

Product-oriented Product–Process–Resource Asset Network and its Representation in AutomationML for Asset Administration Shell

Sára Strakošová^{*‡}, Petr Novák[‡], Petr Kadera[‡]

^{*}*Faculty of Mechanical Engineering, Czech Technical University in Prague, Prague, Czech Republic*
sara.strakosova@fs.cvut.cz

[‡]*Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Prague, Czech Republic*
petr.novak@cvut.cz, petr.kadera@cvut.cz

Abstract—Current products, especially in the automotive sector, pose complex technical systems having a multi-disciplinary mechatronic nature. Industrial standards supporting system engineering and production typically (i) address the production phase only, but do not cover the complete product life cycle, and (ii) focus on production processes and resources rather than the products themselves. The presented approach is motivated by incorporating the impacts of the end-of-life phase of the product life cycle into the engineering phase. This paper proposes a modeling approach coming up from the Product-Process-Resource (PPR) modeling paradigm. It combines requirements on (i) respecting the product structure as a basis for the model, and (ii) incorporates repairing, remanufacturing, or upcycling within cyber-physical production systems. The proposed model called PoPAN should accompany the product during the entire life cycle as a digital shadow encapsulated within the Asset Administration Shell of a product. To facilitate the adoption of the proposed paradigm, the paper also proposes serialization of the model in the AutomationML data format. The model is demonstrated on a use-case for disassembling electric vehicle batteries to support their remanufacturing for stationary battery applications.

Index Terms—Product–Process–Resource, AutomationML, Asset Administration Shell, Mechatronic Systems, Life-cycle management

I. INTRODUCTION

The significant environmental impact of manufacturing industries underscores the importance of addressing sustainability in the context of recycling and remanufacturing of products [1]. Although the term “sustainability” is increasingly used in marketing, leading to skepticism and negative perceptions. Distinguishing genuine sustainability efforts from mere marketing strategies is essential. In this paper, we introduce a formalization to model products and production processes, which has the potential to contribute to sustainability specifically at the end of a product life-cycle.

This paper introduces the Product-oriented Product–Process–Resource Asset Network (PoPAN) formalization, which builds upon the Product–Process–Resource (PPR) [2] and the Product–Process–Resource Asset Network (PAN) [3] approaches. Current modeling approaches (including the original PAN) are frequently focused on production processes. This perspective is substantiated by numerous industrial

standards such as ISA-95 or IEC 62264, Asset Administration Shell, and VDI/VDE 3682. On the contrary, the PoPAN spotlights the product structure as a basis and it assigns production processes/operations to product components.

PoPAN offers a structured description of a product, incorporating its components (represented as products), processes, and resources. This approach enables the design of recycling and remanufacturing processes for products that require disassembly for efficient recycling. Furthermore, PoPAN can function as a digital shadow [4], accompanying the product throughout its entire life-cycle and encapsulating all relevant information. It serves as a record of the product’s evolution, including any modifications or alterations it undergoes during its journey, such as missing components like screws. An additional advantage of employing PoPAN for recycling lies in the data collection aspect. Sustainability initiatives begin at the product design phase, where prioritizing recyclability is paramount. By gathering relevant data from recycling processes, manufacturers can analyze and utilize this information to enhance the recyclability of future product designs. This iterative process leads to more sustainable designs and facilitates easier recycling, contributing to overall environmental sustainability efforts.

To facilitate the adoption of this paradigm, the paper also proposes the serialization of the model in the AutomationML [5] data format. This standardized format enhances interoperability and ease of integration within existing industrial frameworks and systems. An important benefit of AutomationML is the advancing integration with the Asset Administration Shell (AAS) [6]. AAS is a set of standards and recommendations to provide interoperability in industrial systems. AAS is typically used on the resource level (e.g., a motor or a robot). On the contrary, the AAS is used for the product level in the presented approach utilizing PoPAN. Moreover, AAS is used for the entire product life-cycle in this approach, including not only the manufacturing phase but also operations, maintenance, and repair, as well as decommissioning/recycling. The paper [7] highlights the use of AAS in the context of product life cycle management.

This paper addresses the following research questions:

- *RQ1*: How can we adapt the PAN model to primarily reflect the product structure rather than the production process with resource structure?
- *RQ2*: Is it possible to combine production and remanufacturing/disassembling processes into a proposed PPR-based description and if yes, how to do so?
- *RQ3*: To improve the adoption of the proposed approach by industry and academia, how can be the proposed model represented in the AutomationML data format?

The proposed modeling approach is demonstrated in a case study on disassembling electric vehicle (EV) batteries for recycling and remanufacturing [8].

II. STATE OF THE ART

The proposed approach is built on the top of 3 main domains including (i) Asset Administration Shell, (ii) AutomationML data format, and (iii) Model-Driven Engineering and Product-Process-Resource Asset Network. These research areas are described in detail in the following subsections.

A. Asset Administration Shell

One of the key standards in the Industry 4.0 initiative is the Asset Administration Shell (AAS) [9], [10]. AAS was introduced by Platform Industry 4.0 as a promising approach to enable interoperability in various industries. It facilitates data sharing between value chain partners, standardizes data security, and establishes technology-neutral semantic standards, making it crucial to achieving Industry 4.0 goals. The AAS enables diverse communication channels and applications, serving as a bridge between physical objects and the digital world [6].

