Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 59-25 (2025) 89–94

A POL for Modular Plants using Capabilities in Master Recipes

Michael Winter * **Alicia Eve** * **Benedikt Schmetz** * **Tobias Kleinert** *

* *Chair of Information and Automation Systems, RWTH Aachen University, Aachen, Germany (e-mail: {m.winter, b.schmetz, kleinert}@iat.rwth-aachen.de, alicia.eve@rwth-aachen.de)*

Abstract: Modular automation offers solutions for increased flexibility in the process industry by utilizing Module Type Package (MTP). This contribution presents an open-source Process Orchestration Layer (POL) for flexible production, which enables execution of standardized process descriptions in process industry using the Capability, Skill and Service model. A unique feature of the POL is its ability to execute master recipes defined in BatchML that are compliant with ISA-88. These recipes include offered capabilities provided by Process Equipment Assemblies (PEAs) with skills according to implemented MTP services and procedures. Each PEA provides distinct capabilities, e.g., heating or mixing. The POL, labeled Recipol, parses the recipe and MTP files, coordinates the available PEAs, and executes distributed process operations while monitoring their states. The POL was tested and validated on a modular testbed comprising four heterogeneous and vendor-independent PEAs connected via OPC UA. This work demonstrates the potential of MTP-based architectures and provides a validated foundation for future research on interoperable modular automation systems utilizing standardized process descriptions.

Keywords: Modular Automation, Process Orchestration Layer, Master Recipe, Capability, Skill

1. INTRODUCTION

In process industry, the demand for shorter production cycles (Stenger et al., 2016) and the variety of products has increased, along with the rapid pace of innovation and technological development (Boto et al., 2023). Furthermore, modular production and automation, facilitated by the use of Module Type Package (MTP), have increased the adaptability and flexibility of industrial processes (VDI/VDE/NAMUR 2658, 2022). MTP is a standard that describes, among other things, functions as services and procedures as well as the interfaces of a Process Equipment Assembly (PEA) in an XML-based file exchange format. A PEA is an autonomous unit consisting of one or more functional equipment assemblies, representing a process step or providing infrastructure within a modular plant.

Depending on the process description, both the selection of required PEAs and the sequence in which the process steps are executed may vary. Consequently, the modular plant must be capable of handling varying process sequences and executing the individual steps in the correct order, e.g., stirring followed by dosing and heating, on appropriate and available PEAs. The execution of these process descriptions is carried out within the process orchestration layer (POL) by invoking the PEAs and their corresponding procedures defined in the MTP services. According to VDI/VDE/NAMUR 2658 (2022), the POL represents a higher-level automation and information layer responsible for supervising and coordinating the operation of a modular plant. Several vendors of control systems, such as ABB, Honeywell, Siemens, Yokogawa, and software manufacturers, such as COPA-DATA and Semodia, offer commercial, licensed POLs. With the increasing spread of modular plants and the resulting need for cross-component orchestration, the development of a manufacturer-independent and open-source solution for process control is necessary. An example of an open-source POL is Polaris (Lorenz et al., 2021), which, however, does not yet support the execution of modular plants based on flexible and standardized process descriptions.

Considering ISA-88 (2010), IEC 61512 (1997), and the position paper from NAMUR (2024), plant-specific process descriptions can be formalized as machine-readable master recipes using BatchML,[1] which is based on XML. Therefore, the significance of flexible modular plant execution with an open-source POL that supports formalized and standardized plant-specific master recipes becomes evident. Using available PEAs in the plant-specific master recipe serves as a starting point for a flexible and automated execution process, resulting in process orchestration by leveraging the concept of capabilities and skills in manufacturing (Köcher et al., 2023). The resource, i.e., a PEA, provides an executable implementation of an encapsulated function through a skill, which is specified by the corresponding offered capability. Various process sequences can be realized by combining PEAs with provided skills, i.e., implemented MTP services or procedures, thus enabling flexible batch manufacturing. Therefore, this contribution aims to address the following questions:

[1] `https://github.com/MESAInternational/B2MML-BatchML`, accessed Apr. 2025

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2025.11.930

- How can a flexible process sequence be realized using an open-source POL, MTP-files, and master recipes conform to ISA-88?
- How can flexible production be realized in the context of the CSS model using MTP?
- What information must a formalized and standardized master recipe contain to enable execution of changeable process sequences using a POL?

