

## Interview Guide V2 DE

### - Begrüßung

Hallo und danke, dass Sie sich Zeit nehmen, um mit mir dieses Interview durchzuführen.

Möchten Sie sich vielleicht kurz vorstellen und Ihre Verbindung zu BPMN bzw. zur Verfahrenstechnik, Fertigungstechnik oder Prozessmodellierung erklären? Ich möchte Sie bitten, dabei nicht Ihren Namen zu nennen, sondern nur die folgenden Informationen:

- Berufsbezeichnung und Umschreibung des Arbeitgebers
- Basis der Expertise zum Forschungsthema
- Ausbildung bzw. fachlicher Hintergrund
- Berufserfahrung (zB. mehr als 10 Jahre)

### - Allgemeine Einführung

Unsere Forschung konzentriert sich auf die Entwicklung einer Methodik, um kontinuierliche Prozesse in BPMN darzustellen und sie in einer Workflow-Engine ausführbar zu machen. Für diese Aufgabe haben wir an BPMN-Erweiterungen für kontinuierliche Prozesse gearbeitet. Warum kontinuierliche Prozesse? Weil diskrete Prozesse bereits in anderen Forschungsarbeiten behandelt wurden und nicht die gleichen Schwierigkeiten bei der korrekten Darstellung mittels BPMN aufweisen. BPMN ist bereits ein weit verbreiteter Standard im Business Process Management und hat seinen Weg in die Fertigung gefunden. Diskrete Fertigungsprozesse können bereits mit BPMN 2.0 modelliert werden.

Im Grunde wollen wir eine Methodik einführen, um solche Prozesse so darzustellen, dass sie von jeder Person in einem Unternehmen, vom Ingenieur bis zum Manager, verstanden werden können. Dies könnte durch die Verwendung dieser Notation erreicht werden. Ein weiterer Vorteil ist auch, dass es bereits eine Reihe von Workflow-Engines gibt - Anwendungen, die die Ausführung dieser Prozessmodelle auf der Grundlage der für jedes Symbol implementierten Logik ermöglichen. Wir arbeiten mit einer webbasierten Anwendung, die erweiterbar ist und mehrere Kommunikationsschnittstellen implementiert hat. Ein weiterer Vorteil ist daher die Interoperabilität in diesem Zusammenhang im Vergleich zu anderen proprietären, starren Softwareanwendungen. Wir wollen herausfinden, ob diese Technik auch für die Implementierung von digitalen Abbildern eingesetzt werden kann. Da digitale Abbilder dazu dienen, ein physikalisches System oder einen Prozess in digitaler Form darzustellen - meist anhand von Daten oder mathematischen Modellen - mussten wir einen Weg finden, den Ablauf von kontinuierlichen Prozessen, wie sie aus der Prozessindustrie bekannt sind, darzustellen. Aus diesem Grund haben wir uns auf die Modellierung von Regelkreisen konzentriert.

Die Prozessmodelle sollen durch die BPMN für Personen mit unterschiedlichem Hintergrund leicht verständlich sein. Die Interviews werden geführt, um herauszufinden, wie Prozess- und Regelungstechnik und Techniken aus der Business Process Modellierung kombiniert werden können und wie erste Ergebnisse von Experten wie Ihnen wahrgenommen werden. Außerdem wollen wir herausfinden, ob es Schwachstellen gibt, die von Experten identifiziert wurden und wie wir diese beseitigen können.

### - Spezifische Einführung

#### - Begriff "Digitaler Zwilling":

Es gibt verschiedene Methoden, Dinge aus der echten Welt, zum Beispiel echte Maschinen zu simulieren. Teils merkt man aber, dass es mehr Parameter brauchen würde, als bei normalen Simulationsmethoden, um eine Maschine vollkommen so abzubilden, wie sie sich in der Realität verhält. Bei einem digitalen Zwilling wird versucht, möglichst nahe an das reale Verhalten einer Maschine oder anderer Objekte heranzukommen. Das soll dazu führen, dass wenn etwas getriggert wird bei einer echten Maschine, der Digitale Zwilling das gleiche oder ein möglichst ähnliches Verhalten zeigt.

#### - Kontinuierlichen Prozessen:

Kontinuierliche Prozesse möchte ich mit Beispielen erklären.

Wenn man Bierbrauen her nimmt, gibt es 2 Möglichkeiten. Die diskrete - also nicht kontinuierliche Variante - wäre wenn man in einen geschlossenen Kessel die Zutaten hinein gibt, 10 Liter Wasser usw. und den Brauprozess einfach schrittweise ablaufen lässt. Am Ende kommt dann eine begrenzte Menge an Bier heraus.

Die andere kontinuierliche Variante wäre, wenn man keinen vollkommen abgeschlossenen Kessel hat, sondern miteinander verbundene Kessel bei denen immer wieder Zutaten zugefügt werden und immer wieder Bier entnommen wird. Das geht die ganze Zeit so, sodass man nicht nachvollziehen kann, welcher Liter Wasser zu welchem Liter Bier gehört. Dabei läuft ein Teilprozess im ersten Kessel ab, während gleichzeitig im letzten Kessel der letzte Prozessschritt stattfindet bevor das Bier fertig wird.