The AAS is a digital representation of an asset and comprises multiple submodels that represent a specific aspect or set of characteristics of an asset. Within a submodel, various submodel elements are defined. Submodel elements are the individual components or entities that make up the submodel. They represent specific properties, operations, parameters, references, files, or other relevant information associated with the asset. The purpose of submodels is to provide a structured approach to organizing and managing the elements within an AAS. By grouping related elements, it becomes easier to define and enforce standards for specific sets of information.

B. AutomationML

The engineering phase of multi-disciplinary mechatronic systems requires data exchange across software tools belonging to diverse engineering domains (including electric planning, mechanic design, maintenance, etc.). Engineering data can be efficiently captured and exchanged in the data format AutomationML, which is a standardized extension of XML for the interdisciplinary exchange of planning data for production systems and processes. The data format AutomationML is an open, platform-independent standard known as IEC 62714. Detailed information about this standard including various best practices and application recommendations, and how to model

specific information, can be found on the Web pages of the AutomationML association¹.

AutomationML combines and harmonizes already existing XML formats:

- CAEX (IEC 62424) – Topology a hierarchy of objects, properties, and relationships among objects
- COLLADA (ISO/PAS 17506) – Geometry a kinematics of objects
- PLCopen XML (IEC 62714) – Discrete behavior of objects

The basis of the object model in AutomationML is CAEX, which is intended for capturing engineering information in object-oriented form. Each object can have attributes and references to other objects (i.e., internal links or mapping objects, and references via object identifiers). It can also include links to other information, represented in COLLADA or PLCopen XML. In CAEX, a model is represented via the following four basic cornerstones. Role Class Libraries are intended to define object semantics. Each object can have more than one role. Interface Class Libraries provide specifications of object interfaces. Classes of objects are modeled as objects of System Unit Class Libraries. Each system unit class poses a generic object prototype, which can be instantiated when describing a specific system. For better interoperability, semantic mapping to terminology defined as ECLASS based on IEC 61360 can be used in a standardized way. The system description itself is represented in AutomationML/CAEX by instance hierarchy. In the tree structure behind the instance hierarchy root element, Internal Elements, and their sub-elements are defined. Internal elements in most cases reference their prototypes in the frame of the System Unit Class, supported roles from Role Class Libraries, and interfaces from the interface class libraries. Topological relationships are represented as Internal Links.

AutomationML is one of the most complex data formats for system engineering, which is suitable for a wide range of tasks, including the representation of the Asset Administration Shell [11], [12]. Such an AutomationML representation can be imported to the AASX format [9] in the tool AASX Package Explorer². The application of AutomationML in the end-of-life phase is addressed in [13], [14], which are focused on production systems and resources, whereas this paper is mainly focused on products together with production processes.

C. Model-Driven Engineering and Product-Process-Resource Asset Network

Model-driven engineering (MDE) highlights the role of models as core artifacts for engineering and integration development. Similarly, like object-orientation in object-oriented programming, where everything is represented by objects (with specific attributes and methods), in model-driven engineering, everything is a model [15]. Initial efforts in MDE started with the use of the Meta Object Facility and the Unified Modeling Language (UML) [16]. MDE is highly efficient for

¹Online: <https://www.automationml.org/>

²Online: <https://github.com/eclipse-aaspe>

translating information between various representations (i.e., domain models or data formats) [17], [18]; and it provides suitable support for multi-disciplinary engineering projects [19]. One of the plausible models to support the model-driven engineering is a Product-Process-Resource model [20], categorizing artifacts relevant for industrial production into these three categories including mappings or links among these categories.

The Product-Process-Resource Asset Network (PAN) [3] is a graph-based formalism for expressing relationships among products, production processes, and resources. In the domain of Cyber-Physical Production Systems (CPPSs), dependencies among products, processes, and resources are frequently left implicit. The PAN addresses this gap by providing a structured framework for explicitly articulating and overseeing the dependencies among product, process, and resource assets. The PAN modeling is suitable for capturing and sharing knowledge among engineers and respective engineering tools and for identifying changes during engineering processes of multi-disciplinary system engineering projects. PAN usage optimizes engineering processes' efficiency and enhances operations by modeling manufacturing processes, tracking resource usage, and monitoring asset performance. Through data integration, visualization, and analysis, PAN enables informed decision-making [21].

III. PRODUCT-ORIENTED PRODUCT-PROCESS-RESOURCE ASSET NETWORK (PoPAN)

Recycling and remanufacturing products pose challenges due to limited engineering data availability. While manufacturing processes are well-specified and documented, the absence of detailed data for a product in its end-of-life phase complicates recycling and remanufacturing efforts. Leveraging the PAN approach, which has proven efficiency in product manufacturing, presents a promising solution. However, traditional PAN methods describe one specific assembly process, which may not capture the entire variability of assembly and disassembly operations. To address this limitation, this paper proposes extending the PAN paradigm to describe products structurally. By doing so, multiple disassembly processes can be generated from a single description, accommodating the diverse disassembly options frequently encountered in practice. This structural representation enhances flexibility and efficiency in creating recycling and remanufacturing processes.

