Int. J. Appl. Math. Comput. Sci., 2026, Vol. 36, No. 2, 315–327
DOI: 10.61822/amcs-2026-0021

PLUG–AND–PRODUCE INTEGRATION IN INDUSTRIAL CYBER–PHYSICAL SYSTEMS BASED ON IEC 61499 AND THE MODULE TYPE PACKAGE

DAN HUANG [a], LIKUAN ZHANG [b,c], CHUANYANG YU [b,c], HUI ZHANG [b,c], WENBIN DAI [a,c,*]

[a] School of Automation and Intelligent Sensing
Shanghai Jiao Tong University
800 Dongchuan Rd, Minhang District, Shanghai, China
e-mail: `huangdan@sjtu.edu.cn`

[b] Department of Research and Development
Beijing Urban Construction Intelligent Control Co., Ltd
Bldg 4, CTYD 2, Liulijing Rd, Yongdingmenwai St, Dongcheng District, Beijing, China

[c] Beijing Urban Construction Intelligent Control Joint Lab on Universal Automation
Shanghai Jiao Tong University School of Automation and Intelligent Sensing
800 Dongchuan Rd, Minhang District, Shanghai, China
e-mail: `w.dai@sjtu.edu.cn`

With the digital transformation of industrial automation systems, engineers must manage the increasing system size and complexity due to the operation and information technology convergence. To achieve massive customizations, functions provided by various field devices must be integrated flexibly with controllers. This paper proposes a plug-and-produce method for integrating field device functions with edge controllers using the IEC 61499 function block network as the process orchestration layer and the Module Type Package (MTP) as the standardized exchange file format. This seamless integration between IEC 61499 and the MTP standard significantly reduces reconfiguration time by 80% and improves the efficiency of developing industrial cyber-physical systems by up to 50%.

Keywords: industrial cyber-physical systems, Module Type Package, IEC 61499 function blocks, plug-and-produce.

1. Introduction

Industrial automation systems are undergoing digital transformations with support from emerging information and communication technologies (ICTs). With multi-core CPUs and cheaper memory storage equipped on edge computing nodes (ECNs), the traditional industrial automation pyramid based on the ISA-95 standard (International Society of Automation, 2010) is shifting to an open interconnected architecture between industrial cloud platforms and edge computing nodes (Industry IoT Consortium, 2022), as shown in Fig. 1. In the new industrial cyber-physical system architecture, even sensors and drives on the shop floor are capable of exchanging data with industrial cloud platforms while performing complex calculations in real-time.

With the massive product customization introduced by Industry 4.0 (Scheer, 2015), manufacturing processes must be adjusted frequently to adopt operation technologies (OTs) and information technologies (ITs) tasks in industrial edge applications. Equipment on the shop floor needs to be collaborated in a flexible way to address different kinds of requirements. A rapid orchestration method for field devices to adopt changes onsite is necessary to achieve plug-and-produce field devices. Functionalities for each field device need to be updated online in a standard way such that all modifications must be completed before the minimum repetition time of the process.

The IEC 61499 standard (International Electrotechnical Commission, 2012) can be used for service orchestration because it can dynamically reconfigure applications through the standard function

---

[*] Corresponding author

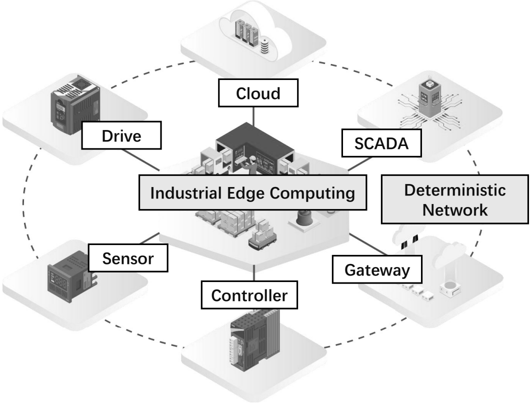


Fig. 1. Industrial cloud and edge computing architecture.

block (FB) interface. Both OT and IT functions can be encapsulated in IEC 61499 FBs, and manufacturing processes can be modeled as an FB network that flows through event connections. Meanwhile, the Module Type Package (MTP) standard (VDI, 2020) describes how to access functions on equipment in a modular way to form process orchestration and provides a deployment model with a standardized file package format for process automation. The main contribution of this paper is to provide a flexible plug-and-produce method for reorganizing functionalities from industrial field devices by combining the MTP standard with IEC 61499 FBs. The IEC 61499 FB networks are used as the process orchestration layer (POL) to integrate various MTP modules rapidly. In addition, mapping between the IEC 61499 management model and the MTP file format is proposed to provide reconfigurability for each device in industrial cyber-physical systems.

The rest of the paper is organized as follows. In Section 2, related works on IEC 61499-based service orchestration, the Module Type Package, and plug-and-produce software components for industrial automation systems are reviewed. Section 3 provides the process orchestration method for plug-and-produce field devices based on IEC 61499 and the MTP. The combined deployment model and mapping rules are proposed in Section 4 for reconfiguring industrial edge devices. In Section 5, a case study of a packaging line is used to prove the concept of the plug-and-produce feature extended to the existing MTP standard, along with a performance analysis. Finally, conclusions and future work are discussed in Section 6.

2. Literature review

Flexible software development and deployment have been important topics for Industry 4.0, the Industrial Internet, and edge computing over the past decade. Data-driven fault identification and analysis approaches are heavily adopted in the industrial domain (Klimek, 2014; Ragot and Kallas, 2022; Fu and Wang, 2025), which requires dynamic reconfiguration during operations.

2.1. IEC 61499-based dynamic reconfiguration. Terzimehic *et al.* (2017) proposed a control software framework that binds IEC 61499 reconfiguration services with the OPC UA (Open Platform Communications Unified Architecture) to achieve orchestration of functionalities. The paper demonstrated that zero-downtime swapping of FBs can be achieved. Their experiments on an aluminum cold-rolling mill demo system showed that module swap times can be maintained under 200 ms.

Prenzel and Steinhorst (2021) provided a dependency graph-based sequence of a safe automatic reconfiguration solution based on the IEC 61499 standard. Without manual intervention, deadlock-free transitions between application variants can be ensured. The proposed planning algorithm can scale well with the increasing system size and is validated on a packaging plant demo.

Lyu *et al.* (2021) applied satisfiability modulo theories (SMTs) to compute an optimal reconfiguration deployment plan. They show that cloud support can significantly reduce reconfiguration latency in distributed edge nodes. The paper's experiments showed a 40% reduction in average orchestration time from 300 ms to 500 ms, which enables faster adaptation in edge devices.

Banerjee and Choppella (2025) introduced a knowledge-driven reconfiguration framework that leverages ontologies and semantic inference to orchestrate IEC 61499 FBs for large-scale industrial cyber-physical systems. The framework can develop no-code self-adaptation and self-healing capabilities. A smart-grid testbed with over 100 FBs is used to show that recovery from a node failure can be completed within 250 ms.

2.2. Module-type package. The Module Type Package (MTP) is a NAMUR standard that formally describes the interfaces and functions of modular process cell automation technology (VDI, 2020). This standard defines the process automation system as module development and process orchestration with pre-defined modules. A manifest file describes the module organization that can be discovered and directly integrated into the process orchestration level.

Wu and Dai (2023) proposed an automatic code generation method based on ISA 88 and the MTP standard. The paper extended physical model definitions to batch control systems as defined in ISA 88, and added entity classes such as equipment and communication, which can describe a modular device assembled in the

MTP. The code generation for IEC 61499 FBs from MTP modules is implemented.