#### - Geschlossenen Regelkreisen (Closed Loop):

Ein geschlossener Regelkreis ist jene Logik in Form von Hardware oder Software, die das kontinuierliche Bierbrauen ermöglicht.

Wenn man einen Prozess wie das kontinuierliche Bierbrauen hat, muss man schauen wie man schlechtes Bier vermeidet während der Prozess läuft. Man möchte die Qualität auf einem gewissen Punkt halten. Beim schrittweisen Bierbrauen hat man nur die 10 Liter, bei denen etwas schief gehen kann und mit den nächsten 10 Litern macht man es dann besser.

Aber was ist, wenn man die Brauanlage dauernd laufen lässt und ständig Bier austritt? Dann muss man währenddessen den Prozess überprüfen und schauen, dass das man die gute Qualität des Bieres erhält. Das heißt man testet oder misst Werte, die die Qualität beschreiben, überprüft wie sich diese Werte von optimalen Werten unterscheiden und reagiert entsprechend - stimmt etwas beim Zucker oder Alkoholgehalt nicht, muss das Mischverhältnis geändert werden.

Das heißt in einem geschlossenen Regelkreis werden, während der Prozess läuft, gewisse Werte überprüft, diese werden mit optimalen Werten verglichen und je nach Abweichung reagiert das System darauf.

#### - Kontinuierliche Prozesse

Aus Informatik Sicht bestehen kontinuierliche Prozesse aus einer sich ständigen wiederholenden Abfolge von Zustandsabfragen und Regulierungen (Z&R). Z&R sind jeweils traditionelle Code-Stücke, die sich auf Sensoren oder Aktuatoren beziehen.

Um solche kontinuierlichen Prozesse konsistent formal zu beschreiben, zu modellieren und in weiterer Folge ausführen zu können haben wir folgende Merkmale identifiziert.

1) Würden Sie die Merkmale für wichtig oder unwichtig einstufen? Bitte geben Sie eine Begründung für Ihre Antwort an:

Eigenschaften	Wichtig (ja)	Unwichtig (nein)	Warum?
1. Verschiedene Z&R Kombinationen sind unabhängig und können parallel ablaufen			
2. Regulierungen folgen immer auf Zustandsabfragen			
3. Die Dauer von jeder Z&R Kombination ist beschränkt			
4. Wenn Zustandsabfragen			

gewisse Ergebnisse liefern wird das System beendet			
5. Bevor das System beendet wird muss es in einen konsistenten Zustand gebracht werden			
6. Das resultierende System soll für Menschen verständlich sein			

2) Können Sie graphische Eigenschaften nennen, die Sie für die Modellierung kontinuierlicher Prozesse wichtig finden? Ergeben sich daraus vielleicht Merkmale die wir vergessen haben?

3) Wo liegen Ihrer Meinung nach die Herausforderungen bei der Modellierung kontinuierlicher Prozesse?

### - BPMN Extension Model

#### Vorstellen der Erweiterungen

Ich werde Ihnen Prozesse zeigen, die mit BPMN 2.0 und mit unseren Erweiterungen modelliert wurden. Die Erweiterungen sollen zum einen vordefinierte Modellierungskonventionen für in der Prozess- und Steuerungstechnik übliche Routinen bereitstellen und zum anderen helfen, die Unterschiede zwischen den parallelen Pfaden in den Prozessmodellen zu visualisieren. Die Prozesse werden in der [REDACTED] modelliert. Für das Verständnis der [REDACTED]-Modelle müssen drei zusätzliche Symbole erklärt werden.

#### Erweiterungen

##### - Closed-Loop Subsystem Gateway



Das Gateway ist eine Kombination aus einem inklusiven und einem ereignisbasierten Gateway. Es enthält Verzweigungen bzw. Kanten, die für die Zustandsabfragen- und Regulierungsphasen des Zyklus ausgelöst werden, sowie Verzweigungen, die beim Empfang von Abbruchereignissen ausgeführt werden. Die Ereignisse und Tasks in den einzelnen Kanten sind unabhängig voneinander. Damit erfüllen wir das erste der oben genannten Features - das einzelne Verläufe unabhängig voneinander sind und sie parallel ausgeführt werden. Das Gateway ermöglicht außerdem die Definition der Intervalldauer jedes Zyklus sowie von Überschreitungsbedingungen (Warten/wait - Abbrechen/cancel) und der Ausführungsreihenfolge für Zustandsabfragen und Regulierungen (Z&R) bzw. Mess- und Steuerungsaufgaben.