The objective of this contribution considers flexible production by providing an open-source POL, labeled *Recipol*, which enables execution of standardized process descriptions in process industry.

The contribution is structured as follows. Section 2 provides fundamentals, requirements for the contents of a master recipe, and a brief overview of related work. Subsequently, Section 3 introduces the system design and implementation of the open-source POL *Recipol*. The validation is shown in Section 4 which demonstrates the results on a multi-vendor modular plant. Finally, Section 5 summarizes and outlines future research work.

2. FUNDAMENTALS AND RELATED WORK

2.1 Fundamentals

The product, process and resource (PPR) model provides a conceptual framework in manufacturing and production engineering, organizing information related to production processes (Schleipen et al., 2015). To enhance flexibility, the strong interconnection within the PPR model is extended by the functional aspects introduced by the Capability, Skill and Service (CSS) model (Köcher et al., 2023). In this contribution, capabilities, skills, and their interface are considered, while services in the context of the CSS model refer to commercial aspects of production and are not considered further. A capability is defined as a description of a function, independent of its implementation (Köcher et al., 2023), to achieve a desired effect (Bock et al., 2024). Furthermore, capabilities are (a) required for the production of products in one or more process steps (required capabilities), without specifying which resource will provide the necessary capability, and (b) offered by manufacturing and operational resources, as defined in resource descriptions (offered capabilities). A skill is defined as the executable implementation of an encapsulated (automation) function, as specified by a capability. In addition, a skill must provide an interface, may include multiple parameters, and operates in accordance with a state machine to control a specific process step. A resource can be equipped with multiple skills.

The MTP standard enables formal, vendor-independent description of interfaces (a) for production execution using modular units, i.e., PEAs, and (b) for communication between different control systems and automation functions. This standard facilitates the development of flexible and scalable production architectures. The use of the MTP standard reduces the required project-specific engineering effort while simultaneously increasing reusability and adaptability. This is achieved by enabling the separation between project-independent module development and module integration into the process orchestration layer (POL engineering), ensuring interoperability across vendors (VDI/VDE/NAMUR 2658, 2022). An MTP file provides a functional description of a PEA, with a particular focus on service execution. This service can be implemented in different ways on a given PEA. For instance, the service “Stirring” may be realized as time-based stirring, i.e., stirring for a defined duration, or as continuous stirring. These different operation modes are defined as procedures within the MTP file. In addition to module-specific services and their procedures, the MTP file defines the parameters and communication interfaces for execution on the PEA. The actual execution logic is implemented within the programmable logic controller (PLC). Furthermore, the PLC contains configuration settings for the communication via OPC UA, the implementation of services and their procedures, as well as the mapping of input and output (I/O) signals to the field devices. According to the CSS model, the automation functions (services and their procedures) implemented on the PLC are referred to as skills. The MTP file represents the interface of these skills, defining the corresponding services and their procedures, parameters, and communication access via OPC UA.

The process orchestration is carried out within the POL to enable the coordinated monitoring and control of PEAs and to support their integration into the overall plant system (Klose et al., 2019). The POL serves as the interface between a higher-level Manufacturing Execution System (MES) and the underlying control level. An overview of the requirements that a POL should fulfill is provided in Klose et al. (2019) and Lorenz et al. (2021).

The ISA-88 standard defines a structured framework for batch control systems, comprising three model types: the process model, the procedural control model, and the physical model (ISA-88, 2010). Within the procedural control model, recipes define the sequence of operations and processing steps required to execute production in batch-oriented industries. The *master recipe* defines general, reusable process instructions, including procedures, steps, and the use of physical equipment units (PEAs) as resource types. According to the CSS model, these process steps are represented as *offered capabilities* of the PEAs. In contrast, a *control recipe* is a plant-specific, executable instance derived from a master recipe, tailored to a particular batch run and the actual equipment assignments. To support adaptability across various production scenarios, the master recipe adopts a modular and hierarchical structure that allows individual steps or units to be flexibly reused, adapted, or replaced without the need for a complete rewrite. A formal representation of both master and control recipe is enabled using BatchML, an ISA-88-compliant XML schema that supports automated control code creation and execution as well as standardized communication in batch manufacturing environments. The markup language enables the description of data in both human-readable and machine-processible formats. In BatchML, a set of tags and a hierarchical structure are defined to describe the concepts of ISA-88.

