NAMUR – User Association of Automation Technology in Process Industries

WG POSITION

State Model Alignment of ISA 88 and Module Type Package

In this position you find information on:

- **Seamless Integration**: The MTP standard in combination with ISA 88 batch systems enhances “plug and produce” capabilities that streamline integration across diverse modular components.
- **Enhanced Orchestration**: Process Orchestration Layers (POL) coordinating modular Process Equipment Assemblies (PEA) can optimize the execution of complex batch recipes.
- **State Model Alignment**: Alignment between MTP and ISA 88 state models enables consistent command and state management for improved operational efficiency.
- **Future-Proof Solutions**: Position your plant for success in both brownfield and greenfield scenarios by leveraging the aligned capabilities of MTP and ISA 88, ensuring scalable and adaptable automation solutions.

Status: 2024-09-09

WG 2.3 BATCH Control, WG 2.12 POL, WG 2.4.2 MES
in cooperation with ISA 88 WG2 and PNO JWG MTP Core Specification

Authors:
Joe Alford (ISA 88)
Dave Board (ISA 88), Yokogawa
Christian Bramsiepe (NAMUR), Evonik
Dennis Brandl (ISA 88)
Nirav Chokshi (ISA 88), Fluor
Henry Daigle (ISA 88)
Christie Deitz (ISA 88), Emerson
Lieven Dubois (ISA 88), Lieven Dubois Consulenza
Dave Emerson (ISA 88), Yokogawa
Guido Fischer (NAMUR), Boehringer Ingelheim
Douglas Fisher (ISA 88), Rockwell Automation
Alexander Fröhlich, COPA-DATA
Charlie Fu (ISA 88)
Harry Gianakis (ISA 88), Dupont
Charlie Gifford (ISA 88)
Lukas Grahl (NAMUR), Semodia
Allen Hatcher (ISA 88), Honeywell
Terry Douglas Jayasuriya (ISA 88), TDJ Consultancy Group
Bruce Kane (ISA 88)
Ronald Kleber (ISA 88), Honeywell
Willem de Kruijf (ISA 88), Yokogawa
Mathias Maurmaier (PNO), Siemens
Ernst Matthies (NAMUR), BASF
Karl Mocha (NAMUR), Bayer
Chris Monchinski (ISA 88)
Mohamed Moujahed (ISA 88), Siemens
Christoph Münch (NAMUR), Spiratec
Herbert Oberauer (PNO), COPA-DATA
Svj Pawa-Kumar (ISA 88)
André Pomraenke (NAMUR), Fraunhofer IFF
Christopher Rocker (NAMUR), Spiratec
Günther Schätzle (NAMUR), CHT
Andreas Stutz (PNO), Siemens
Kazuo Sueyoshi (ISA 88), Yokogawa
Derrick Tapscott (ISA 88, PNO), Rockwell Automation
Thomas Tauchnitz (NAMUR), Tautomation
Rüdiger Trobisch (ISA 88)
Jean Vieille (ISA 88)

WG Managers: Anselm Klose (NAMUR, ISA 88, PNO), Process-to-Order Lab TU Dresden
Christian Schäfer (NAMUR), Merck
Nils Richter (NAMUR), BASF

WG POSITIONS reflects the experience gathered by the listed authors and has been agreed within the Working Group. This document does not enjoy the same degree of consensus as a NAMUR Recommendation or NAMUR Worksheet. By publishing an WG POSITION, the Working Group can almost immediately inform interested readers of their own experience.

Foreword

Established in 1995, ISA 88 (ANSI/ISA-88.00.01-2010), along with its adaptation to an international standard IEC 61512, is the de facto model for batch systems. Thousands of plants follow ISA 88 principles in their implementation. The standard intentionally leaves flexibility for adaptation to specific use cases. The *Module Type Package* (MTP) is an emerging standard for interfacing automation systems, equipment, and procedural services. It specifies how modularized process units, so-called *Process Equipment Assemblies* (PEA) or "modules" are orchestrated by superordinated systems called *Process Orchestration Layers* (POL). As MTP is driven by the desire to facilitate integration of (any sort of) modular components it naturally includes use cases with and without relation to other than batch applications. Therefore, the two standards have overlapped, but originated from different needs. MTP can be implemented to be fully in line with the ISA 88 standards. This paper recommends an approach for implementing MTP PEAs with a batch system based on the ISA 88 model.