Interval duration overrun	⇒ cancel	Interval duration overrun	⇒ wait
Measure-control cycle execution	⇒ parallel	Measure-control cycle execution	⇒ sequential
<b>Execution Probability</b>		<b>Execution Probability</b>	
Min times	⇒ Times	Min times	⇒ Times
Max times	⇒ Times	Max times	⇒ Times
Avg times	⇒ Times	Avg times	⇒ Times

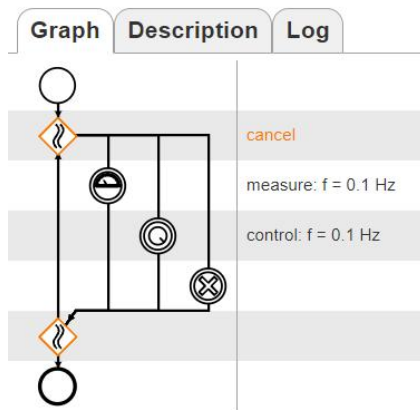
<p><b>wait</b></p>	<p><b>cancel</b></p>	<p>Wenn <b>wait</b> gewählt wird, beginnt die nächste Iteration, wenn alle Verzweigungen beendet sind und die festgelegte Intervalldauer erreicht ist. Bei <b>cancel</b> definiert die Intervalldauer genau die Zeit, in der jeder Zweig zu beenden ist. Wenn die Tasks in einem Zweig schneller beendet werden, wird der Zweig warten. Wenn noch nicht alle Tasks beendet sind, werden sie abgebrochen.</p>
<p><b>parallel</b></p>	<p><b>sequential</b></p>	<p>Bei <b>parallel</b> werden die Tasks nach Measure und Control Events parallel ausgeführt. Bei <b>sequential</b> werden die Tasks nach Control Events erst ausgeführt, nachdem alle Tasks nach Measure Events beendet sind.</p>

### - Intermediate Catching Events

In einem Closed-Loop Subsystem werden spezifische Ereignisse erwartet, die in eine der drei folgenden Kategorien fallen - Ereignisse für Zustandsabfragen/Messungen, Ereignisse für Regelungen bzw. Regulierungen und Ereignisse für die Unterbrechung des Closed-Loop Subsystems. Es gibt für jede Ereigniskategorie zumindest eine Kante, die vom Gateway ausgeht. Die Kanten zeigen an, welche Tasks nebeneinander ablaufen.

Sobald diese Ereignisse eintreten, werden auch die Tasks, die in der Kanten danach angeordnet sind, ausgeführt.

Hier sehen Sie ein Bild eines Closed-Loop Subsystems, in dem nur Ereignisse der drei Kategorien ohne darauffolgende Tasks modelliert sind.



Die drei Ereigniskategorien, die wir definiert haben, sind wie folgt:

### Measure



Empfängt Events für die Ausführung von Tasks in Messzyklen

### Control



Empfängt Events für die Ausführung von Tasks in Regelzyklen

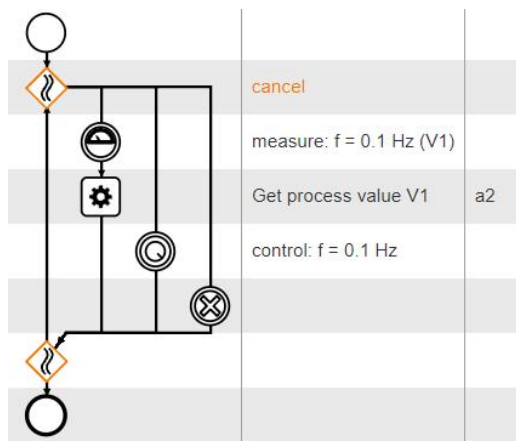
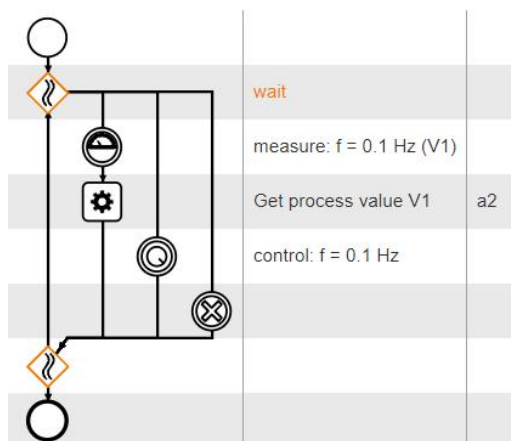
### Cancel



Empfängt Events für das Abbrechen von Closed-Loop Systemen

Diese Symbole geben den Zweck der nachfolgenden Tasks/Aufgaben an. Diese Tasks werden nur ausgeführt, wenn die Ereignisse ausgelöst werden. Das bedeutet, dass das Messereignis angibt, dass die nachfolgenden Symbole nur Messabläufe bzw. Zustandsabfragen anzeigen. Das Gleiche gilt für Regulierungs- (Kontroll-) und Abbruchereignisse. Für Zustandsabfragen und Regulierungen (Mess- und Steuerereignisse) können wir eine Zykluszeit definieren. Dadurch kann die Dauer von Anpassungen im System definiert werden. Je nachdem, ob das Closed-Loop-Subsystem einen parallelen oder sequentiellen oder einen Wait- oder Cancel-Ansatz verfolgt, läuft die Ausführung unterschiedlich. Mit diesen Bedingungen kann man definieren, inwiefern Anpassungen beim System erfolgen.

