution description with its elements and dependencies, and (iv) implications of the pattern, such as benefits and limitations. Software design patterns can serve as a blueprint for PSE design patterns but need to be adapted to the multi-disciplinary context of PSE.

In this paper, we build on the guideline VDI 3695 to describe the reuse maturity in PSE and the idea of mechatronic units as reusable entities [28]. Furthermore, we build on *design patterns* as a concept to identify reusable patterns in PSE.

III. RESEARCH QUESTIONS AND APPROACH

In this paper, we follow the *Design Science* methodology [31] to investigate how to improve identifying I4.0 assets for reuse in PSE. Therefore, we (i) conducted a domain analysis in the automotive industry, (ii) condensed a representative use case, and (iii) elicited requirements on I4.0 asset reuse with domain experts at medium-to-large European PSE companies (cf. Section IV).

Considering identified gaps in the related work and requirements in PSE, we formulate the following research questions.

RQ1a. *What model and elements facilitate identifying I4.0 assets for reuse in PSE?* The systematic reuse of I4.0 assets in engineering fosters quality and efficiency [8]. However, in PSE, the knowledge required for reuse often consists of heterogeneous information and implicit knowledge, scattered across the engineering landscape. To address RQ1a, we investigated recurring engineering artifacts from the domain analysis to identify knowledge elements that help engineers in efficiently identifying reusable assets. Our contribution is the *Industry 4.0 Asset Network (I4AN)* as a foundation to explore assets suitable for reuse.

RQ1b. *What connections between system parts and engineering artifacts represent dependencies in an I4.0 asset network as a foundation for identifying sets of reusable assets?* Connections and relationships between I4.0 assets provide data to understand internal and external dependencies of CPPS

assets. These dependencies are crucial to coherently identify and explain which potentially reusable assets can be reused *as-is* or require further assets to be included to correctly reuse them. To address RQ1b, we build on the *Industry 4.0 Asset Network (I4AN)*, coming from RQ1a, and investigated which links represent internal and external dependencies that are relevant to facilitate the reuse of assets. Our contributions focus on the classification of dependencies in the I4AN that are crucial to identify sets of reusable assets.

RQ2. *Which basic patterns for reuse facilitate identifying best-practice pattern candidates for PSE?* For identifying patterns for reuse, engineers require a starting point in their particular context. For instance, engineers are likely to recognize a pattern as an initial set of assets and their dependencies to other assets. Basic patterns, which occur independently from the particular context of the PSE project, can represent such a starting point. From domain analysis and discussions with engineers, we identify basic patterns that regularly occur in PSE. These patterns provide blueprints to help engineers identify reusable assets in I4AN instances.

Each research question addresses parts of the overarching question (cf. Section I) tying together model elements, their dependencies, and patterns for the efficient reuse of CPPS knowledge. We evaluate the I4AN in a feasibility study for the use case “*Car Body with Screwed-on Parts*” (cf. Section IV). Therefore, we use data from a sample of artifacts from the domain analysis. We investigate to what extent advanced reuse scenario questions can be answered by queries to the I4AN.

IV. ILLUSTRATIVE USE CASE

This section introduces the use case “*Car Body with Screwed-on Parts*”. We condensed the use case from a domain analysis in the automotive manufacturing domain. The analysis was conducted in a setting with 80 types of screwing robot cells and 27 robot types.

In automotive manufacturing, human workers collaborate with industrial robots in mounting lines to place and screw various components onto a car body using screwdrivers. Typical mid-class cars contain screws of 80 screw types at 1,500 to 1,800 screw positions. Figure 1 shows the use case with its I4.0 assets and their connections. The left-hand side shows a screwing process consisting of two steps: (i) *positioning* the dashboard and the screws and (ii) *fastening* the screws. Both steps are characterized by process requirements, defining the necessary skills of the resources including technical or economic parameters. In PSE, relevant resources, i.e., resource hierarchies, (see right-hand side of Figure 1) are selected and orchestrated to provide the required skills [17], [19].

In theory, one can engineer an optimized robot-screwdriver combination for each screw type to maximize production effectiveness and efficiency. Yet, this approach might lead to around 80 different robot and screwdriver types, adding significant costs for installation, maintenance, and expert knowledge.