To create a product description that offers a comprehensive understanding of the product's structure, including the specific processes necessary for both assembly and disassembly, along with the resources required for each process, the initial step is to construct a Product Structure graph depicting dependencies between product's components (see Fig. 1 on the left). Since the PPR approach is used, to preserve the name of elements, the product's components are depicted also as products. Within this framework, three types of products are distinguished (1) Elementary Product: a product that cannot be further disassembled, (2) SubProduct: a product that is considered elementary in terms of assembly/disassembly procedures, but

is composed of several parts and can be further disassembled, and (3) Fastener Product: a product that serves as a fastener for other products (e.g., screws). The graph delineates two types of edges connecting individual products, forming the overall structure. The solid line edge illustrates physical continuity between products and the dashed line edge, on the other hand, forms a structural continuity between the products, i.e., dependencies that indicate certain products cannot be assembled or disassembled until others are. Moreover, both types of edges end with arrows on both ends representing the assembly/disassembly direction. Open arrows represent assembly direction, full arrows represent disassembly direction, and *Initial Product* and *Last Product* are highlighted for better orientation in the graph. For example, when disassembling *Product 3*, it is essential to consider all three structural dependencies. Considering the orientation by the full arrows, *Product 1* and *Product 2* are not blocking *Product 3* from being disassembled. However, the *Last Product* is blocking the disassembly of *Product 3*, meaning that the *Last Product* must be disassembled before *Product 3*. Additionally, frequently used terms *final product* and *semi-product* in the sense of a classic PAN are highlighted to show intermediate steps in an assembly operation. However, the PoPAN approach diverges from relying on these terms.

Expanding the Product Structure graph with the PAN/PPR approach involves incorporating processes and resources alongside products. This structured visualization facilitates understanding the relationships between products and corresponding processes, which are supported by the required resources. As depicted in Fig. 1 on the right, in the extended Product Structure graph called Product-oriented Product-Process-Resource Asset Network (PoPAN), each product is connected with a corresponding process by a simple line edge or by a simple line edge with a question mark above it. The question mark indicates conditional linking between the product and process, where the process sequence might not correspond with the structure of a product, which is then resolved by a red dotted line edge determining the sequence of the processes. Lastly, the graph contains green circle marks with a letter which depict edges connecting processes with their required resources. As with the previous Product Structure graph, the arrows form the sense of orientation in the graph and the connections between products, processes, and resources form the edges.

To find the correct sequence of processes for the assembly/disassembly operation within the PoPAN, it is necessary to follow a set of certain rules. In addition to the assembly/disassembly direction of orientation according to the arrows, special attention must be paid to the priority of the edges. For example, the steps taken when following the assembly direction from *Initial Product* to the *Fastener Product i* and *Product 1* (Fig. 1) are as follows:

- 1) Upon arriving at *Fastener Product i*, the conditional linking product to the process edge is examined.
- 2) Continuing to *Fastener Process i*, the sequence of processes edge is checked, revealing a discrepancy in