This is a position paper which is based on currently released guidelines, namely VDI VDE NAMUR 2658-4 2022, VDI 2776-1 2020, ANSI/ISA-88.00.01-2010, and IEC 61512. This paper represents the opinion of the listed NAMUR, ISA and PNO members.

1 Intro on MTP and ISA 88

MTP and ISA 88 both address structure and integration of equipment on many levels. However, MTP specifically addresses “plug and produce” scenarios for both continuous and batch processes. The standard allows for multiple applications, ranging from interfacing of single components up to potentially interfacing complete process plants or batch systems. We present how an implementation of MTP compliant equipment can be made to align with the exemplary state transition diagram laid out in ISA 88 for procedural elements to allow easy integration in existing batch systems.

To begin such an alignment, some fundamental concepts of ISA 88 and MTP need to be described.

In batch control according to ISA 88 there are two major vantage points. One is that of the process which must be followed to make a product, and the other is that of the equipment which must have suitable capabilities to accomplish processing tasks. In terms of process, we reference ISA 88’s *Procedural Control Model* in this paper. The *Procedural Control Model* has the same hierarchical composition as the ISA 88 *Process Model* but is more applicable in this context because of its equipment dependence. It describes the process for a specific piece, or class of equipment. The resulting equipment dependent recipe *Procedures* (master recipes) are defined using the hierarchical levels of the *Procedural Control Model*.

In ISA 88 the recipe procedural elements (*Procedures*, *Unit Procedures*, *Operations*, and *Phases*) sequence the equipment control of the equipment entities (*Process Cells*, *Units* and *Equipment Modules*). The equipment entities provide the capability to execute tasks ‘assigned’ to them. MTP works similarly: the POL orchestrates the *service* execution of the PEAs, and the PEAs provide the processing capability. In the kitchen at home, a cookbook – containing recipes made up of parameters and sequences – defines the orchestration, and the kitchen containing the equipment and human cook provides the capabilities. In many cases the person preparing the food provides equipment control – the person in combination with the equipment is then the equipment entity. A bread maker may, however, have a program of its own, one that it executes through when called for. The bread maker’s program is a good example of equipment control in ISA 88, or a *service* in MTP.

Fully modular MTP systems have a POL, which orchestrates individual *services* of PEAs. The logic of the *services* is executed on the controller of the PEA and is designed in advance, independent from the POL. For interaction, the POL sends commands to the PEA and reads the respective states and state changes in the PEA *services*. *Services* of PEAs can be combined to form recipes.

In brownfield sites, newly introduced PEAs and proprietary units are likely operated by existing ISA 88 based batch systems, which will act as POL in MTP terms. Here, the usage of MTP and its *services* within the ISA 88 structure needs to be defined.

To use MTPs in an ISA 88 based batch system, the control-recipe execution system should take the role of orchestrator and interact with PEAs by sending commands and expecting respective state changes (see Figure 1). The challenge here is to make a PEA ISA 88 compliant, to support seamless interaction with different batch systems.