2.2 Requirements for the contents of a master recipe

The POL presented in Section 3 utilizes a master recipe according to the ISA-88 standard to define the executable process sequence. Although BatchML specifies the XML schema structure and permissible XML elements for master recipes, specifications for the contents of all of these elements are not explicitly defined. Therefore, content-specific requirements are defined to ensure correct interpretation and execution by the POL. The required content of specific XML elements in BatchML is defined as follows:

- The *ProcedureLogic* element defines the process sequence and contains the elements *Step*, *Transition*, and *Link*. A valid sequence consists of alternating steps and transitions, beginning with an initial step and ending with a terminating step.
- The *Step* element represents a process step. Each step includes a unique identifier (ID) and a reference to a corresponding *RecipeElement* element.
- The *RecipeElement* element specifies the details of the referenced process step. This element defines the MTP procedure and their parameters, which are executable on a specific resource (PEA).
- The *Transition* element defines the transition between process steps and includes an ID. A specific syntax is required for the condition in order to enable correct interpretation by the POL. A transition occurs either upon termination of the associated MTP procedure or when a predefined threshold is reached, e.g., as indicated by a sensor.
- The *Link* element defines the connections between steps and transitions within the *ProcedureLogic* structure.

This contribution considers only sequential process descriptions, incorporating step-to-transition and transition-to-step connections. Within the BatchML file, the identifiers of the MTP procedures and parameters may be reused. To maintain uniqueness of XML element identifiers, a prefix is applied. The prefix is user-defined, must not contain spaces, and must be separated from the MTP identifier by a colon. While not part of this contribution, the graphical and user-friendly recipe editor,[2] developed by the Chair of Information and Automation Systems (IAT), enables the creation of master recipes based on these requirements. The editor enables the import of MTP files and the mapping of MTP procedures and parameters to process steps within the master recipe.

2.3 Related Work

The research-oriented POL Polaris, introduced by Lorenz et al. (2021), employs micro frontends to support task-specific interfaces during plant engineering and operation. In Köcher et al. (2022), Business Process Model and Notation (BPMN) is used as the modeling language to specify and execute process descriptions based on capabilities. During execution, each capability in the modeled process is substituted with a concrete skill. However, both approaches do not support standardized process description formats, e.g., as defined by ISA-88, thereby limiting their applicability in modular production environments that require formalized and reusable recipe structures.

Wrede et al. (2016) and Frommel et al. (2020) address flexible manufacturing in robotic environments. Wrede et al. (2016) propose a systemic approach based on reconfigurable modular production cells, using a non-standardized execution of BPMN for runtime service orchestration. In contrast, Frommel et al. (2020) present an enhanced automated preforming process that incorporates a POL and a database infrastructure. Their system autonomously coordinates sub-processes such as cutting, transport, fixation, draping, and quality assurance, while continuously logging execution data and sensor values. Together, these approaches illustrate the diverse potential of orchestration technologies to enable adaptability, modularity, and transparency in robotic manufacturing systems.

In Stutz et al. (2023), an approach is presented that integrates non-choreography-enabled equipment assemblies into service choreographies by employing decentralized orchestration. While choreography refers to the distributed interaction among autonomous modules without a central control instance, decentralized orchestration introduces orchestrator functionalities within individual modules to manage procedural sequences and control functionalities. However, the approach necessitates specific software component extensions on PEAs, which restricts its applicability to existing modular plants utilizing MTP.

Existing approaches in the process industry do not yet provide an open-source POL capable of realizing flexible process sequences through the CSS model and standardized process descriptions, such as master recipes conform to ISA-88.

3. SYSTEM DESIGN AND IMPLEMENTATION

In order to realize flexible process sequences based on the CSS model and standardized process descriptions, an open-source POL, *Recipol*,[3] has been developed at IAT. The functional interaction of the developed POL is shown in Figure 1. Starting with an order request and continuing through the execution of the production process, a production order is initiated based on a master recipe. The POL converts the master recipe into a control recipe and derives corresponding control commands, which are transmitted to the respective PEAs by utilizing their available skills. During execution, the PEAs provide relevant process information to both the POL and the operator, enabling coordinated and informed process control.

