

### Transcript - Group 1 "Engineers", Interview 3

I ... Interviewer (BLINDED)

B ... Expert

(Unv.)... Incomprehensible passage

(...) ... Pause longer than 3 sec.

( ) ... Comment

// ...// ... Speaker overlap

#### Transcript

1 1. I: Gut. Hallo und danke, dass Sie sich Zeit nehmen, um mit mir dieses Interview durchzuführen.  
2 Ich heie Diana Strutzenberger. [REDACTED]  
3 [REDACTED] Und mchten Sie sich vielleicht kurz vorstellen und Ihre Verbindung  
4 zu BPMN beziehungsweise Verfahrenstechnik, Fertigungstechnik oder Prozessmodellierung  
5 erklren? Also je nachdem, aus welcher Fachrichtung Sie kommen. Ich mchte Sie bitten, dabei  
6 nicht Ihren Namen zu nennen, sondern nur die folgenden Informationen. Die Berufsbezeichnung  
7 und Umschreibung des Arbeitgebers, Basis der Expertise zum Forschungsthema, Ihre Ausbildung  
8 beziehungsweise Ihren fachlichen Hintergrund und Ihre Berufserfahrung. #00:00:52-9#

9 2. B: Gerne. [REDACTED]  
10 [REDACTED]  
11 [REDACTED]  
12 [REDACTED]  
13 [REDACTED]  
14 [REDACTED]  
15 [REDACTED]  
16 [REDACTED]  
17 [REDACTED]  
18 [REDACTED]  
19 [REDACTED]  
20 [REDACTED] #00:02:20-6#

21 3. I: Wunderbar. Dankeschn. Noch kurz zu meinem Forschungsthema beziehungsweise zu meiner  
22 Arbeit. Meine Forschung konzentriert sich auf die Entwicklung einer Methodik, um  
23 kontinuierliche Prozesse in BPMN darzustellen und sie in einer Workflow Engine ausfhrbar zu  
24 machen. Fr diese Aufgabe haben wir an BPMN-Erweiterungen fr kontinuierliche Prozesse  
25 gearbeitet. Warum jetzt gerade kontinuierliche Prozesse? Weil diskrete Prozesse bereits in  
26 anderen Forschungsarbeiten behandelt wurden und nicht die gleichen Schwierigkeiten bei der  
27 korrekten Darstellung mittels BPMN aufweisen. BPMN ist bereits ein weit verbreiteter Standard  
28 im Business Process Management und hat auch schon seinen Weg in die Fertigung gefunden.  
29 Das heit, diskrete Fertigungsprozesse knnen bereits mit BPMN 2.0 modelliert werden. Im  
30 Grunde wollen wir eine Methodik einfhren, um solche Prozesse so darzustellen, dass sie von  
31 jeder Person in einem Unternehmen vom Ingenieur bis zum Manager verstanden werden  
32 knnen, und dies knnte durch die Verwendung von BPMN erreicht werden. Ein weiterer Vorteil  
33 ist auch, dass es bereits eine Reihe von Workflow Engines gibt. Das sind Anwendungen, die die  
34 Ausfhrung dieser Prozessmodelle auf der Grundlage der fr jedes Symbol implementierten  
35 Logik ermglichen. Wir arbeiten mit einer webbasierten Anwendung, die erweiterbar ist und  
36 mehrere Kommunikationsschnittstellen implementiert hat, wie beispielsweise OPC UA, HTTP  
37 Requests, dergleichen. Ein weiterer Vorteil ist daher die Interoperabilitt in diesem  
38 Zusammenhang im Vergleich zu anderen proprietren, starren Softwareanwendungen. Wir  
39 wollen herausfinden, ob diese Technik auch fr die Implementierung von digitalen Abbildern  
40 eingesetzt werden kann. Und da digitale Abbilder dazu dienen, ein physikalisches System oder  
41 einen Prozess in digitaler Form darzustellen, meist anhand von Daten oder mathematischen  
42 Modellen, mssen wir einen Weg finden, den Ablauf von kontinuierlichen Prozessen, wie sie aus  
43 der Prozessindustrie bekannt sind, darzustellen. Aus diesem Grund haben wir uns auf die  
44 Modellierung von Regelkreisen konzentriert. Die Prozessmodelle sollen durch BPMN fr

Personen mit unterschiedlichen Hintergrund leicht verständlich sein. Die Interviews werden geführt, um herauszufinden, wie Prozess- und Regelungstechnik und Techniken aus der Business Processes 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 werden und wie wir diese auch beseitigen können. So als kurze Einführung. (...) So. Zu Beginn des Interviews eine kurze Begriffseinführung, was wir unter den folgenden Begriffen verstehen im Zuge dieser Arbeit. Digitaler Zwilling. Es gibt verschiedene Methoden, Dinge aus der echten Welt, zum Beispiel echte Maschinen, zu simulieren. Teils merkt man aber, dass es natürlich mehr Parameter brauchen würde als bei herkömmlichen Simulationsmethoden, um eine Maschine vollkommen abzubilden, wie sie sich in der Realität wirklich verhält. Bei einem digitalen Zwilling wird versucht, möglichst nahe an das reale Verhalten einer Maschine oder andere Objekte heran zu kommen. Und das soll dazu führen, dass, wenn etwas getriggert wird, ein spezielles Verhalten bei einer echten Maschine, der digitale Zwilling, das gleiche oder in möglichst ähnlicher Form auch tut oder das gleiche Verhalten zeigt. Kontinuierliche Prozesse. Dafür möchte ich vielleicht mit Beispielen das Ganze erklären. Wenn man jetzt Bierbrauen her nimmt, gibt es zwei verschiedene Möglichkeiten. Es gibt einerseits die diskrete, also die nicht-kontinuierliche Variante. Die wäre, wenn man in einem geschlossenen Kessel die Zutaten hinein gibt, zum Beispiel zehn Liter Wasser und so weiter, und den Brauprozess einfach genau mit diesen Zutaten schrittweise ablaufen lässt. Am Ende kommt dann eine begrenzte Menge an Bier heraus und an gewissen Ausschuss Produkten. Die andere kontinuierliche Variante wäre, wenn man keinen vollkommenen abgeschlossenen Kessel hat, sondern miteinander verbundene Kessel, bei denen immer wieder Zutaten hinzugefügt werden und immer wieder Bier entnommen wird. Das würde die ganze Zeit so gehen, so dass man nicht nachvollziehen kann, welcher Liter Wasser zu welchem Liter Bier am Ende gehört. Dabei läuft ein Teilprozess im ersten Kessel ab, während gleichzeitig im letzten Kessel der letzte Prozessschritt stattfindet bevor das Bier fertig wird. Das heißt, wir haben mehrere Prozesse, die nebeneinander laufen, die ganze Zeit. Dann geschlossene Regelkreise. 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 zehn Liter, bei denen etwas schief gehen kann und mit den nächsten zehn Litern könnte man es dann besser machen. Aber was ist, wenn man die Brauanlage dauernd laufen lässt und wenn ständig Bier austritt? Dann muss man währenddessen den Prozess überprüfen und schauen, dass 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 dann natürlich auch entsprechend. Und stimmt etwas zum Beispiel beim Zuckergehalt nicht oder beim Alkoholgehalt, dann müsste man am Anfang das Mischverhältnis ändern oder anpassen. Das heißt, in einem geschlossenen Regelkreis werden während der Prozess läuft gewisse Werte überprüft. Diese werden mit optimalen Werten verglichen und je nach Abrechnung reagiert das System darauf. Kommen wir zum ersten Abschnitt, zu den ersten Fragen. Aus Informatik-sicht bestehen kontinuierliche Prozesse aus einer sich ständigen wiederholenden Abfolge von Zustandsabfragen und Anpassungen beziehungsweise Regulierungen. Zustandsabfragen und Regulierungen 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 auszuführen, haben wir folgende Merkmale identifiziert. Diese Merkmale werde ich jetzt kurz nur vorlesen und ich würde Sie gerne bitten, dass Sie mir sagen, ob Sie diese Merkmale für wichtig oder unwichtig einstufen und vielleicht auf Ihre Antwort auch eine Begründung folgen lassen. Das heißt, Merkmal, halten Sie es für wichtig oder unwichtig und eine kurze Begründung, warum, bitte. Das erste wäre, dass verschiedene Zustandsabfragen und Regulierungen-Kombinationen sind, die unabhängig funktionieren und die auch parallel ablaufen können. Würden Sie das für wichtig oder für unwichtig halten? #00:10:05-7#

4. B: Für wichtig. Weil viele kontinuierliche Prozesse kann man schlussendlich in Unterprozesse wieder aufgliedern und wieder parallel laufen (unv., lassen?) und die zeitlich eine Abhängigkeit voneinander (unv., haben?). #00:10:23-2#

- 100 5. I: Okay, danke. Zum zweiten Punkt. Regulierungen. Regelungsprozesse an sich folgen immer auf  
101 Zustandsabfragen. Würden Sie das für wichtig oder für unwichtig halten? #00:10:40-3#
- 102 6. B: Ja, für wichtig. Ja. Die Regulierung basiert auf den Feedbacks der Zustandsabfragen. Sonst  
103 wäre es ja eine Steuerung. #00:10:53-5#
- 104 7. I: Was würden Sie, wenn Sie diesen Begriff jetzt hier einführen, als Steuerung in Ihrem  
105 Verständnis definieren? #00:11:01-2#
- 106 8. B: Eine Steuerung reagiert also auf Input-Parameter und folgt modellbasiert und generiert  
107 Stellgrößen ohne, dass man auf Rückwirkungen ein/ also zurückgreift. #00:11:18-5#
- 108 9. I: Und im Falle einer Regelung ist es so, wie beschrieben, nehme ich an. Dass das System auf  
109 Parameter reagiert, die während des Prozesses abgefragt werden. #00:11:31-4#
- 110 10. B: Genau. #00:11:32-4#
- 111 11. I: Gut. #00:11:33-4#
- 112 12. B: Also ist irgendwo ein Ist-Soll-Vergleich oder eine Rückführung eines Messwertes, der dann  
113 direkt in die Regulierung eingreift. #00:11:41-1#
- 114 13. I: Okay, sehr gut. Dann passt das mit unserem Verständnis überein. Die Dauer von jeder  
115 Zustandsabfrage beziehungsweise Regulierung, von dieser Kombination, ist beschränkt. Halten  
116 Sie das für wichtig oder unwichtig? #00:11:56-8#
- 117 14. B: Die Dauer als Zeit meinen Sie? #00:12:00-4#
- 118 15. I: Genau. Die Dauer, dass es gewisse Bedingungen gibt, die die zeitliche Abfolge beziehungsweise  
119 die Reaktionszeit eventuell beschränken oder definieren. #00:12:15-5#
- 120 16. B: Bei den Erfahrungen, wo ich gemacht habe, ist die Taktung, sagen wir es mal so, sehr wichtig.  
121 Dass man die definieren kann. Vor allem wieviel Zeiteinheit, das so ein Regler oder eine  
122 Anpassung erfolgt. Je nach Prozess kann das halt/ Ja. Bei Reglern im Antriebsbereich spricht man  
123 davon Mikrosekunden. Bei Temperaturen sind es dann irgendwo Sekunden. Bei größeren  
124 Prozessen sind es dann zum Teil Minuten. Also die ZEITLICHE Folge ist sehr wichtig und aber  
125 auch die Latenzzeit, die man dazwischen hat. Und die Periodizität. Schlussendlich basieren die  
126 meisten Modelle auf einer kontinuierlichen Abtastzeit. Und wenn diese Abtastung zu fest jittert,  
127 dann funktioniert es nicht mehr. #00:13:11-4#
- 128 17. I: Hm. (zustimmend) Danke. Punkt Nummer Vier. Wenn Zustandsabfragen gewisse Ergebnisse  
129 liefern, wird das System beendet. Das heißt, Abbruchbedingungen. Würden Sie sagen, dass das  
130 wichtig oder eher unwichtig ist? #00:13:29-0#
- 131 18. B: Ja, ich glaube die typischen Zustandsregler in dem Sinne, also jetzt nicht im  
132 regelungstechnischen Sinne, aber diese Regelungseinheiten, die werden meistens von außen  
133 instrumentiert. So ein Temperaturregler wird irgendwann gestartet, hat ein Hochlaufverhalten,  
134 dann ein zyklisches Verhalten und ein stabiles Verhalten und damit ein Endverhalten. Und das  
135 wird von außen gesteuert. Was es sicher braucht, ist im Modell drinnen gewisse Bedingungen,  
136 die man setzen kann, um nach außen zu kommunizieren, dass diese Regelungen nicht mehr  
137 funktioniert. Aber dass der Prozess von sich aus definiert, wann abgebrochen wird, das glaube  
138 ich nicht. Das wird von außen gesteuert. #00:14:21-0#
- 139 19. I: Von außen, okay. Wir sehen das Ganze also allgemein natürlich als Bedingungen, die auch von  
140 außen eingebracht werden können, aber natürlich auch intern schon modelliert werden können.  
141 Das heißt, wenn das jetzt von einem Datenpunkt abhängig gemacht werden könnte, der von  
142 einer anderen Quelle gelesen wird, könnte man das schon hinein modellieren. Also so ungefähr