Köcher *et al.* (2022) investigated the MTP and skill-based modeling approaches in discrete and process manufacturing. The paper identified key semantic and structural mismatches that hinder unified asset integration. An automated mapping algorithm is developed and verified to transform MTP metadata and service definitions into a capability-skilled ontology in the Web Ontology Language (OWL), enabling seamless semantic queries, reasoning, and module reuse across both domains without manual re-engineering.

Bittorf *et al.* (2022) investigated the trade-offs between single-operation and multi-operation MTP service interfaces for downstream processing units. An extraction-column case study evaluated impact on engineering effort, runtime performance, and flexibility. The results showed that coarse-grained services can reduce orchestration latency by 30% and increase the reusability of individual process steps.

2.3. Plug-and-produce modular automation. Naumann *et al.* (2007) introduced a plug-and-play software module gateway that ingests device and process description files in the XML format. Those files can automatically match available hardware, such as robots, grippers, and sensors, with high-level commands. This architecture enables plug-and-play. When a new station is connected, its capabilities are parsed and offered via a unified API, allowing the controller to reschedule tasks dynamically. The gateway is tested on a small-batch production cell and change times can be under 5 s.

Hoffman and Basson (2016) developed a holonic manufacturing architecture implemented entirely on IEC 61131-3 PLCs (International Electrotechnical Commission, 2013). Each holon runs as a separate virtual PLC thread in a Beckhoff TwinCAT PLC. By dynamically instantiating and destroying FBs at runtime and migrating state status via PLC global variables, plug-and-play reconfiguration is achieved for a pick-and-place unit with the average reconfiguration latencies of 300 ms.

We previously proposed a microservice architecture (Dai *et al.*, 2019) in which IEC 61499 FBs can be accessed via HTTP RESTful services and the OPC UA (OPC Foundation, 2017). The orchestration engine queries the ontological knowledge base to discover compatible FBs and compose them at runtime automatically. We demonstrated that new microservice compositions can be achieved under 150 ms on an automotive assembly line.

Overall, the IEC 61499 standard has proved its ability to enable dynamic reconfiguration during runtime operation, and FB networks can be used to orchestrate services. The MTP standard can also increase the reusability of software components and provide a standard interface for discovering and integrating field devices. This paper will bridge both standards to achieve rapid service orchestration for OT-IT convergence industrial cyber-physical system applications to meet customized requirements for flexible manufacturing.

3. Plug-and-produce engineering design based on IEC 61499 and the MTP

In the current engineering design process for industrial automation systems, all software components are pre-defined in the integrated development environment (IDE). Engineers must complete the development process with the vendors' fixed set of libraries. Since functionalities provided by third-party suppliers may need to be updated frequently, it will be very convenient for engineers if updated functionalities can be automatically discovered from field devices and directly integrated into the process design. On the other hand, there is no standardized module interface like the USB for industrial automation systems available yet, which makes it even difficult to maintain all versions for different IDE platforms.

The MTP standard is introduced to solve integration challenges between different vendors for process automation. Produced by the specialist committee named Future Architectures in Automation of the VDI/VDE Society for Measurement and Automatic Engineering, together with NAMUR and the German Electrical and Electronic Manufacturers' Association (ZVEI), the MTP standard defines module specifications as the manifest. It aims to reduce downtimes and the time-to-market for both new and legacy plants.

The MTP manifest file format is based on AutomationML (International Electrotechnical Commission, 2018). As shown in Fig. 2, the core parts of the MTP manifest are services, communication, a human-machine interface (HMI), and a state model. Additional hierarchical levels can be attached to those core parts. For example, the OPC UA server is one of the communication interfaces that can be used to communicate between controllers and field devices. Definitions can be extended freely as long as rules comply with the AutomationML standard.

Once each component (such as valves, pumps, and tanks) is defined as a manifest, the next step is to orchestrate them into a workflow according to the actual procedures. The MTP standard provides a process orchestration layer for service orchestration as illustrated in Fig. 3.

In the POL, modules are integrated into a process flow according to manifests. For example, inputs and outputs from the field devices can be defined as OPC tags, and the HMI screen can be pre-defined and quickly inserted into the supervisory control and data acquisition (SCADA) system. In addition, control algorithms can

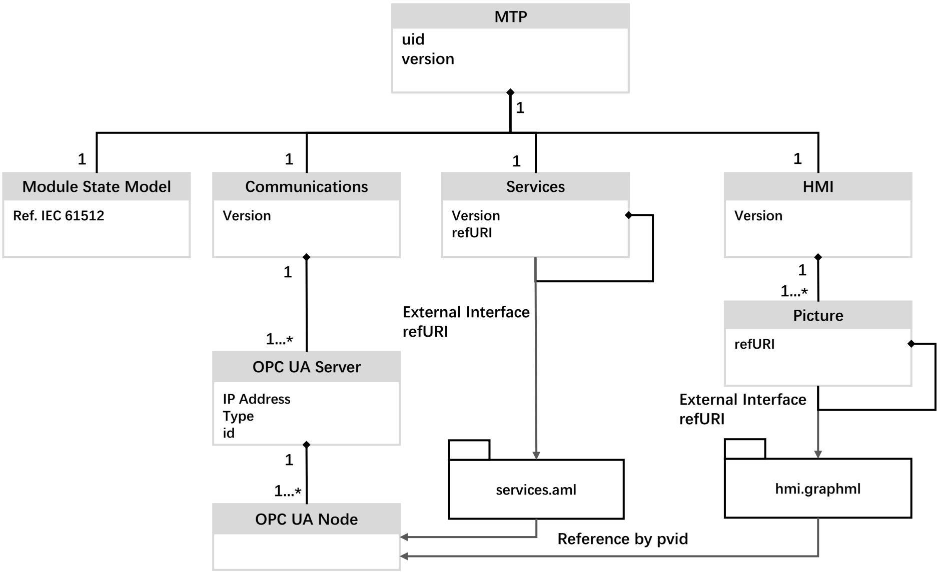


Fig. 2. MTP manifest file definition (NAMUR-2658, 2020).

be exposed to the POL only through the interface to protect intelligence properties. Conversely, services can be updated via the manifest to provide easy over-the-air update (OTA) for site equipment, significantly reducing engineering efforts.

The IEC 61499 standard provides a system-level component-based design for OT-IT convergence industrial edge applications. FB networks can represent service workflow in the low-code development of industrial edge applications. Both IEC 61131-3 functions and high-level programming languages such as C++ and Python can be encapsulated in IEC 61499 FBs since programming languages inside algorithms are not limited.

However, the management model in the IEC 61499 standard did not specify the file format for integrating field devices. Communication to field devices and HMI definitions are only considered as service interface FBs (SIFB) in the IEC 61499 software model without further details. On the other hand, the MTP standard covers all the missing pieces from the IEC 61499 standard. However, how to achieve the POL is not covered.

This paper explores the possibility of extending the boundary of the MTP from the process industry to discrete manufacturing to achieve a universal plug-and-produce (PnP) device for industrial cyber-physical systems, including both discrete manufacturing and process automation. This paper will bridge both standards to provide a complete plug-and-produce procedure for rapidly engineering industrial cyber-physical systems. The IEC 61499 software model is used to implement the MTP standard to achieve PnP devices.

