The final authenticated version is available online at https://doi.org/10.1007/978-3-030-79108-7_3

BPMN Extensions for Modeling Continuous Processes *

Diana Viktoria Strutzenberger[1,2], Juergen Mangler[3], Stefanie Rinderle-Ma[3]

[1] Austrian Center for Digital Production, Vienna,
[2] Faculty of Computer Science, University of Vienna, Austria
[3] Department for Informatics, Technical University of Munich, Germany
diana.strutzenberger@acdp.at,
{juergen.mangler, stefanie.rinderle-ma}@tum.de

Abstract. Business process management has focused on discrete processes so far, i.e., processes with identifiable distinct outcomes (e.g., in manufacturing). By contrast, processes known from process and control engineering, e.g., chemical synthesis, have not been fully considered yet. Such processes can be discrete or continuous, i.e., require real-time control systems with constant inlet and outlet flows as well as temporally stable conditions. This paper models continuous processes with existing and standardized means, i.e., BPMN, and provides an exact definition of the parameters and loop conditions. The capabilities of BPMN for modeling continuous processes are analyzed and necessary extensions are provided. The concepts are applied to several real-world use cases from process and control engineering.

Keywords: Process and Control Engineering, Continuous Processes, Process Modeling and Execution, BPMN Extensions

1 Introduction

In process engineering, the design of control systems focuses on the formal description of processes that deal with measuring and controlling complex systems, such as chemical reactors [4] or heat exchangers [5], which are typically applied in mining, production, electricity, gas and water supply as well as waste management. While closed-loop systems take a measured value into consideration for the next control operation (e.g. thermostat), open-loop systems ignore the effect of their output on the system (e.g. temperature control knob on a radiator) [12].

Open-loop systems can be represented as **discrete processes**, closed-loop systems as **continuous processes**. Brewing beer, for example, can be operated as a discrete process, in which a reactor is filled with the ingredients, then started, and at some point in time the next batch of beer is ready. For the continuous operation of the reactor ingredients are continuously added on one side, while

* **Acknowledgements.** This work has been partially supported and funded by the Austrian Research Promotion Agency (FFG) via the Austrian Competence Center for Digital Production (CDP) under the contract number 854187.

beer continuously comes out on the other side. The reactor can run forever, its inside is in certain defined states, but it is not possible to track the contents of a glass of water that is added on one side, while it becomes beer.

With ongoing efforts to introduce semantically rich modeling notations such as BPMN[1] into domains like manufacturing [8], it becomes apparent that for reducing the complexity of a system, it is beneficial to model all of its behavior in a single notation, instead of having a collection of modeling and planning artefacts. BPMN is a widely applied standard for modeling business processes and therefore proves to be the right medium to communicate complex processes across industries. Manufacturing processes have already been modeled and orchestrated using BPMN in the Cloud Process Execution Engine (CPEE)[2] [8], enabling dynamic models of complex processes and respective flow description. By contrast, continuous processes still lack such a digital representation despite its advantages for interoperability and for creating digital twins in process engineering; the latter constitutes a research topic which has strongly gained interest in the last years [2,9]. Based on a set of requirements derived from real-world scenarios, the following artefacts are elaborated and evaluated based on real-world use cases: 1) A BPMN extension to simplify the modeling of continuous processes, while increasing their understandability through custom gateways and events. 2) A set of examples that cover the commonly used control engineering patterns, in order to exemplify the expressiveness of our approach. The remainder is structured as follows. Section 2 introduces a set of examples as an evaluation baseline. In Sect. 3 requirements for modeling continuous processes are described. Section 4 introduces BPMN extensions to realize the requirements. In Sect. 5 the solution is evaluated. The paper closes related work in Sect. 6, and conclusion in Sect. 7.

2 Scenario Analysis

To provide a clear understanding of the approach, it is important to analyse how the terms discrete and continuous are used in the fields that specialize on continuous processes, i.e., process and control engineering.