- 143 haben wir uns das auch vorgestellt als Beispiel, als Anwendungsbeispiel. Aber natürlich auch,  
144 dass man auf eine User-Eingabe oder etwas von extern wartet. Punkt Nummer Vier. Bevor das  
145 System beendet wird, muss es in einen konsistenten oder sicheren Zustand gebracht werden.  
146 Halten Sie das für wichtig oder unwichtig? #00:15:10-5#
- 147 20. B: Ja (lacht) die Frage (unv.) doch, gehört hier hinein oder nicht? Also ich glaube jeder Regler für  
148 sich wird von außen instrumentiert. Das heißt, diese Instrumentierung muss so erfolgen, dass es  
149 in einem stabilen Zustand gebracht wird. Und ob jetzt das dazu gehört oder nicht ist ja die große  
150 Frage. Aber sicher muss ein Regler, wie vorher gesagt, muss eine Phase haben, wie man so ein  
151 System hochfährt. Wie man es stabil hält. Und am Schluss muss es auch wieder in einen  
152 definierten Endzustand gebracht werden, so dass man diese Regelung eigentlich abschalten  
153 kann. #00:15:55-5#
- 154 21. I: Und dann der letzte Punkt. Für die Darstellung solcher Systeme. Das resultierende System soll  
155 für Menschen verständlich sein. #00:16:07-6#
- 156 22. B: Ja, auf dieser grafischen Ebene sehr wichtig. (lacht) Wenn es zu abstrakt ist, dann kann man es  
157 auch mit syntaktischem Text beschreiben. Also ich glaube, es ist wichtig, dass man dieses  
158 Regelmodell, was da dahinter abläuft, dass man das in verschiedenen grafischen einzelnen  
159 Elementen sieht. Also wenn es nur ein Block ist und es ist alles dort drinnen versteckt, dann ist es  
160 nicht nützlich ist. Also man muss, muss irgendwie/ Den Ablauf dieser Regelung muss man  
161 grafisch darstellen können. Also sprich. Bei wenn man jetzt einen Antriebsregler anschaut, da  
162 gibt es die Position, gibt die Geschwindigkeit, gibt den Strom. Das sind die physikalischen  
163 Einheiten, die es dort drinnen gibt und die muss man irgendwo sehen. Entsprechend muss man  
164 diese physikalischen Elemente oder Komponenten von so einem Modell muss man irgendwo  
165 greifbar haben. #00:17:16-4#
- 166 23. I: Natürlich zielt die Erweiterung der Notation insofern auch darauf ab, dass man das auf  
167 verschiedenste Arten von Prozessen anwenden kann. Wir möchten es natürlich auch eher für  
168 Beispiele aus der Industrie erweitern, aber wir überlegen uns auch, ob es für  
169 Managementprozesse anwendbar sein kann. Aber da Sie das jetzt erwähnt haben, die grafische  
170 Verständlichkeit auch, kommen wir eigentlich gleich auch zur nächsten Frage. Die heißt. Können  
171 Sie grafische Eigenschaften nennen, die Sie für die Modellierung kontinuierlicher Prozesse  
172 wichtig finden und ergeben sich vielleicht daraus Merkmale, die wir hier in unserer Liste  
173 eventuell vergessen haben? #00:18:03-4#
- 174 24. B: Ja, also, was ich vermisst habe, vielleicht kommt es nachher noch, sind, wenn man  
175 kontinuierliche Prozesse hat, hat man ja gewisse Eigenschaften vom System, die man  
176 modellieren will. Und jetzt in physikalischen Modellen sind es physikalische Eigenschaften wie  
177 eine Temperatur, ein Füllstand, eine Geschwindigkeit, ein Volumenstrom und irgendsolche  
178 Themen. Und dass man auf diese Größen das ganze Modell anwenden kann, dass man diese  
179 Größen auch irgendwo grafisch sieht und an bestimmten Punkten. #00:18:51-3#
- 180 25. I: Also eine aktuelle oder/ ja, doch. Eine aktuelle Möglichkeit, um den Zustand wirklich grafisch  
181 darstellbar zu machen für den Endanwender. #00:19:04-2#
- 182 26. B: Genau. Der Zustand ist ein Vektor von physikalischen Größen schlussendlich. Jetzt bei REALEN  
183 Prozessen. Wenn man jetzt andere Businessmodelle hat, dann ist es vielleicht irgendwie Geld,  
184 Liquidität oder irgendwas oder/ Das ist dann/ Ob es eine physikalische Einheit ist, der Euro,  
185 weiß ich nicht. Nein, glaube ich nicht. (lacht) Aber bei realen Maschinen ist immer wichtig, dass  
186 man einen Bezug hat zur Physik oder zu den realen Größen und nicht irgendwelche abstrakte  
187 Größen hat, wie zum Beispiel den digitalen Wert eines Temperatursensors in einem binären  
188 Format. Das nützt niemandem was. Also, dass man dort wirklich Grad Celsius hat oder einen  
189 Volumenstrom oder ein Fülllevel in Millimeter oder solche Einheiten. #00:19:56-5#
- 190 27. I: Verstehe, ja. Wenn wir das jetzt abschließen von der Sichtweise her, dass wir jetzt die  
191 Erweiterungen noch nicht kennen und noch keine Beispielsmodelle gesehen haben, wie wir das

192 jetzt modellieren würden. Würde Ihnen vielleicht vorab einfallen, welche Herausforderungen es  
193 geben kann, wenn man kontinuierliche Prozesse modellieren möchte? #00:20:28-1#

194 28. B: Ja, was bei mir also immer wieder kommt ist/ Das eine ist das Modellieren vom  
195 kontinuierlichen Prozess. Aber der kontinuierliche Prozess ist nicht über die ganze Zeit  
196 kontinuierlich. Sondern der kontinuierliche Prozess ist während einer gewissen Phase  
197 kontinuierlich. Und parallel dazu gibt es halt dann das Zustandsmodell, oder die Abfolge dieses  
198 Prozesses. Und die Kombination, das weiß ich, dass ist immer ein Problem auch bei  
199 Zustandsautomaten. Es gibt so ein Zustandsmodell, wenn die Maschine zum Beispiel in einem  
200 normalen operativen Modus ist. Aber sobald ein Fehler passiert, wird das Zustandsmodell  
201 entscheidend anders. Oder bei einer Wartungsarbeit ist das Modell auch anders. Und es gibt  
202 immer eine zweite Dimension. Also es gibt ein kontinuierliches Modell. Das gilt bei einem  
203 gewissen Zustand der Maschine. Und in anderen Zuständen wie das Hochfahren, Herunterfahren,  
204 vielleicht Putzen des Gerätes, Wartungsarbeiten, gelten andere Modelle. Und dieses Switchen  
205 vom Modell in bestimmten Bereichen ist wie eine zusätzliche Dimension. Und das ist immer  
206 brutal schwierig das abzubilden. Weil oft wird diese zusätzliche Dimension, wird ins Modell  
207 hinein gekocht und dann wird das Modell extrem komplex. Das ist was, was ich spannend finde,  
208 wie Sie so etwas lösen. Denn wenn das Modell all diese Zustände oder diese Phasen im Modell  
209 drinnen hat, dann explodiert das Modell. Und eigentlich sind die Modelle, kann man hier  
210 eigentlich vereinfachend sagen, dieses Modell gilt dann. Dieses Modell gilt da, gilt dann. Dieses  
211 Modell gilt dann. Weil die Maschine ist ja nicht gleichzeitig im operativen Modus und im  
212 Wartungsmodus. Das sind ja exklusive Zustände und diese Exklusivität kann man eigentlich wie  
213 da durch das Modellieren von verschiedenen Modellen, die dann wirken. Und dann muss man  
214 parallel aber definieren, wie man von einem Modellzustand in den anderen wechseln kann. Das  
215 sind die zwei Dimensionen und zwei unterschiedliche Sachen. Wie ist die Transition von einem  
216 Modellzustand in den anderen? Dazu muss es definierte Regeln geben, aber dann ist die  
217 Modellierung für einen bestimmten Zustand ist dann viel einfacher, weil man die beiden Dinge  
218 nicht ineinander mixt. #00:23:14-2#

219 29. I: Das ist spannend, dass Sie das ansprechen. Ich würde das vielleicht gerne dann nach den  
220 Beispielen nochmal aufgreifen, vielleicht, dass Sie das so als Bewertungspunkt auch nochmal  
221 hinzunehmen. Das schreibe ich mir auch auf, wenn wir die Modelle, die Beispiele gesehen haben.  
222 Es sind eigentlich zwei. Ein etwas einfacheres und ein etwas komplexeres. Und vielleicht, dass  
223 man das so mit diesen Gedanken im Hinterkopf auch betrachtet. Weil die Modelle zeigen  
224 eigentlich nur wirklich den kontinuierlichen Part, aber nicht wirklich die Prozessmodelle für das  
225 Anfahren, das Abfahren einer Anlage könnte man sagen. Aber das ist auf jeden Fall guter Input.  
226 (...) Dann komme ich jetzt zu unseren Erweiterungen. Ich werde Ihnen Prozesse zeigen, die mit  
227 BPMN 2.0 und mit unseren Erweiterungen modelliert wurden. Die Erweiterungen sollen zum  
228 einen vordefinierte Modellierungskonventionen für in der Prozess- und Steuerungstechnik  
229 übliche Routinen bereitstellen und zum anderen helfen, die Unterschiede zwischen den  
230 parallelen Pfaden in den Prozessmodellen zu visualisieren. Die Prozesse werden in der [REDACTED]  
231 [REDACTED], modelliert und für das Verständnis der [REDACTED] Modelle müssen  
232 drei zusätzliche Symbole erklärt werden. Das sind die Symbole für Service Calls, also Service  
233 Requests, für Script und für Service Request mit Script. Die habe ich Ihnen vor unserem Interview  
234 kurz erklärt. Würden Sie sagen, dass Sie das verstanden haben und dass Ihnen diese Symbole im  
235 Kopf geblieben sind? Und Sie wissen, wie die zu definieren sind? #00:25:07-9#

236 30. B: Ja, die drei Symbole. Kann (unv.) #00:25:11-9#

237 31. I: Okay, wunderbar, gut. Dann können wir eh weitermachen. Dann komme ich direkt zu unseren  
238 Extensions. Und zwar haben wir am Anfang das Closed Loop Subsystem Gateway. Das Gateway  
239 ist eine Kombination aus einem inklusiven und einem ereignisbasierten Gateway. Es enthält  
240 Verzweigungen beziehungsweise Kanten, die für die Zustandsabfragen und Regulierungsphasen  
241 des Zyklus ausgelöst werden, sowie Kanten, die beim Empfang von Abbruchereignissen  
242 ausgeführt werden. Die Ereignisse und Tasks in den einzelnen Kanten sind unabhängig  
243 voneinander. Damit erfüllen wir das erste oder oben genannten Feature, quasi, dass einzelne  
244 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 beziehungsweise Mess- und Steuerungsaufgaben, wie man es auch in der Technik sagen würde. Wir haben die Möglichkeit zu sagen, wir haben bei der Interval Duration Overrun einerseits Cancel und Wait. Und bei Measure Control Cycle Execution parallel oder sequentiell. Als Beispiele. Wenn Wait gewählt wird, beginnt die nächste Iteration, wenn alle Verzweigungen beendet sind und die festgelegte Intervalldauer erreicht ist. Bei Cancel definiert die Intervalldauer genau die Zeit, in der jeder Zweig zu beenden ist. Wenn die Tasks in einem Zweig schneller beendet werden und der Zweig warten. Wenn noch nicht alle Tasks beendet sind, werden sie abgebrochen. Also das heißt, Wait würde warten. Wirklich jeder einzelne Strang, jede einzelne Kante muss fertig werden. Cancel würde aber beenden. Je nachdem, wie die Zeitvorgabe ist. Bei parallel und sequentiell haben wir die Möglichkeit, einerseits zu entscheiden mit parallel, die Tasks werden nach Measure und Control Events parallel zueinander ausgeführt. Oder bei sequentiell. Die Tasks werden nach Control Events erst ausgeführt, nachdem alle Tasks nach Measure Events beendet sind. Das heißt, das würde auch ein bisschen die Funktion erfüllen, wenn man bei einer SPS (Speicherprogrammierbare Steuerung) zum Beispiel das erste Abbild macht und alle Input-Variablen einmal geschrieben hat, alle Werte einmal speichert und damit erst den ganzen Zyklus dann ablaufen lässt. Das wäre quasi erst einmal messen und warten, bis alle Messaufgaben fertig sind. Und dann erst die entsprechenden Regulierungen oder Regelaufgaben anwenden. Das wäre die Variante. Die drei weiteren Erweiterungen, die wir gemacht haben, kommen aus der Reihe der Intermediate Catching Events. In einem Closed Loop Subsystem werden spezifische Ereignisse erwartet, die in eine der drei folgenden Kategorien fallen. Ereignisse für Zustandsabfragen beziehungsweise Messungen, Ereignisse für Regelungen beziehungsweise Regulierungen und Ereignisse für die Unterbrechung des Closed Loop Subsystems. Das heißt, wenn der kontinuierliche Verlauf abgebrochen wird. 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 den Kanten danach angeordnet sind, ausgeführt. Das heißt, die Ereignisse zeigen an, 'Okay. Ab sofort wird der Prozess-Flow in dieser Kante weiter verfolgt.' Und das Bild, das jetzt gleich kommt, in dem sehen Sie ein Closed Loop System, in dem nur Ereignisse der drei Kategorien, ohne darauf folgende Task modelliert sind. Das ist ein Ausschnitt aus der . Das heißt, wir haben einerseits das Closed Loop Subsystem, das durch dieses Symbol dargestellt wird. Einerseits die Einführung des Symbols für Measure-Aufgaben beziehungsweise Zustandsabfragen. Kurze Zwischenfrage. Können Sie das erkennen? Ist das für Sie groß genug oder? #00:29:43-7#

32. B: Ja, ist gut. Ist gut lesbar. #00:29:45-1#

33. I: Passt. Dann. Das zweite Symbol für Control, also für Regulierungen, und das dritte Symbol mit dem Kreuz in der Mitte für Abbruchbedingungen. Das sind jetzt nur die Ereignisse, nur die Events, ohne jegliche Task-Abfolge, die in den Kanten danach folgt. Die drei Ereigniskategorien, die ich erwähnt habe, die sind hier im Folgenden definiert. Wir haben einerseits Measure Events. Das empfängt Events für die Ausführung von Tasks in Messzyklen oder Zustandsabfragezyklen. Wir haben Control Events. Dieses Symbol empfängt Events für die Ausführung von Tasks in Regelzyklen oder Regulierungszyklen. Und wir haben Cancel Events für das Abbrechen von Closed Loop Subsystems. Diese Symbole geben den Zweck der nachfolgenden Aufgaben an. Die Tasks werden nur ausgeführt, wenn die Ereignisse ausgelöst werden. Das bedeutet, dass das Messereignis angibt, dass die nachfolgenden Symbole für Messabläufe beziehungsweise Zustandsabfragen zuständig sind oder diese darstellen. Und das Gleiche gilt dann für Regulierungs- beziehungsweise Kontroll- und Abbruchereignisse. Das heißt, alle Aufgaben, die danach laufen, zählen zu diesem jeweiligen Event. Für Zustandsabfragen und Regulierungen, Mess- und Steuerungsereignisse also, 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. Im nächsten Bild sehen Sie ein Closed Loop Subsystem mit

300 einem Task für eine Messung. In diesem Fall wird das Ereignis für die Messung alle zehn  
301 Sekunden getriggert und danach wird der Wert V 1 geholt beziehungsweise gemessen. Wait  
302 bedeutet hier, dass ein neuer Zyklus erst starten würde, wenn die Messung erfolgt. Das heißt,  
303 wenn der Prozess in dieser Kante abgeschlossen wäre. Mit Cancel wird nach zehn Sekunden  
304 automatisch der neue Zyklus gestartet. Das wäre diese Abbildung hier. Das heißt, wir haben im  
305 Grunde den gleichen Grundaufbau wie in der letzten Abbildung. Jedoch dass hier eine Aufgabe  
306 angefügt wurde. Nämlich ein HTTP Call oder ein Service Call für das Messen eines Prozesswertes  
307 V 1. Das heißt, hier würde nur eine Messung abfolgen. #00:32:50-5#