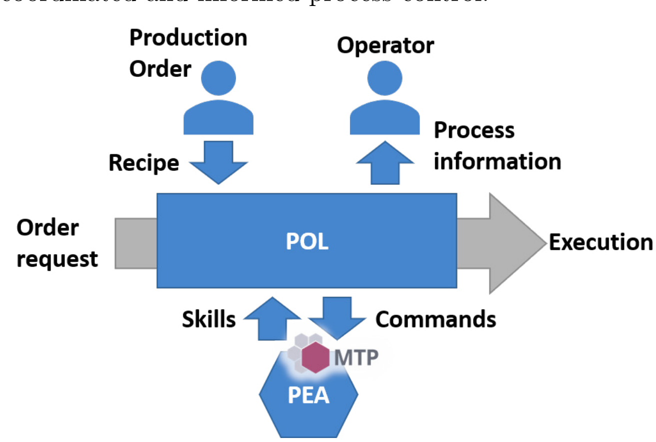

Fig. 1. Interaction of the open-source POL *Recipol*

[2] `https://github.com/rwth-iat/RecipeEditor/`, accessed Apr. 2025

[3] `https://github.com/rwth-iat/Recipol/`, accessed Apr. 2025

A formal representation of the process sequence is required for the POL to execute it on the respective PEAs. For this purpose, a master recipe is used, which contains all necessary information for carrying out the individual process steps. The master recipe is utilized as input to enable automated process execution through the POL. The creation of the master recipe is not part of this contribution. Instead, the master recipe is already provided as a BatchML file conform to ISA-88 and the aforementioned content requirements. This ensures interoperable and non-proprietary dissemination and processing. To enable execution, the information contained in the master recipe must be extracted. A parser reads and processes the elements and their associated data from the BatchML file, providing the necessary information for controlling the PEAs using the POL *Recipol*. To ensure seamless and unambiguous mapping between the process steps of the master recipe and the available skills of the PEAs, the identifiers within the master recipe are aligned with and verified against the procedure definitions specified in the corresponding MTP file. In cases where multiple procedures exist for a given service, the master recipe explicitly references the corresponding procedure of the target resource. The procedures, defined within the MTP file as the standardized interface of these skills, are implemented as executable logic within the PLC of the respective PEAs.

The selection and verification of the master recipe are performed prior to process execution and thus belong to the planning phase. During the execution phase, communication for process control is ensured via OPC UA. The POL sends commands for the execution of skills to the respective PEAs in accordance with the offered capabilities (process steps) defined in the recipe. In the MTP file of the PEAs, in addition to the services, interfaces for the sensors and actuators of the respective components are defined. This allows the POL to monitor current process values, such as fill levels or temperatures. Furthermore, each service includes a state machine through which the POL can query and monitor the current state of the services and procedures. In addition to the functionalities mentioned, further functionalities of the developed POL, *Recipol*, include:

- **Recipe interface:** The sequential process execution at the modular plant is described by a master recipe, provided in the form of a BatchML file. A corresponding interface is implemented in the POL to load and process these recipes.
- **MTP interface:** The MTP files are imported as AML (Automation Markup Language) files, a neutral data format based on XML. The MTP interface translates this information into a data structure interpreted by the remaining POL software modules.
- **OPC UA connectivity:** Communication between the POL and the PEAs is realized via OPC UA. Each PEA hosts an OPC UA server on its respective PLC, which is accessed by the POL to enable process control. To facilitate this connectivity, an OPC UA client is implemented within the POL.
- **Process monitoring:** The current states of the services and signal values of the PEAs, as defined in the MTP file, are queried and continuously monitored. Additionally, an interface is provided to allow external applications — such as graphical user interfaces (which are not part of the currently developed POL) — to retrieve these signals.
- **Monitoring of process sequence execution:** The execution of the process sequence is visualized as a graphical representation, such as a state diagram, on the command line. This representation can then be forwarded to a graphical user interface for further interaction.
- **HMI integration:** The MTP files contain graphical information for the creation of an operator interface, typically in the form of a piping and instrumentation diagram (P&ID). Although the actual development of the graphical user interface is beyond the scope of this contribution, an interface has been implemented to extract and preprocess this information for integration into an HMI.
- **Process orchestration and execution:** Process orchestration is achieved by verifying the process steps defined in the recipe are aligned with the defined IDs provided by the MTP files. Once the sequence is established, process execution is initiated via OPC UA, triggering the corresponding modules.

Figure 2 illustrates the relationship between these functionalities through a simplified representation of data processing. To enable the implementation and the depicted data exchange within the POL, four distinct interfaces are required: (i) an interface for importing and interpreting recipes based on the procedures provided by the available PEAs, (ii) an interface for importing MTP files to verify against the master recipe and to provide information about the PEAs, (iii) an interface for communication and control of the PEAs via OPC UA, and (iv) an interface for monitoring the execution of the recipe.