In practice, PSE aims at cost-optimized system designs [32]. Hence, a sufficiently effective and efficient robot-screwdriver combination to each screw type can be assigned, minimizing

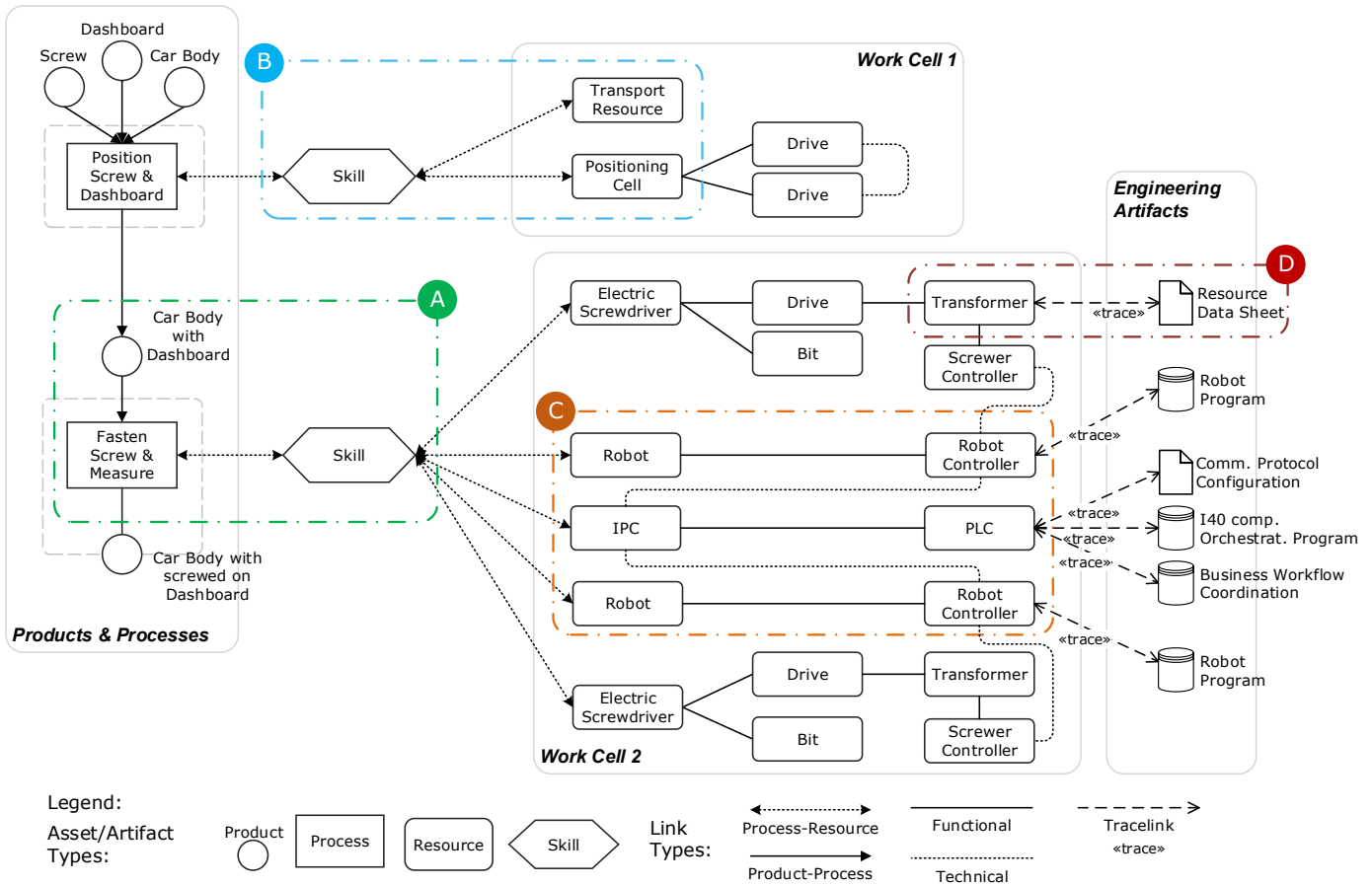


Fig. 1. Reuse Patterns (dashed boxes) in an asset network for the use case “Car Body with Screwed-on Parts” (in adapted VDI 3682 notation [10], [29]).

the number of robot types and investments in spare parts and know-how. This approach may significantly reduce costs in comparison to a high-variety approach.