308 34. B: Das A 2 heißt was? #00:32:55-1#

309 35. I: Das A 2 ist nur der Indikator für die zweite Aufgabe hier. Also, das ist eigentlich nur ein  
310 laufender Index. #00:33:02-2#

311 36. B: Okay. #00:33:03-5#

312 37. I: Das sind die Bezeichner für Measure Events. Man kann einen Bezeichner seitlich auch einfügen  
313 für die jeweilige Aufgabe, also ein Label, könnte man sagen. Und das entsprechende Label auch  
314 für das Kontrollereignis. Wie gesagt, die Zyklen, die Zyklusdauer kann definiert werden. Wir  
315 haben uns hier für die Darstellungsweise über Frequenz, also mit Hertz, entschieden. Wenn man  
316 das Measure Event genauer definieren möchte, kann man mit einem seitlichen Fenster die  
317 Attribute hier festlegen. Also hier könnte man die Frequenz anpassen, erhöhen oder auch  
318 herunter setzen beziehungsweise kann man hier auch definieren, wenn man eine globale  
319 Variable hat. Um welchen Wert es sich hier handeln würde, der sich ändert. Das heißt, wenn ich  
320 V 1 definiert habe, würde genau dieser Wert damit sich ändern im Verlaufe dieser Kante. (...)   
321 Wenn man mehrere Calls hätte, die verschiedene Werte hintereinander abfragen, könnte man  
322 hier auch noch, wenn man das möchte, einen weiteren Wert hinzufügen. Ansonsten, wenn die  
323 Werte gleichzeitig ausgelesen werden sollen, könnte man hier eine neue Kante einfügen und  
324 hätte ein zweites Measure-Ereignis drinnen. Man kann bei diesem Measure-Ereignis die gleiche  
325 Frequenz angeben. Man kann aber auch eine andere Frequenz angeben, wenn man das möchte.  
326 Bei Measure kann man neben der Frequenz des Messereignisses den Wert, der sich während des  
327 Prozesses ändert, definieren, wie eben gesehen. Mit Hilfe von Regelungsereignissen kann man  
328 ferner festlegen, welches Reglermodell man verwenden möchte. Also zum Beispiel PID, PI, PD.  
329 Wie auch immer. Diese Regler werden aber in ihrer mathematischen Form dargestellt. Die Tasks  
330 für sie sind im Grunde Berechnungen, die in festen Teilprozessen dargestellt werden. In unserem  
331 Fall würde ein Script-Symbol hergenommen werden, in dem die mathematische Formel für das  
332 jeweilige Regelmodell drinnen steht und abgearbeitet werden würde. Wie in der Art und Weise,  
333 wie wenn man einen Regler zum Beispiel mithilfe von MATLAB darstellen möchte und berechnen  
334 möchte. Nach diesen Berechnungen kann der Benutzer Tasks zu weiteren Datenverarbeitung  
335 auch hinzufügen. Dies kann auch nach Mess-Tasks geschehen, die man auch als  
336 Datenerfassungs-Tasks bezeichnen könnte. Hier sehen Sie ein Prozessmodell mit einem Wert,  
337 der gemessen wird und einer darauf folgenden Regelung. Das heißt, wir haben wieder in der  
338 ersten Kante unseren Wert, der gemessen wird. Hier haben wir dann wieder den Index, je  
339 nachdem wie das eingefügt wurde. Bei Control hätten wir bei dem ersten Skript mal, dass die  
340 Differenz von V optimal und V 1, also  $V_{opt} - V_1$  berechnet wird. Dann hätten wir den Code für  
341 den PID-Regler. Und dann könnte man sagen, dass man mit einem Service Call definieren  
342 möchte, dass gleich wieder ein Wert nach außen geschickt wird als Befehl an einen Aktuator  
343 oder wie man auch immer das definieren möchte. Also man könnte hier frei nach dem Ereignis,  
344 nach dem jeweiligen Event-Symbol, den Verlauf des jeweiligen Prozesses, der hier ablaufen soll,  
345 weiter modellieren mit den herkömmlichen Methoden. (...) Wir haben vorher über die globalen  
346 Datenelemente beziehungsweise die globalen Variablen gesprochen. Das würde man in einem  
347 eigenen Fenster hier definieren bei der . Als Beispiel. Also es gibt einen Reiter Data  
348 Elements. Gibt es auch noch Endpoints. Bei Data Elements kann man wie gesagt, entweder einen  
349 Default Value, der von Anfang an drinnen stehen soll, definieren oder man kann es auch einfach  
350 mit Null definieren und das wird dann sowieso im Laufe des Prozesses überschrieben. Diese  
351 Möglichkeit gibt es halt. Wenn man Control, also dieses Ereignis hier, für Regulierungen genauer  
352 definieren möchte, kann man genauso die Frequenz hier festlegen. Man kann aber auch wie

353 gesagt beim jeweiligen Regler sagen, welcher Wert würde dann ausgerechnet werden und  
354 welches Upper Limit beziehungsweise welches Lower Limit hätten wir. #00:38:07-5#

355 38. B: Das sind Parameter, die dann für das Script entsprechend passen müssen, oder? #00:38:13-9#

356 39. I: Genau, die könnte man/ #00:38:14-8#

357 40. B: Das Regler-Script wäre jetzt/ wäre halt/ stellen die zur Verfügung und das arbeitet mit diesen  
358 Parametern. #00:38:22-0#

359 41. I: Genau. Einerseits natürlich mit denen, die hier oben/ Entschuldigung. Wo habe ich das? Ja,  
360 genau. Einerseits mit den Parametern, die wir hier festgelegt haben und andererseits würde es  
361 aber auch jeweils, also wirklich das Script vom jeweiligen Regler, mit diesen Parametern hier  
362 arbeiten. Also das kann man hier auch definieren. Haben wir so vorgesehen. Wait bedeutet  
363 wieder, dass für den nächsten Zyklus auf das Beenden aller Tasks gewartet wird. Das heißt,  
364 einerseits muss die Messung natürlich fertig sein, andererseits muss die Regulierung erfolgt sein.  
365 Sequentiell heißt, dass die Tasks nacheinander ausgeführt werden. Das heißt, es müssen erst alle  
366 Tasks in der Messungskante, in der Zustandsabfragekante, erledigt sein. Und dann würden erst  
367 die Control-Tasks erfolgen. Dabei wird vom, wie gesagt/ Bei diesem Prozess wird vom optimalen  
368 Wert der aktuelle Wert V 1 abgezogen. Die Differenz dafür ist die Basis, um den neuen Stellwert  
369 dann auszurechnen. Und dieser wird mit einem Service Request an das jeweilige Stellglied, wie  
370 auch immer, geschickt. Also hier geht es wirklich nur um die Prozesslogik. Hier geht es aber nicht  
371 darum, was dann jeweils im Backend geschieht. Wenn man möchte, kann man die  
372 Differenzberechnung, das mathematischen Modell des Reglers und das Aussenden des Befehls in  
373 einem Prozess zusammenfassen und das als Subprozess in diese Prozesskante einfließen lassen.  
374 Bei Control können zusätzlich die Art der Regelung sowie der neue Stellwert und dessen Limits  
375 eingetragen werden. Das haben wir gerade gesehen. Und würde hier parallel verwendet werden,  
376 würde der letzte Wert von V 1 genommen werden, für den natürlich dann noch keine  
377 Zeitgarantie besteht. Das heißt, man würde jetzt nicht warten, dass die Messung, die zuletzt  
378 gemacht wurde, wie bei einer SPS zum Beispiel, dass man dieses Abbild nimmt und damit  
379 arbeitet. Auf das würde man nicht warten, sondern man würde einfach den zuletzt  
380 überschriebenen, den aktuellsten Wert von/ den jeweiligen V 1 hier übernehmen für die  
381 Berechnung. Zustandsabfragen und Regulierungen sollen in regelmäßiger Frequenz ausgelöst  
382 werden. Abbruchereignisse natürlich nicht. Abbruchereignisse werden nur durch ihre  
383 Abbruchbedingungen ausgelöst, die der Benutzer definieren kann. Ein Beispiel für ein  
384 Abbruchereignis wäre, wenn etwas den Abbruch eines Zyklus auslöst, zum Beispiel  
385 Watch-Dog-Funktion zur Überwachung der maximalen Zykluszeit. Oder es könnte also, wenn  
386 man hier sagt, das würde jetzt etwas abbrechen, zum Beispiel also der Zyklus dauert einfach zu  
387 lang. Oder es könnte auch der Befehl eines Not-Stopp-Schalters oder so etwas sein, wenn man  
388 das so hinein modellieren möchte. In dem Bild sehen Sie einen Prozess mit einem zu messenden  
389 Wert, eine Regelung und eine Abbruchbedingung. Das heißt, wir haben wieder einen Wert  
390 gemessen. Wir haben unsere Scripts, die ausgeführt werden, nach dem Control-Ereignis und  
391 dann den Befehl, der den jeweiligen neuen Werte nach außen schickt, nach außen an den  
392 Aktuator weiter gibt. Und unsere Abbruchbedingung wäre, wenn der Emergency-Stop auf True  
393 gesetzt wird. Das heißt, wenn der Emergency-Stop aktiviert wird. Als Beispiel. (...) Wenn ich zu  
394 schnell bin, sagen Sie es bitte. #00:42:03-8#

395 42. B: Nur eine Frage. Dieses Process Control Symbol mit dem Pfeil nach oben, das ist dann  
396 eigentlich der Loop? Oder wie muss ich das verstehen? #00:42:14-4#

397 43. I: Meinen Sie, dass hier mit den Wellen? #00:42:17-6#

398 44. B: Ja, genau. #00:42:18-9#

399 45. I: Ja, das ist die Loop. Genau. Das würde dann immer wieder ablaufen. Also dieser Pfeil nach  
400 oben soll indizieren, dass das hier wieder dann zum Anfang an zurück geht. #00:42:30-6#


---



401 46. B: Und jetzt wie/ Also jetzt beim Notaus. Bei diesem Abbruch-Event fällt er ja eben unten heraus,  
402 oder? #00:42:43-6#

403 47. I: Genau. Er würde hier heraus springen. Und wir haben jetzt für diese Bedingung hier keine  
404 Aufräumaufgaben definiert. Das heißt, das Ereignis wird zwar ausgelöst. Da aber hier keine  
405 zusätzlichen Aufgaben mehr angehängt sind, wird er direkt einfach den Prozess beenden. Und  
406 danach würde dann ein, ich sage dazu, ein diskreter Prozess erfolgen, der dann gewisse Befehle  
407 ausführen würde oder gewisse Aufgaben ausführen würde. Das könnte man machen.  
408 #00:43:16-2#

409 48. B: Okay. #00:43:19-1#

410 49. I: Hier haben wir wieder eine Variable hinzugefügt, also mit zum Beispiel Emergency-Stop, die  
411 eigentlich normalerweise nicht aktiviert ist, also normalerweise auf false stehen würde. Jetzt  
412 kommt das. Nachdem das Ereignis ausgelöst wurde, können Task zur Aufräumroutine  
413 abgearbeitet werden, bevor der Zyklus komplett beendet wird. Oder wie gesagt der Prozess  
414 vollständig beendet wird. Bei diesem Prozess hier, bei dem Beispiel, sehen wir, dass wir  
415 Aufräum-Tasks eingeführt haben und definiert haben. Das heißt, wir haben wieder ganz normal  
416 unseren Prozess, wo etwas gemessen wird, wo ein PID-Regler angewendet wird und wo wir  
417 dann, nachdem die Abbruchbedingung erfolgt, also nachdem diese Abbruchbedingung erfüllt  
418 wird, die Aufgabe Initiate shutdown routine in vessel one durchgeführt wird. Das heißt, hier wird  
419 dann ein Subprozess gestartet, der für die Shutdown/ für den Herunterfahrzyklus oder die  
420 Herunterfahrroutine im ersten Kessel, im ersten Tank, zuständig ist. Nur so als Beispiel. Also ich  
421 hab das jetzt hier ganz einfach dargestellt. Man könnte es aber natürlich auch noch komplexer  
422 definieren beziehungsweise wirklich einen Standardsubprozess hier anhängen, wenn man das  
423 möchte. Die vorgestellten Erweiterungen sollen bei der Modellierung von kontinuierlichen  
424 Prozessen helfen, indem Vorlagen für die Erstellung von Prozessmodellen vorgegeben werden.  
425 Und andererseits durch die Darstellung als Closed Loop Subsystem mit eigenen Symbolen für  
426 Zustandsabfrage-, Regulierungs- und Abbruchereignisse helfen, solche Prozesse leichter  
427 nachvollziehen und darstellen zu können. Hinzu kommt, dass man für eine übersichtlichere  
428 Darstellung des gesamten Prozesses auch Subprozesse zur Unterteilung nutzen kann. Und damit  
429 erfüllen wir dann auch natürlich unseren Wunsch oder unsere Bedingung, dass man auch zur  
430 Verständlichkeit des Gesamtmodells beiträgt. Jetzt kämen wir auch zu konkreten Beispielen. Wir  
431 haben zwei Beispiele. Die werde ich Ihnen jetzt zeigen. Die haben wir in der  mit unseren  
432 Erweiterungen modelliert. Ich hätte gerne, dass Sie sich die Modelle ansehen und mir sagen, was  
433 Sie aus ihnen herauslesen können und ob die Modelle den notwendigen Informationsgehalt für  
434 die Modellierung der zugrundeliegenden Regelungsprozesse erfüllen. Vorab wird Ihnen zum  
435 jeweiligen Prozess erklärt, was abgebildet werden soll. Und ich würde Sie einfach bitten, geben  
436 Sie mir offenes Feedback. Beziehungsweise nachdem wir die jeweiligen Beispiele genauer  
437 betrachtet haben, gibt es dann auch noch so eine Tabelle, die vorgibt, wie man das Ganze  
438 bewerten kann. Also. So ein paar Anhaltspunkte, wo ich Sie bitten würde, einfach Ihre Meinung  
439 dazu zu sagen oder zu nennen. Das erste Beispiel. Bitte bewerten Sie das folgende Modell. Wir  
440 haben hier eine einfache PI-Temperaturregelung für einen Wärmetauscher basierend auf einem  
441 Beispiel aus der MathWorks Bibliothek. Die Temperatur einer Flüssigkeit/ Also hier sehen Sie das  
442 Flowchart des Prozesses. Die Temperatur einer Flüssigkeit in einem Rührkessel wird mittels  
443 Wärmetauscher kontrolliert. oder verändert. Der über den Wärmetauscher eingebrachte  
444 Wärmestrom wird über ein Ventil, das den Dampfstrom kontrolliert, gesteuert. Der zu beachten  
445 störende Umgebungseinfluss ist die schwankende Temperatur der zugeführten Flüssigkeit. Das  
446 heißt, wir gehen nicht davon aus, dass die äußere Temperatur die Temperatur im Tank, im  
447 Rührkessel, irgendwie beeinflusst. Also wir gehen davon aus, dass der Tank isoliert ist. Hier  
448 haben wir wieder die Darstellung der Datenelemente. Sind verschiedene Variablen. Die  
449 Variablen werden dann entsprechend auch im mathematischen Modell des Reglers verwendet.  
450 Wir können dann gerne nochmal in diese Grafik zurückspringen zur Verständlichkeit, wenn das  
451 hilft. Wir haben in dem Fall auch Endpoints, das heißt, die jeweiligen Adressen nach außen hin  
452 definiert. Also woher wir unsere Werte vielleicht holen oder wohin wir gewisse Werte schicken.  
453 Diese werden mir hier unterscheiden mit HTTP Get oder HTTP Post. Also man kann auch gleich  
454 die Adressen hier definieren. Das Erste, was wir hier vorgegeben haben, ist, dass wir wait und