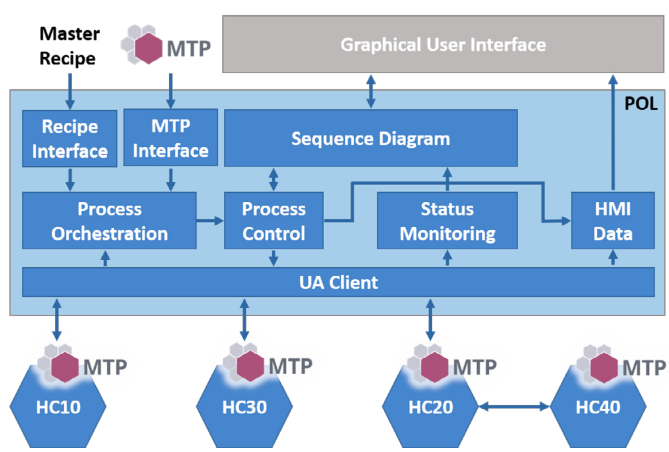

Fig. 2. Data processing of the open-source POL *Recipol*

After the master recipe and the associated MTP files have been imported and prepared for execution using the corresponding interfaces, process orchestration is invoked in order to validate the master recipe and check the availability of required the PEAs. If all required PEAs are available for recipe execution, the user can initiate the process control after confirmation. Using the OPC UA client, the PEAs are controlled by the POL. In Figure 2, the PEAs are represented as HC10, HC20, HC30, and HC40, with HC20 and HC40 being connected via a fieldbus. The subsequent validation demonstrates the orchestration of the PEAs HC10 and HC20 based on the presented POL, *Recipol*. Additionally, the OPC UA client provides information for status monitoring and HMI data. The software components for process control and status monitoring also provide information for a sequence diagram of the executed recipe.

4. VALIDATION

As part of ongoing research in the field of modular automation in process industry, the modular plant Honeycomb[4] (HC) is used to execute various production processes and validate the developed POL. The POL *Recipol* has been validated using an industry-oriented, multi-vendor modular testbed consisting of four PEAs, designated HC10, HC20, HC30, and HC40. These PEAs are flexibly connectable and incorporate diverse commercial components — including actuators, sensors, and control systems — thus providing a realistic representation of a production environment (see Figure 3). The objective of this testbed is to enable MTP-based operation of the modules and orchestrate them via the developed POL and other POLs, thereby establishing a robust foundation for future research and a suitable framework for validating modular automation.

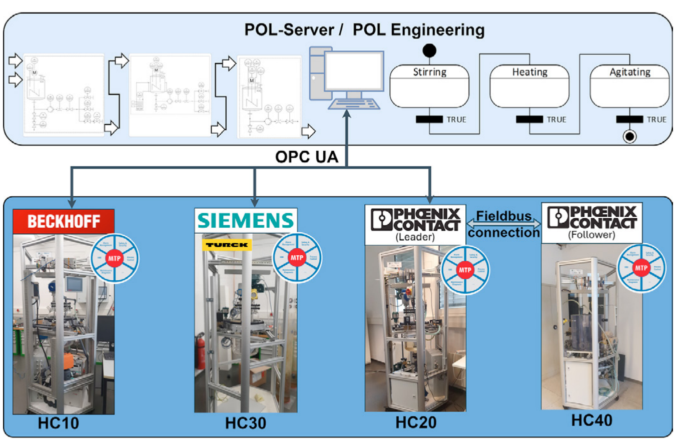

Fig. 3. Validation through PEA execution using *Recipol*

Each of the PEAs includes a tank for holding liquid and a drainage system comprising a pump in combination with both analog and digital valves. Three of the four PEAs are each equipped with a PLC from one of the manufacturers BECKHOFF, PHOENIX CONTACT, SIEMENS, or TURCK, thereby representing individual automation units. The fourth PEA (HC40) represents a special case, as it does not include a dedicated PLC but instead functions as a follower module within a leader/follower control structure, sharing a common fieldbus network with HC20. Additionally, the three PEAs equipped with a PLC each feature an integrated stirrer. One of these modules is also equipped with a temperature control jacket, enabling heating of the contained liquid. Consequently, the PEAs serve as heating and mixing units with offered capabilities and implemented skills, all physically interconnected and communicating via OPC UA.