Hier sehen Sie ein Closed-Loop Subsystem mit einem Task für eine Messung. In diesem Fall wird das Ereignis für die Messung alle 10 s getriggert. Danach wird der Wert V1 geholt bzw. gemessen. "wait" bedeutet hier, dass ein neuer Zyklus erst startet, wenn die Messung erfolgt dh. der Prozess in dieser Kante abgeschlossen ist. Mit "cancel" wird nach 10 s automatisch der neue Zyklus gestartet.



Interval frequency in Hz  $\Rightarrow 0.1$

### Values Expected to Change

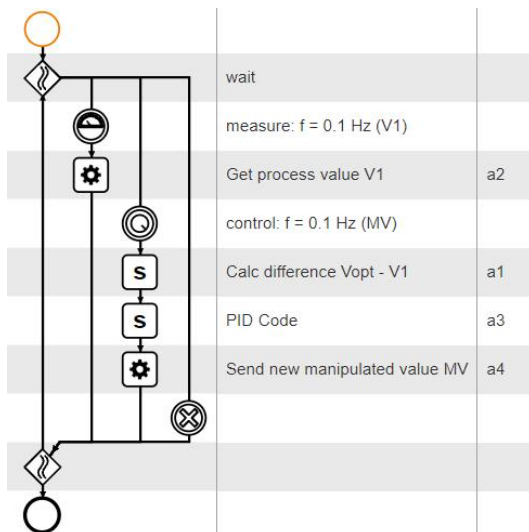
X  $\updownarrow$  Value  $\Rightarrow V1$

Add Value

Bei Measure kann man, neben der Frequenz des Messereignisses, den Wert, der sich während des Prozesses ändert, definieren.

Mit Hilfe von Regelungsereignissen kann ferner festgelegt werden, welches Reglermodell verwendet wird - PID, PI, PD, - diese Regler werden in ihrer mathematischen Form dargestellt. Die Tasks für sie sind im Grunde Berechnungen, die in festen Teilprozessen dargestellt werden. Nach diesen Berechnungen kann der Benutzer Tasks zur weiteren Datenverarbeitung hinzufügen. Dies kann auch nach Mess-Tasks geschehen, die man auch als Datenerfassungs-Tasks bezeichnen kann.

Hier sehen Sie ein Prozessmodell mit einem Wert, der gemessen wird, und einer darauffolgenden Regelung.



Interval duration overrun  $\Rightarrow$  wait  
Measure-control cycle execution  $\Rightarrow$  sequential

	Data Elements	Endpoints	Attributes
New	X $\updownarrow$ Vopt	$\Rightarrow 100$	
	X $\updownarrow$ MV	$\Rightarrow 200$	

Interval frequency in Hz  $\Rightarrow 0.1$

### Values Expected to Change

X  $\updownarrow$  Control type  $\Rightarrow$  :PID  
Value  $\Rightarrow$  MV  
Upper Limit  $\Rightarrow 250$   
Lower Limit  $\Rightarrow 100$

Add Value

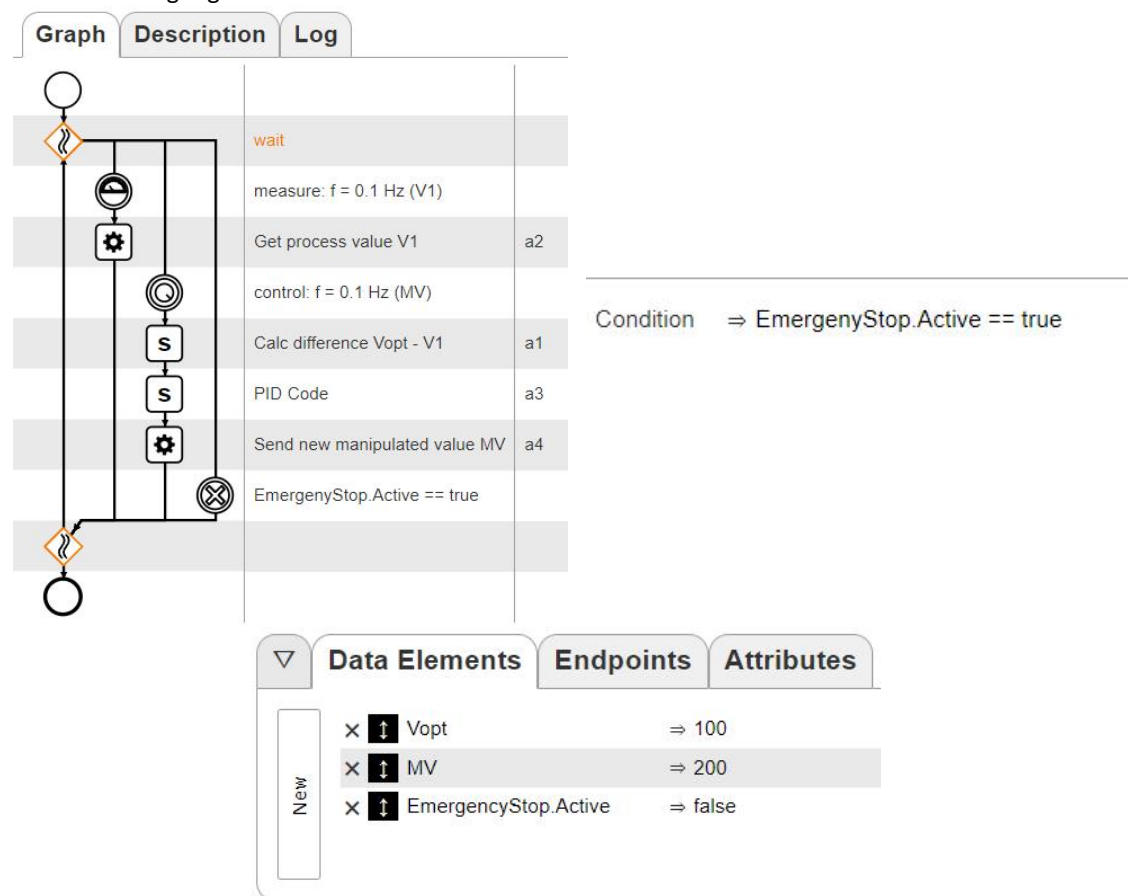
“Wait” bedeutet wieder, dass für den nächsten Zyklus auf das Beenden aller Tasks gewartet wird - auch auf die Regulierungstasks. “Sequential” heißt, dass die Tasks nacheinander ausgeführt werden, dh. es wird erst gemessen bzw. der Zustand abgefragt und mit diesem gemessenen Wert wird die Regelung durchgeführt. Dabei wird vom optimalen Wert Vopt der aktuelle Wert V1 abgezogen und mit dieser Differenz die neue Stellwert MV ausgerechnet. Dieser wird dann mit einem Service Request an das entsprechende Stellglied - also ein Element, das aktiv Einfluss auf den Prozess ausübt - geschickt. Wenn man möchte, kann man die Differenzberechnung, die Regelungsberechnung PID