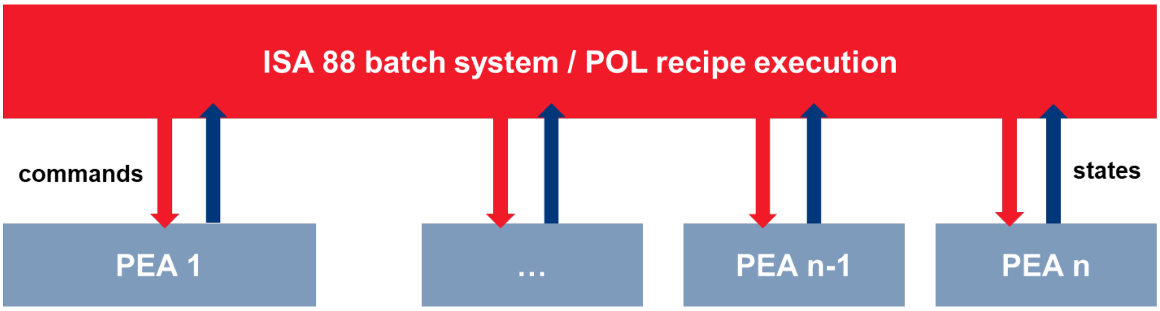


Figure 1: Interaction of batch system and PEAs

In alignment of ISA 88 and MTP, the key elements to be examined are on the interface between recipe (orchestration layer in MTP), and capabilities of PEAs encapsulated as *services* (engineered equipment entity in ISA 88). First, due to the hierarchical nature of the applicable ISA 88 models, it is necessary to consider the levels at which the linking of the control recipe and MTP services or PEA might occur, and second, the state-related information and commands exchanged via this *service* interface shall be taken into consideration.

2 Comparison of MTP and ISA 88 Models

The structures as proposed by ISA 88 describe a decoupling of recipe and equipment for batch systems. The control recipe structured according to the *Procedural Control Model* would be similar to such functionality as provided by the POL. The *Procedural Control Model* is structured in four hierarchical levels: *Procedures*, *Unit* P*rocedures*, *Operations*, and *Phases*. For equipment, a similar 4-layer hierarchy known as the *Equipment Entity Model* exists in ISA 88. It is decomposable into *Process Cell Entities*, *Unit Entities*, *Equipment Module Entities* and *Control Module Entities*. These nested equipment entities are formed from the combination of equipment control and physical equipment.

For the chosen scenario of a batch system orchestrating one or multiple PEAs, MTP as an automation interface and the respective plant structure for modular plants can be aligned to these structures of ISA 88; most importantly to the *Equipment Entity Model* and the *Procedural Control Model*. With the approach of MTP to encapsulate complexity, PEAs may be any of the following physical equipment groupings defined in the physical model of ISA 88: *Process cell*, *Unit* or *Equipment Modules*. During batch design, the mapping of the implemented PEA configuration to the ISA 88 structure must be addressed. There are also applications in which PEAs are acting as *Control Modules* but linking of procedural elements to *Control Modules* is not recognized in ISA 88, and as such not relevant for this paper. It is also not intended to have a mapping on the level of *Functional Equipment Assemblies* (FEA) and *components* from the modular plant hierarchy following VDI 2776, since there should be no direct interaction with the batch system. The automation of PEAs internally can be structured in a way that ISA 88 proposes for classical process plants using FEAs and *components*.