Identifying an optimized set of robot-screwdriver combinations for the high number of different screws types and positions requires the identification of (i) *resources* that can execute several varying screwing tasks and (ii) *engineering artifacts* that can be reused, such as control programs. However, identifying suitable solutions is difficult due to the scattered information and the engineers’ implicit knowledge.

Achieving these advantages requires a set of reusable patterns. This set of patterns can be completed by (i) *identifying similar components* within existing engineering projects, (ii) *mapping* these components to expected future requirements, and (iii) *abstracting* these components with respect to possible adaptations for application-case related parameterization [8].

Requirements for I40 Asset Reuse. From the domain analysis, we elicited the following *requirements Rx* towards asset reuse in PSE with eight domain experts from five medium-to-large European PSE companies⁴.

R1. I40 Asset Map. Domain experts require an *Industry*

4.0 Asset Map, i.e., an overview on the assets in the planning phase to explicitly represent implicit knowledge and relevant information as a context for reuse, currently scattered across various engineering artifacts. This requirement is adapted from software engineering, i.e., documenting the project structure and software artifacts, to multi-disciplinary PSE assets in the *Product-Process-Resource (PPR)* scope.

R2. I40 Dependency Network. As a basis to identify patterns for reuse in PSE, domain experts require an explicit representation of the links and dependencies of and between assets coming from several engineering disciplines. This concerns mainly three different views. Product engineering requires links between product components, processes, and their required skills that a process requires from a resource to automate the process. Systems engineering concerns relationships between resources and their provided capabilities. The assets and dependencies can be represented in an *Industry 4.0 Dependency Network* that adds information and knowledge required for reuse to the *Industry 4.0 Asset Map*.

R3. System Boundary. For reuse, a system or subsystem containing the reusable assets needs to have a clearly defined boundary. System boundaries are a means to group assets into a meaningful set of assets that can be reused. A boundary also allows to investigate incoming and outgoing dependencies.

⁴The experts rated their company’s maturity level of asset reuse, using the VDI 3695 classification, at reuse levels RL-C (controlled from a central point) or RL-E (reuse based on internal and external standards).

Thus, system boundaries serve as a basis for systematically reusing (parts of) a solution that was used in previous projects. Without a clear boundary, it is unclear which elements can, should, or have to be included in a set of reusable assets. Furthermore, system boundaries enable developing and using metrics, like complexity, to compare patterns.

R4. Solution Design Abstraction. As a foundation to identify reusable patterns, domain experts need a representation of solution design candidates at a suitable level of abstraction. This abstraction is required to allow the adaptability and portability of a pattern to similar problems with varying characteristics. For example, to make a solution for a *position* task reusable requires hiding unnecessary attributes and dependencies. In the use case, the robot positioning accuracy is a relevant characteristic, while the way how the robot moves might be irrelevant. *Solution Design Abstraction* facilitates (i) generalizing from a particular solution instance to a more general level of problems and (ii) finding reusable solution candidates in similar or historic designs.

The following section builds on this use case to illustrate a novel knowledge representation model for Industry 4.0 Assets for identifying patterns for reuse.

V. PATTERNS FOR REUSE IN PSE

This section presents the Industry 4.0 Asset Network and four basic patterns to identify concrete patterns for reuse.

A. I40 Asset based Network with Dependencies

To address *RQ1a* and *RQ1b*, we investigated the data of robot cells with up to two robots from the use case context with domain experts. From this data, we determined knowledge elements that we can use for identifying abstract patterns for reuse. These elements were used to build a condensed metamodel as the foundation for the I4AN. This section illustrates the metamodel and the I4AN using the *car body with screwed-on parts* use case from Section IV.

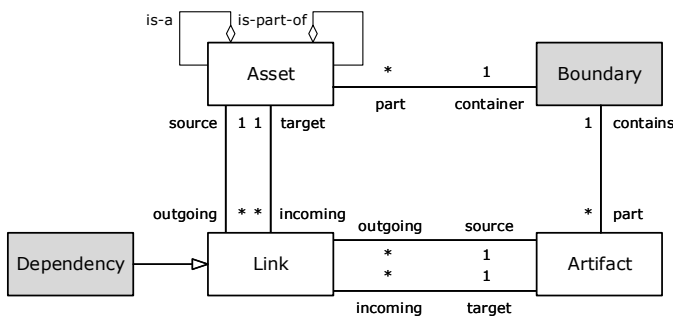


Fig. 2. *Asset*, (Engineering) *Artifact* and *Link* meta model, based on [33]