Code und das Aussenden des Befehls an das System in einen Subprozess zusammenfassen. Bei Control können zusätzlich die Art der Regelung (hier PID) sowie der neue Stellwert und dessen Limits eingetragen werden.

Würde hier “parallel” verwendet werden, würde der letzte Wert von V1 genommen werden, für den keine Zeitgarantie besteht.

Zustandsabfragen und Regulierungen sollten in regelmäßiger Frequenz ausgelöst werden - Abbruchereignisse hingegen werden nur durch ihre Abbruchbedingungen ausgelöst, die der Benutzer definieren kann. Ein Beispiel für ein Abbruchereignis wäre, wenn etwas den Abbruch eines Zyklus auslöst, wie z.B. eine Watchdog-Funktion zur Überwachung der maximalen Zykluszeit, oder es könnte ein Notstopp sein.

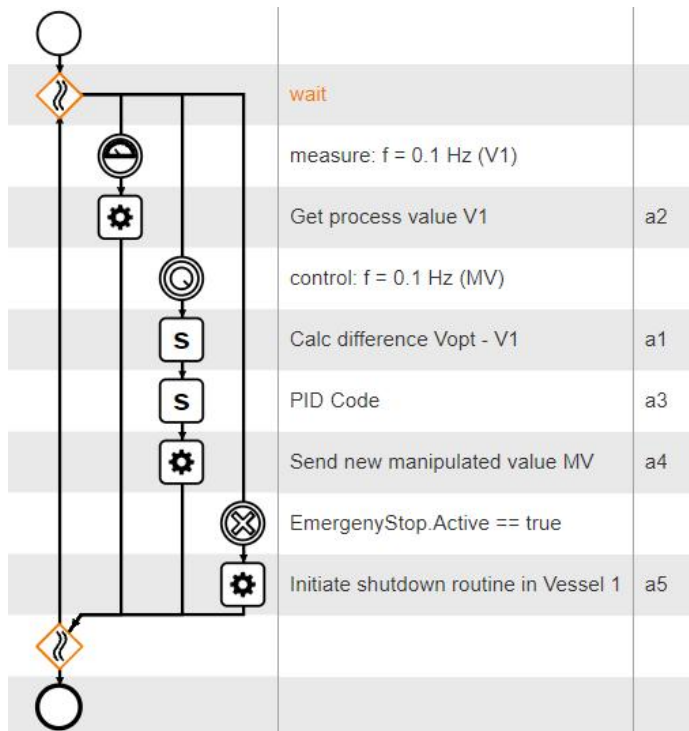
Hier sehen Sie einen Prozess mit einem zu messenden Wert, einer Regelung und einer Abbruchbedingung.



Sobald die Abbruchbedingung “EmergencyStop.Active” true wird, werden repetitive Tasks beendet. Abbruchbedingungen werden bei jedem Zyklus neu evaluiert.

Nachdem das Ereignis ausgelöst wurde, können Tasks zur Aufrämroutine abgearbeitet werden, bevor der Zyklus beendet oder der Prozess vollständig beendet wird. Damit wird auch das fünfte Feature für Aufräumprozesse erfüllt.

Hier sehen Sie einen Prozess, bei dem Aufräumtasks definiert wurden.