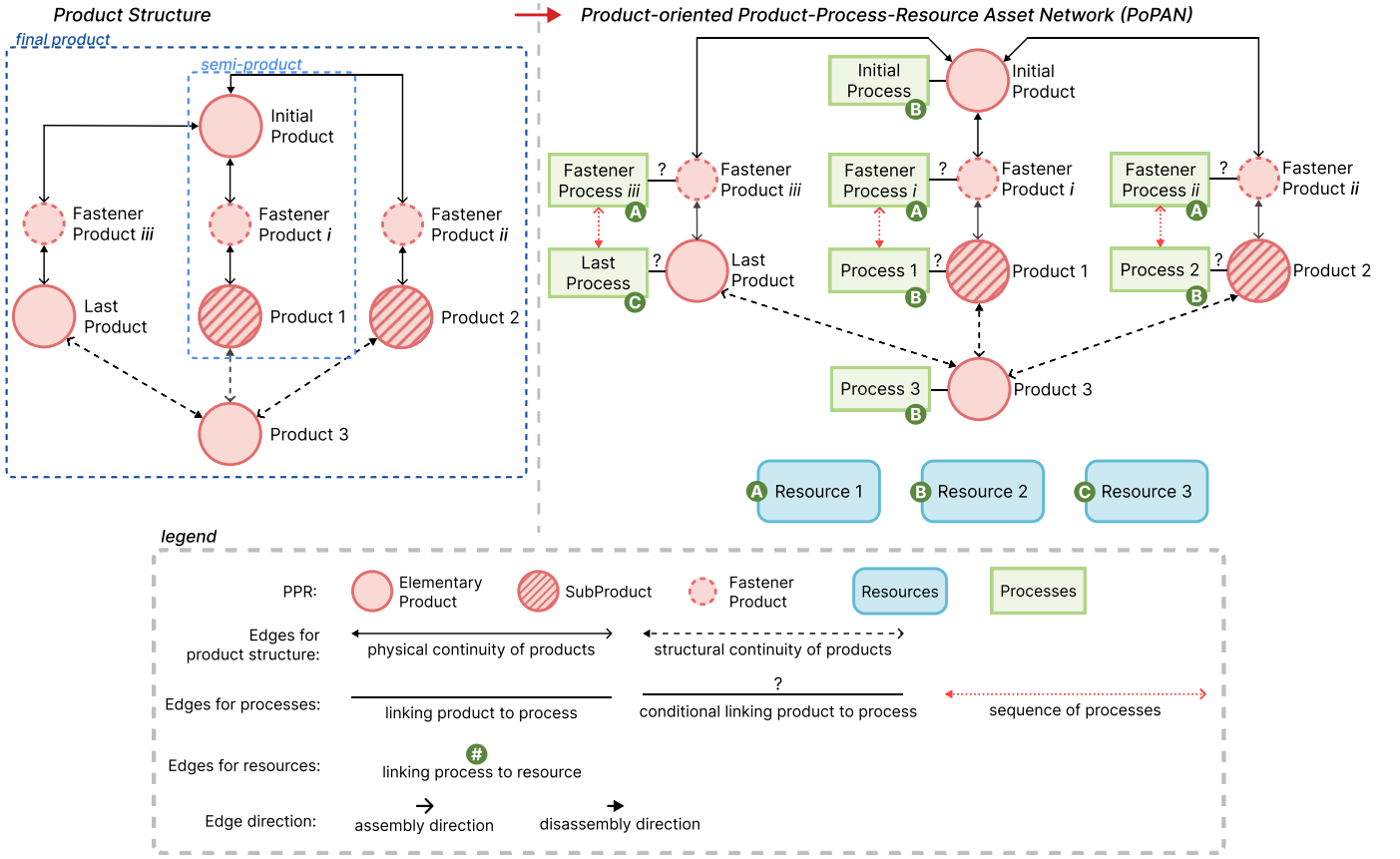


Fig. 1. The proposed PoPAN model for a generic product (on the right), which has been created based on the Product Structure model (on the left). The PoPAN graph is substantiated by products, which have assigned processes, and the edges (their types, orientation, and arrow types) have a crucial role.

direction.

- 3) Returning to *Fastener Product i*, the physical continuity of the product's edge is verified in the correct direction, leading to *Product 1*.
- 4) Upon reaching *Product 1*, the structural continuity of the product's edge is examined, indicating the *Product 3* is to be assembled subsequent to *Product 1*.
- 5) Next, the conditional linking product to process edge is followed, leading to *Process 1*.
- 6) Verifying the edge sequence of processes at *Process 1*, it's confirmed to be in the right direction.
- 7) *Process 1* with *Product 1* is noted simultaneously with *Resource 2*, and the sequence of processes edge is followed back to *Fastener Process i*.
- 8) *Fastener Process i* is noted with *Fastener Product i* simultaneously with *Resource 1*.

As outlined in the preceding steps, when assembling a product the initial edge to examine, if present, is the structural continuity of the product's edge. This edge determines whether the product can be assembled before other products. Once the feasibility of assembly/disassembly is established, the next edges to consider are the linking products to the process edge and conditional linking products to the process edge, which assign processes to products. Additionally, for the conditional

linking products to the process edge, the sequence of the process edge is presented which ensures the processes are sequenced correctly. Afterward, the linking product to the process edge is followed to assign a resource to a product after which the physical continuity of the product's edge can be taken to continue to the next product where the whole process is repeated.

These rules, which specify the order in which edges should be traversed along with the assembly/disassembly direction, enable PoPAN to determine the correct path for creating assembly/disassembly operations. This capability facilitates the efficient design of recycling or remanufacturing processes of products.

IV. REPRESENTATION OF POPAN IN AUTOMATIONML

Having the PoPAN model, it is necessary to serialize such a graph into a computer-understandable form to capture the knowledge and support its exchange across involved stakeholders. The following requirements on the data format were postulated: (i) widely adopted and standardized data format, (ii) support for AAS, (iii) modularity and scalability of the description, and (iv) platform neutrality and openness. Considering these requirements, we decided to use the AutomationML data format for serialization of the PoPAN description.

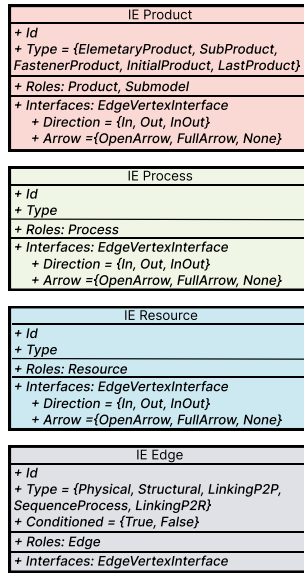


Fig. 2. System Unit Classes of the AutomationML description for PoPAN representation.

The AutomationML representation aims at capturing the PoPAN network as a graph. First, we specified the Interface-ClassLib with the Interface Class *EdgeVertexInterface*. This interface allows the connection of edges and vertices via AutomationML Internal Links. The Interface Class has an attribute *Direction*, which can have values *In* for the incoming edge, *Out* for the outgoing edge, and *InOut* for the undirected edge. Since PoPAN distinguishes assembly and disassembly directions, the *EdgeVertexInterface* Class has also attribute *Arrow*, which can have values *OpenArrow* for the assembly direction, *FullArrow* for the disassembly direction, or *None* for none arrow.

In the second step, we have defined the Role Class Library with the following terms for PoPAN specifications: (i) *Product*, (ii) *Process*, (iii) *Resource*, and (iv) *Edge*. All these Role Classes have attributes *Id* and *Type* (to unambiguously specify these assets) and one or more *EdgeVertexInterface* (to enable connecting edges via Internal Links).

In the third step, we have defined exemplary/generic System Unit Class, which are depicted in Fig. 2. The figure shows four Internal Elements (IE) from the System Unit Class, in compliance with their Roles from the Role Class Library. The Internal Elements are as a matter of fact instances of the aforementioned Role Classes, accompanied by *EdgeVertexInterface* to enable the PoPAN representation.