455 sequential eingestellt haben. Und unser Prozessmodell sehen Sie dann hier. Das werde ich  
456 vielleicht ein bisschen heraus zoomen. Genau. So sieht man es auf einer Seite. Das heißt, wir  
457 haben zuerst einmal Wait. Wir haben jetzt eigentlich sogar zwei Kanten mit Measure Events.  
458 Einerseits die Temperatur aus Tank Eins. Also wir überprüfen einmal den Zustand Temperatur  
459 Tank Eins. Dann verwandeln wir, wenn wir das wollen, diesen Prozesswert noch irgendwie um  
460 beziehungsweise rechnen ihn um, je nachdem wie wir ihn vom jeweiligen Sensor bekommen.  
461 Vielleicht müssen wir ihn noch irgendwie post processen (nachbearbeiten). Dann bekommen wir  
462 die Temperatur von Tank Zwei. Vielleicht müssen wir hier nichts weiter tun. Vielleicht hat der  
463 jeweilige Temperatursensor ein eigenes Modul, mit dem der Wert noch irgendwie umgewandelt  
464 werden kann und die jeweilige Schnittstelle schickt schon den richtigen Wert. Oder wir haben  
465 das Ganze einfach in einem Subprozess dargestellt. Wir haben dann Control. Wir haben einen  
466 PI-Controller. Wir haben die Umwandlung von PI-Controller MW Two Output, also den  
467 Manipulating Value in den jeweiligen Ausgangswert, der wirklich auch nach außen geschickt wird.  
468 Und diesen nennen wir dann Manipulating Value U Eins. Und der wird dann wirklich an den  
469 Aktuator jeweils geschickt. Das heißt, in dem Fall ist das die Input-Spannung, also der Input-Wert  
470 für die Spannung, für den Motor am Ventil für den Dampfstrom, für die Dampfstromzufuhr.  
471 Dann haben wir eine Abbruchbedingung. Die habe ich einfach Stop Activated genannt. Das ist,  
472 wenn der Stopp irgendwie aktiviert wird. Das heißt, wenn von außen das Kommando  
473 kommt, 'Halte den Prozess an.' Wenn der auf true gesetzt wird, wird dann die jeweilige  
474 Shutdown-Sequenz des Prozesses durchgeführt. Das ist in dem Fall ein einfaches Script. Man  
475 kann aber auch das natürlich mit Service Calls definieren. Man kann wieder einen Subprozess  
476 hier modellieren, wie man das auch möchte. #00:50:31-5#

477 50. B: Ich habe mal eine Frage. Die zwei Input Values. Das eine ist die Tanktemperatur und das  
478 andere ist? #00:50:38-9#

479 51. I: Das ist die Disturbance. Entschuldigung, das habe ich vielleicht oben nicht erwähnt. Und zwar  
480 vom Input Flow. #00:50:47-8#

481 52. B: Der Volumenstrom vom Input Flow? #00:50:51-5#

482 53. I: Das ist die Temperatur vom Input Flow. #00:50:53-6#

483 54. B: Die Temperatur. Und der Volumenstrom selber, den kennt man? #00:51:01-2#

484 55. I: Vom Einfluss, vom jeweiligen Zufluss nicht. Man kennt jetzt hier nur die Temperatur. Also hier  
485 wird nur die Temperatur gemessen. Man kann aber natürlich den Input Flow, wenn man das als  
486 Sensor auch zur Verfügung hat, auch messen. Es ist wirklich nur ein ganz einfaches Beispiel, das  
487 dafür stehen soll, dass hier eine Störgröße in das System eingebracht wird, über diesen  
488 Volumenstrom. Aber die Steuergröße soll halt dargestellt werden durch die Störung der fremden  
489 Temperatur, also dieses Zuflusses. Aber natürlich kann man auch den Zuflussstrom überwachen,  
490 wenn man das möchte. #00:51:41-7#

491 56. B: Weil die ja/ Genau. Ist/ (lacht) Ein vereinfachtes Modell. #00:51:48-2#

492 57. I: Ja. #00:51:49-1#

493 58. B: Weil jetzt der Volumenstrom ist natürlich in dir/, also ist/ #00:51:53-5#

494 59. I: Wäre natürlich für die // B: Maßgebend. // Level-Haltung auch wichtig ja. #00:51:57-5#

495 60. B: Auch abhängig. Ja. // I: Natürlich, ja. // Die Temperatur/ Ja. Die Temperatur alleine ist jetzt  
496 relativ. // I: Ja, nat/ // Außer der Volumenstrom sei konstant. Dann. #00:52:08-8#

497 61. I: Das sind Bedingungen, die man natürlich dann hier auch entsprechend modellieren müsste,  
498 aber dann müsste man das Modell dann auch genauer definieren. Also das ist die Vorgabe, die  
499 man auch dazu hat. #00:52:22-3#

---

500 62. B: Ja, ja. Nein, passt schon. Passt schon. #00:52:24-9#

501 63. I: Aber das ist gut, dass Sie das sagen, weil gewisse Annahmen muss man dann einfach  
502 detaillierter dazu auch hier her schreiben. Das stimmt schon. #00:52:32-5#

503 64. B: Ja, also den Volumenstrom braucht man um die Füllmenge // I: Natürlich, ja. // des Tanks.  
504 Und, ja. Das Rührwerk kommt vermutlich auch nirgends vor. Aber dann nehmen wir, dass sich  
505 die Temperatur homogen verteilt. #00:52:50-9#

506 65. I: Es gibt gewisse ideale Annahmen, die wir hier treffen für die einfache Darstellung des Modells,  
507 ja. Weil auch in dem Beispiel, das wir als Basis für dieses Modell hergenommen haben, auch  
508 nicht genauere Angaben natürlich getroffen werden. Also man kann/ Die Vorgabe dieses  
509 Beispiels aus der MathWorks Bibliothek ist leider nicht so detailliert, dass wir darauf eingehen  
510 können, sondern es geht eher darum, dass man grundsätzlich sieht, man könnte diese Logik  
511 dieses Prozesses so darstellen. Also diese Möglichkeit gibt es. Aber ja, natürlich, vollkommen  
512 technisch darstellbar nur durch dieses Modell wäre es nicht, weil gewisse zusätzliche  
513 Informationen fehlen. Hier nur so als Beispiel, was man seitlich auch noch definieren kann bei  
514 den Attributen der einzelnen Elemente. Die ID A Drei. Das würde zum Beispiel hier zu Conversion  
515 to Process Value gehören. Hier habe ich einfach etwas gleichgesetzt, also eine Variable auf die  
516 andere Variable überschrieben. Aber natürlich kann man hier alles mögliche hineinschreiben im  
517 Script, was man möchte. Und als Beispiel halt natürlich das mathematische Modell des Reglers.  
518 Wenn man weiß, wie es sich verhält und wenn man weiß, wie man es in Code darstellen könnte,  
519 kann man das hier auch hinein programmieren und kriegt dann ein gewisses Ergebnis auch  
520 heraus. Also als Beispiel/ #00:54:19-2#

521 66. B: Der Integralwert, braucht es da noch einen zusätzlichen Speicher? Ist der in den globalen  
522 Variablen drinnen? #00:54:23-6#

523 67. I: Den kann man in den globalen Variablen modellieren. // B: Kann es auch Teil des Scripts sein?  
524 // Haben Sie jetzt die Variable dazu/ Bitte? #00:54:31-6#

525 68. B: Kann es auch Teil des Skripts sein, dieser lokale // I: Ja, natürlich. // Integralwert?  
526 #00:54:39-0#

527 69. I: Sagen wir es so. Wenn Sie das in einem anderen Script/ Also wenn Sie zum Beispiel das Ganze  
528 im Python Code oder in irgendeiner anderen Programmiersprache darstellen können, dann  
529 könnte man das in Ruby Code wahrscheinlich von der Logik ja genauso hier abbilden. Aber das  
530 ist nur eine Möglichkeit, weil unsere Workflow Engine halt natürlich mit Ruby funktioniert. Aber  
531 das Script könnte man mit anderen Möglichkeiten, mit anderen Programmiersprachen auch  
532 ausführen. Also wenn es in Code darstellbar wäre, könnte man es hier auch umsetzen. Das soll  
533 die Information daraus sein. #00:55:16-9#

534 70. B: Okay, aber der PID-Controller, das ist eine Struktur und die ist global definiert, oder? In Ihrem  
535 Fall. #00:55:25-3#

536 71. I: Was meinen Sie mit global definiert? #00:55:28-4#

537 72. B: Ja, die muss ja persistent sein für das Executen (Ausführen) dieses Modells. Der Integralwert,  
538 den man hier in PID, also im PID-Regler hat. Der muss ja remanent sein, so dass es beim  
539 nächsten Zyklus, beim nächsten Controller-Zyklus, wieder vorhanden ist. #00:55:49-0#

540 73. I: Ach so, das meinen Sie. Ja, das kann man auf eine Variable überschreiben, die dann im  
541 nächsten Zyklus wieder vorhanden ist und mit der man dann wieder weiter arbeitet. Ja.  
542 #00:55:58-1#

543 74. B: Genau. #00:55:58-9#

---

544 75. I: Das geht. Ja. Andere Variablen/ #00:56:01-7#

545 76. B: Das muss (unv.) sein, sonst geht es nicht. #00:56:03-3#

546 77. I: Ja, klar. (lacht) Das stimmt, ja. Aber andere Variablen natürlich, die sich jetzt speziell im Zyklus  
547 ändern. Die würden dann überschrieben werden. Ja. (...) Ja, das ist mal die einfache Variante  
548 oder ein einfacheres Modell. Und da würde ich Sie vielleicht dann kurz bitten. Wir können auch  
549 gerne wieder hinauf zur graphischen Darstellung wechseln. Vielleicht können Sie mir auf einer  
550 Skala von Eins bis Fünf, also Eins, wir gehen jetzt mal davon aus, dass mehr Punkte positiver sind.  
551 Eins, von sehr schlecht, bis Fünf, sehr gut, bewerten. Und zwar einerseits bei diesem speziellen  
552 Modell die Verständlichkeit, ob Sie sagen würden, dass man versteht, im Groben, was für ein  
553 Prozess hier abläuft? Dann das zweite wäre die/ Ich gehe jetzt einmal kurz alle Kriterien durch.  
554 Dann das zweite wäre die Übersichtlichkeit. Ob man das Gesamtsystem in einem Blick erkennen  
555 kann, also ob es für einen möglich ist, auf einen Blick das Ganze nachzuvollziehen, was hier  
556 abläuft, beziehungsweise wie das Gesamtsystem sich abbildet. Die Einfachheit. Also könnte man  
557 das Modell vielleicht sogar noch einfacher gestalten oder ist das eigentlich schon sehr einfach?  
558 Dann die Logik. Wird klar, was parallel und was sequentiell passiert? Und dann der letzte Punkt  
559 wäre noch die Erweiterbarkeit. Könnte man dem Modell noch etwas hinzufügen, was den  
560 Informationsgehalt verbessern würde? Sie haben schon erwähnt, man müsste jetzt hier natürlich  
561 für diese speziellen physikalischen Prozessen noch einige zusätzliche Parameter einfügen  
562 beziehungsweise einige andere Einflüsse natürlich auch noch mit berücksichtigen. Aber wenn  
563 man es jetzt nicht von dem Beispiel hier konkret vom physikalischen her sehen würde, sondern  
564 generell was, wenn wir jetzt davon ausgehen, wir hätten den Rührer berücksichtigt, wir hätten  
565 den Volumenstrom drinnen. Könnte man eventuell rein von der Modellierungsvariante könnte  
566 her noch etwas hinzufügen? Das wäre die Frage eigentlich. Das heißt, zurück zum ersten Punkt.  
567 Verständlichkeit. Würden Sie sagen, dass das Modell verständlich war? #00:58:23-0#

568 78. B: Ja, also wo ich Mühe habe ist mit dem Abbruchereignis. Also mit der Darstellung. Die drei  
569 oder, die Kanten, die vier, die gehen alle unten hinein, oder? #00:58:37-9#

570 79. I: Genau, ja. #00:58:37-9#

571 80. B: Jetzt, der Abbruch führt eigentlich zum Stopp. Und die anderen führen, dass es wieder oben  
572 beginnt. #00:58:44-8#

573 81. I: Genau ja. #00:58:45-5#

574 82. B: Und wenn ich das das erste Mal sehe, ich sehe keinen Unterschied zwischen den vier Kanten,  
575 die unten ankommen. Ich sehe nur die Parallelität. Wieso jetzt das Abbruchkriterium, das  
576 nachher gefolgt wird von einem Script. Wieso, dass das zu einem Stopp führt und die anderen  
577 Kanten, aber zu einem Start. Das ist für mich/ Entweder man muss es wissen, aber so grafisch  
578 sichtbar ist es nicht. #00:59:15-1#

579 83. I: Also nicht intuitiv quasi, könnte man sagen. #00:59:18-1#

580 84. B: Ja, genau. #00:59:18-5#

581 85. I: Okay, gut. Das heißt, auf der Skala von Eins bis Fünf, wobei Fünf das Beste ist. Was würden Sie  
582 sagen? #00:59:26-5#

583 86. B: Ja, also Vier ist okay. Weil der Abbruch ist eh etwas Spezielles. Ist die Frage, ob man den  
584 vielleicht nicht anders darstellt. Aber jetzt was/ von der Parallelität her, vom Trigger, vor allem  
585 vom Messen und Controller, das ist verständlich, die Sequenz der Scripte. #00:59:45-3#