A PnP device $\mathit{Dev}_{\mathrm{pnp}}$ is defined as a tuple of four components based on the MTP definitions including a set of $\mathit{Service}$, a state model M_{state}, a set of communication interfaces $\mathit{CommIntf}$, and a set of HMI:

$$\mathrm{Dev}_{\mathrm{pnp}} = (\mathit{Service},\ M_{\mathrm{state}},\ \mathit{CommIntf},\ \mathit{HMI}).$$

The first component is the service. As defined in the NAMUR 2658-4 modeling of module services, a $\mathit{Service}$ contains a $\mathit{ServiceSet}$, which may include zero or more $\mathit{Service}$, as shown in Fig. 4. In each $\mathit{Service}$, there shall be one or more $\mathit{ServiceProcedure}$.

Rule 1. The IEC 61499 service interface FB type represents the service in MTP:

$$F_{\mathrm{si}} : \mathit{ServiceSet} \to \mathit{SIFB}.$$

Each `ServiceSet` in the MTP service interface is mapped to the `ServiceSequence` in IEC 61499 SIFBs:

$$\forall s \in \mathrm{ServiceSet},\ F_{\mathrm{ss}}(s) \in \mathrm{SSsifb}.$$

Each MTP `ServiceProcedure` is mapped to the `ServiceTransaction` in a `ServiceSequence` of SIFBs:

$$\forall s \in \mathrm{ServiceProcedure},\ F_{\mathrm{sst}}(s) \in \mathrm{SSTsifb}.$$

Input and output parameters used in $\mathit{ServiceProcedure}$ can be listed as input and output primitives in SIFB service transitions. The SIFB may contain algorithms written in high-level programming languages that can be deployed to field devices or as an

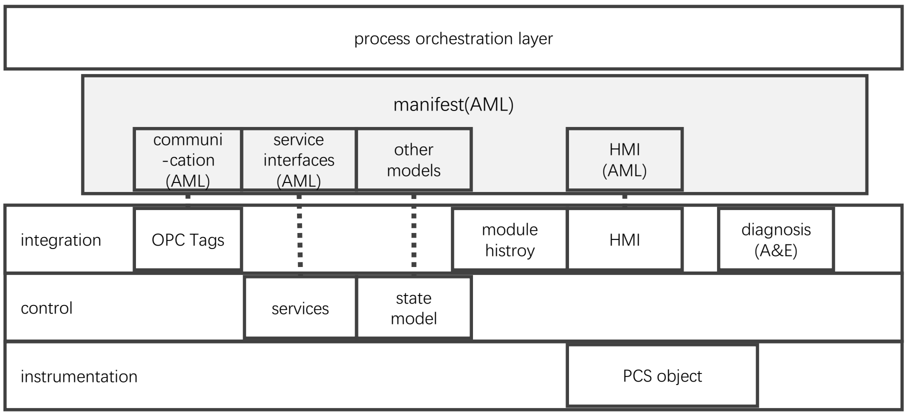


Fig. 3. MTP process orchestration layer definition (NAMUR-2658, 2020).

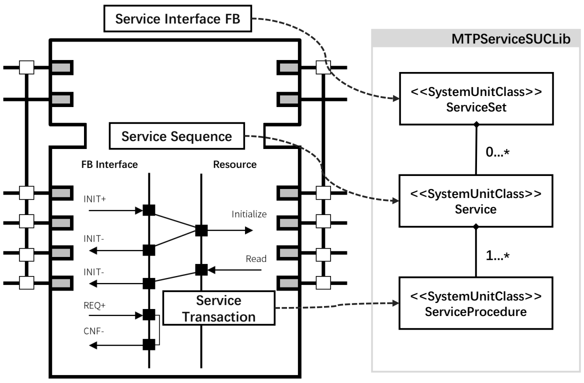


Fig. 4. MTP service in IEC 61499 SIFBs.

invoking interface to the algorithm protected in the field device.

The second component is the state model in the manifest, which maps the IEC 61499 standard to a basic FB (BFB).

Rule 2. The MTP state model is implemented by the execution control chart (ECC) inside an IEC 61499 BFB:

$$F_{\text{state}} : \mathit{ECState} \rightarrow \mathit{ServiceState}.$$

In the MTP, each service consists of a state machine that contains 16 states, including transient (`IDLE`, `EXECUTE`, `HELD`, `PAUSE`, `COMPLETED`, `STOPPED`, `ABORTED`) and non-transient ones (`RESETTING`, `STARTING`, `RESUMING`, `PAUSING`, `UNHOLDING`, `COMPLETING`, `HOLDING`, `STOPPING`, `ABORTING`), as shown in Fig. 5. These states are directly converted into EC ones in the new basic FB:

$$\begin{aligned}\mathit{ServiceState} = \{&\texttt{IDLE, EXECUTE, HELD, PAUSE,}\\ &\texttt{COMPLETED, STOPPED, ABORTED,}\\ &\texttt{RESETTING,STARTING, RESUMING,}\\ &\texttt{PAUSING,UNHOLDING,COMPLETING,}\\ &\texttt{HOLDING, STOPPING,ABORTING}\}.\end{aligned}$$

The POL can send commands to field devices by using one of the following: `Reset`, `Pause`, `Resume`, `Hold`, `Unhold`, `Stop`, `Abort`, and `Restart`,

$$\begin{aligned}\Sigma_{\texttt{command}} = \{&\texttt{Reset, Pause, Resume, Hold,}\\ &\texttt{Unhold, Stop, Abort, Restart}\}.\end{aligned}$$

These commands are used as transition conditions in the ECC. The event input REQ and output CNF are used to read and write I/O values:

$$\delta_{\text{ectran}} : \mathit{ServiceState} \times \Sigma_{\text{command}} \rightarrow \mathit{ServiceState}.$$

This new BFB can be used in the IEC 61499 FB network as the POL for controlling the states of the field devices. The MTP service state control is only applied to process automation systems. To extend the MTP concept to discrete manufacturing, the state machine in the basic FB can be customized for various equipment available from the shop floor.

The third element is the communication interface to field devices. As the MTP states, the communication interface can use the OPC UA or other protocols. In the IEC 61499 version, network segments can be implemented as SIFBs in the FB network. To access values from the network segment SIFBs, adapters can exchange input/output values.

Rule 3. The communication interface $\mathit{CommIntf}$ is modeled as adapters to the IEC 61499 network segment

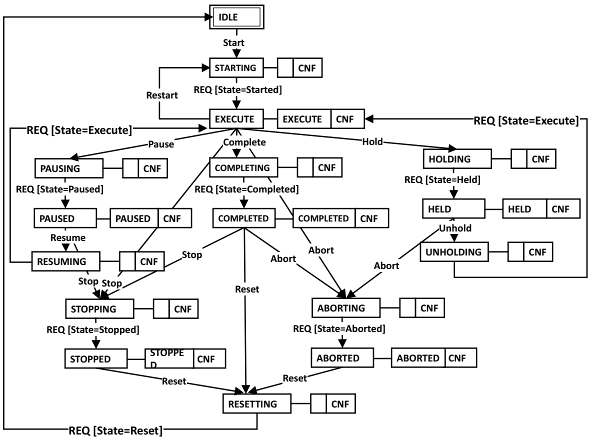


Fig. 5. MTP service state machine in the IEC 61499 ECC.

SIFBs:

$$F_{\text{comm}} : \mathit{ServiceSet} \to \mathit{CommIntf}.$$

The adapter is designed as a group of input InVar and output OutVar variables with their events in the IEC 61499 standard. The MTP communication is also a set of variables, Var, that can be accessed externally:

$$\mathit{Adapter} = (\mathit{InVar}, \mathit{OutVar}),$$

where $\mathit{Var} \in \mathit{InVar} \cup \mathit{OutVar}$.