Discrete and Continuous Processes in Process Engineering: In terms of process engineering, processes are divided into two groups, i.e., batch and continuous processes. According to [4], continuous processes are characterized by constant inlet and outlet flows as well as temporally stable conditions. This steady state approach implies a constant progression of the process variables, which can only be achieved after the start-up phase. In contrast, batch or discrete processes present themselves as a one-time input of the materials to be processed. The process steps to be performed mostly run sequentially or are least limited in time by a certain condition or state.

Discrete and Continuous Processes in Control Engineering: According to [12] continuous systems are characterized by parameters which may take any value in a defined boundary. Further [12] conclude that the frequency in which

[1] BPMN: Business Process Modeling and Notation, `www.bpmn-standard.org`

[2] `cpee.org`

data access and control tasks are performed determines a discontinuous behavior which needs to be counteracted by finding a fitting control strategy. Due to hardware performance constraints truly continuous behavior may not be realized as physical sensors can only provide data in short time intervals. Among others, control engineering mainly deals with the following three frequently used patterns explained using sample processes.

Feedback and Feedforward Control - Heat Exchanger: As described in [5] there are different options for the implementation of the control system for a heat exchanger. A simple feedback controller such as a PID controller measures the system output, compares the value to the set point and reacts accordingly. Feedforward control is another option for controlling a heat exchanger and reacts to disturbances before they influence the system. A coupled feedback controller compensates the remaining errors [5]. The process model in BPMN is shown in Fig. 1.

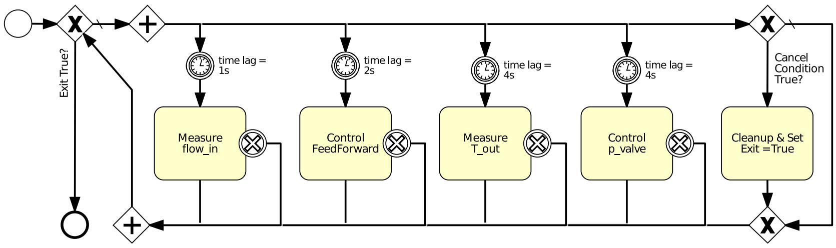


Fig. 1. BPMN Model of a Feedforward Control System for a Heat Exchanger [5].

Cascade Control - Position Control in Machine Tool: For the position control of drives in machine tools, the cascade control method is usually used. The control model consists of control loops that are nested within each other [11]. The output variable of one control loop is the input variable of the following control loop. Therefore, a direct time dependency between the individual control loops is evident and must be displayed in the workflow. The BPMN workflow model of the control procedure is shown in Fig. 2.

3 Modeling Requirements

Modeling languages and implementation environments must support the realistic representation of continuous processes. Based on the application scenarios presented in Sect. 2, the following modeling requirements for continuous processes are derived that serve as basis for assessing existing modeling languages:

Req.1 - Continuity Continuity needs to be presented in form of a loop. The process model shall imply a continuous flow without having to set a limited number of repetitions or a time limit from the beginning. BPMN supports loop characteristics for tasks and sub-processes [1]. However, this modeling option is confined to individual tasks and sub-processes and thus may lead to complex, multi-level process flows.

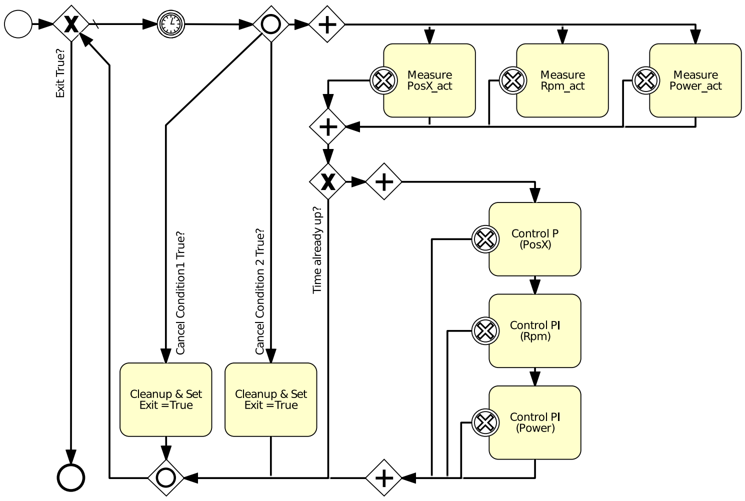


Fig. 2. BPMN Model of a Cascade Control System for Position Control [11].