The IE Product type represents physical products within the whole product. Each IE Product is uniquely identified by an *Id* and has a *Type* including *ElementaryProduct*, *SubProduct*, *FastenerProduct*, *InitialProduct*, or *LastProduct*. Additionally, IE Product has assigned Roles *Product* and *Submodel* (to correspond with AAS). The IE Process type represents processes performed within the assembly/disassembly operations. IE Process is uniquely identified by an *Id*, has assigned Role *Process*, and has a *Type*. The IE Resource type represents

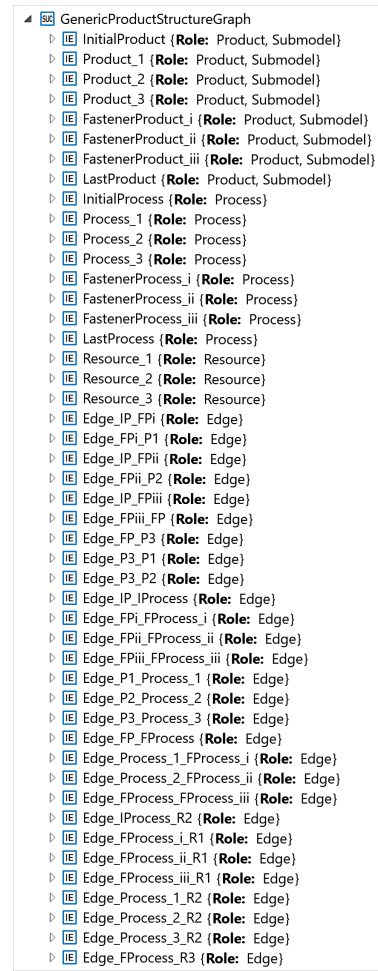


Fig. 3. Example of a generic product structure in the AutomationML Editor.

resources needed to perform processes, such as tools and stations. Similar to IE Process, each IE Resource has a unique identifier *Id*, has an assigned Role *Process*, and has a *Type*. The IE Edge type represents connections between PPR components within the whole product. IE Edge is uniquely identified by *Id*, the Role *Edge* and has a *Type* that can have values *Physical* to represent the physical continuity of products, *Structural* for the structural continuity of products, *LinkingP2P* for linking a product to a process, *SequenceProcess* to represent dependencies/sequences of processes, and *LinkingP2R* for linking a process to a resource. For all four IE Product types, the *EdgeVertexInterface* is defined to ensure connection to other IEs within the PoPAN.

The IE Product supports the Role *Submodel* from the *Asset Administration Shell Role Class Library*³. In this place we should note that we are missing a “SubModelElement/Entity” in this AutomationML library, which would be useful for transforming the description into the AASX format.

The example of a generic product structure according to the

³Online: https://www.automationml.org/wp-content/uploads/2022/04/Asset-Administration-Shell-Representation-V1_0_0.zip

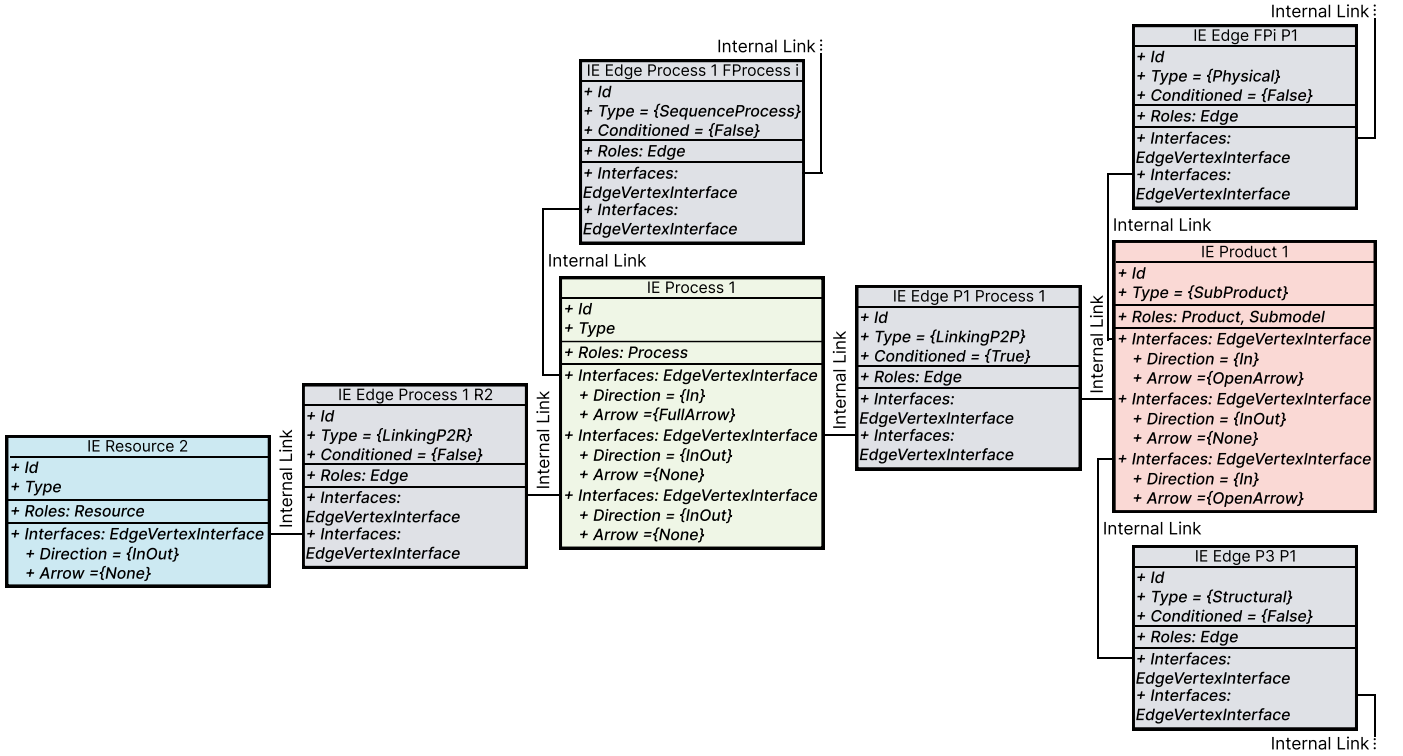


Fig. 4. Example of a specific PoPAN represented in the AutomationML data format. It includes Internal Elements (IE) for Resources (blue), Edges of PoPAN (grey), Process (green), and Products (salmon).