586 87. I: Okay, Dankeschön. Super. Was würden Sie von der Übersichtlichkeit her sagen? Also Sie haben  
587 gerade gemeint, auf den ersten Blick ist für Sie nicht erkenntlich, dass es zum Beispiel auch einen  
588 Abbruch im System gibt. Das heißt, es würde eigentlich ein wichtiges Element des Systems

589       fehlen. Aber, was würden Sie von der Gesamtheit des Prozesses sagen? Würden Sie sagen, man  
590       kann es gut auf einen Blick eigentlich erkennen, was passiert? #01:00:15-9#

591   88.   B: Das schon. Ja. #01:00:18-2#

592   89.   I: Okay, das heißt Vier oder Fünf oder? #01:00:23-6#

593   90.   B: Ja, auch Vier. Ein wenig überladen ist es dann schon. (lacht) #01:00:28-9#

594   91.   I: Okay, ja dann. #01:00:30-6#

595   92.   B: So etwas einfaches, ja. #01:00:32-7#

596   93.   I: Dann bin ich gespannt, was Sie zum nächsten Modell dann sagen. Gut. Einfachheit. Wenn Sie  
597       schon meinen, es ist eher überladen. Könnte man vielleicht das Modell noch etwas einfacher  
598       darstellen? #01:00:46-2#

599   94.   B: Ja, ich sehe noch keinen Bedarf für das Unterscheiden von Messen und Steuern. Für diese  
600       zwei Symbole. Weil eigentlich ist es nur wie ein zeitkontinuierlicher Trigger, den man auch  
601       abstrahieren könnte als zeitkontinuierlicher Trigger. Ich habe jetzt noch keine Differenzierung  
602       gesehen im Verhalten dieser zwei Elemente. #01:01:09-7#

603   95.   I: Okay. Also im Verhalten. Ich/ Nur als Frage. Wie würden Sie/ Was hätten Sie erwartet, wenn  
604       man die unterschiedliche Verhaltensweise darstellen möchte? Oder welches Verhalten hätten  
605       Sie erwartet? Wenn man jetzt sagt, das ist ein/ das System, das dieses Prozessmodell umsetzt,  
606       ist eigentlich nur die übergeordnete Steuerung, die das los stoßen soll. Welches Verhalten  
607       hätten Sie erwartet? #01:01:48-4#

608   96.   B: Ja, was ich/ was man bei Regelungen braucht, ist ein zeitkontinuierlicher Trigger. Diese  
609       Abtastzeit, dieses zeitlich konstante Triggern. Und diese Eigenschaft haben beide, beide dieser  
610       Elemente. Also sowohl Messen wie Control haben ja als Parameter diese Zeit, diese Frequenz.  
611       Aber sonst haben sie außer dem grafischen Aussehen haben sie ja keine unterschiedliche  
612       Eigenschaft. #01:02:16-3#

613   97.   I: Na ja, sie haben ein bisschen unterschiedliche Attribute. #01:02:21-1#

614   98.   B: Die habe ich noch nicht gesehen. (lacht) #01:02:24-9#

615   99.   I: Okay, dann gehen wir kurz nochmal hinauf. (...) So, ganz kurz. Tut mir leid. Dass ich so weit  
616       springen musste. Nochmal zu diesem Abschnitt hier, wo ich die Erweiterungen erkläre. Das hier.  
617       Diese Grafik ist die genaue Definition von Measure, von diesem Ereignis. #01:02:53-9#

618   100. B: Genau, ist ein Attribut mit der Einheit Hertz. #01:03:00-7#

619   101. I: Genau. Wir können definieren, welcher Wert hier verarbeitet wird oder geändert wird oder  
620       überschrieben wird. Sagen es so. Als Differenz DAZU oder im Kontrast dazu. Wo war das jetzt?  
621       Bin ich schon wieder darüber gesprungen? Haben wir die Möglichkeit, bei Control auch einiges  
622       zu definieren. Genau ja, das war diese Grafik hier. Das heißt, DAS sind die/ #01:03:28-1#

623   102. B: Das sind die Attribute vom Script, nicht vom Control. #01:03:32-4#

624   103. I: Das sind die/ das sind die Attribute vom Control. Control definiert, welcher PID-Regler/ Dann  
625       habe ich das falsch verstanden, Ihre Frage, und habe es falsch beantwortet. Das tut mir leid.  
626       Control kann hier definieren, welche Regler danach kommt. Man kann bei Control gleich  
627       hinzufügen, okay, man hat einerseits einen PID-Regler nach diesem Ereignis. Und zu diesem  
628       PID-Regler gehören die Attribute, die man hier noch weiter definiert. Das heißt, ja, natürlich. Im

629 Script drinnen ist das mathematische Verhalten, das mathematische Modell vom PID-Regler  
630 definiert, aber diese Attribute gehören eigentlich zum Control-Ereignis. #01:04:20-4#

631 104. B: Und das Measurement-Ereignis hat keine Attribute? #01:04:27-3#

632 105. I: Das Measurement-Ereignis hat nur die Möglichkeit zu definieren, genauso wie das Control, in  
633 welcher Frequenz es arbeitet und zu definieren, welcher Wert sich ändert. (...) Das ist der  
634 Unterschied. #01:04:43-7#

635 106. B: Welcher Wert sich ändert? #01:04:45-5#

636 107. I: Genau. Welcher Wert überschrieben wird. #01:04:47-9#

637 108. B: Und das Script kriegt dann diesen Wert als Parameter, damit es diesen dann überschreiben  
638 kann? #01:05:06-8#

639 109. I: Das Script arbeitet mit diesem Parameter weiter, je nachdem, was man definiert. Ja. Wie  
640 gesagt, man kann jetzt auch das gesamte mathematische Verhalten des Prozesses in ein Script  
641 zusammenfügen. Man kann aber auch sagen, 'Ich möchte alleine das mathematische Modell des  
642 PID-Reglers in einem Script haben.' Oder ich möchte wirklich jeden Einzelschritt vielleicht mit  
643 einem eigenen Script haben. Je nachdem, wie man das modellieren möchte. #01:05:40-0#

644 110. B: Aber entscheidend ist, man/ diese Liste von (unv.) kann ich selber definieren. Die kann ein  
645 oder mehrere oder keine Elemente enthalten. #01:05:48-9#

646 111. I: Ja. #01:05:49-8#

647 112. B: Und beim Control-Element kann ich die Attribute, kann ich eigentlich auch selber definieren?  
648 #01:05:57-6#

649 113. I: Genau, ja. #01:05:59-2#

650 114. B: Da ist jetzt so ein konkretes Beispiel mit Controller Type, Value, Upper und Lower Limit.  
651 #01:06:04-9#

652 115. I: Nur als Beispiel. Ja. (...) Das soll/ #01:06:14-1#

653 116. B: (Unv.) Als böse Frage, wenn ich jetzt ein Measurement-Ereignis nehme und dort Controller  
654 Type, Value, Upper Limit und Lower Limit verwende und das gleiche Script unters Measurement  
655 hänge, dann würde technisch das gleiche passieren. #01:06:34-0#

656 117. I: Die Rechnung würde natürlich gleich ablaufen, aber es würde dann wenn man keinen/ Wenn  
657 man jetzt kein Service Call daran hängen kann mit einem Aktuator, wenn das nicht vorgesehen  
658 ist, dass man hier etwas nach außen hin schreibt. Dann würde das auf das System eigentlich  
659 keinen Einfluss haben. Das ist der Hinweis. Also diese Ereignisse sollen vor allem dazu dienen,  
660 den Modellierer zu sagen, ich kann in diesen Strang für Mess- oder Zustandsabfrageaufgaben  
661 Flows definieren oder Symbole einhängen. Und bei Control kann man dem User sagen, hier  
662 sollen eigentlich Ereignisse stattfinden, die dazu führen, dass etwas umgerechnet wird, dass man  
663 einen Wert bekommt, den man auch nach außen hin schickt. (...) Von der Definition her, also,  
664 wie es dann wirklich ausgeführt wird in der ■■■■■, sind wir noch nicht soweit, dass wir sagen, wir  
665 haben jetzt diese Restriktion eingeführt. Aber von der Modellierung her wäre das die Vorgabe an  
666 den User. Also DAS ist nur Zustandsabfrage. Wie schaut das System aus, das ich überhaupt für  
667 mein Subsystem erfassen kann? Okay, das ist jetzt der Zustand. Und das andere ist wirklich, wie  
668 interagiere ich. Also DAS sind wirklich die Anpassungen. Das soll eigentlich der Unterschied sein.  
669 Aber von der Granularität her, wie ich es dann machen möchte, das versuchen wir jetzt auch  
670 noch ein bisschen herauszufinden, was klüger ist. Wie gesagt, das Script ist ein Mittel zum Zweck,  
671 um darzustellen, dass hier mathematische Berechnungen erfolgen. Aber man könnte auch sagen,

672 dass vielleicht das mathematische Verhalten von einem anderen System definiert wird. Also,  
673 dass ich hier wirklich eine andere externe Komponente habe, die mir das ausrechnet und dann  
674 erst wieder den eigentlichen Wert, den ich an den Aktuator schicke, zurück gibt. Dass ich  
675 sage, 'Okay, ich kriege jetzt den Wert zurück und den muss ich dann einfach nur an den richtigen  
676 Aktuator schicken.' Diese Möglichkeit gibt es auch, dass man das so modelliert. Das ermöglicht  
677 halt natürlich auch dieses Interface, also das User Interface, dass man das auf verschiedenste  
678 Arten und Weisen macht. Wie gesagt, auch mit Subprozessen, wenn man das möchte.  
679 #01:09:08-3#

680 118. B: Ich glaube, entscheidend ist ja, das was Sie erreichen wollen, ist, Measurement heißt wirklich  
681 quasi vom System Werte in diesen Regelprozess hineinbringen. Und Control heißt eigentlich  
682 Werte definieren dann, die auf den/ nach außen wirken, also wie Inputs und Outputs. Und die  
683 Verrechnung erfolgt halt innerhalb und typischerweise halt im Controller. Die Separierung dieser  
684 zwei/ ja, diese zwei Funktionalitäten, vom Lesen, und dann Verarbeiten und Ausgeben. Und das  
685 sind/ aber eigentlich sind es eben drei, oder? Wenn man so spricht. Das eine ist das Einlesen,  
686 was Sie mit Measurement definieren. Oder? #01:10:05-3#

687 119. I: Ja. #01:10:06-0#

688 120. B: Und Control. Da mischen, also dann sag ich jetzt, oder da mischen Sie Berechnen von  
689 Ausgabeparameter und das Schreiben der Ausgabeparameter. Und jetzt in DER Formulierung  
690 sind/ eigentlich sind es drei Dinge. Einlesen, Verarbeiten und Ausgeben. Verarbeiten und  
691 Ausgeben wird jetzt bei Ihnen durch das Control-Ereignis getriggert. Das müsste man vielleicht  
692 überlegen, ob das immer/ ob die zwei immer zusammen gehören. #01:10:42-6#

693 121. I: Das Verarbeiten und das eigentliche Ereignis nach außen setzen meinen Sie? // B: Ganz genau.  
694 // Dass man das vielleicht separiert? Ja. #01:10:51-2#

695 122. B: Es wird vielleicht Prozesse geben, wo der Takt der Verarbeitung schneller erfolgt, wie das  
696 ausgeben der Werte. Weil das interne Modell/ vielleicht arbeitet das interne Modell mit der  
697 gleichen Taktfrequenz wie das Einlesen. Und der rechnet dann, berechnet irgendwas. Aber  
698 zeitkontinuierlich im gleichen Abtasttakt, wie man die Daten einliest. Und das Ausgeben erfolgt  
699 aber langsamer. Und so etwas könnten Sie, ich glaube, in Ihrem Meta-Modell so nicht darstellen.  
700 Natürlich kann man das Script des Reglers auch unter das Messereignis hängen, aber ich glaube,  
701 dann wäre es nicht in Ihrem Sinne. #01:11:42-3#

702 123. I: Also man kann natürlich trotzdem mit zeitlichen Events weiterhin arbeiten und den weiteren  
703 Verlauf auch in weiterer Folge so definieren, dass man diese zeitlichen Bedingungen hineinbringt  
704 ins Modell. Das kann man natürlich auch weiter ziehen. Also man muss nicht in einem Strang  
705 jetzt nur oben ein Ereignis haben, sondern man kann auch intern sagen, im weiteren Verlauf  
706 nach Control oder nach Measure, dass man trotzdem zeitliche Bedingungen hat. Das kann man  
707 auch machen. Wir haben es jetzt nicht in diesen Beispielen drinnen, aber wir haben  
708 grundsätzlich von der Logik her vorgesehen, dass man das auch machen könnte, wenn man das  
709 möchte. Also von den zeitlichen Bedingungen her sind wir nicht eingeschränkt. Aber diese  
710 speziellen Symbole, die wir hier einführen, wenn ich zum Beispiel ein Control-Event habe. Und  
711 das definiert jetzt einen PID-Regler. Und danach passiert noch etwas, dass jetzt noch eine  
712 Zwischenstufe in der Verarbeitung wäre und die muss auch in einem gewissen Takt, also mit  
713 einer gewissen zeitlichen Bedingung erfolgen. Dann könnte ich das einerseits als Subprozess  
714 machen, der intern vielleicht auch wieder zeitliche Bedingungen hat. Oder ich könnte noch  
715 einmal ein Timer Event einfügen und könnte sagen, 'Okay, das muss aber jetzt wirklich in den  
716 nächsten, keine Ahnung, so und so vielen Millisekunden passieren.' Also BPMN an sich von der  
717 Notation her bietet diese Möglichkeit an. Das ist halt das Schöne daran, dass der Standard auch  
718 so etwas vorsieht. #01:13:23-8#

719 124. B: Ich glaube hier geht es ja nicht darum/ Hier geht es ja um eine Erweiterung, dass es für  
720 prozessorientierte denkende Leute einfacher wird. Und elementar ist tatsächlich, wie oft  
721 passiert etwas. #01:13:42-0#

---

722 125. I: Genau, ja. #01:13:43-0#

723 126. B: Und das ist ja das Attribut Hertz. Und dann fügen Sie beim Auslösereignis noch die  
724 Hauptparameter, die den nachfolgenden Regelungsprozess steuern, fügen Sie als Attribute zum  
725 Ereignis hin. Weil jemand, der dann das auslöst, oder, der wird jetzt nicht denken/ der wird vor  
726 allem schauen, was werden da bei diesem Control-Ereignis für Attribute mitgeschickt? Diese  
727 Attribute sind entscheidend für das Verhalten der Prozessregelung, nicht Scripte, die danach  
728 kommen oder Parameter, die in diesen Scripten sind. Also die hauptbeeinflussenden Attribute  
729 für die Prozessregelung sollten bei diesem Control-Ereignis drinnen stehen. #01:14:38-7#