Figure 2 shows the metamodel (in UML notation) containing the *Asset* class, one of the Industry 4.0 Asset types *product*, *process*, *resource*, or *skill*. An *Asset* can be a specialization (*is-a* relation, e.g., an electric screwdriver is a type of screwdriver) and/or a part (*is-part-of* relation, e.g., a bit is part of a screwdriver) of another asset. An *Artifact* is an engineering object created during design time, e.g., an electrical plan or

robot program, or during runtime, e.g., a set of qualitative data. A particular *Link* can connect assets with each other or to artifacts. *Links* can have different forms (cf. Figure 1) realized using typed properties (not shown in the meta-model): *Functional links* between production resources may represent a resource composition. *Technical links* may represent a wired connection from an *Industrial PC (IPC)* to a robot. To model a connection between an *Asset* and an *Artifact*, we use *Trace Links*, e.g., a robot controller requires a robot program. A *Link* can be manifested as *Dependency*, if the link is strictly required by an *Asset*. *Assets*, *Artifacts*, and *Links* can have attributes that describe characteristics of the particular object. These properties follow the I4.0 Asset Administration Shell (AAS) [3] design to facilitate the standardized representation of property views coming from several engineering disciplines. A *Boundary* object represents a pattern boundary that contains *Assets* and *Artifacts*, e.g., boundary (*A*).

These concepts provide the foundation to build an I4AN that explicitly represents PSE information and knowledge for a wide range of applications, such as change impact analysis. Figure 1 illustrates an I4AN with the relevant engineering artifacts and the links between the assets. This model can be created automatically by exploiting appropriate engineering data logistic systems [12]. This overall model can be the starting point to identify common reusable patterns [19].

B. Patterns for I40 Asset Reuse

This section describes four basic patterns for identifying best-practice candidates for reuse in their context. These identification patterns can be used as a starting point to identify patterns in the particular PSE contexts of domain experts.

The reuse of assets requires considering the asset itself and, beyond that, its embedding in the surrounding system and functional intentions [6], [9], [28]. As described, PSE comprises two main phases, rough and detail planning.

The rough planning phase consists of matching process skills required by products and provided by resources. This comparison shall be based on *product creation* (P1) and *process execution* (P2) patterns.

P1. Product-Process-Skill Pattern. Product creation in PSE aims at providing the combination of products with their requirements and processes to manufacture them. **Aim:** The *Product-Process-Skill* pattern (cf. Figure 1, tag A) supports product engineers in selecting appropriate processes for their products. This product creation pattern contains production processes with their input and output materials, boundary conditions, and required skills. **Solution:** The pattern can be identified by collecting all assets connected to the related processes by product-process-related links: For an output product isolate the input products and determine their relevant properties. For each input product determine the required process steps and build the aggregated required skills of the steps according to [10]. Group the products, process steps, and skills into a boundary object. For the outgoing and incoming links, determine whether they are strict dependencies. For dependencies, decide if you need to either expand the boundary

or create a depending pattern object. **Example:** An example are screw-screwing combinations. We identified different reuse patterns from equivalence classes based on the screwing bit, the applicable torque, and the screw material (magnetic vs. non-magnetic) with industry partners.

P2. Skill-Resource Pattern. Process execution in PSE is to identify resources able to execute a production process based on their functional skills. **Aim:** The *Skill-Resource* pattern (cf. Figure 1, tag *B*) helps to select appropriate resources matching to the *Product-Process-Skill* pattern. This process execution pattern contains resources with their properties, boundary conditions, and provided skills. **Solution:** The pattern can be identified by collecting all assets connected to the related resource links: For a set of connected resources, determine their provided skills and properties. From the skills, build the aggregated provided resources skills according to [10]. Group the resources and skills into a boundary object and determine the dependencies. For dependencies, decide if you need to either expand the boundary or create a depending pattern. **Example:** The pattern supports the definition of skills, e.g., positioning, with predefined attributes, like positioning accuracy, which are fulfilled by a set of resources.

The main concern within the detail planning phase is realizing production resources providing all necessary functionalities to fulfill the required skills. Here, patterns related to resource structuring and functionality are relevant. Thus links shall be considered depending on the use case.