Die vorgestellten Erweiterungen sollen bei der Modellierung von kontinuierlichen Prozessen helfen, indem Vorlagen für die Erstellung von Prozessmodellen vorgegeben werden. Und andererseits durch die Darstellung als Closed-Loop Subsystem mit eigenen Symbolen für Zustandsabfrage-, Regulierungs- und Abbruchereignisse helfen, solche Prozesse leichter nachvollziehen können. Hinzu kommt, dass man für eine übersichtlichere Darstellung des gesamten Prozesses auch Subprozesse zur Unterteilung nutzen kann. Damit erfüllen wir auch das letzte Feature - Verständlichkeit der Modelle von kontinuierlichen Prozessen.

### Beispiele für Prozessmodelle

Ich werde Ihnen nun Prozessbeispiele zeigen, die mit den in unserer Arbeit vorgestellten Erweiterungen modelliert sind. Ich möchte, dass Sie sich die Modelle ansehen und mir sagen, was Sie aus ihnen herauslesen können und ob die Modelle den notwendigen Informationsgehalt für die Modellierung der zugrundeliegenden Regelungsprozesse erfüllen. Vorab wird Ihnen zum jeweiligen Prozess erklärt, was abgebildet werden soll. Geben Sie bitte offenes Feedback zu den Modellen in BPMN.

4) Bewerten Sie das folgende Modell: Einfache PI-Temperaturregelung (Simple Feedback control) für einen Wärmetauscher basierend auf dem Beispiel aus der MathWorks Bibliothek<sup>1</sup>.

Die Temperatur einer Flüssigkeit in einem Rührkessel wird mittels Wärmetauscher geregelt. Der über den Wärmetauscher eingebrachte Wärmestrom wird über ein Ventil, das den Dampfstrom kontrolliert, gesteuert. Der zu beachtende störende Umgebungseinfluss ist die schwankende Temperatur der zugeführten Flüssigkeit. Der Tank ist als isoliert anzunehmen.

<sup>1</sup> Siehe <https://de.mathworks.com/help/control/ug/temperature-control-in-a-heat-exchanger.html>, abgerufen am 05/05/2021



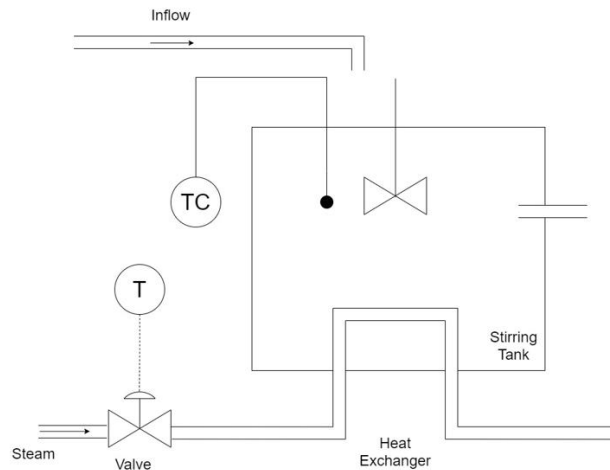


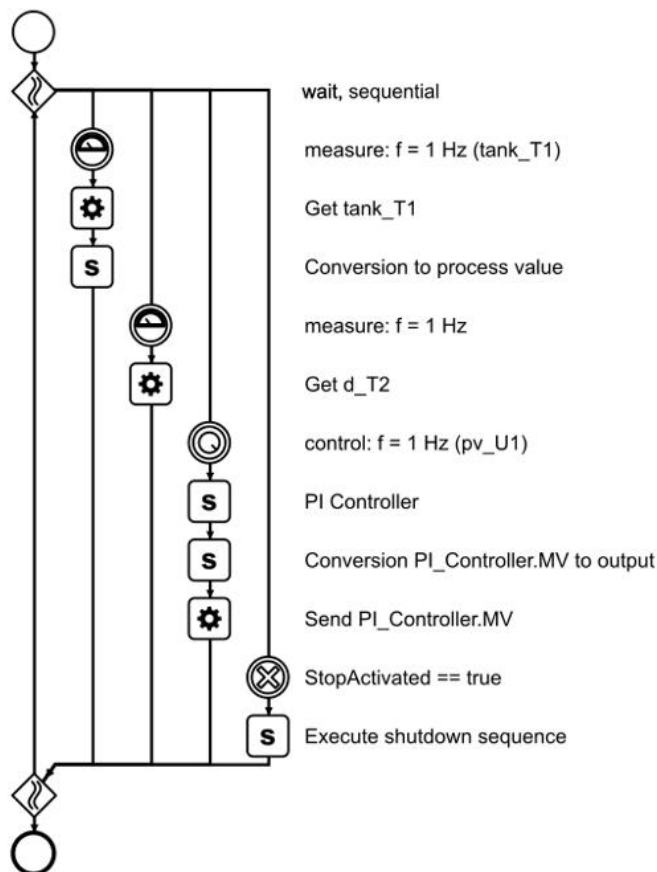
Bild 1: Flow Chart des Prozesses<sup>2</sup>