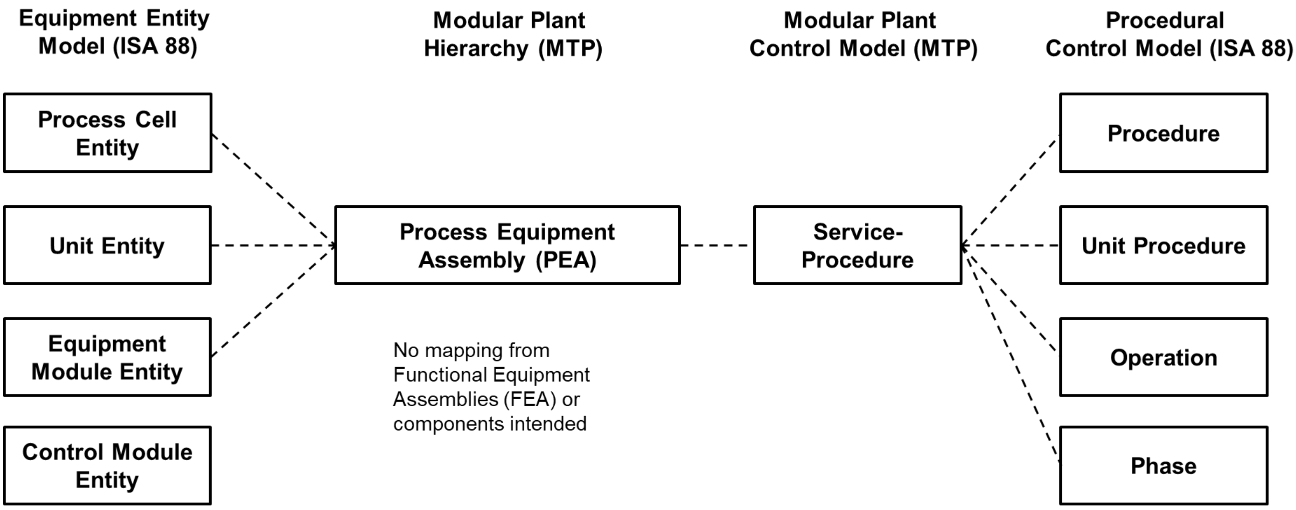


Figure 2: Alignment of modular concepts following VDI VDE NAMUR 2658 and VDI 2776 to ISA 88 Equipment Entity Model and Procedural Control Model

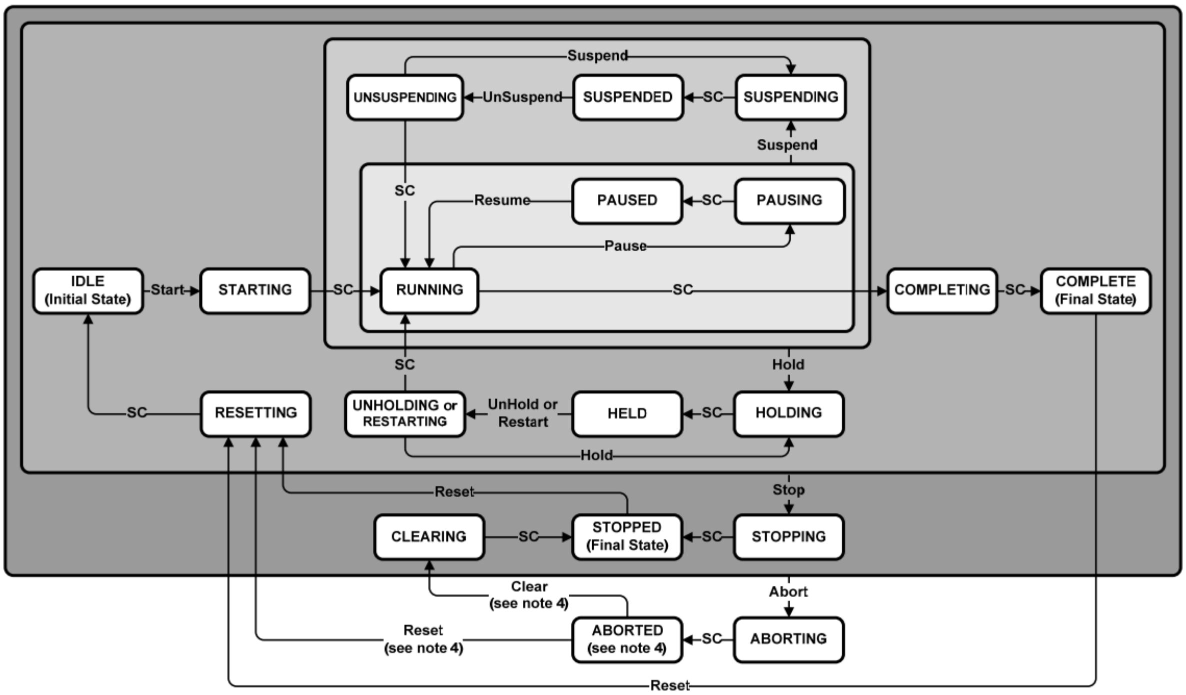
The automation functions of PEAs are encapsulated in *services*. These *services* are essentially the *Equipment Procedural Elements* of ISA 88; they may be referenced by *Recipe Procedural Elements*. Mostly a *service* would be an *Equipment Procedural Element* for a *Unit Procedure*, *Operation* or *Phase*. In exceptional cases, the *services* could also link to *Procedures* of ISA 88. This would bring PEAs on the level of *Process Cells.* On the other hand, *services* can be used as a *Phase*. The usage of *services* in an ISA 88 batch system depends on the implementation of the *service* in the PEA and the specific application in the recipe.