PoPAN model from the previous section is depicted in Fig. 3. If we need to assign more than one edge to a vertice, we have to increase the number of *EdgeVertexInterfaces*. In Fig. 4, a network of PoPAN elements represented in AutomationML is shown for a *Product 1* from a generic product (Sec. III).

AutomationML supports annotating objects relevant to the Asset Administration Shell (AAS). AAS is becoming an important industrial artifact to be exchanged in the entire supply chain. For creating the AAS, we are using the aforementioned tool AASX Package Explorer, which can import an AutomationML file and transform it into the AASX format, which can accompany the product during its entire life-cycle.

This comprehensive product description, when used in conjunction with the AAS, can accompany the product throughout its entire life-cycle. It can assist in scenarios where defective parts need replacement, while also facilitating the tracking of new or missing products (e.g., a missing screw) to streamline the recycling process.

V. ELECTRIC VEHICLE BATTERY REMANUFACTURING USE-CASE

To illustrate the advantages of the PoPAN approach and to test its validity and benefits for industrial applications, electric vehicle (EV) battery remanufacturing serves as a use-case. From a circular economy perspective, where resources are reused to minimize waste and enhance sustainability, EV battery recycling/remanufacturing is a significant concern. This is due to the valuable materials contained within EV batteries

and the environmental risks posed by improper management of EV batteries in their end-of-life phase.

In our laboratory conditions, utilizing the Industry 4.0 Testbed⁴ at CTU in Prague – CIIRC, we employ KUKA robots to manipulate the real EV battery according to predefined parameters (see Fig. 5). The primary goal of this use-case is to disassemble the entire EV battery with a flexible robotic system. This approach offers the potential use of modules from the EV battery for secondary applications. Additionally, the secondary goal involves replacing broken modules, which necessitates a combination of assembling and disassembling operations. We were seeking a formalism that allows to accommodate all possible operations in our use-case, and PoPAN fits our requirements perfectly for all considered applications. The visualization of PoPAN for a simplified description of the EV battery is depicted in Fig. 6. It illustrates all relevant data for creating assembly/disassembly operations. As outlined in Section III, this visualization contains products, processes, and resources interconnected by different types of edges, with two types of arrows indicating assembly/disassembly orientations. Following the rules described in Section III, remanufacturing and recycling operations can be effectively designed. Through integration with the AAS, PoPAN serves as a digital shadow for the EV battery, accompanying it throughout the entire life-cycle. Such an integration enhances data management and data sharing between involved stakeholders.

⁴Online: <https://www.ciirc.cvut.cz/teams-labs/testbed/>

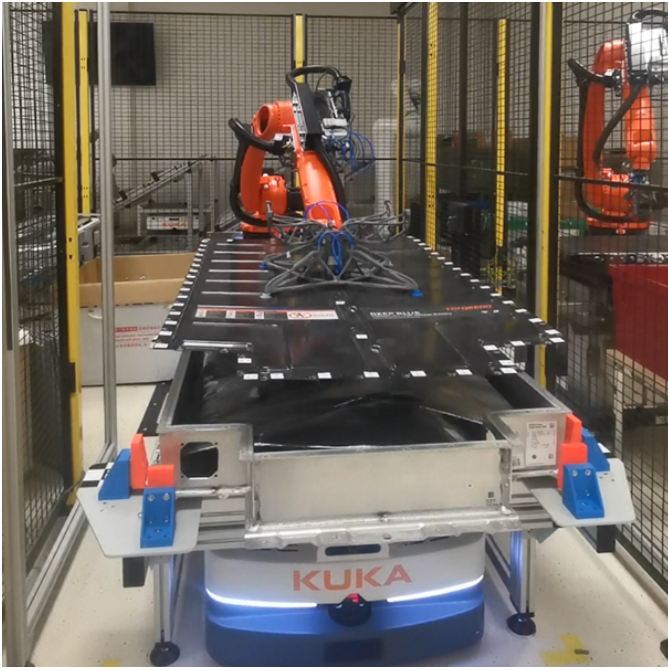


Fig. 5. A KUKA robot is lifting the lid of an EV battery from a BMW i3 in the Industry 4.0 Testbed at CIIRC

From the PoPAN presented in Fig. 6, we can query the set of operations to disassemble the battery in the robotic workcells. We can see that the *Last Product* is the Lid, which has to be manipulated/removed from the EV battery. However, the conditioned edge between the Lid product and Manipulation in this PoPAN requires to evaluate the edges connected to the vertex Manipulation. Since we are currently disassembling, we cannot proceed to the process Screwing. The search algorithm proceeds in the full arrow direction to the Bolts M6 with the assigned Screwing process. When finished, it can continue to the Manipulation process, which was previously skipped. Then the search algorithm can proceed to the product Battery Box, which has more than one incoming edge. Prior to executing the assigned process Manipulation, the incoming edges have to be passed. Therefore, the search algorithm proceeds to the remaining branches and starts from the vertex that has all dependencies satisfied. During this systematic search, it goes through the entire graph until all processes are done to get to the *Initial Product*, which is the empty battery box in this case.