P3. Resource-Resource Composition Pattern. The goal of detailed engineering in PSE is detailing and programming the selected resources. **Aim:** The *Resource-Resource composition* pattern (cf. Figure 1, tag *C*) represents the composition of a resource from sub-components, with the knowledge on technical parameters and dependencies on the type and instance levels. A quality ensured resource tree pattern could be applied at this point, reflecting the optimized orchestration of resources. **Solution:** For a group of connected resources (*part-of* relation) determine which resources are required to either fulfill a particular skill or if they require each other for functionality. Group the strictly required resources into a boundary. For dependencies, decide if you need to either expand the boundary or create a depending pattern object. **Example:** Screwdrivers can be driven, e.g., electrically or pneumatically. Depending on the drive, the screwdriver requires a transformer for the current or not, which can be expressed in an *RR* pattern.

P4. Resource-Artifact Pattern. Within the commissioning phase of PSE the detailed resource system is established according to the relevant engineering artifacts, e.g., relevant for operation. **Aim:** The *Resource-Artifact* pattern (cf. Figure 1, tag *D*) aims at binding the required engineering artifacts to the resources used in the production system. This helps engineers to reuse resources and their corresponding data or programs as a bundle. **Solution:** From a resource, follow the trace links to the engineering artifacts. For the resource and the necessary engineering artifacts, use a boundary object to group them. For incoming or outgoing dependencies from resources or engineering artifacts, decide whether to expand the boundary

or create a depending pattern object. **Example:** Screwdrivers have a minimum, maximum, and yield torque for a screwing process. The screwdrivers and function blocks controlling the torque of the screwdrivers can be expressed as a pattern and reused in future projects.

The use case *Car Body with screwed-on parts* can benefit from reuse patterns in (at least) four ways: (i) The *product-process-skill* pattern can support product engineers in selecting appropriate screwing processes for their car body parts (see tag *A* in Figure 1). (ii) The *skill-resource pattern* facilitates selecting appropriate screwdrivers to screwing processes (see tag *B* in Figure 1). (iii) The *resource-resource composition* pattern can be applied for the optimized combination of screwing resources, e.g., robots and robot controllers (see tag *C* in Figure 1). (iv) The *resource-artifact* pattern can be applied for reusing engineering artifacts, e.g., robot controllers and robot control programs (see tag *D* in Fig. 1).

VI. FEASIBILITY STUDY AND DISCUSSION

This section presents a preliminary feasibility study and discusses the contributions with a focus on the research questions raised in Section III.

A. Preliminary Feasibility Study

As a proof of concept, we used a part of the production system for the investigated use case “*Car Body with Screwed-on Parts*” from the initial domain analysis to design and instantiate the *Industry 4.0 Asset Network (I4AN)* in a *Neo4J*⁵ graph database. The I4AN was found easy to extract from existing engineering information, which has to be integrated according to the the I4.0 AAS design [3].

The graph database facilitated the effective and efficient exploration, querying, and visualization of the linked assets. In addition to the *technical links* between assets coming from engineering models, we instantiated *dependency links* between the assets. Deep domain expert knowledge has to be added to the I4AN manually. The concepts in the I4AN facilitated adding previously implicit domain knowledge to the graph.

The I4AN instance associated to Figure 1 enables identifying I4.0 Assets that belong to a pattern for I4.0 Asset reuse (cf. Section VI-B). To investigate the functionality, we issued queries onto the I4AN to track the dependencies. We used iterative queries, similar to cause-effect graph exploration [33], starting at a selected I4.0 Asset, such as a skill, and followed the multi-model links to neighboring assets of a specified type until reaching a stopping condition. We were able to efficiently isolate parts of the I4AN that correspond to the basic patterns introduced in Section VI-B. This approach also worked for the reuse scenario *system boundary analysis* that can be translated into the question: *Which set of dependency links connects a selected set of assets to their immediate neighboring assets?* This capability indicates that engineers can utilize the I4AN to investigate the network to identify familiar patterns of assets as candidates for reuse.

⁵Graph database *Neo4J*: <https://neo4j.com>

B. Discussion

We conducted a domain analysis with 80 types of robot cells and 27 robot types. Further, we elicited requirements from domain experts at five European PSE companies. The requirements showed that a key aspect is modeling the multi-disciplinary dependencies between assets and engineering artifacts that need to be considered to identify reusable asset patterns. It is also essential to thoroughly model the boundaries of the patterns to allow suitable reuse in practice among the involved engineering disciplines.