730 127. I: Das stimmt. Ja. #01:14:39-5#

731 128. B: Weil man dort hineinschaut. Wie wird der Prozess geregelt? Also ist es ein PI-Regler, PID, und  
732 dann entsprechend sind die Regelungsparameter vielleicht drinnen, vielleicht nicht, kann man ja  
733 dann entscheiden. Aber ich glaube wenn jemand jetzt dieses Modell sieht grafisch, das BPMN,  
734 dann wird er zuerst diese zwei Trigger anschauen und dort schauen, was lösen die aus und was  
735 machen die. Und beim Measurement steht eine Liste, welche Variablen oder welche  
736 Datenpunkte werden da zyklisch gelesen oder gemessen. WIE die dann gemessen werden, das  
737 ist eigentlich den nächsten egal, oder? Und beim Control-Ereignis, dort soll ja auch beim Control,  
738 bei diesem Value, bei dieser Liste, die wir jetzt sehen, soll ja drinnen stehen, die  
739 Hauptargumente oder die Hauptparameter, die auf diesen Controller einwirken. #01:15:40-2#

740 129. I: Die sind im Script dann natürlich auch definiert, ja. #01:15:44-8#

741 130. B: Ja, im Script verwendet man diese Werte, oder? Aber eigentlich steht hier drinnen, 'Auf diese  
742 Werte reagiert mein Controller.' Dass man dann im Hintergrund über globale Variablen noch  
743 zugreifen kann und die Scripte austauschen kann und da Dinge machen kann, die man eigentlich  
744 nicht sollte, das ist immer (lacht) ein Thema der Modellierung. Also. Das sagen da/ die Erfahrung  
745 habe ich auch gemacht. Sobald man scripten kann, kann man natürlich auch einen Prozessablauf  
746 rein im Script implementieren. Also. Durch die Möglichkeit vom Scripten kann man eigentlich die  
747 Modellierungs/ das Meta-Modell der Modellierung, dass man jetzt mitbringen will, komplett  
748 zerstören. #01:16:35-2#

749 131. I: Ja, das stimmt natürlich. #01:16:38-9#

750 132. B: Und das ist ja nicht das Problem vom Meta-Modell. Das ist das Problem, wie man es  
751 implementiert. Also so ein Subprozess, wie Sie sagen. Durch das Hinzufügen eines  
752 BPMN-Subprozesses. So einen Subprozess kann man/ formal könnte man den auch rein in Ruby  
753 implementieren. Oder das/ der Ablauf vom Messdaten-Holen, Controller-Aufrufen und dann auf  
754 Output setzen. Das kann man auch mit einem einfachen Timer Event in BPMN auch lösen. Aber  
755 damit ist es zwar funktional das gleiche, aber man verliert diese Transparenz, die Sie ja hier  
756 reinbringen wollen. #01:17:19-9#

757 133. I: Ja, das stimmt. Man kann es dann natürlich auch noch komplexer modellieren und dadurch  
758 eigentlich die Vorteile zunichte machen. #01:17:27-7#

759 134. B: Genau. #01:17:29-2#

760 135. I: Ja, das stimmt natürlich. Ja. Wir sind jetzt halt noch gerade auch in der Phase, um ein bisschen  
761 zu definieren, inwieweit wir die Modellierungsmöglichkeiten auch für die User einschränken  
762 sollten, so dass auch von Anfang an Fehler vermieden werden können. Aber ja, das müssen wir  
763 jetzt auch noch herausfinden. (...) Okay. Ich bin mir jetzt nicht mehr ganz sicher. Ich glaube, wir  
764 sind bei Einfachheit dann stehen geblieben. Könnte man das Modell/ Genau, das haben wir  
765 gerade diskutiert. Der nächste Punkt war Logik. Wird klar, was parallel und sequentiell passiert?  
766 Aus den Indikatoren heraus? #01:18:23-3#

767 136. B: Das auf alle Fälle, ja. #01:18:25-4#

---



768 137. I: Okay. Erweiterbarkeit. Könnte man dem Modell noch etwas hinzufügen, was den  
769 Informationsgehalt verbessern würde? Wir haben vorhin über die physikalischen zusätzlichen  
770 Bedingungen geredet. Das heißt, wir haben einen Rührer, den Motor, der irgendwie angetrieben  
771 werden muss, natürlich, und gesteuert werden muss. Wir haben natürlich noch den  
772 Volumenstrom durch den Zufluss. Diese Dinge fließen jetzt nicht unbedingt mit ein in das  
773 Prozessmodell. Aber wenn man jetzt rein von der Modellierungsvariante, sich das ganze also  
774 vom Modellierungsablauf anschauen würde, würden Sie sagen, dass man hier noch etwas  
775 einfügen kann oder dass es recht vollständig abgebildet ist? #01:19:10-2#

776 138. B: Also so ist es vollständig. Einfach von der Lesbarkeit her. Wenn Sie vielleicht noch mal so hoch  
777 gehen. (...) Zum Beispiel beim Controller gibt es jetzt zwei Scripts. Die zwei Scripts könnte man  
778 technisch auch in einem Script lösen. Aber so macht man zwei Scripts, weil man/ diese Scripts  
779 sind variiert im Sinne von Wiederverwendbarkeit oder Separation of Concerns. Also das eine ist  
780 ein Regler und dieses Script kann ich immer wieder verwenden.// I: Zum Beispiel. Ja. // Und die  
781 Transformation und Einheitenanpassung an das Dampfventil ist halt ein zweiter Schritt. Darum  
782 macht es Sinn, das zu separieren. Aber jetzt eben in der Grafik, aber das weiß ich nicht, wie das  
783 Tooling ist, oder? Cool wäre, wenn man bei den Scripts noch irgendwas hinzufügen könnte. Also  
784 in der Grafik. Jetzt in Ihrer Darstellung hat man rechts diesen Descriptor drinnen. Ich weiß nicht,  
785 ob das bei Ihnen immer so der Fall ist. Das funktioniert nur dann, wenn man eine lineare/ also  
786 wenn man alle Prozessschritte linear hintereinander abbildet. Jetzt, wenn ein Flussdiagramm  
787 hundert parallele, also hundert einzelne Komponenten hat oder so Prozess-Icons, dann gebe es  
788 eine Liste von hundert. Auch wenn die parallel vielleicht maximal eine Länge von Fünf haben.  
789 Also angenommen man hat fünf Schritte in jedem parallelen Pfad und zwanzig parallele Pfade.  
790 Dann gebe es/ In der Notation gebe es eine Liste von hundert Elementen. #01:20:56-3#

791 139. I: Genau. Also die Labels/ Sie meinen damit die Bezeichnungen hier seitlich. Und dann könnte  
792 man sich die Scripts noch extra anschauen. Meinen Sie das? #01:21:05-4#

793 140. B: Genau. Vor allem die Labels. #01:21:07-0#

794 141. I: Ja, also, die // B: Dass man/ // würden dann/ Genau. #01:21:10-0#

795 142. B: Dass man sieht, ja, was ist jetzt das erste S und das zweite S. #01:21:14-0#

796 143. I: Genau, ja. #01:21:14-8#

797 144. B: Das ist sicher was, was/ Ich weiß jetzt nicht, in welcher/ Das wird vermutlich Tooling abhängig  
798 sein, wie man das dann/ wie man so zusätzliche Textpassagen, oder Tooltips ist oft so ein Name,  
799 wie man das dann hinzufügen kann. #01:21:33-7#

800 145. I: Okay, verstehe. #01:21:35-4#

801 146. B: Und die technische Frage ist dann, wo dieses Script sitzt. Aber das ist so eine Detailfrage. Also  
802 wenn man jetzt verschiedene Flows hat und eben dann das gleiche Script mehrfach verwenden  
803 will, ich sag PI-Controller. Ist das so wie eine Library Funktion, die man aufruft? Oder ist es eine  
804 Code-Kopie? Dann wäre es schlecht. Weil wir/ Der Sinn ist ja eigentlich, dass man diesen  
805 PI-Controller als eigenen Prozessschritt darstellt hier, ist, weil der öfters irgendwo vorkommt.  
806 #01:22:11-0#

807 147. I: Genau, ja. #01:22:12-6#

808 148. B: Wenn man so eine wiederverwendbare Script-Passage hätte, dann müsste sie irgendwie eine  
809 externe Funktion sein, die man referenziert oder/ Das weiß ich nicht genau, was da Ihr Tool tut.  
810 #01:22:27-2#

811 149. I: Das müssten wir uns auch noch überlegen, wie wir das auch wirklich als so etwas wie eine  
812 Standardbibliothek vielleicht zur Verfügung stellen. Aber so ist es natürlich auch gedacht, dass

813 man das deswegen separiert. Das stimmt. (...) Okay, das zweite Modell, das wäre das etwas  
814 komplexere von beiden. Hierbei handelt es sich auch um einen Heizprozess, der aber aus  
815 Schulungsunterlagen der Firma Siemens entnommen wurde. Auch eine Temperaturregelung für  
816 einen Rührreaktor. Die Regelung wird in diesem Beispiel mit einem PID-Regler durchgeführt.  
817 Grundsätzlich werden in der Dokumentation auch eine Handsteuerung sowie  
818 Pulsweitenmodulation erwähnt. Die Heizung erfolgt nicht über einen Wärmetauscher, sondern  
819 über ein Heizelement. Und weiters gibt es auch Verriegelungsbedingungen, die wirklich in den  
820 Unterlagen auch aufscheinen. Als Basis für die Prozessmodellierung wie gesagt wurden diese  
821 Schulungsunterlagen der Firma Siemens für die Modellierung mit Simatik PCS 7 herangezogen  
822 und unser Prozessmodell wird in dem Fall aber mit einer automatischen Steuerung modelliert,  
823 die mit Umschalten auf Fernsteuerung aus dem Closed Loop System ausbrechen würde. Wir  
824 gehen davon aus, dass das System bereits angelaufen ist und automatisch gesteuert wird und  
825 weiters wird der Prozess für einen einzelnen Reaktor und nicht wie in den Unterlagen  
826 beschrieben für zwei Reaktoren umgesetzt. Die Bezeichnungen PV In steht entsprechend für  
827 PCS 7 analog Input PV In für die Temperaturregelung. Da geht es ein bisschen um die  
828 Übersetzung, welcher Wert jetzt wirklich wie übertragen wird. (...) Haben Sie sich die Unterlagen  
829 vorab angeschaut oder wollen Sie sich/? #01:24:21-9#

830 150. B: Nein, nicht im Detail. Aber ich glaube, ich kann mir vorstellen, was Sie sich/ // I: Okay, gut.  
831 //also von der Beschreibung des Modells. #01:24:32-8#

832 151. I: Wir haben wieder verschiedene Datenelemente, die wir definiert haben. Wir haben ein/  
833 Diesmal natürlich mit cancel und sequential. Das heißt, wir haben eine maximale Zykluszeit, die  
834 wirklich auch abbricht, wenn diese Zykluszeit überschritten werden würde. Und zuerst werden  
835 die Messungen durchgeführt, dann werden die Regelungsaufgaben durchgeführt. Das Modell ist  
836 leider um etwas umfangreicher. Ich versuche es mal halbwegs/ #01:25:08-2#

837 152. B: Das ist jetzt genau die lange Liste, die ich erwähnt habe. (lacht) #01:25:12-2#

838 153. I: Genau. Das ist es dann. Ja. #01:25:13-7#

839 154. B: (Unv.) dass man die Attribute oder die Bezeichnungen in einer Spalte hat. #01:25:19-2#

840 155. I: Wie Sie es beschrieben haben, das stimmt, ja. Wir haben verschiedene Messaufgaben. Wir  
841 haben eine Regelungsaufgabe und wir haben verschiedene Abbruchbedingungen. Ich fange jetzt  
842 mal kurz mit den Abbruchbedingungen an. Es wird einerseits natürlich von einem Hauptschalter  
843 gesprochen, das heißt der Main Switch. Wenn der auf Off geschaltet wird, bricht natürlich der  
844 kontinuierliche Prozess ab und es gibt in dem Fall, das ist jetzt nicht unbedingt in den  
845 Siemens-Unterlagen dargestellt, aber in dem Fall gebe es eine Nachricht an den Operator. Es gibt  
846 einen Emergency Stop. Wenn der auf true gesetzt wird, wieder Nachricht an den Operator. Die  
847 Bedingung, die dann interessant ist. Wir haben in den Unterlagen stehen, dass wenn die  
848 Temperatur des Reaktors über eine gewisse vorgegebene Maximaltemperatur steigt, dass wir  
849 den Wert, dass wir die Heizung halt null setzen. Also das es wirklich herunter geregelt wird auf  
850 Temperatur Null zum Beispiel. Das ist auch jetzt nicht in den Siemens-Unterlagen definiert. Jetzt  
851 genau mit null, aber ich habe das jetzt als Beispielwert wirklich dafür angegeben, dass das  
852 herunter geregelt wird. Und natürlich wollen wir jetzt kein Wasser zum Gefrieren bringen oder  
853 dergleichen, aber wir wollen möglichst die Anlage in einen sicheren Zustand fahren als Beispiel.  
854 Dann haben wir als nächste Bedingung Level Reaktor, also Füllstand des Reaktors. Wenn der  
855 unter einen gewissen, einen Mindestfüllstand kommt, haben wir wieder das Gleiche, dass auf  
856 einen sicheren Wert geregelt werden soll. Als Beispiel hier mal mit null aber natürlich würde  
857 dafür ein anderer Prozesswert, entsprechend hinein kommen. Für das letzte, was ich erwähnt  
858 habe mit manual und automatic. Grundsätzlich ist eine Handsteuerung im Modell von Siemens  
859 vorgesehen. Wir haben jetzt aber gesagt, wir lassen das ganze mal im Automatikbetrieb. Wenn  
860 wir aber in einen anderen Betriebszustand wechseln/ Das haben Sie am Anfang unseres  
861 Interviews erwähnt. Also wenn wir einen Zustandswechsel haben sich/ das soll heißen wenn der  
862 Zustand des Systems sich ändert, würde das Prozessmodell natürlich auch ein anderes werden.  
863 Damit würde der kontinuierliche Verlauf des Prozessmodells so nicht mehr gültig sein und wir