Services themselves can be structured by re-using internal *services* of a PEA and using *DataAssemblies* which realize the interaction with e.g., field devices. It is not intended to have a reference of *DataAssemblies* to the *Procedural Control Model*.

The MTP orchestration is not integrated into the picture since this is equivalent to the batch engineering of the procedures. For operation, all interactions between batch systems and PEAs are only done via *services*.

3 MTP Service Implementation for ISA 88 Alignment

ISA 88 proposes a state model that defines the behavior of procedural elements (Figure 3). The states, commands and transition of recipe procedural elements should align with those of the equipment procedural elements to effectively orchestrate a control recipe. Furthermore, the ISA 88 notes that state changes may be propagated to other equipment entities or procedural elements through what ISA 88 terms coordination control. It is generally accepted that states can be propagated in either direction, from a lower to a higher level or vice versa. Explicit propagation rules have, however, not been defined in the standard and are also not subject of this paper. With different documents referring to the state model and the international standard from 1997 describing it in the normative annex, the ISA 88 state model became a de-facto standard for batch implementations. Other adaptions, e.g. for packaging industry as PackML standards have been made.
For MTP, there is a very similar state model for the behavior of *services* (Figure 4). With the implementation of the logic within the states of *services*, the behavior of the state models can be aligned. For the alignment, the state models in Figure 3 and Figure 4 are used and referred to.



Notes: 1. SC = State Change as a result of state actions completed.
2. Actions of an equipment procedural element are generally defined by its Acting States.
3. The light, light+medium, light+medium+dark, and light+medium+dark+very dark grey boxes represent collections of states that can be preempted using the Hold, Stop, and Abort commands, respectively.
4. In any procedural elements using a collapsed version of this model that does not include CLEARING, the Clear command is invalid and ABORTED is a Final State; in those that include CLEARING, Reset is invalid while in the ABORTED state.

Figure 3: Exemplary ISA 88 state model (version 2010)

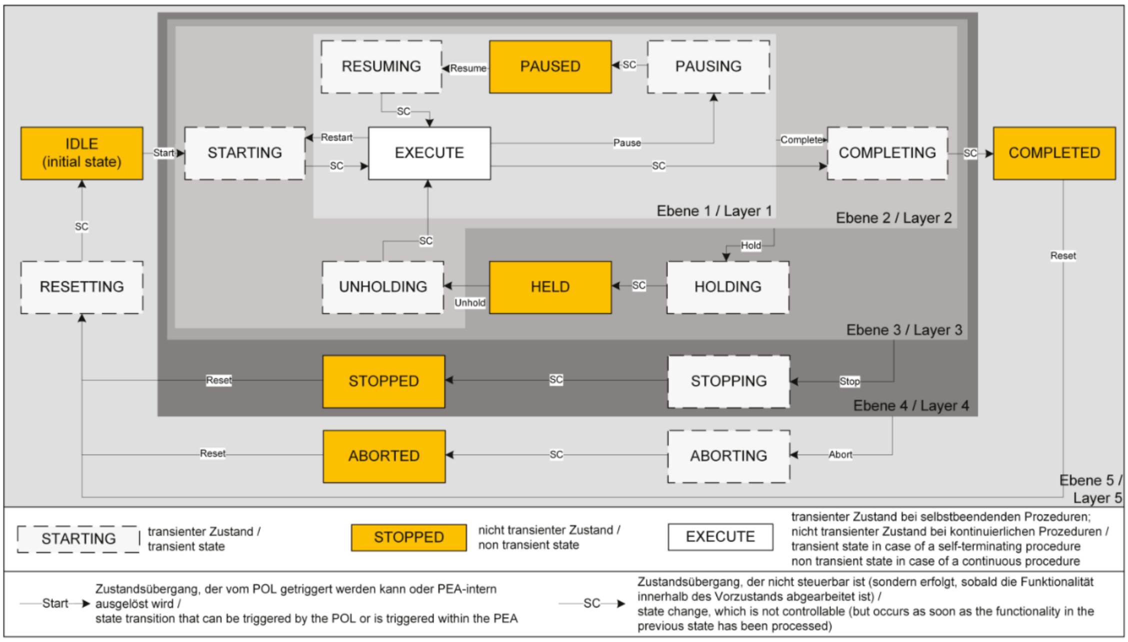


Figure 4: MTP State machine for Services (version 2022)