We can directly map MTP communication $\mathit{DataItems}$ to variables in adapters. To distinguish the access attribute of the $\mathit{DataItems}$, inputs at adapters are set as read-write (write to field devices), and outputs are set as read-only (read back to the POL) for the access:

$$\forall s \in \mathit{Var}, \quad F_{\text{adp}}(s) \in \mathit{DataItem}.$$

Finally, the HMI component needs to be mapped to IEC 61499 FBs. IEC 61499 did not specify detailed descriptions or a dedicated FB type for HMI under the current version. HMI FBs are usually considered to be SIFBs. On the other hand, the MTP standard provides a comprehensive definition of HMI screen elements:

$$F_{\text{hmi}} : \mathit{HMISet} \to \mathit{HMIFB}.$$

As shown in Fig. 7, HMISet is the MTP HMI's root node, containing several $\mathit{Picture}$ with three attributes: Width, Height, $\mathit{HierarchyLevel}$. HMISet denotes the root set of MTP HMI, and $\mathit{Picture}$ denotes the set of all pictures contained within HMISet:

$$\mathit{Picture} \subseteq \mathit{HMISet},$$

$$\mathit{Picture} = (\text{Width}, \text{Height}, \text{HierarchyLevel}).$$

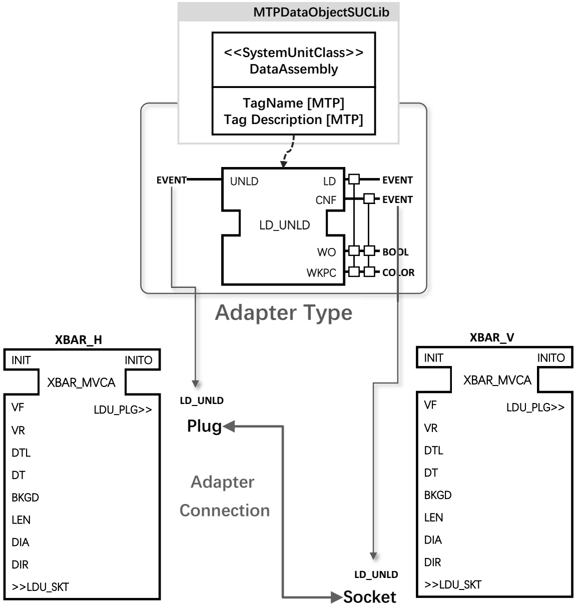


Fig. 6. Flexible MTP communication interface in IEC 61499 adapters.

Each picture p has the following attributes: Width$(p) \in \mathbb{R}^+$, Height$(p) \in \mathbb{R}^+$, and HierarchyLevel$(p) \in \mathbb{N}$. Each picture also contains a subset of visual objects:

$$\mathit{VisualObject}(p) \subseteq \mathit{VisualObject}.$$

Each visual object $v \in \mathit{VisualObject}$ has the following attributes: Width$(v) \in \mathbb{R}^+$, Height$(v) \in \mathbb{R}^+$, $X(v) \in \mathbb{R}$, $\mathit{ZIndex}(v) \in \mathbb{Z}$, Rotation$(v) \in \mathbb{R}$, and ViewType$(v) \in \mathit{ViewTypeSet}$. Each visual object $v \in \mathit{VisualObject}$ has associated ports:

$$\mathit{PortObject}(v) \subseteq \mathit{PortObject}.$$

A connection can be made between two ports:

$$\mathit{Connection} \subseteq \mathit{PortObject} \times \mathit{PortObject}.$$

A topology object is required to present a junction or a termination for multiple ports: $\forall t \in \mathit{TopologyObject}$, $\exists P_t \subseteq \mathit{PortObject}$ such that $|P_t| \geq 2$.

The new HMI FB type extends the existing definitions of IEC 61499 to provide a standardized way to describe HMI screens. The XML scheme for the HMI FB type is defined in Fig. 7 according to the MTP HMI. An HMI FB may contain one or more pictures. The attribute $\mathit{HierarchyLevel}$ indicates the home screen (as $= 0$) or popup windows (> 0). $\mathit{VisualObject}$ can be used for a button, a textbox, a pump, a valve, etc. Each $\mathit{VisualObject}$ may bind variables on the FB interface to provide animations using the parameter tag in the XML definition.

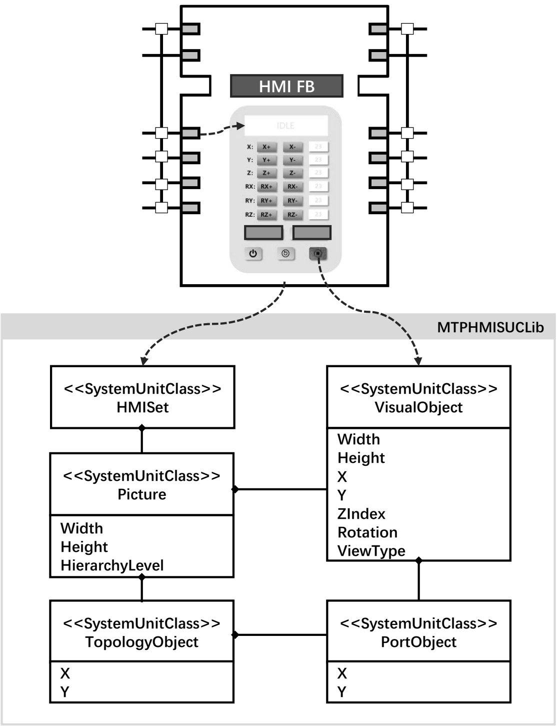


Fig. 7. Extended IEC 61499 HMI FB type based on MTP HMI definitions (NAMUR-2658, 2020).

Hence, all major MTP elements are mapped to IEC 61499 FBs. The next step is to use these definitions to integrate field devices rapidly.

4. Plug-and-produce service orchestration with the MTP and IEC 61499

Industry vendors like Beckhoff (Beckhoff Automation, 2025) and Siemens (Siemens, 2024) are developing dedicated runtimes to support MTP standards in process automation. However, the MTP runtime must be stopped before the POL is updated in the engineering environment. For the safety-critical domain in process automation, like hot rolling in steel-making, stopping the manufacturing line to deploy the new orchestration plan will cause several days of downtime and massive restart costs, which is not acceptable. With the combined IEC 61499 and MTP model, a unique runtime can be used for field devices and automation controllers. In addition, the dynamic reconfiguration of IEC 61499 FBs gives software-defined ability to field devices.

However, there are some restrictions for dynamic reconfiguration for process automation. The reconfiguration actions must be completed before the minimum repetition time for each process, unless an individual process stage can be decoupled and stopped without affecting the entire continuous manufacturing process. Dynamic reconfiguration for a fully continuous process without any decoupling points is feasible only if all modifications can be completed less than the minimum repetition time of the process, subtracting the last execution time. During the MTP POL reconfiguration, FBs will be switched to idle mode to prevent event triggering.

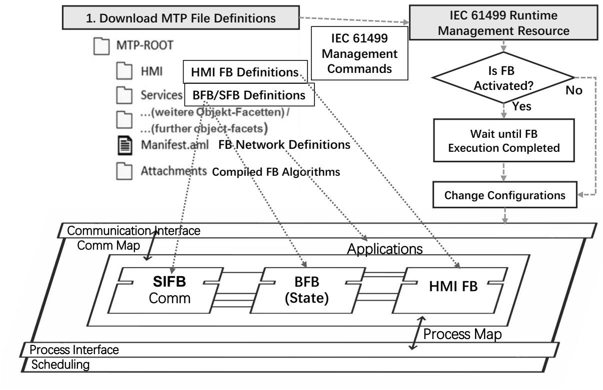


Fig. 8. Packaging of MTPs for IEC 61499 deployment (NAMUR-2658, 2020).