RQ1a and **RQ1b** concerned models and dependencies that facilitate the identification of assets suitable for reuse. To address RQ1a and RQ1b, Section V-A introduced the *Industry 4.0 Asset Network (I4AN)* that addresses requirements *R1* to *R3* identified in Section IV. The I4AN builds on I4.0 assets and uses their administration shell to integrate property views from several engineering disciplines. In comparison to patterns in software engineering, this multi-disciplinary aspect adds complexity to identifying patterns for reuse in PSE.

We go beyond the state of the art [4], [18] by modeling *skills* as I4.0 assets using their digital representation for linking multi-disciplinary assets and identifying boundaries for reusable assets. We build on and go beyond [25] by integrating multi-disciplinary multi-model links between I4.0 Assets.

RQ2 asked which basic patterns for reuse facilitate the identification of patterns for reuse. To address RQ2, Section IV identified four basic patterns addressing requirement *R4* (cf. Section IV). These patterns specifically incorporate regularly occurring connected assets in PSE that can be reused for similar problems. Therefore, they provide guidance for reuse design and management with the I4AN. In this sense, the I4AN provides designers with the capability to describe partial solutions and integrate partial solutions into a complete solution from production processes to automation devices that automate the production process.

Limitations. The following limitations require further investigation. The research in this paper focused on the reuse of production processes and associated automation system elements in a typical use case of automotive manufacturing, the *Car Body with Screwed-on Parts* use case. As we assume the findings of this paper to be relevant in the broader scope of production processes and automation system elements, e.g., for discrete production and continuous production, the approach should be investigated in a broader range of application areas.

The *domain analysis* was conducted by one of the paper authors with consultation from domain experts and checked for plausibility by the author team. While the *feasibility study* focused on a I4AN for a robot cell of typical complexity, the authors of this paper, consulting with domain experts in car manufacturing, conducted the design of the I4AN including dependencies that are missing in traditional PSE design. This reflects the current practice of PSE engineering only partially and introduced bias to the study, requiring validation in a range of traditional and advanced PSE environments.

VII. CONCLUSION AND FUTURE WORK

The Industry 4.0 (I4.0) vision of production systems that are easy to adapt depends on advanced capabilities for reusing proven production processes, I4.0 assets and software-intensive components that automate these production processes. In Production Systems Engineering (PSE), the reuse of I4.0 assets requires understanding the dependencies of these assets in multi-disciplinary systems-of-systems engineering with heterogeneous models.

This paper investigated the information requirements for advanced multi-disciplinary reuse scenarios, such as *process and resource identification* and for *system boundary analysis*. To address the challenges of scattered and implicit domain expert knowledge that may lead to overlooking risky dependencies of reusable system elements, we introduced the *Industry 4.0 Asset Network (I4AN)*. The I4AN builds on the I4.0 Asset Administration Shell [3] design to integrate system element properties and dependencies from several engineering disciplines, such as mechanical, electrical, and software interfaces and technical links.

Therefore, the I4AN enables designing a knowledge graph that represents for a reuse scenario important multi-disciplinary dependencies between system elements as neighborhoods of I4.0 Assets. Further, the I4AN concepts facilitate representing domain expert knowledge that was implicit, e.g., to recommend using a resource type with a process type.

We presented the use case “*Car Body with Screwed-on Parts*” to illustrate typical I4.0 Assets and links in production processes and robot cells widely used in car manufacturing. In the I4AN of the use case (cf. Figure 1), we identified four types of patterns for reuse.

In a feasibility study, we evaluated the I4AN with reuse scenarios by instantiating an I4AN knowledge graph formulating scenario concepts and questions as data in and queries to the knowledge graph. The study results indicate that the I4AN model is a good foundation for PSE domain experts to identify patterns for reuse in their contexts.

The research results advance the state of the art in knowledge engineering in PSE by modeling the *Skill* concept as an I4.0 Asset. The I4AN provides a lens for analyzing similarities and differences in production process and system designs. To this end, we are providing the foundations for advanced reuse design and management with the I4AN and patterns.