	Data Elements	Endpoints	Attributes
New	x <input type="checkbox"/> tank_T1	⇒ 0	
	x <input type="checkbox"/> theta	⇒ 14.7	
	x <input type="checkbox"/> tau	⇒ 21.3	
	x <input type="checkbox"/> tauc	⇒ 24.5582	
	x <input type="checkbox"/> PI_Controller.ER	⇒ 0	
	x <input type="checkbox"/> PI_Controller.PV	⇒ 0	
	x <input type="checkbox"/> PI_Controller.SP	⇒ 210	
	x <input type="checkbox"/> PI_Controller.GAIN	⇒ 1.2341	
	x <input type="checkbox"/> PI_Controller.TC	⇒ 24.5582	
	x <input type="checkbox"/> PI_Controller.PV_Out	⇒ 0	
	x <input type="checkbox"/> PI_Controller.IntER	⇒ 0	
	x <input type="checkbox"/> PI_Controller.prevER	⇒ 0	
	x <input type="checkbox"/> d_T2	⇒ 0	
	x <input type="checkbox"/> PI_Controller.MV	⇒ 0	
	x <input type="checkbox"/> MV_U1	⇒ 0	

<sup>2</sup> Siehe <https://de.mathworks.com/help/control/ug/temperature-control-in-a-heat-exchanger.html>, abgerufen am 05/05/2021

▼	Data Elements	Endpoints	Attributes
	x	timeout	⇒ http://gruppe.wst.univie.ac.at/~mangler/services/timeout.php
	x	get_tank_T1	⇒ http-get://127.0.0.1:8080/process1/parameters/tank/t1
	x	post_U1	⇒ http-post://127.0.0.1:8080/process1/parameters/valve/v1
	x	get_d_T2	⇒ http-get://127.0.0.1:8080/process1/parameters/disturbance/t2

Interval duration overrun	⇒ wait
Measure-control cycle execution	⇒ sequential



ID ⇒ a3  
 Label ⇒ Conversion to process value  
 Script ⇒  $PI\_Controller\_PV = tank\_T1$

ID ⇒ a1  
 Label ⇒ PI Controller  
 Script ⇒  $PI\_Controller.prever = PI\_Controller.ER;$   
 $PI\_Controller.ER = PI\_Controller.PV - PI\_Controller.SP; PI\_Controller.IntER = PI\_Controller.IntER +$   
 $PI\_Controller.DerER = (PI\_Controller.ER - PI\_Controller.prever) * interval\_frequency;$   
 $PI\_Controller.MV = (PI\_Controller.GAIN * PI\_Controller.ER) + (PI\_Controller.GAIN/tauc * PI\_Control$

ID ⇒ a7  
 Label ⇒ Conversion PI\_Controller.MV to output  
 Script ⇒  $MV\_U1 = PI\_Controller.MV$

Ich bitte Sie, das Modell nach folgenden Kriterien auf einer Skala von 1 (schlecht) bis 5 (sehr gut) zu bewerten:

Kriterium	Frage	1 (sehr schlecht)	2 (nicht gut)	3 (weder noch)	4 (gut)	5 (sehr gut)
Verständlichkeit	Was passiert?					
Übersichtlichkeit	Kann ich das Gesamtsystem auf einen Blick erfassen?					
Einfachheit	Könnte man das Modell noch einfacher darstellen?					
Logik	Wird klar was parallel und was sequentiell passiert?					
Erweiterbarkeit	Könnte man dem Modell noch etwas hinzufügen, was den Informationsgehalt verbessern würde?					

5) Bewerten Sie das folgende Modell:

Die Modell basiert auf der Beschreibung des Heizprozesses, entnommen aus Schulungsunterlagen der Fa. Siemens<sup>3</sup>.

Es handelt sich hierbei ebenfalls um eine Temperaturregelung für einen Rührreaktor, die Regelung wird in diesem Beispiel mit einem PID-Regler, einer Handsteuerung sowie einem Pulsgenerator realisiert. Die Heizung erfolgt nicht über einen Wärmetauscher sondern über ein Heizelement. Weiters sind Verriegelungsbedingungen definiert. Als Basis für die Prozessmodellierung wurden die Beschreibungen aus den Schulungsunterlagen für die Prozessmodellierung mit SIMATIC PCS7 herangezogen<sup>4</sup>. Unser Prozessmodell wird mit einer automatischen Steuerung modelliert, die mit Umschalten auf Handsteuerung aus dem Closed-Loop System ausbricht. Wir gehen davon aus, dass das System bereits angelaufen ist und automatisch gesteuert wird. Weiters wird der Prozess für einen Reaktor und nicht wie in den Unterlagen für 2 Reaktoren beschrieben. PV\_In steht entsprechend für Pcs7AnIn.PV\_In für die Temperaturregelung.