864 müssten ein anderes Prozessmodell herannehmen. Das heißt, hier würde halt dann zum Beispiel  
865 die Manual Routine als dieser Wechsel aufgerufen werden und wir würden hier heraus wechseln  
866 aus diesem System. Ganz kurz nur. Temperatur Max, maximal sechzig Grad. Level Reaktor Min  
867 sind 200 Milliliter. Die sind hier oben halt auch als Parameter vorgegeben, die wirklich von  
868 Anfang an gelten und die auch nicht mehr überschrieben werden. Das sind diese  
869 Grenzbedingungen, kann man sagen. Hier steht dann natürlich auch seitlich cancel. Die  
870 Messungen, die durchgeführt werden, sind einerseits die Temperatur des Reaktors. Dann wird  
871 etwas umgewandelt in einem Script. Dann je nachdem, wie gesagt, was der jeweilige Sensor  
872 liefert beziehungsweise welches Interface der jeweilige Sensor hat. Dann haben wir noch den  
873 Füllstand des Reaktors, der natürlich überprüft werden muss. Wir haben die Überprüfung des  
874 aktuellen Betriebsmodus, der immer wieder abgefragt wird, regelmäßig. Wir haben auch eine  
875 Überwachung für den Emergency Stop oder auch eine Überwachung für den Hauptschalter. Und  
876 der eigentliche Kontrollprozess, der dann abläuft, ist der PID-Controller. Die Basis der  
877 Siemens-Unterlagen. Das wird alles in einem Continuous Function Chart dargestellt. Hier hätten  
878 wir auch zuerst einmal den Funktionsblock für den PID-Controller mit gewissen Variablen, die  
879 angegeben sind, also Inputs und Outputs. Wir haben eine Pulsweitenmodulation. Für die hab ich  
880 jetzt auch das mathematische Verhalten hier ausgelassen. Aber man kann es natürlich auch im  
881 mathematischen Verhalten beschreiben, wenn man das möchte. Oder einen Subprozess oder  
882 wie auch immer man das darstellen möchte. Datenumwandlung, wenn es vorgesehen ist. Und  
883 dann auf den Prozesswert, der nach außen hingeschickt wird. Der an den jeweiligen Aktor  
884 geschickt wird. Und wie gesagt, die Abbruchbedingungen. Das heißt, ein bisschen ein  
885 komplexeres Modell. Mathematisches Verhalten des PID/ #01:30:10-5#

886 156. B: Jetzt mal Out of Comment. #01:30:12-1#

887 157. I: Ja? #01:30:12-9#

888 158. B: Wenn man Pulsweitenmodulation macht, dann geht man auf einen digitalen Schalterausgang,  
889 null eins. Und steuert den Analogwert über die Zeit. Dann braucht es keine Datenkonversion weil  
890 es ein digitaler Ausgang ist. #01:30:25-2#

891 159. I: Ja also, die Data Conversion habe ich mir eigentlich als allgemeine Umwandlungs-Task oder so  
892 etwas hier hinein geschrieben. Also das soll wirklich wie auch immer der Wert dann noch  
893 manipuliert wird, oder was auch noch immer mit dem Wert geschieht, kann man quasi da noch  
894 hinein programmieren. Das ist so ein bisschen der Placeholder. #01:30:44-2#

895 160. B: Die Reihenfolge ändern, dann macht es Sinn. In dieser Reihenfolge macht es technisch wenig  
896 Sinn. #01:30:49-9#

897 161. I: Okay, also Sie würden erst die/ Wie würden Sie es dann anordnen? #01:30:55-4#

898 162. B: Das kann man so machen. Beim PWM, das geht ja auch den Heizschalter. Und der  
899 Heizschalter wird typischerweise über irgendeine Pulsmodulation geschaltet. Also über Zeitwert.  
900 Das Tastverhältnis wird mit Ein und Aus jetzt gewandelt. Aber der Wert ist schon digitalisiert  
901 zeitdiskret und es gibt keine ein/ also den muss man nicht mehr datenwandeln. #01:31:16-9#

902 163. I: Okay, danke für die Anmerkung. #01:31:23-9#

903 164. B: Einfach drehen und dann gibt es die Diskussion nicht mehr. #01:31:32-9#

904 165. I: Okay, Dankeschön. Ja, ich bin/ ich muss sagen, ich weiß leider nicht, wie man in einer SPS das  
905 normalerweise programmieren würde. Es soll halt mal so auf Basis eines normalen Ablaufs wie  
906 beispielsweise bei einer SPS nachvollziehbar sein. Es soll einfach gezeigt werden, man könnte,  
907 wenn man wollte, von der Logik her diese Prozessschritte abbilden. Aber/ #01:32:01-0#

908 166. B: Passt schon. Nur umdrehen. Dann/ #01:32:02-3#

---

909 167. I: Okay. Danke. // B: (Unv.) // Danke für den Hinweis. Ja, es ist ein etwas komplexeres Modell als  
910 das, was wir zuvor gesehen haben. Wir haben hier in den Siemens-Unterlagen wie gesagt, wenn  
911 Sie sich durchgeschaut haben, wird Ihnen aufgefallen sein, wir haben um einiges mehr Vorgaben  
912 zum Prozess. Also wir wissen genauer, wie die Logik abläuft aufgrund des CFCs. Wir wissen  
913 gewisse Grenzwerte, die für gewisse Parameter herrschen. Wir kennen damit auch gewisse  
914 Abbruchbedingungen natürlich. Ich würde Sie jetzt auch bitten, ähnlich wie beim vorigen Modell,  
915 dass Sie mir sagen, wie sieht es mit diesen Eigenschaften aus? Verständlichkeit, Übersichtlichkeit,  
916 Einfachheit? Das Sie es einfach auf ähnliche Art und Weise bewerten. Also ob Sie zum Beispiel,  
917 ob es für Sie verständlich war mit dem Hintergrundwissen, was für ein Prozess hier abgebildet  
918 wird, was im Modell auch wirklich abläuft. Würden Sie sagen, dass das hier auch verständlich  
919 war, obwohl es doch um einiges komplexer war? #01:33:10-1#

920 168. B: Sie haben gesagt, das ist jetzt sequentiell programmiert. #01:33:13-5#

921 169. I: Genau, ja. #01:33:14-4#

922 170. B: Wo sehe ich das? Dass es jetzt sequentiell zu interpretieren ist? #01:33:20-0#

923 171. I: Das war hier eines der Attribute für das Closed Loop Subsystem. (...) Es ist nicht grafisch  
924 dargestellt. Das stimmt. Also es könnte genauso gut eigentlich parallel hier ablaufen. Aber es ist  
925 durch diesen Indikator definiert. #01:33:39-4#

926 172. B: Okay. (...) Ja, wenn man jetzt diese differenziert. Für mich ist die elementar, diese  
927 Unterscheidung, ob es sequentiell oder parallel ist. Und wenn man es grafisch nicht sieht, dann/  
928 Das wird man nicht sehen. // I: Also, Sie meinen/ // Es heißt eigentlich, dass diese  
929 Differenzierung, die Sie jetzt machen, mit Measurement und Control, wo ich ja gesagt habe, 'Das  
930 könnte man vielleicht sogar abstrahieren und nur ein Element machen, weil es eigentlich nur  
931 Zeittrigger ist.' Für mich wäre es viel wichtiger, dass dieses Controller-Symbol, dass das die Raute,  
932 dass es da zwei Varianten gibt. Dass man in der Raute sieht, ob es jetzt ein parallel interpretiert  
933 oder sequentiell interpretiertes Flow-Diagramm ist. Auch wenn man es parallel zwar zeichnet.  
934 Aber dass dieses parallele Zeichnen am Schluss sequentiell abgearbeitet wird, das muss man  
935 groß sehen. Weil wenn man darauf schaut, dann ist es immer parallel. Also wenn Sie noch/ Ich  
936 sehe hier nur das Cancel und das kann ich nicht/ da habe ich auch noch Mühe zu interpretieren.  
937 Aber wenn ich das sehe, dann ist es ein paralleles Netzwerk, wie das vorige auch. Und weil die  
938 Interpretation dieser Grafik entscheidend ist, wie man diese Raute konfiguriert hat, muss man  
939 es/ das muss man sehen. Also diesbezüglich ist es/ // I: Darf ich/ // Dass das jetzt sequentiell  
940 abgearbeitet wird, das sieht man nicht. #01:35:31-7#

941 173. I: Okay. Darf ich nur fragen, wenn Sie auch meinen, Sie haben auch Mühe bei Cancel, obwohl es  
942 hier drinnen steht. Meinen Sie, dass man das auch irgendwie prägnanter vielleicht darstellen  
943 sollte? #01:35:47-9#

944 174. B: Ja. Jetzt ist die Frage, was cancel heißt. Also da müssen Sie mir vielleicht nochmal nochmals  
945 genau sagen, was der Unterschied ist, wenn da cancel steht. Gegenüber dem vorhergehenden.  
946 #01:36:01-8#

947 175. I: Bei cancel würde, wenn die Zykluszeit, also wenn diese Dauer, die durch die Timer Events  
948 definiert wäre, überschritten werden würde, dass wir hier einen Abbruch der jeweiligen Kante  
949 hätten. Also wir warten hier nicht darauf, dass alles fertig wird, jeder einzelne Strang. Sondern  
950 das muss zu einem vorgegebenen Zeitpunkt passieren. Das soll cancel heißen. Also eine strikte  
951 Grenze. Bei Wait/ #01:36:34-4#

952 176. B: Also die maximale Zeit aller Zeiten, die bei diesem Trigger-Ereignissen definiert ist?  
953 #01:36:40-5#

954 177. I: Ja. Das soll dafür stehen. Bei wait, das Gegenteil dazu, könnte man definieren, wenn man das  
955 möchte, dass man auf jeden einzelnen Strang wartet. Das heißt, wenn ich zum Beispiel ein

956 einfacheres Modell habe und nach einem Measure Event eine einzelne/ ein Get Request. Das  
957 soll einfach irgendeinen Wert holen von außen. Und das dauert viel zu lang, weil einfach keine  
958 Antwort kommt. Dann würde man auf dieses Get Request warten. Also man gibt dem  
959 Modellierer die Möglichkeit, es so auch zu machen. Ich bin mir nicht ganz sicher. Wenn man jetzt  
960 konkret eine SPS her nimmt, wie viele Funktionen es gibt, die genau das verhindern würden.  
961 Aber ich glaube, wenn man es per default modellieren würde, hätte man ja eigentlich immer  
962 eine vorgegebene Zykluszeit natürlich. Und die würde immer sagen, okay, bis zu dem Punkt geht  
963 es maximal. Und dann würde einfach wieder der Zyklus von vorne starten. Das heißt, man hätte  
964 sowieso, wie bereits erwähnt am Anfang, das, ich glaube, das heißt Prozessabbild, wo man alle  
965 Parameter einmal abgelesen hat und mit denen dann eigentlich durch den Zyklus geht, also  
966 durch diesen ganzen Prozess durchgeht und das bearbeitet. Und dann halt wieder auf einmal  
967 alles schreibt. #01:38:16-4#

968 178. B: Bei der SPS dieses Holen der Werte und Abbilden in einer Prozessvariable wird unabhängig  
969 eigentlich vom Programmieren der eigentlichen Logik wird das konfiguriert und läuft parallel ab.  
970 Und damit hat sich der Anwender eigentlich gar nicht zu kümmern. Der konfiguriert nur diese  
971 Prozessvariable in diesem Takt. Von diesem digitalen (unv.). Und dann macht das das System im  
972 Hintergrund. #01:38:51-5#

973 179. I: Also diese Möglichkeit da zu unterscheiden, wie man das machen möchte, möchten wir dem  
974 Modellierer hier auch geben mit cancel und wait. #01:39:01-0#

975 180. B: Ja, die Frage ist. Bei Ihnen ist jetzt das Verhalten, wie man, wie man dieses Netzwerk oder  
976 diesen Graphen interpretiert und wie man die Zeiten interpretiert bezüglich cancel und wait und  
977 parallel und seriell, sind zwei Parameter. Und die zwei Parameter sieht man zwar als Text, aber  
978 man sieht sie nicht wirklich in der Grafik. Und die Frage ist dann immer bei so Meta-Modellen,  
979 gibt es einen Use Case, dass man einen solchen Graphen bezüglich dem Verhalten einfach durch  
980 Umparametrieren, dass man das Verhalten ändern will. Oder weiß der Anwender schon am  
981 Anfang, wenn man so ein Netzwerk, also ein Graphen zeichnet, was er für ein Verhaltensmodell  
982 will? Das heißt, er überlegt sich zuerst, ja, ich will jetzt seriell und cancel. Und da, wenn er so mal  
983 begonnen hat, das System zu zeichnen, dann wird er das nicht mehr ändern. Oder gibt es den  
984 Fall und dann müssen Sie vielleicht noch tiefer schauen. Gibt es den Fall, dass er mal seriell  
985 beginnt und dann plötzlich auf die Idee kommt, 'Nein, ich nehme jetzt den Graphen' und ich/ Die  
986 Abarbeitung soll jetzt sequentiell sein. Nein, doch parallel. Die Zeiten jetzt doch wait oder/ Nein,  
987 doch cancel. Also sind das Parameter, die man wirklich während der Entwicklung immer hin und  
988 her schaltet? Geht das überhaupt? Oder wenn es eben nicht geht, dann ist es gefährlich, wenn  
989 man das als Parameter löst. Weil dann sind es so Initialdefinitionen. Und dann würde ich eher  
990 die Symbolik dieser Raute, würde ich im Extremfall vier Symbole der Raute machen, wo man es  
991 anhand der Raute sieht. #01:41:03-4#

992 181. I: Okay. #01:41:04-9#

993 182. B: Und vielleicht sagt die obere Raute seriell parallel und die unter Raute sagt, wie man abbricht.  
994 #01:41:11-6#

995 183. I: Das ist sehr guter input. Dankeschön. #01:41:16-6#

996 184. B: Und ändern müsste man/ Das ist dann auch eine Frage von grafischen Editor. Ob man dieses  
997 Rautensymbol ersetzen kann, ohne dass der ganze Baum verfällt. Ich meine jetzt da der  
998 Anfangsknoten. Aber wenn man es so darstellen könnte, ich glaube, das würde helfen.  
999 #01:41:36-4#

1000 185. I: Ich meine, im Editor wäre das an sich, denke ich, kein Problem. Das gebe dann/ wir könnten  
1001 das Ganze dann so lösen. Allein von Modellierungsprozesse, dass wir mit Fehlermeldungen an  
1002 den User arbeiten und hier sagen, 'Nein, von der Logik her kann man das nicht so machen. Und  
1003 Sie müssten natürlich ein weiterer Prozessfolge das so und so noch abändern, weil diese  
1004 Bedingungen gelten.' Das könnten wir so lösen. Danke, ja. #01:42:07-1#

---

1005 186. B: Gerne. Und das andere ist/ aber das ist vielleicht die Einfachheit oder Logik gehört eher dort  
1006 dazu. So wie Sie es beschrieben haben, gibt es jetzt also Cancel Events, die den Sollwert auf null  
1007 setzen. #01:42:27-5#