The research results advance the state of the art by adapting blueprints for design pattern to a multi-disciplinary engineering environment where multi-model links are crucial. The I4AN provides designers with the capability to describe partial solutions and integrate these partial solutions into a complete solution, from production process to automation devices that automate the production process. The I4AN facilitates identifying risky external systems dependencies across several engineering disciplines as input to assess the reuse effort and risk of candidate solution designs.

Future Work. *Validation of patterns for reuse.* We plan to investigate I4AN applications for reuse to improve PSE tools, e.g., with knowledge on multi-model dependencies.

Scalability. We see the need to investigate the scalability of the I4AN in a larger context and with additional engineering disciplines to evaluate the impact on the multi-disciplinary dependencies and boundaries beyond the scale of work cells.

Skills. We consider examining the extended use of *skills* as an advanced method to abstract from process requirements to resource capabilities and their role in reusable process and resource assets, e.g., using standardized catalog search.

Extension of the I4AN with Semantic Web content. For the PSE domain, the I4AN seems well represented in a graph database as this technology is increasingly well accepted in PSE, while Semantic Web technology is mainly used in research. We envision extending the I4AN with knowledge organized with Semantic Web technologies, e.g., issues, recommendations as natural text. The I4AN knowledge graph can collect knowledge instances that can be converted efficiently to Semantic Web technologies to facilitate research on industrial data for Semantic Web researchers.

Security. Aggregating domain knowledge in an I4AN creates a high-value knowledge graph. This graph requires research on security concerns, e.g., theft of intellectual property or using it to plan attacks on systems that represent critical infrastructure.

ACKNOWLEDGMENT

The financial support by Christian Doppler Research Association, Austrian Federal Ministry for Digital and Economic Affairs, and National Foundation for Research, Technology and Development is gratefully acknowledged. The competence center CDP is funded within the framework of COMET – Competence Centers for Excellent Technologies by BMVIT, BMDW, and the federal state of Vienna, managed by the FFG.