Req.2 - Break Conditions Break conditions can be applied to tasks and sub-processes with loop characteristics [1]. For defining the termination handling of a continuous process and allowing the option to define clean up sequences, Cancel Events can be used. However, for Intermediate Cancelling Events only Boundary Interrupting Events are defined.

Req.3 - Real-Time Process Due to the critical impact of time regarding continuous processes the role of time needs to be clearly defined. According to [12], a real-time system reacts to simultaneously occurring process signals in time with a corresponding output. BPMN supports Timer Events which need to be applied correctly and comprehensibly in order to understand the implied constraints and display them correctly.

Req.4 - Parallelism Parallel processing of tasks and task sequences needs to be supported by the chosen modeling environment. Parallelism can be modeled in a way similar to loops in form of attributes for tasks and sub-processes [1]. The orientation of the attribute marker indicates whether the multiple sequences are processed in parallel or sequentially. Again, increasing complexity of the process leads to an incomprehensible model.

Req.5 - Exception Handling Mechanisms for exception handling have to be available as assurance for real-time processing and determinism. For exception handling BPMN already implies the usage of Intermediate Events [1]. Timer events can be applied to deal with time restrictions which are fundamental for continuous processes.

Req.6 - Limited Complexity If all necessary details of a continuous process are included in the model, the level of complexity must not exceed to a point at which users no longer understand the process behind the model. To prevent this drawback, modeling conventions need to compensate complex relations, but still lead to a detailed and comprehensible process models.

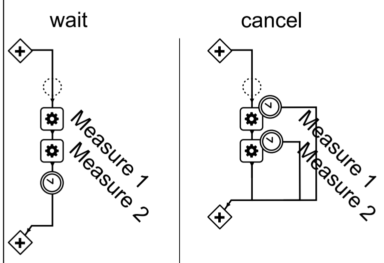
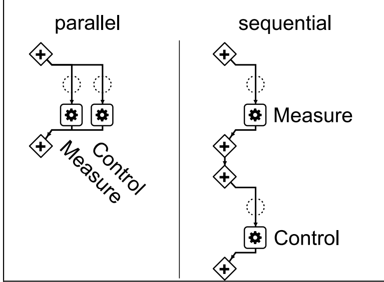
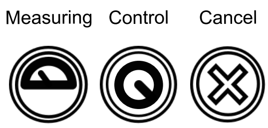
4 BPMN Extensions for Modeling Continuous Processes

The newly defined symbols are based on common BPMN symbols which have been used to depict similar process models as introduced in Sect. 2. Achieving a completely consistent representation of real control logic for continuous processes is difficult to depict at a reasonable level of complexity. In addition, the requirements set out in Section 3 have to be met. Using the standard modeling capabilities in BPMN might result in a complex sequence behind a single control task as demonstrated by the example in Fig. 2.

Closed-Loop Sub-System Gateway: The Closed-Loop Sub-System is a combined version of an Inclusive and an Event-Based Gateway. A Closed-Loop Sub-System combines the advantages of the described gateways and the attributes of loop and multiple-instance characteristics defined in [1]. The symbol and basic definitions of a Closed-Loop Sub-System is introduced in Tab. 1. Event-based Gateways do not allow Cancel triggers for Intermediate events used in branches after the gateways. Closed-Loop Sub-Systems let tokens traverse each branch which allows processing multiple parallel branches simultaneously (Req. 4). Each branch starts with newly introduced Intermediate Event markers – Measuring, Control and Cancel. Measuring and Control are Intermediate Timer Events indicating an execution interval for each branch as well as individual sections of a sequence to illustrate gradually changing intervals and also dependencies (Req. 3). Cancel is an Intermediate Catching Event which includes conditions for ending the loop (Req. 2). Similar to features of Inclusive and Event-Based Gateways, the Closed-Loop Sub-System is passed through in cycles, which is indicated by an arrow leading back from the converging marker after the branches have been traversed to the first diverging marker (Req. 1). A return to the first marker is only allowed as long as no Cancel Catching Intermediate Event is triggered (Req. 5). The Gateway Direction of a Closed-Loop Sub-System is diverging. It MUST have at least two outgoing Sequence Flows, one starting with a Measuring Intermediate Catching Event and one starting with a Cancel Intermediate Catching Event. It MAY further have multiple outgoing Sequence Flows but MUST have no more than one incoming Sequence Flow. The Closed-Loop Sub-System allows to model continuous processes in a simple structure with necessary attributes, but clearly arranged at one level (Req. 6). Further modeling conventions are described in Tab. 1.