Through the performed evaluation, we have confirmed the efficacy of PoPAN, as it seamlessly integrates all tasks developed in this use-case. Based on this lessons-learned use-case, we believe PoPAN can be also well-suited for other use-cases.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a specific adaptation of the PAN paradigm called Product-oriented PPR Asset Network (PoPAN), having the product structure as the core basis of the descriptive model. Such an adaptation enables to express not only the assembling processes but also the disassembling

ones in just one compact form. The PoPAN model can be afterward queried to get the right sequence of production processes/operations for various production, reparation, remanufacturing, or recycling purposes during later phases of the whole product life-cycle.

The PoPAN model serves as a model-driven engineering backbone for process management within the whole product life-cycle. It enables large-scale system integration and scalability of the models and respective systems. The proposed approach is demonstrated in a case study on disassembling EV batteries to upcycle them from automotive applications towards stationary battery storages in industry or households.

Addressing the research question *RQ1*, this paper proposes a PPR-based model coming up from the PAN model, which however primarily reflects the product structure. The model allows to seamlessly combine the production and remanufacturing/disassembling processes into one unified description, addressing the research question *RQ2*. This overall model was presented in Sec. III.

During the presented research, we were aware of the necessity to foster the adaptability of this new modeling approach by industrial and academic stakeholders. Therefore, we have from the very beginning considered an export to the AutomationML data format and Asset Administration Shell, addressing the research question *RQ3*, providing the serialization of the model into AutomationML in Sec. IV.

In future work, we would like to implement a user-friendly search algorithm for generating the right directed acyclic graph of processes for the queried situation and export it into BPMN.

ACKNOWLEDGMENT

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS24/125/OHK2/3T/12 and the project “Regeneration of used batteries from Electric Vehicles” (Slovak ITMS2014+ code 313012BUN5), the project is part of the Important Project of Common European Interest (IPCEI) called the European Battery Innovation (code OPII-MH/DP/2021/9.5-34), which was announced as a part of Operational Program Integrated Infrastructure (EZOP ID 71235). This work was co-funded by the European Union under the project Robotics and advanced industrial production – ROBOPROX (reg. no. CZ.02.01.01/00/22_008/0004590).

REFERENCES

- [1] J. Oláh, N. Aburumman, J. Popp, M. A. Khan, H. Haddad, and N. Kitukutha, “Impact of industry 4.0 on environmental sustainability,” *Sustainability*, vol. 12, no. 11, 2020.
- [2] M. Ahmad, B. R. Ferrer, B. Ahmad, D. Vera, J. L. Martinez Lastra, and R. Harrison, “Knowledge-based ppr modelling for assembly automation,” *CIRP Journal of Manufacturing Science and Technology*, vol. 21, pp. 33–46, 2018.
- [3] S. Biffl, J. Musil, A. Musil, K. Meixner, A. Lüder, F. Rinker, D. Weyns, and D. Winkler, “An industry 4.0 asset-based coordination artifact for production systems engineering,” in *IEEE 23rd Conference on Business Informatics (CBI 2021)*, vol. 01, 2021, pp. 92–101.
- [4] T. Bergs, S. Gierlings, T. Auerbach, A. Klink, D. Schraknepper, and T. Augspurger, “The concept of digital twin and digital shadow in manufacturing,” *Procedia CIRP*, vol. 101, pp. 81–84, 2021.