In the IEC 61499 standard, runtime reconfiguration is classified into three levels:

- level 0: only connections between FB instances and parameters can be modified;
- level 1: FB instances, connections, and parameters can be dynamically reconfigured;
- level 2: all elements, including FB types and data types and resources, can be dynamically reconfigured.

We extend the IEC 61499 runtime to support MTP features rather than having a separate MTP runtime. With the level 2 dynamic reconfiguration abilities provided by the IEC 61499 standard, MTP files can be integrated with the entire process at design time and downloaded to the runtime. This provides a feasible solution to integrate MTP services written in different high-level programming languages with IEC 61131-3 controllers by adopting the generic FB interface.

As defined in Fig. 8, in the MTP file format (`*.mtp` or `*.zip`), the `Manifest.aml` is the table of contents describing the three folders: `HMI`, `services`, and `Attachments`. Our approach will place HMI FB type definitions in the `HMI` folder as `*.fbt`. All other FB types shall be placed in the `Services` folder as `*.fbt`. All related library files, including dynamic link files or other executable services, can be placed in `Attachments`. If the field device has limited

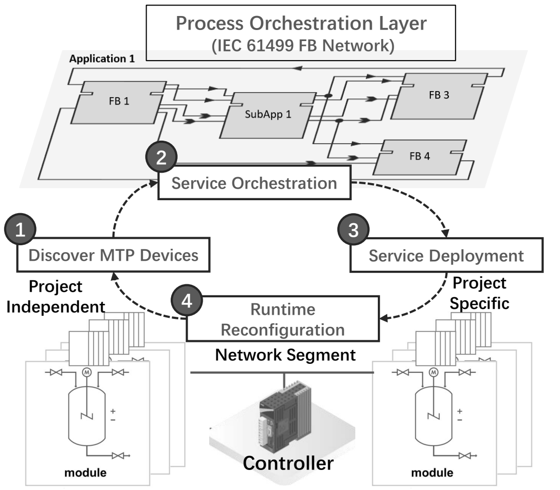


Fig. 9. IEC 61499-based MTP design process (NAMUR-2658, 2020).

resources and computing power, level 0 or level 1 reconfiguration shall be applied, allowing only updating parameter changes and process flows. FB types can also be updated if the field device has enough resources. Such FB types can be service interface FBs used for hardware drivers, including sensors, valves, motors, or computational tasks, such as calibration, data acquisition, etc.

The reconfiguration process starts with the deployment of MTP files to the runtime. For workflow changes (IEC 61499 level 0 reconfiguration) and data interface changes (IEC 61499 level 1 reconfiguration), only the `*.fbt` will be sent to the runtime. However, files compiled from new FB algorithms must be placed in the Attachments section, to deploy new functionalities for MTP devices, and transferred to the runtime.

The management command can transfer the MTP file to IEC 61499-based MTP runtimes for field devices. Each MTP node is considered a `Resource` type in the IEC 61499 that can be mapped to any device in the deployment model. As shown in Fig. 8, the MTP file is attached as the internal contents of the resource type that can be used in the create and query actions when downloading and uploading the MTP file to/from the runtime.

The next step is interpreting MTP files in the IEC 61499 runtime. Management commands will be executed in sequence by the management resource in the IEC 61499 runtime to update FB instances, parameters, or type definitions according to the MTP file definition. If the current FB instance is in running mode, this command will be queued temporarily and wait until the execution is completed. To update an FB type, the runtime must ensure no instance of this FB type is activated. Events triggered to those FB instances shall be paused until the update process is completed.

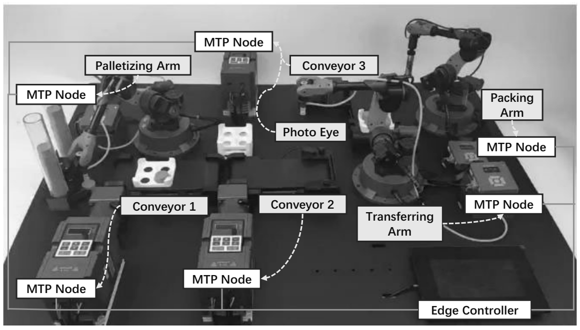


Fig. 10. Packaging demo system layout.

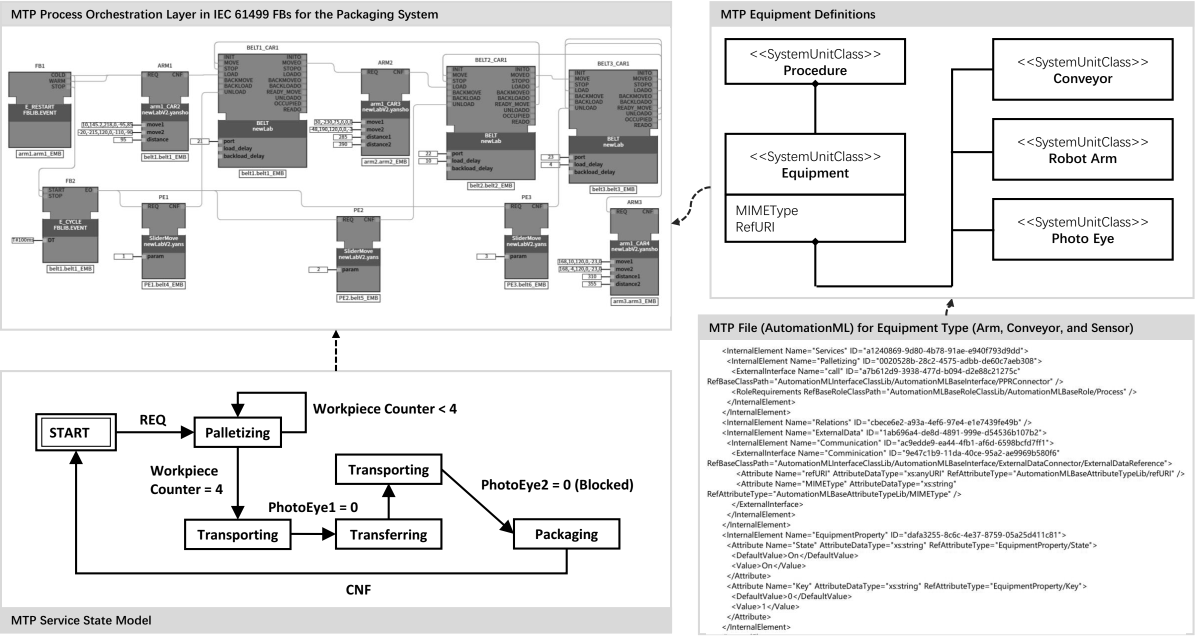
Enhancing the IEC 61499 standard with MTP support, the engineering design process can be described as a closed-loop one, as shown in Fig. 9.

MTP files provided by sensor and drive vendors can be imported into the IEC 61499 FB network design. Engineers can create FB instances by dragging FBs from imported MTP definitions and placing them in order according to the system process. The FB network can be deployed to controllers and MTP-enabled field devices simultaneously. SIFBs are proxy interfaces that exchange input and output values between controllers and field devices. These FBs can be the OPC UA or other communication protocols that can link to field devices as given in the MTP definitions. As a result, the service workflow can be automatically bridged between field devices and controllers.