To validate the orchestration logic, the POL *Recipol* was evaluated using a predefined master recipe in BatchML. This recipe specifies a linear process sequence with individual steps and transitions, and their mapping to MTP-compliant procedures, as shown in Figure 4. It comprises three steps: *001:Stirring:Duration_Stirring*, *002:Dosing: Dosing_Flow*, and *003:Heating:Controlled_Heating*, using the procedures *Duration_Stirring*, *Dosing_Flow*, and *Controlled_Heating* of the MTP services Stirring, Dosing, and Heating. Each step defines a transition condition that must be fulfilled before proceeding: *Stirring completed*, *Dosing completed*, and *Heating completed*, respectively. These conditions refer to the completed state of the procedure as defined by the MTP state machine.

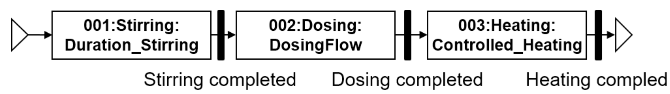

Fig. 4. Validation through master recipe execution using *Recipol* on Honeycombs

The PEAs are implemented in accordance with the MTP standard and are equipped with appropriate interfaces, enabling both the acquisition of process values and the actuation of process functions. After successful parsing and verification of the recipe via the integrated recipe interface, orchestration of all module instances was initiated based on validation against the MTP files and by the user through the command line. Subsequently, process control was executed, allowing for monitoring and control of cross-vendor PEAs. The POL translated the recipe into control commands, which were dispatched to the corresponding PEAs. An internal OPC UA client within the POL managed communication and continuously monitored execution progress by retrieving status information from the modules. The execution sequence was confirmed through status monitoring and evaluated using log outputs (currently as CSV exports) and observed state transitions, as shown in Table 1.

Table 1. State transition of MTP services

Time	**HC10: Service: Heating**	**HC20: Service: Stirring**	**HC20: Service: Dosing**
16:01:09	Idle	Execute	Idle
16:01:30	Idle	Idle	Execute
16:02:02	Execute	Idle	Idle

The correct orchestration of HC10 and HC20 demonstrated that the recipe steps were successfully mapped to the corresponding MTP procedures. Furthermore, deviations and state changes were detected and relayed to the operator via the command line, which could be extended with a graphical user interface in future work. The validation confirms that the POL is capable of executing standard-compliant master recipes on PEAs with linear process descriptions.

5. SUMMARY AND OUTLOOK

In this contribution, the open-source POL *Recipol* is presented that enables standardized and flexible execution of linear batch production processes in modular plants

[4] `https://www.iat.rwth-aachen.de/cms/iat/forschung/anlagen/~zrnbp/honeycomb/?lidx=1`, accessed Apr. 2025

based on the CSS model and master recipes according to ISA-88. The results demonstrate the applicability of the developed system as a foundation for flexible and modular automation architectures, prioritizing modularity in both design and implementation. The developed POL is able to execute ISA-88-conform master recipes in BatchML. For orchestration, content requirements for a master recipe are defined and must be aligned with the corresponding MTP files. Through OPC UA communication and the use of MTP-defined PEAs, the POL translates master recipes into executable commands for individual PEAs, showcasing modular automation. The validation of the POL demonstrated the feasibility and reliability of this approach. Recipe-based orchestration, execution monitoring, and correct handling of state transitions were confirmed using the industry-oriented, multi-vendor modular testbed HC. The implementation, testing, and validation of the developed POL on MTP-conform PEAs evaluates the potential for industrial application.

In future work, several enhancements are planned: First, a front-end implementation for more user-friendly use of the POL is necessary. This front-end needs to offer functionality for manual control of individual modules to facilitate testing of PEA services and procedures, and to enable intervention in case of execution errors. This will be achieved by leveraging the state machine defined within the MTP, which allows the POL to invoke and execute specific procedures independently of an automated recipe sequence. Second, the execution of parallel and serial process sequences, as well as alarm and event management, is planned. Third, the integration of dynamic recipe adaptation during runtime using the concept of VoR (Verification of Request) (NAMUR, 2025) could enable more flexibility in handling process disturbances or changes in production requirements. Finally, further standard conform process descriptions, e.g., formalized process description (FPD), should be considered as input for the developed POL. Overall, the presented POL *Recipol* contributes to the advancement of modular production automation and supports the transition toward flexible process automation systems based on standardized process descriptions.