Intermediate Catching Event Types: To indicate which tasks are executed in one of the parallel branches under the Closed-Loop Sub-System Gateway, three new symbols based on Intermediate Catching Events are proposed in this work. The symbols are shown and described in Tab. 1.

Table 1. Closed-Loop Sub-System Attributes and Model Associations

Attribute Name	Description/Usage
Interval duration overrun: 	When *wait*, the following iteration starts when all branches are finished and the defined interval duration is reached. When *cancel*, the interval duration defines exactly the time each branch takes to finish. If the tasks in a branch are finished early, the branch waits. If not all tasks are finished yet, they are terminated.
Measure-control cycle execution: 	When *parallel*, tasks after Measuring and Control Intermediate Catching Events are performed in parallel. When *sequential*, the tasks after Control Intermediate Catching Events are performed only after all tasks after Measuring Intermediate Catching Events are finished.
Marker	Description
Closed-Loop Sub-System Gateway: 	**Closed-Loop Sub-System Gateway:** contains branches which are triggered for the measuring and control phases of the cycle, as well as branches executed when cancellation events are received.
Intermediate Catching Events: 	**Measuring:** Receiving events to perform measuring cycles **Control:** Receiving events to perform control cycles **Cancel:** Receiving events to abort closed-loop systems

5 Application

In addition to standard functionalities for modeling with BPMN and support for common workflow patterns the suggested extensions to BPMN are implemented in the CPEE. In order to show the advantages of the proposed extensions in modeling and understanding processes, the process examples identified by the literature study from Sec. 2 were implemented. In the following, some examples are presented. The model of a heat exchanger with a combination of feedback and feedforward control in CPEE is shown in Fig. 3 on the left. The Closed-Loop Sub-System is implemented with the *cancel* attribute set. The control system cancels the execution of every branch in which the tasks are not finished in the given time interval. A position control system with cascade controller is shown in Fig. 3 on the right. The model is implemented in CPEE with the *cancel* attribute set in order to guarantee real-time behavior. Multiple control tasks with different execution frequencies are modeled sequentially to show the order for the execution of the controller elements. Regarding Fig. 2 the effort involved in changing the model can lead to errors in the model and thus the intended semantics of the real process. The modeling convention depicted in Fig. 3 allows the user to modify the model much easier (insert into one branch vs. insert into a combination of loop and parallel as well as inserting additional events and connections).