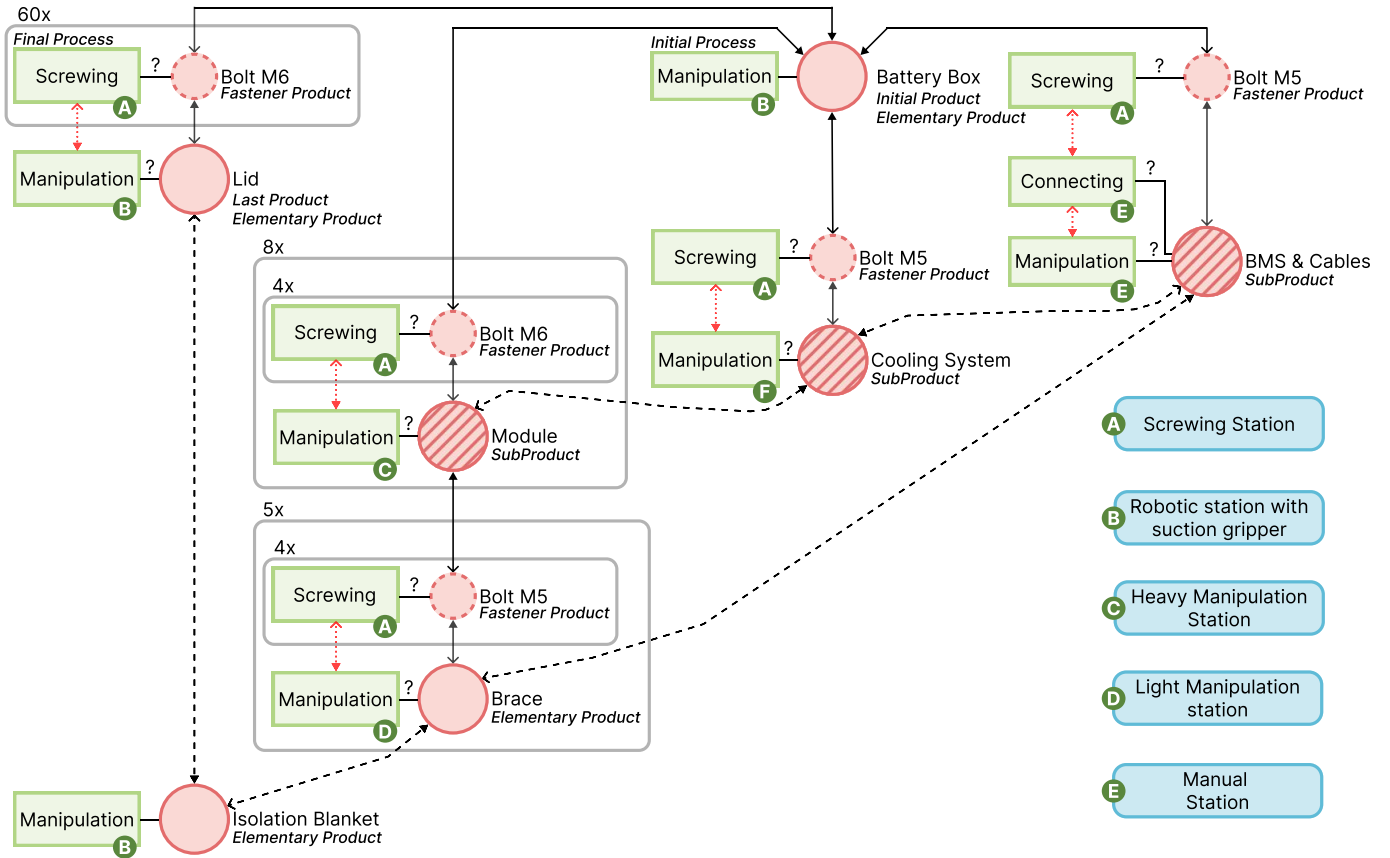


Fig. 6. PoPAN describing a simplified structure of the EV battery from BMW i3. Products are accompanied by production processes and resources, which enables to find assembling and disassembling operations from one graph.

- [5] R. Drath, A. Luder, J. Peschke, and L. Hundt, "AutomationML - the glue for seamless automation engineering," in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 616–623.
- [6] "Details of the administration shell - from idea to implementation [update]," Federal Ministry for Economic Affairs and Climate Action (BMWK), Plattform Industrie 4.0, Berlin, 2019, available online: <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/vws-i40-detail-presentation.pdf> [Accessed April 4, 2024].
- [7] C. Plociennik, M. Pourjafarian, A. Nazeri, W. Windholz, S. Knetsch, J. Rickert, A. Ciroth, A. do Carmo Precci Lopes, T. Hagedorn, M. Vogelgesang, W. Benner, A. Gassmann, S. Bergweiler, M. Ruskowski, L. Schebek, and A. Weidenkaff, "Towards a digital lifecycle passport for the circular economy," *Procedia CIRP*, vol. 105, pp. 122–127, 2022, the 29th CIRP Conference on Life Cycle Engineering, April 4 – 6, 2022, Leuven, Belgium.
- [8] A. Beaudet, F. Larouche, K. Amouzegar, P. Bouchard, and K. Zaghib, "Key challenges and opportunities for recycling electric vehicle battery materials," *Sustainability*, vol. 12, no. 14, 2020.
- [9] "Details of the asset administration shell - part 1," Federal Ministry for Economic Affairs and Climate Action (BMWK), Plattform Industrie 4.0, Berlin, 2022, available online: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.html [Accessed April 4, 2024].
- [10] "Details of the asset administration shell - part 2," Federal Ministry for Economic Affairs and Climate Action (BMWK), Plattform Industrie 4.0, Berlin, 2021, available online: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.html [Accessed April 4, 2024].
- [11] R. Drath, M. Rentschler, and M. Hoffmeister, "The AutomationML Component Description in the context of the Asset Administration Shell," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1278–1281.
- [12] A. Lüder, A. Graf, M. Müller, M. Schleipen, M. Wiegand, S. Biffl, and R. Drath, *Serialization of the Asset Administration Shell by AutomationML*. Berlin, Boston: De Gruyter Oldenbourg, 2021, pp. 365–378.
- [13] N. Schmidt and A. Lüder, "Development of a generic model for end-of-life scenarios of production systems," *Procedia Manufacturing*, vol. 8, pp. 385–392, 2017, 14th Global Conference on Sustainable Manufacturing, GCSM 3-5 October 2016, Stellenbosch, South Africa.
- [14] N. Schmidt and A. Lueder, "Recovery planning method as end-of-life support for production systems," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 1103–1108.
- [15] J. Bézin, "In search of a basic principle for model driven engineering," *Novatica/Upgrade*, vol. 5, 01 2004.
- [16] S. Kent, "Model driven engineering," in *Integrated Formal Methods*, M. Butler, L. Petre, and K. Sere, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 286–298.
- [17] E. Guerra, J. de Lara, D. S. Kolovos, R. F. Paige, and O. M. dos Santos, "Engineering model transformations with transML," *Software & Systems Modeling*, vol. 12, pp. 555–577, 2013.
- [18] K. Lano, "Program translation using model-driven engineering," in *IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2022, pp. 362–363.
- [19] L. Berardinelli, A. Mazak, O. Alt, M. Wimmer, and G. Kappel, *Model-Driven Systems Engineering: Principles and Application in the CPPS Domain*. Cham: Springer International Publishing, 2017, pp. 261–299.
- [20] K. Meixner, J. Decker, H. Marcher, A. Lüder, and S. Biffl, "Towards a domain-specific language for product-process-resource constraints," in *25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, pp. 1405–1408.
- [21] D. Winkler, P. Novák, K. Meixner, J. Vyskočil, F. Rinker, and S. Biffl, "Product-process-resource asset networks as foundation for improving cpps engineering," in *26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2021)*, 2021, pp. 1–4.