REFERENCES

- [1] B. Vogel-Heuser, T. Bauernhansl, and M. Ten Hompel, "Handbuch Industrie 4.0 Bd. 4," *Allgemeine Grundlagen*, vol. 2, 2020.
- [2] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn, and K. Ueda, "Cyber-physical systems in manufacturing," *CIRP Annals*, vol. 65, no. 2, pp. 621 – 641, 2016.
- [3] Plattform Industrie 4.0 and ZVEI, "Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01 Review)," German BMWI, Standard. Nov. 2020, <https://bit.ly/37A002L>.
- [4] J. Pfrommer, D. Stogl, K. Aleksandrov, V. Schubert, and B. Hein, "Modelling and orchestration of service-based manufacturing systems via skills," in *19th IEEE ETFA*, 2014, pp. 1–4.
- [5] A. Lüder, N. Schmidt, K. Hell, H. Röpke, and J. Zawisza, *Identification of Artifacts in Life Cycle Phases of CPPS*. Cham: Springer International Publishing, 2017, pp. 139–167.
- [6] H. Johannesson, *Emphasizing Reuse of Generic Assets Through Integrated Product and Production System Development Platforms*. New York, NY: Springer New York, 2014, pp. 119–146.
- [7] A. Lüder, N. Schmidt, K. Hell, H. Röpke, and J. Zawisza, *Fundamentals of Artifact Reuse in CPPS*. Cham: Springer International Publishing, 2017, pp. 113–138.
- [8] N. Jazdi, C. R. Maga, P. Göhner, T. Ehben, T. Tetzner, and U. Löwen, "Improved Systematisation in Plant Engineering and Industrial Solutions Business - Increased Efficiency through Domain Engineering," *Automatisierungstechnik*, vol. 58, no. 9, pp. 524–532, 2010.
- [9] F. Stallinger, R. Neumann, R. Plosch, P. Hehenberger, B. Bohm, A. Kohlein, and N. Gewald, "Improving mechatronical engineering: An artifact-assessment-based approach," in *16th IEEE ETFA*. IEEE, 2016, pp. 813–820.
- [10] K. Meixner, A. Lüder, J. Herzog, H. Röpke, and S. Biffl, "Modeling expert knowledge for optimal CPPS resource selection for a product portfolio," in *25th IEEE ETFA*. IEEE, 2020, pp. 1687–1694.
- [11] A. Lüder, N. Schmidt, K. Hell, H. Röpke, and J. Zawisza, *Description means for information artifacts throughout the life cycle of CPPS*. Cham: Springer International Publishing, 2017, pp. 169–183.
- [12] A. Lüder, A. Behnert, F. Rinker, and S. Biffl, "Generating industry 4.0 asset administration shells with data from engineering data logistics," in *25th IEEE ETFA*. IEEE, 2020, pp. 867–874.
- [13] A. Strahilov and H. Hämmerle, "Engineering Workflow and Software Tool Chains of Automated Production Systems," in *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*. Springer, 2017.
- [14] K. Paetzold, *Product and Systems Engineering/CA* Tool Chains*. Cham: Springer International Publishing, 2017, pp. 27–62.
- [15] R. Drath, M. Barth, and A. Fay, "Offenheitsmetrik für engineering-werkzeuge," *atp edition*, vol. 54, no. 09, pp. 46–55, 2012.
- [16] A. Lüder, N. Schmidt, and R. Drath, *Standardized Information Exchange Within Production System Engineering*. Cham: Springer International Publishing, 2017, pp. 235–257.
- [17] J. Pfrommer, M. Schleipen, and J. Beyerer, "PPRS: Production skills and their relation to product, process, and resource," in *18th IEEE ETFA*. IEEE, 2013, pp. 1–4.
- [18] G. Cândido and J. Barata, "A multiagent control system for shop floor assembly," in *Holonic and Multi-Agent Systems for Manufacturing*, V. Mařík, V. Vyatkin, and A. W. Colombo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 293–302.
- [19] J. Herzog, H. Röpke, and A. Lüder, "Allocation of PPRS for the plant planning in the final automotive assembly," in *ETFA*. IEEE, 2020, pp. 813–820.
- [20] M. Sabou, F. J. Ekaputra, T. Ionescu, J. Musil, D. Schall, K. Haller, A. Friedl, and S. Biffl, "Exploring enterprise knowledge graphs: A use case in software engineering," in *European Semantic Web Conference*. Springer, 2018, pp. 560–575.
- [21] G. Booch, *The Unified Modeling Language Use Guide*. Addison-Wesley Object Technology, 2017.
- [22] D. F. Eugenio Brusa, Ambra Calà, Ed., *Systems Engineering and Its Application to Industrial Product Development*. Springer, 2018.
- [23] C. Legat, C. Seitz, S. Lamparter, and S. Feldmann, "Semantics to the shop floor: Towards ontology modularization and reuse in the automation domain," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3444–3449, 2014, 19th IFAC World Congress.
- [24] U. Lindemann, M. Maurer, and T. Braun, Eds., *Structural complexity management: An approach for the field of product design*. Springer, 2009.
- [25] S. Feldmann, M. Wimmer, K. Kernschmidt, and B. Vogel-Heuser, "A comprehensive approach for managing inter-model inconsistencies in automated production systems engineering," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 1120–1127.
- [26] S. Fischer, L. Linsbauer, R. E. Lopez-Herrejon, and A. Egyed, "Enhancing Clone-and-Own with Systematic Reuse for Developing Software Variants," in *2014 IEEE International Conference on Software Maintenance and Evolution*, sep 2014, pp. 391–400.
- [27] *VDI Guideline 2206: Design methodology for mechatronic systems*, VDI-Verlag, Düsseldorf Std., 2004.
- [28] A. Lüder, L. Hundt, M. Foehr, T. Holm, T. Wagner, and J. Zaddach, "Manufacturing system engineering with mechatronical units," in *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, 2010, pp. 1–8.
- [29] *VDI Guideline 3695: Engineering of industrial plants - Evaluation and optimization*, Beuth Verlag Std., 2009.
- [30] E. Gamma, R. Helm, R. E. Johnson, and J. Vlissides, *Design patterns: elements of reusable object-oriented software*. Pearson Education India, 1995.
- [31] R. J. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014.
- [32] R. Mehr and A. Lüder, *Managing Complexity Within the Engineering of Product and Production Systems*. Cham: Springer International Publishing, 2019, pp. 57–79.
- [33] S. Biffl, A. Lüder, K. Meixner, F. Rinker, M. Eckhart, and D. Winkler, "Multi-view-model risk assessment in cyber-physical production systems engineering," in *MODELWARD*. SCITEPRESS, 2021, pp. 163–170.