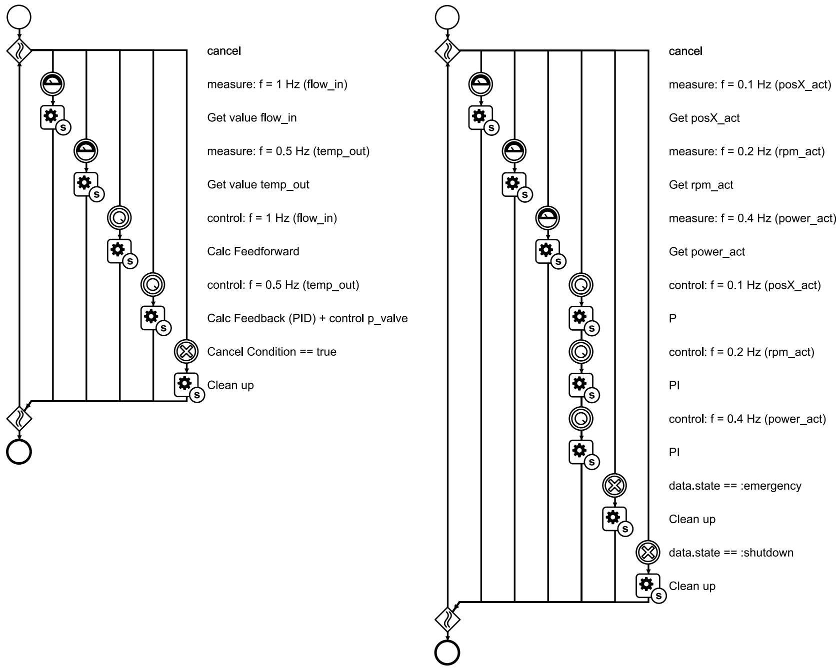


Fig. 3. Feedforward (left) and Cascade Control (right) - Process Model with Extensions in CPEE.

6 Related Work

Tools for process and control engineering include Aspen Plus [7] and Matlab/Simulink [13]. In general, a common practice is the separation of modeling and execution environment. The linkage between both environments is realized via a code generating solution as presented in [3]. Process patterns provide (partly complex) constructs for describing process flows such as time [6] and resource patterns [10]. Based on the features described in Sect. 3 the usability of the patterns has been assessed. Overall, the most limited support is provided for Req.3 on modeling real-time processes and Req.6 on limited complexity.

7 Conclusion

Process and control engineering constitutes a major industry including mining, gas and water supply, but the continuous processes in this field have not been considered from a business process management perspective yet. This work explains and distinguishes the characteristics of discrete and continuous processes. BPMN is analyzed for representing continuous processes based on set of requirements derived from real-world scenarios. The challenge is to express continuity with break conditions, real-time processing, parallelism, and exception handling in balance with taming the complexity of the resulting models. BPMN extensions in terms of symbols are proposed. The executability in CPEE allows the use of the models also as non-proprietary digital twins.

References

1. Business Process Model and Notation (BPMN), Version 2.0 p. 538
2. Bamberg, A., Urbas, L., Bröcker, S., Kockmann, N., Bortz, M.: What makes the digital twin an ingenious companion? Chemie Ing. Technik 92(3), 192–198 (2020)
3. Chindris, G., Muresan, M.: Deploying simulink models into system-on-chip structures. In: Electronics Technology. pp. 313–317. IEEE (2006)
4. Hertwig, K., Martens, L.: Chemische Verfahrenstechnik: Berechnung, Auslegung und Betrieb chemischer Reaktoren. Oldenbourg (2007)
5. Khare, Y.B., Singh, Y.: Pid control of heat exchanger system. International Journal of Computer Applications 8(6), 0975–8887 (2010)
6. Lanz, A., Reichert, M., Weber, B.: Process time patterns: A formal foundation. Inf. Syst. 57, 38–68 (2016)
7. Luyben, W.L.: Distillation design and control using Aspen simulation. John Wiley & Sons (2013)
8. Mangler, J., Pauker, F., Rinderle-Ma, S., Ehrendorfer, M.: centurio.work - industry 4.0 integration assessment and evolution. In: BPM Ind. Forum. pp. 106–117 (2019)
9. Pfeiffer, B.M., Oppelt, M., Leingang, C.: Evolution of a digital twin for a steam cracker. In: Emerging Technologies and Factory Automation. pp. 467–474 (2019)
10. Russell, N., Hofstede, ter, A., Aalst, van der, W., Mulyar, N.: Workflow control-flow patterns : a revised view. BPM reports, BPMcenter. org (2006)
11. Schmid, D., Kaufmann, H., Pflug, A., Strobel, P., Baur, J.: Automatisierungstechnik - Grundlagen, Komponenten, Systeme. Europa Lehrmittel, Nourney, Vollmer GmbH & Co. KG (2015)
12. Tröster, F.: Steuerungs- und Regelungstechnik für Ingenieure. Oldenbourg Lehrbücher für Ingenieure, Oldenbourg
13. Yang, C.h., Vyatkin, V.: Transformation of simulink models to iec 61499 function blocks for verification of distributed control systems. Control Engineering Practice 20(12), 1259–1269 (2012)