<sup>3</sup> Siehe

<https://www.automation.siemens.com/sce-static/learning-training-documents/pcs7/v9-0/p01-06-control-loop-v9-tud-0719-de.pdf>, abgerufen am 05/05/2021

<sup>4</sup> Siehe

<https://www.automation.siemens.com/sce-static/learning-training-documents/pcs7/v9-0/p01-06-control-loop-v9-tud-0719-de.pdf>, abgerufen am 05/05/2021

▼		Data Elements	Endpoints	Attributes
New	×	↑ ↓ T_Reactor_In	⇒ 0	
	×	↑ ↓ T_Reactor_Max	⇒ 60.0	
	×	↑ ↓ Level_Reactor_In	⇒ 0	
	×	↑ ↓ Level_Reactor_Min	⇒ 200.0	
	×	↑ ↓ PID_Controller.GAIN	⇒ 1.0	
	×	↑ ↓ PID_Controller.TI	⇒ 100.0	
	×	↑ ↓ PID_Controller.TD	⇒ 0.0	
	×	↑ ↓ PID_Controller.DiffGAIN	⇒ 1.0	
	×	↑ ↓ PID_Controller.ER	⇒ 0.0	
	×	↑ ↓ PID_Controller.PV	⇒ 0.0	
	×	↑ ↓ PID_Controller.SP	⇒ 50.0	
	×	↑ ↓ PV_Out	⇒ 0.0	
	×	↑ ↓ PV_In	⇒ 0.0	
	×	↑ ↓ OperationMode	⇒ Automatic	
	×	↑ ↓ MainSwitch	⇒ On	
	×	↑ ↓ PID_Controller.prevER	⇒ 0	
	×	↑ ↓ PID_Controller.IntER	⇒ 0	
	×	↑ ↓ PID_Controller.DerER	⇒ 0	
	×	↑ ↓ PID_Controller.MV	⇒ 0	
	Interval duration overrun ⇒ cancel			
	Measure-control cycle execution ⇒ sequential			



ID ⇒ a7

Label ⇒ PID Controller

Script ⇒ 

```
PID_Controller.prevER = PID_Controller.ER;
PID_Controller.ER = PID_Controller.PV - PID_Controller.SP;
PID_Controller.IntER = PID_Controller.IntER + (PID_Controller.ER * 1/interval_frequency);
PID_Controller.DerER = (PID_Controller.ER - PID_Controller.prevER) * interval_frequency;
PID_Controller.MV = PID_Controller.GAIN * (PID_Controller.ER + (PID_Controller.IntER / PID_Controller.interval_frequency));
```

Ich bitte Sie, das Modell nach folgenden Kriterien auf einer Skala von 1 (schlecht) bis 5 (sehr gut) zu bewerten:

Kriterium	Frage	1 (sehr schlecht)	2 (nicht gut)	3 (weder noch)	4 (gut)	5 (sehr gut)
Verständlichkeit	Was passiert?					
Übersichtlichkeit	Kann ich das Gesamtsystem auf einen Blick erfassen?					
Einfachheit	Könnte man das Modell noch einfacher darstellen?					
Logik	Wird klar was parallel und was sequentiell passiert?					
Erweiterbarkeit	Könnte man dem Modell noch etwas hinzufügen, was den Informationsgehalt verbessern würde?					

6) Wären Sie aufgrund dieser Erweiterungen bereit, diese Modellierungsmethode in Ihrem Arbeitsalltag einzuführen, wenn Sie ein Modell eines kontinuierlichen Prozesses entwickeln?

7) Wie gut beschreiben diese Erweiterungen Ihrer Meinung nach ein Kontrollsystem für diese Beispiele?

Features	1 (sehr schlecht)	2 (nicht gut)	3 (weder noch)	4 (gut)	5 (sehr gut)
Modell 1 (Frage 3)					
Modell 2 (Frage 4)					

Bitte begründen Sie Ihre Antworten.

8) Was fehlt für eine detaillierte Prozessbeschreibung?

9) Wenn Sie Erfahrung in der Regelungstechnik haben - was würden Sie empfehlen, um diese Erweiterungen zu ergänzen, um sie für Ingenieure attraktiver zu machen?

10) Ich möchte zum Schluss noch etwas genauer auf die Modelle mit unseren Erweiterungen von oben eingehen und Sie bitten, diese nochmals zu bewerten - auf einer Skala von 1 bis 5, wobei 5 der beste Wert ist.

Frage	1 (sehr schlecht)	2 (nicht gut)	3 (weder noch)	4 (gut)	5 (sehr gut)
Wie einfach ist in den gezeigten Modellen nachzuvollziehen, dass die einzelnen Abläufe parallel und unabhängig voneinander laufen?					
Wie einfach ist es, zu definieren, wann eine Anpassung am System erfolgt?					
Wie einfach ist es die max. Dauer einer Anpassung zu definieren?					
Wie einfach ist es zu definieren, unter welchen Bedingungen sämtliche repetitiven Aufgaben beendet werden sollen?					
Wie einfach ist es, zu definieren, dass danach Aufräumaufgaben einmalig zu erfolgen haben?					
Wie einfach ist es, komplexe Abläufe im Kontext von kontinuierlichen Prozessen mit diesen Erweiterungen zu beschreiben?					

Vielen Dank, dass Sie sich dafür Zeit genommen haben!  
Ich bin dankbar für Feedback zum Interview!

Was sagen Sie zu den Fragen? Waren sie verständlich formuliert? Waren sie zu lang? Zu geschlossen?  
War die Dauer des Interviews unangenehm?