The guidelines for MTP *service* implementation can be described with the following rules. Note, that these recommendations address maximum compliance with the above printed ISA 88 state model:

- The *restart* command should be disabled, since a *restart*, as defined by MTP, of the ISA 88 state model is not foreseen
- The *complete* command should be disabled, and therefore a *service* shall be implemented as self-completing, since there is no *complete* command foreseen in ISA 88
- All information required by the *service* for execution and completion should be provided by the ISA 88 system prior to entering the EXECUTE state.
- The UNHOLDING state should perform similar logic and checks as STARTING and also should consider previous states. This is required to ensure that if *hold* occurred while in STARTING or COMPLETING, the *unhold* recovers appropriately.

Likewise, there are also best practices for the ISA 88 batch system:

- The ISA 88 ABORT loop should use the *reset* path instead of CLEARING path (see also Note 4 of Figure 3)
- The ISA 88 SUSPENDING loop should not be used. MTP *services* and ISA 88 have similar behavior on their PAUSING loop and therefore this loop represents the preferred way of pausing an EXECUTE/RUNNING state.

Finally, there are differences in the naming of states and commands between ISA 88 and MTP which are translated in the following table:

Table 1: Mapping of State and Command Names of ISA 88 and MTP

ISA 88	**MTP**
RUNNING	EXECUTE
COMPLETE	COMPLETED
UnHold / Restart	*Unhold*
Not Used	*Restart*
Not Used	*Complete*

Table 1 highlights the differences in states and commands for the state models from ISA 88 and MTP. The ISA 88 state RUNNING is represented as EXECUTE in the MTP model. There is also a difference in the ISA 88 COMPLETE which is named COMPLETED in MTP. The commands *unhold* and *restart* in ISA 88 both address the transition from HOLD via UNHOLDING/RESTARTING to RUNNING. In MTP this is done with only *unhold*. Furthermore, the commands *restart* and *complete* are not available in ISA 88.

From the implementation guidelines and the differences listed in Table 1 the ISA 88 and MTP functionality is specialized to make it compliant:

- *Services* cannot be completed by the ISA 88 system via the *complete* command. Note that approaches other than an MTP *complete* command may be employed for the batch system to signal the PEA to complete. In exceptional cases the *stop* command could be used to end a *service*. During the design of the process and the recipe, it should be stated which *stop* would lead to a loss of product. It is up to the implementation scope of the ISA 88 system vendor to offer possibilities for continuous *service procedures.*
- Within the *EXECUTE* logic, the *service* needs to check for its complete conditions e.g., via *ProcessValueIn*, *ProcedureParameter, ConfigurationParameter,* or internal conditions.

With POL Systems, batch processes can be implemented using continuous services. During the design of recipes, suitable services need to be selected and orchestrated. It is up to the PEA vendor to design services for either application. Both ways, continuous and self-completing can be implemented in parallel e.g. as procedures of a *service*.
Remaining differences from the ISA 88 state model is the possibility to *hold* from STARTING or COMPLETING. Both are transient states which means that there are no actions from the ISA 88 system required. A *hold* command might be triggered which would lead to a transition to HOLDING and HOLD. Thus, the batch system should be able to process the HOLDING and HOLD state independent from the underlying state model. The ISA 88 and MTP working groups are working together to potentially align the state models.

With these implementation rules, an alignment with the extended ISA 88 state model example can be achieved. Further prerequisites are dependent on the technical specifications of batch systems, e.g., communication via OPC UA. This alignment of MTP and ISA 88 represents a significant leap forward for Modular Automation and enhances its applicability for both brownfield and greenfield scenarios. As an outlook, the alignment of the behavior for different modes of operation as well as transitions and scaling scenarios of recipes will be continued to be discussed.