As shown in Algorithm 1, the PnP algorithm for MTP deployment can be divided into three steps. The first one is to discover all available MTP devices and compare all definitions from the MTP runtime. In the second step, adapters and deployed FB instances on each field device will automatically be added, updated, or deleted according to the FB network of the POL. The final step is to create communication interfaces between controllers and MTP runtimes on field devices by using publish-subscribe SIFBs over the OPC UA or other protocols based on the MTP definitions. By applying this approach, the plug-and-produce software components can be obtained.

5. Case study and discussions

A small packaging system is used to validate the concept of combining IEC 61499 and the MTP for rapid industrial cyber-physical systems prototyping. The packaging system consists of three robotic arms and three conveyor belts. The conveyors transport products (workpieces) to be packed from the in-fill workstation to the docking workstation, as shown in Fig. 10. The three robotic



```
    <InternalElement Name="Services" ID="a1240869-9d80-4b78-91ae-e940f793d9dd">
     <InternalElement Name="Palletizing" ID="0020528b-28c2-4575-adbb-de60c7aeb308">
      <ExternalInterface Name="call" ID="a7b612d9-3938-477d-b094-d2e88c21275c"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/PPRConnector" />
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Process" />
     </InternalElement>
    </InternalElement>
    <InternalElement Name="Relations" ID="cbece6e2-a93a-4ef6-97e4-e1e7439fe49b" />
    <InternalElement Name="ExternalData" ID="1ab696a4-de8d-4891-999e-d54536b107b2">
     <InternalElement Name="Communication" ID="ac9edde9-ea44-4fb1-af6d-6598bcfd7ff1">
      <ExternalInterface Name="Comminication" ID="9e47c1b9-11da-40ce-95a2-ae9969b580f6"
RefBaseClassPath="AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector/ExternalDataReference">
       <Attribute Name="refURI" AttributeDataType="xs:anyURI" RefAttributeType="AutomationMLBaseAttributeTypeLib/refURI" />
       <Attribute Name="MIMEType" AttributeDataType="xs:string"
RefAttributeType="AutomationMLBaseAttributeTypeLib/MIMEType" />
      </ExternalInterface>
     </InternalElement>
    </InternalElement>
    <InternalElement Name="EquipmentProperty" ID="dafa3255-8c6c-4e37-8759-05a25d411c81">
     <Attribute Name="State" AttributeDataType="xs:string" RefAttributeType="EquipmentProperty/State">
      <DefaultValue>On</DefaultValue>
      <Value>On</Value>
     </Attribute>
     <Attribute Name="Key" AttributeDataType="xs:string" RefAttributeType="EquipmentProperty/Key">
      <DefaultValue>0</DefaultValue>
      <Value>1</Value>
     </Attribute>
    </InternalElement>
```

Fig. 11. Equipment and procedure control model of the MTP.

Algorithm 1. Plug-and-produce algorithm for deployment.

```
$ServiceSet \leftarrow QueryRuntime()$
$updateList \leftarrow \emptyset$
for each $Service$ in $ServiceSet$ do
    if not $Exist(Service, ServiceSet)$ then
        $updateList \leftarrow updateList \cup \{Service\}$
    end if
end for
for each $Service$ in $downloadList$ do
    $SendMGTCommand("Create", Service)$
    // <Request ID="1"
    Action="Create"><FB
    Name="Service.Name" /></Request>
end for
```

arms are responsible for the palletizing, transferring, and packing process. A small controller with the ARM Cortex-A7+ CPU is equipped with each conveyor, robotic arm, and proximity sensor to simulate MTP nodes. The IEC 61499 runtime FBSRT (Yueyi Automation, 2025) with MTP file format support is installed on each controller. A central controller with the ARM Cortex-A53+ CPU coordinates all MTP controllers. All controllers are connected via the standard Ethernet.

The MTP service model design is presented in Fig. 11. The entire system consists of eight MTP service FBs: three for conveyors, three for robotic arms, and two for photoelectronic (PE) sensors. The state model of the robotic arm is designed as a three-state ECC model: start-pick-place. The conveyor control is designed as an ECC with only several states: idle and run. The conveyor will switch back from the run state to the idle state if a workpiece blocks the attached PE sensor. Finally, the POL workflow FB design is given as an event chain of FBs, including all MTP service FBs, the POL state machine FB, and other communication interface FBs such as IEC 61499 publish/subscribe SIFBs, to exchange input and output values between the main controller and all MTP ones.

This case study covers two experiments. Firstly, the system development time with the same engineer is measured by comparing the bottom-up and top-down approaches. In the former, FBs are fetched from the MTP manifest files and imported into the IDE. The engineer must complete the service orchestration by connecting all events and data variables. In the latter approach, the engineer must manually create the FB instances according to the workflow. From the results, we can find out that the development time for the bottom-up approach is only 7 minutes and 20 seconds, compared to the top-down one, which uses 15 minutes and 40 seconds, i.e., a roughly 50% time reduction.

The second test is to measure the functionality reconfiguration time of the IEC 61499-based MTP runtime for field devices. The reconfiguration process is performed by modifying the POL of the IEC 61499 FB network in the engineering environment and dynamically deploying new system configurations to the field device runtime online. Since the real-time constraint is one of the most important indices for industrial automation systems, this test is set to ensure that online functionality changes

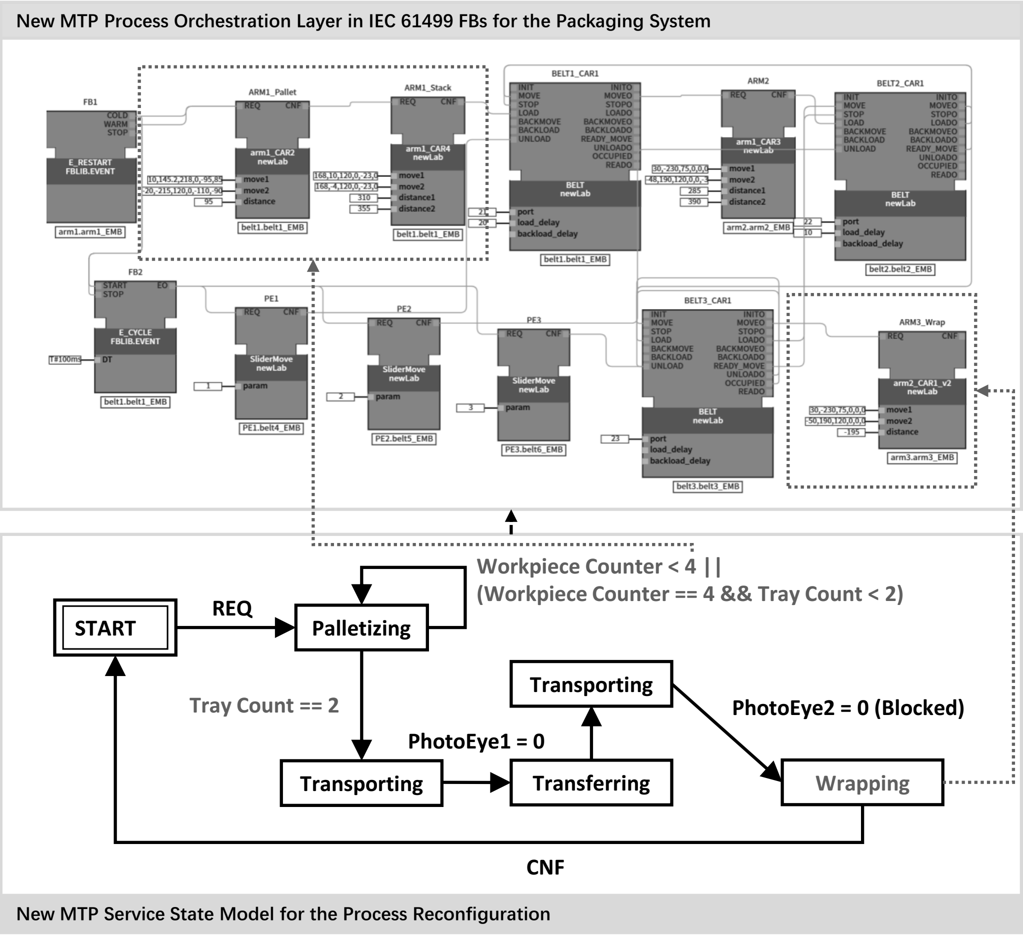


Fig. 12. POL reconfiguration for the packaging system.

will still meet the real-time constraints.