REFERENCES

Bock, J., Klausmann, T., Kleinert, T., Köcher, A., Simon, M., and Winter, M. (2024). A formal approach to defining effects of manufacturing functions. In *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8. IEEE. doi:10.1109/ETFA61755.2024.10710847.

Boto, S.T., Bardl, B., Harnisch, F., and Rosenbaum, M.A. (2023). Microbial electrosynthesis with Clostridium ljungdahlii benefits from hydrogen electron mediation and permits a greater variety of products. *Green chemistry: an international journal and green chemistry resource: GC*, 25(11), 4375–4386. doi:10.1039/D3GC00 471F.

Frommel, C., Krebs, F., Haase, T., Vistein, M., Schuster, A., Larsen, L., Körber, M., Malecha, M., and Kupke, M. (2020). Automated manufacturing of large composites utilizing a process orchestration system. *Procedia Manufacturing*, 51, 470–477. doi:10.1016/j.promfg.2020.10.0 66.

International Electrotechnical Commission (1997). IEC 61512-1: Batch control part 1: Models and terminology.

International Society of Automation (2010). ANSI/ISA–88.00.01: Batch control part 1: Models and terminology.

Klose, A., Merkelbach, S., Menschner, A., Hensel, S., Heinze, S., Bittorf, L., Kockmann, N., Schäfer, C., Szmais, S., Eckert, M., Rüde, T., Scherwietes, T., Da Silva Santos, P., Stenger, F., Holm, T., Welscher, W., Krink, N., Schenk, T., Stutz, A., Maurmaier, M., Stark, K., Hoernicke, M., Unland, S., Erben, S., Kessler, F., Apitz, F., and Urbas, L. (2019). Orchestration requirements for modular process plants in chemical and pharmaceutical industries. *Chemical Engineering & Technology*, 42(11), 2282–2291. doi:10.1002/ceat.201900298.

Köcher, A., Belyaev, A., Hermann, J., Bock, J., Meixner, K., Volkmann, M., Winter, M., Zimmermann, P., Grimm, S., and Diedrich, C. (2023). A reference model for common understanding of capabilities and skills in manufacturing. *at - Automatisierungstechnik*, 71(2), 94–104. doi:10.1515/auto-2022-0117.

Köcher, A., Da Silva, L.M.V., and Fay, A. (2022). Modeling and Executing Production Processes with Capabilities and Skills using Ontologies and BPMN. In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8. IEEE. doi:10.1109/ETFA52439.2022.9921564.

Lorenz, J., Lohse, C., and Urbas, L. (2021). Microfrontends as opportunity for process orchestration layer architecture in modular process plants. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 01–04. IEEE. doi:10.110 9/ETFA45728.2021.9613474.

NAMUR (2024). State Model Alignment of ISA 88 and Module Type Package. Position paper. URL `https://www.namur.net/de/publikationen/news-archiv/wg-position-state-model-alignment-of-isa-88-and-module-type-package-wurde-veroeffentlicht.html`.

NAMUR (2025). NAMUR NE 178 - Open Architecture NOA Verification of Request.

Schleipen, M., Lüder, A., Sauer, O., Flatt, H., and Jasperneite, J. (2015). Requirements and concept for Plug-and-Work. *at - Automatisierungstechnik*, 63(10), 801–820. doi:10.1515/auto-2015-0015.

Stenger, F., Schmalz, D., Bieringer, T., Brodhagen, A., Dreiser, C., and Schweiger, A. (2016). Flexible Chemical Production by Modularization and Standardization: Status Quo and Future Trends. *Chemie Ingenieur Technik*, 88(9), 1217. doi:10.1002/cite.201650240.

Stutz, A., Fay, A., Barth, M., and Maurmaier, M. (2023). Automation service choreographies using decentralized orchestration to integrate non-choreography-enabled equipment assemblies. In *2023 IEEE 21st International Conference on Industrial Informatics (INDIN)*, 1–8. doi:10.1109/INDIN51400.2023.10218097.

VDI, VDE, and NAMUR (2022). VDI/VDE/NAMUR 2658 - Automation engineering of modular systems in the process industry. Berlin: Beuth Verlag GmbH.

Wrede, S., Beyer, O., Dreyer, C., Wojtynek, M., and Steil, J. (2016). Vertical Integration and Service Orchestration for Modular Production Systems Using Business Process Models. *Procedia Technology*, 26, 259–266. doi:10.1016/j.protcy.2016.08.035.