The POL reconfiguration for the packaging system is described here. As shown in Fig. 12, the original workflow starts with palletizing four workpieces on a tray by the first robotic arm and running towards the end of Conveyor 2. Then the transferring arm will pick the tray and place it onto Conveyor 3. Finally, the packing arm will stack the trays together. The original FB chain is designed according to this workflow, where each step is intended as a separate FB.

To simulate the customization of the manufacturing process, the reconfiguration of the FB network is designed as follows: the first robotic arm will repeat palletizing twice and stack two trays together, before moving them to the packaging station. The packaging will add a new action for wrapping the trap before unloading trays. The new MTP service state model is given in Fig. 13, and the new FB network is provided.

To replace the existing FB network in Fig. 11 with the new one in Fig. 12, the MTP-based deployment is achieved by sending only the MTP file to all nine controllers. In the original approach, at least three management commands are needed for each controller to create the new deployment by cleaning all existing configurations, deploying new FB types, and creating new FBIs. As shown in Figs. 13 and 14, the worst case of the reconfiguration time in 50 times can be controlled under 96 ms (22 ms maximum download time and 74 ms maximum reconfiguration time for executing the MTP file). The traditional IEC 61499 deployment method uses management commands to create all elements individually. The worst case of the reconfiguration time in 50 times is 504 ms. By reducing the number of management command messages in the MTP-based approach, the reconfiguration time can be saved by 80%.

Table 1. Engineering design time.

Method	Design time
Bottom-up (MTP)	7 mins 20 secs
Top-down (manual)	15 mins 40 secs

Test results show that functionalities can

Table 2. System deployment and reconfig time.

Method	Download [ms]	Reconfig [ms]
MTP file	22	74
IEC61499 MGT	127	504

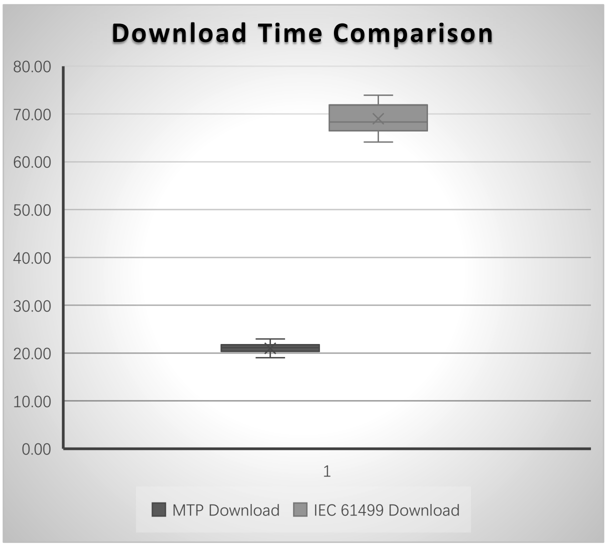


Fig. 13. Download time of IEC 61499 management commands and the MTP.

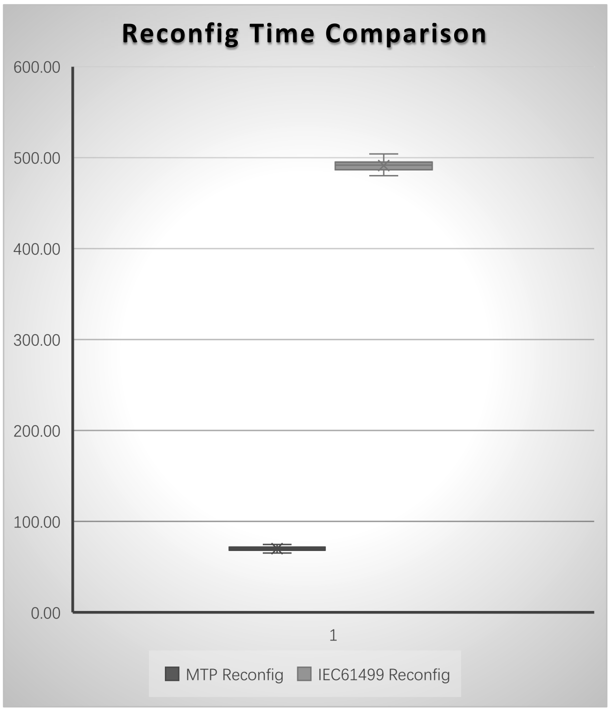


Fig. 14. Reconfiguration time of IEC 61499 management commands and the MTP.

be dynamically updated to field devices using combined IEC 61499 and MTP standards without interrupting manufacturing operations. Based on those software-defined field devices, online process optimization can be achieved based on real-time monitoring, simulation, and validation to enhance production quality. In addition, the downtime can be avoided, and production time can be reduced by automatically generating new solutions and updating components from directly connected industrial AI agents and AI models.

6. Conclusions and future work

This paper presented a rapid integration method between controllers and field devices for industrial cyber-physical systems. The IEC 61499 FB model was enhanced with the MTP standard to achieve discovering and reconfiguring plug-and-produce software components in a closed-loop design process. The core elements of the MTP were mapped to their FB versions, and the FB network was used as the process orchestration layer for the MTP. The generic FB interface design provided great interoperability between industrial controllers and field devices. Also, the MTP file format was used as the standard for all equipment, which improved design portability. The case study shows that the reconfiguration and deployment time can be significantly reduced by using one unique runtime for both MTP and IEC 61499.

Within future work, the scalability of the proposed approach needs to be investigated to support next-generation virtualized edge controllers for industrial automation systems. To validate the system behavior with integrated functionalities of field devices, contracts will be introduced to provide formal constraints and automatic validation for each MTP-enabled device. Finally, the industrial AI agent can be applied to MTP-enabled field devices to achieve software-defined automation on the shop floor.

Acknowledgment

This research work is sponsored by the National Natural Science Foundation of China (project no. 92467301) and the Shandong Province Research and Development Key Program (project no. 2025CXPT073).

References

Banerjee, A. and Choppella, V. (2025). A knowledge-driven approach for dynamic reconfiguration of control design in Internet of things and cyber-physical systems, *IEEE Internet of Things Journal* **12**(5): 5615–5641.

Beckhoff Automation (2025). TwinCAT MTP: Modular automation with Type Module Package, *Type Release 1.00.*, Beckhoff Automation, Hülshorstweg, `https://w` `ww.beckhoff.com/en-en/products/automation/module-type-package-mtp/`.

Bittorf, L. Oeing, J., Kock, T., Garreis, R. and Kockmann, N. (2022). Design of Module Type Package services for modular downstream units and process analytic technology, *Chemie-Ingenieur-Technik* **94**(9): 1217.

Dai, W., Huang, W. and Vyatkin, V. (2019). Semantic integration of plug-and-play software components for industrial edges based on microservices, *IEEE Access* **7**: 125882–125892.

Fu, X. and Wang, X. (2025). Fault detection based on interval observers and dynamic event-triggered fault-tolerant control for morphing aircraft, *International Journal of Applied Mathematics and Computer Science* **35**(3): 429–442, DOI: 10.61822/amcs-2025-0030.

Hoffman, A.J. and Basson, A.H. (2016). IEC 61131-3-based holonic control of a reconfigurable manufacturing subsystem, *International Journal of Computer Integrated Manufacturing* **29**(5): 520–534.

International Electrotechnical Commission (2012). *Function blocks- Part 1: Architecture, International Standard IEC 61499-1* International Electrotechnical Commission, Geneva, `https://webstore.iec.ch/en/publication/5506`.

Industry IoT Consortium (2022). The industrial Internet reference architecture (IIRA), *Technical Report IIC:TC:IN:V1.10*, Industry IoT Consortium, New York, `https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf`.

International Electrotechnical Commission (2018). IEC 62714-1: Engineering data exchange format for use in Industrial automation systems engineering—Automation Markup Language. Part 1: Architecture and general requirements, *Iternational Standard*, International Electrotechnical Commission, Geneva, `https://webstore.iec.ch/en/publication/32339`.

International Society of Automation (2010). Enterprise-control system integration. Part 1: Models and terminology, *ANSI/ISA-95.00.01-2010 (IEC 62264-1 Mod), American National Standard*, International Society of Automation, Research Triangle Park, `https://www.isa.org/standards-and-publications/isa-standards/isa-95-standard`.

International Electrotechnical Commission (2013). Programmable Controllers—Part 3: Programming Languages, *International Standard IEC 61131-3*, International Electrotechnical Commission, Geneva, `https://webstore.iec.ch/en/publication/68533`.

Klimek, R. (2014). A system for deduction-based formal verification of workflow-oriented software models, *International Journal of Applied Mathematics and Computer Science* **24**(4): 941–956, DOI: 10.2478/amcs-2014-0069.

Köcher, A., Beers L. and Fay, A. (2022). A mapping approach to convert MTPS into a capability and skill ontology, *Proceedings of the 27th International Conference on Emerging Technologies and Factory Automation (ETFA), Palermo, Italy*, pp. 1–8.

Lyu, T., Blech, J.O. and Vyatkin, V. (2021). SMT-based deployment calculation for IEC 61499 control applications, *Proceedings of the IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), Valencia, Spain*, pp. 172–178.

Naumann, M., Wegener, K. and Schraft, R.D. (2007). Control architecture for robot cells to enable plug'n'produce, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy*, pp. 287–292.

OPC Foundation (2017). *OPC 10000-1: Unified Architecture. Part 1: Overview and Concepts*, Release 1.04, OPC Foundation, Wilmington, `https://reference.opcfoundation.org/specs/OPC-10000-1/v1.04`.

Prenzel, L. and Steinhorst, S. (2021). Automated dependency resolution for dynamic reconfiguration of IEC 61499, *Proceedings of the IEEE Emerging Technologies & Factory Automation (ETFA), Frankfurt, Germany,* pp. 305–312.

Ragot, J., Mourot, G. and Kallas, M. (2022). A data driven fault isolation method based on reference faulty situations with application to a nonlinear chemical process, *International Journal of Applied Mathematics and Computer Science* **32**(4): 635–655, DOI: 10.34768/amcs-2022-0044.

Scheer, A.-W. (2015). Industry 4.0: From vision to implementation, *Whitepaper No. 5*, AWS Institute for Digital Products and Processes GmbH, Saarbrücken, `https://www.scheer-ids.com/fileadmin/scheer/Dokumente/Whitepaper/Whitepaper-Industry-4-0-Implementation.pdf`.

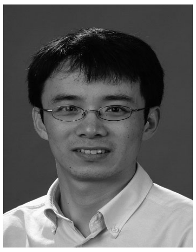
Siemens (2024). *SIMATIC: STEP 7 Professional/WinCC Unified SIMATIC MTP Integrator for WinCC Unified*, V1.0, Siemens, Nuremburg, Germany, `https://cache.industry.siemens.com/dl/files/132/109827132/att_1277113/v1/ReadMeMTPPOLEngMODAUTenUS.pdf`.

Terzimehic, T., Wenger, M., Zoitl, A., Bayha, A., Becker, K., Müller, T. and Schauerte, H. (2017). Towards an Industry 4.0 compliant control software architecture using IEC 61499 & OPC UA, *Proceedings of the IEEE International Conference on Emerging Technologies & Factory Automation (ETFA), Limassol, Cyprus,* pp. 1–4.

Wu, X. and Dai, W. (2023). A process orchestration and deployment method for industrial edge applications based on IEC 61499 and MTP, *Proceedings of the 49th Annual Conference of the IEEE Industrial Electronics Society (IECON), Singapore*, pp. 1–6.

VDI (2020). VDI/VDE/NAMUR 2658 Blatt 4: Automation engineering of modular systems in the process industry: Modelling of module services, Düsseldorf, `https://www.vdi.de/en/home/vdi-standards/details/vdivdenamur-2658-blatt-4-automation-engineering-of-modular-systems-in-the-process-industry-modelling-of-module-services`.

Yueyi Automation (2025). IEC 61499 Integrated Development Environment and Runtime: Function Block Builder and Function Block Service Runtime, Yueyi Automation, Shanghai, `http://www.iec61499.cn`.



Dan Huang received his BS degree in thermal and power engineering from Shanghai Jiao Tong University, China, in 2002, and his MS and PhD degrees in electrical and computer engineering from the University of Auckland, New Zealand, in 2004 and 2008, respectively. In 2009, he was a lecturer with the Department of Electrical and Computer Engineering, University of Auckland. In 2010, he joined Shanghai Jiao Tong University. He is currently an associate professor of system and information control with the School of Aeronautics and Astronautics there. His present research interests include systems engineering, functional modeling, and aviation human factors.

Likuan Zhang is a senior engineer, serving as a product director at Beijing Urban Construction Intelligent Control Co., Ltd. China. He received his master's degree in software engineering from the Renmin University of China in 2014, and his MBA degree from the Corvinus University of Budapest, Hungary, in 2020. He joined Beijing Urban Construction Intelligent Control Co., Ltd. in 2023. His research interests are industrial edge computing and real-time control systems.

Chuanyang Yu is a product manager at Beijing Urban Construction Intelligent Control Co., Ltd., specializing in smart city solutions, and has been engaged in the Internet of things industry for a decade. He offers some unique insights in the field of industrial automation control.

Hui Zhang is an expert in intelligent control for urban construction, specializing in smart cities and intelligent transportation. With years of industry experience, he has led multiple technological innovations, significantly advancing the intelligence of urban transportation. Due to his outstanding leadership and keen market insight, H. Zhang has successfully guided his team through challenges and has implemented numerous projects, making valuable contributions to smart city construction.

Wenbin Dai is a professor at the Department of Automation, Shanghai Jiao Tong University, China, which he joined in 2014. He received his PhD degree in electrical and electronic engineering at the Department of Electrical and Computer Engineering, University of Auckland, New Zealand, in 2012. His research interests are IEC 61499 function blocks, industrial cyber-physical systems, automatic code generation and industrial edge computing.

Received: 30 June 2025
Revised: 26 October 2025
Re-revised: 4 December 2025
Accepted: 5 February 2026