1008 187. I: Ja, auf einen anderen Wert. Also/ #01:42:30-5#

1009 188. B: Auf einen Wert. Aber die Regelung selber bricht nicht ab. Im Gegensatz zu einem Notaus.  
1010 #01:42:41-5#

1011 189. I: Der zyklische Prozess würde abbrechen. Das letzte Kommando an den Aktuator wurde nicht  
1012 beeinflusst. Also ich sage es so. Ich kann natürlich hinter dem Abbruchereignis hinein  
1013 modellieren, 'Setze jetzt den Aktuator auf diesen Wert' aber der/ #01:43:13-8#

1014 190. B: Den Aktuator stellen Sie? #01:43:20-9#

1015 191. I: Genau, ja. Es wäre natürlich eine Aktion/ #01:43:24-5#

1016 192. B: Weil geschrieben steht eben Setpoint, SP. Das ist für mich der Stellwert. #01:43:29-1#

1017 193. I: Ja, es wäre aber natürlich als Kommando nach außen gedacht gewesen, ja. #01:43:34-8#

1018 194. B: Genau also mache/ also 'Stelle die Heizung ab'. #01:43:37-9#

1019 195. I: Genau ja. #01:43:38-8#

1020 196. B: Nicht, 'Regle auf eine Temperatur zwanzig' #01:43:43-2#

1021 197. I: Also 'Regle auf einen sicheren Zustand'. Das ist immer so gedacht, dass man hier einen Prozess  
1022 definiert, der/ oder einen Zustand definiert, auf den das System fahren kann, ohne dass  
1023 irgendetwas Kritisches passiert. Also dass hier der Modellierer definieren kann, das ist jetzt der  
1024 Übergangszustand, indem ich das System lassen kann, ohne dass etwas passiert im Falle, WENN  
1025 zum Beispiel der Betriebsmodus wechselt oder wenn es so weit kommt, dass das Fülllevel ein  
1026 kritisches Niveau erreicht. Also es sind eigentlich immer/ Hier ist vorgesehen, wirklich die  
1027 Routine zu definieren, dass man das System in einen sicheren Zustand überführt. Oder wie man  
1028 halt in diesem Fall reagieren möchte. Es ist vielleicht etwas falsch formuliert mit Set Setpoint to  
1029 zero. Aber es soll dafür stehen, dass der Modellierer hier etwas definiert, das genau in diesem  
1030 Fall zu tun ist, ohne dass irgendetwas passiert. Also wirklich die Maßnahmen in diesem Fall.  
1031 #01:44:51-2#

1032 198. B: Aber er springt dann in den Stoppzustand? #01:44:54-6#

1033 199. I: Er springt aus dem kontinuierlichen Prozess heraus. #01:44:57-8#

1034 200. B: Ja, okay. Ja, dann war vielleicht Ihr Wording ungeschickt. #01:45:02-0#

1035 201. I: Ja, das tut mir leid. Das muss ich umformulieren im Modell. #01:45:06-3#

1036 202. B: Ja, es hat so getönt wie man den Stellwert, den Sollwert, für den Regler auf null setzt, aber die  
1037 Regelung dann nicht mehr mit dem Regler, also dem Sollwert von außen arbeitet, sondern mit  
1038 einem fix gesetzten. Aber die Regelung eigentlich noch aktiv bleibt und das wäre ja, // I: Ja das  
1039 stimmt. Verstehe ich. // Wenn das so gewesen wäre, dann wäre es ja schwierig nachzuvollziehen,  
1040 wieso das jetzt gewisse Exceptions quasi nur einen Sollwert setzen, der Regel bleibt aber aktiv  
1041 und gewisse brechen ab. Aber wenn das nicht so ist, dann passt schon. #01:45:44-0#

1042 203. I: Tut mir leid für die Verwirrung. Aber dann muss ich das Modell natürlich ändern. Danke für  
1043 den Hinweis. Aber dann würde hier stattdessen stehen, 'Schick den Befehl so und so an den  
1044 Aktuator' und dann geht es aus der Kontinuität heraus. #01:45:58-6#

---

1045 204. B: Ja genau. Also, 'Setz den Pulsweitenmodulationsausgang auf null'. Zum Beispiel. #01:46:03-7#

1046 205. I: Oder das. Danke ja, dann schreibe ich stattdessen das. #01:46:12-0#

1047 206. B: Oder Heizung aus oder irgend so etwas. (lacht) #01:46:14-7#

1048 207. I: Ja, etwas Einfacheres, ja. So war das gedacht. #01:46:19-4#

1049 208. B: Ich glaube, es geht in der Modellierung dann darum, dass man hinter einer Exception auch  
1050 noch einmalig eine Aktion ausführen kann. #01:46:25-9#

1051 209. I: Genau ja, das wäre auch eine Frage in den letzten paar Fragen, die dann noch kommen,  
1052 gewesen. Ob man das auch erkennt und ob das auch so ersichtlich ist, dass einmalig diese Aktion  
1053 ausgeführt werden soll. Aber wenn Sie das jetzt schon sehen, dann ist das natürlich auch sehr  
1054 gut. Tut mir leid, wenn es technisch nicht ganz korrekt dargestellt ist. #01:46:46-5#

1055 210. B: Nein, nein. Es geht ja nur darum zu fragen, wie das Modell wirklich// I: Gemeint ist. //  
1056 funktioniert. #01:46:53-9#

1057 211. I: Das heißt, Einfachheit würden Sie sagen, grundsätzlich eher wieder einfacher dargestellt oder  
1058 doch recht komplex? #01:47:04-6#

1059 212. B: Nein, ich denke in DER Notation ist das nachvollziehbar. Ja. #01:47:09-3#

1060 213. I: Von der Logik her, haben Sie gemeint, kann man das nicht so gut erkennen, ob es parallel oder  
1061 sequentiell passiert. Wenn man jetzt nicht genau auf diesen Indikator schaut, der halt leider  
1062 nicht die ganze Zeit über einem ins Auge springt, kann man sagen. #01:47:25-6#

1063 214. B: Ja. Es sind ja zwei Attribute, die man setzt. Sequentiell. Parallel. Und cancel und wait. Und  
1064 jetzt in der Grafik, die Sie gezeigt haben, sieht man ja nur das Attribut Cancel. #01:47:35-8#

1065 215. I: Das ist zu wenig. Ja. Was würden Sie hier sagen/ #01:47:41-0#

1066 216. B: Weil es eben so entscheidend ist, wie man diesen Graphen interpretieren muss, ist es sehr,  
1067 sehr wichtig, dass man das wirklich auf den ersten Blick sieht. #01:47:52-2#

1068 217. I: Das letzte wäre die Erweiterbarkeit. Ob man dem Modell noch etwas hinzufügen könnte vom  
1069 Informationsgehalt. Wenn man jetzt ein bisschen einen komplexeren Prozess darstellen möchte.  
1070 Also das mit sequentiell und parallel haben wir jetzt. Das wäre sinnvoll, es auf den ersten Blick  
1071 erkenntlich zu machen. Aber würde Ihnen hier noch etwas einfallen eventuell? #01:48:23-6#

1072 218. B: Ja nur noch, am Schluss gibt sehr viele S-Blöcke. #01:48:34-4#

1073 219. I: Okay. #01:48:37-6#

1074 220. B: Und hinter diesen S'en versteckt sich ja dann irgendein Skript. Ich weiß nicht, ob es da noch  
1075 Bedarf gibt noch hier Varianten zu haben oder dass man die besser oder geschickter taggen kann  
1076 oder grafisch aufpeppen kann, oder/ #01:48:56-9#

1077 221. I: Dass man die Scripte, wofür sie wirklich stehen, noch unterscheiden kann? #01:49:01-0#

1078 222. B: Dass man dort noch Typen von Scripten im Sinne von kontinuierlicher Prozessmodellierung.  
1079 Schon in die Richtung, dass man da auch noch irgendetwas hinschreibt. Oder dass das jetzt  
1080 vielleicht eben ein Regler ist oder dass das eine Datentransformation ist. #01:49:20-1#

1081 223. I: Verstehe, ja. #01:49:22-6#

---

1082 224. B: So in die Richtung. #01:49:24-7#

1083 225. I: Guter Input. Danke. #01:49:27-8#

1084 226. B: Typisierung von Scripten. Weil Sie haben sie ja typisiert im Sinne von/ Jetzt muss ich auf den  
1085 Spickzettel schauen. (...) Wir haben ja drei Typen von Scripten, oder? // I: Meinen Sie/ //Die zwei,  
1086 drei Symbole. #01:49:51-9#

1087 227. I: Meinen Sie damit // B: Also zwei sicher. // die Kombination, also, dass man einmal nur das  
1088 Script hat, einmal den Service Call und dann die Kombination aus beiden, was ich Ihnen am  
1089 Anfang gezeigt habe? #01:50:04-2#

1090 228. B: Ja, genau. #01:50:05-5#

1091 229. I: Ja, das. Ja, das sind einerseits Service Call, also wirklich nur die Endpunkte, die man auch extra  
1092 definieren kann. Das zeige ich Ihnen noch einmal ganz kurz, was ich meine. Also hier gibt es den  
1093 Reiter Endpoints. Und irgendwo gab es auch noch eine Liste, die ich hier angelegt habe. Einen  
1094 Moment bitte. Genau. Wo man eigentlich wie bei den Data Elements auch noch verschiedene  
1095 Adressen definieren könnte, mit denen dieses Orchestrierungs-Tool wirklich kommuniziert oder  
1096 über die das Orchestrierungs-Tool kommuniziert. Und die Service Calls, also das ganz einfache  
1097 Zahnrad ist eigentlich dafür da, wirklich nur diese Befehle umzusetzen, diese Calls abzusetzen.  
1098 Entweder einfache Get Calls zum Beispiel oder auch Calls mit Argumenten, die mitgeschickt  
1099 werden. #01:51:07-1#

1100 230. B: Aber diese Modellierung, die kommt ja jetzt nicht vom/ Die ist jetzt nicht prozessspezifisch?  
1101 Das sind drei Elemente, die kommen von diesem Framework mit. Die sieht man auch bei  
1102 Continuous, bei diskreten Modellierungen sieht man die auch. #01:51:25-5#

1103 231. I: Genau. Die wurden generell bei der Entwicklung dieser [REDACTED]  
1104 angelegt. Es sind zwar auch Erweiterungen, aber die drei gehören eigentlich nicht zu dem  
1105 Umfang dieser speziellen Arbeit, sondern die wurden schon in früheren Arbeiten hier auch  
1106 eingeführt. Sind im Grunde auch Erweiterungen. Aber wurden schon um einiges früher  
1107 eingeführt und veröffentlicht. #01:51:50-5#

1108 232. B: Also es geht mehr/ Für den Prozessmodellierer ist das schon brutal abstrakt. Ob er jetzt einen  
1109 Mehrwert hat, wenn man jetzt sieht, dass es ein Script ist oder ein Script mit Service Call, das  
1110 interessiert ihn eigentlich nicht. Auf Management-Ebene, weil das geht auch gar nicht zur  
1111 technischen Realisierbarkeit. #01:52:21-2#

1112 233. I: Also es gibt natürlich auch die Möglichkeit, dass man einerseits nur das Service Request und  
1113 danach einfach das Script ausführt. Also man müsste jetzt nicht unbedingt die Kombination aus  
1114 beiden auch als eigenes Symbol anbieten. Es wurde aber im Sinne der Usability schon früher hier  
1115 eingeführt. Und es gab auch gute Bewertungen dazu. Soweit mir bewusst ist. Ich kenne jetzt  
1116 keine grundsätzlichen Ergebnisse aus diesen früheren Arbeiten, weil sie nicht zu meiner  
1117 speziellen Arbeit oder zu diesem Umfang hier gehören. Aber zur Verständlichkeit natürlich der  
1118 [REDACTED], weil diese Symbole nicht zum Standard von BPMN gehören, wollte ich sie hier natürlich  
1119 auch erwähnt haben, dass es für Sie verständlich ist. #01:53:10-0#

1120 234. B: Ja genau. // I: Was hier abläuft. // Nein, ich überlege mir, wenn jetzt ein Prozesstechnologe  
1121 Ihre Darstellungen anschaut. Dann sieht er viele S. Und dann sieht er noch S mit einem Zahnrad.  
1122 Und das ist für ihn irgendetwas. Aber hat mit dem Prozess wenig zu tun. #01:53:31-1#

1123 235. I: Ja, hier kommt natürlich auch ein bisschen eher die informationstechnische Art und Weise zu  
1124 denken mit hinein, könnte man sagen. Wie gesagt, ich kenne leider die früheren Arbeiten// B:  
1125 Kann man vielleicht nochmal hochgehen? // dazu nicht so im Detail. Zum Modell? #01:53:49-1#

1126 236. B: Weil Sie haben ja jetzt bewusst parallele Pfade verwendet. #01:53:53-6#

---



1127 237. I: Ja. #01:53:54-6#

1128 238. B: Die Frage ist jetzt, wieso wählt man parallele Pfade, wenn überall das Trigger-Ereignis, sage  
1129 ich jetzt wieder, ein wenig provokant, die gleiche Zeit ist? #01:54:11-4#

1130 239. I: Ach so, das ist jetzt wieder eine neue Frage. Entschuldigung, ich habe gedacht, das baut ein  
1131 bisschen. #01:54:18-6#

1132 240. B: Das ist eine neue Frage. #01:54:19-3#

1133 241. I: Okay, ja. #01:54:20-9#

1134 242. B: Also mir geht es darum, wenn man jetzt die Grafik hier ansieht, dann kann das irgendein  
1135 Prozess sein, der dann noch fünf, eins, zwei drei, vier, fünf Messwerte, ein Controller und vier  
1136 Exceptions. #01:54:40-0#

1137 243. I: Ja. Also auf fünf, aber ja. #01:54:44-3#

1138 244. B: Der ist total abstrakt, oder? Das kann jetzt irgendwas sein. (lacht) #01:54:48-5#

1139 245. I: Ja, das kann rein vom physikalischen her irgendetwas sein, aber von der Logik her/ Also das ist  
1140 ja eigentlich die Frage. Ich versuche ja herauszufinden, ob es für Sie speziell ersichtlich ist,  
1141 welche Logik hier hinter abläuft, hinter diesem Prozessmodell oder wofür dieses Prozessmodell  
1142 steht. Ich habe da jetzt natürlich eine ganz eigene Sichtweise darauf und traue mich da jetzt  
1143 gerade auch nichts dazu zu äußern, weil ich Sie nicht beeinflussen möchte. #01:55:20-1#

1144 246. B: Genau. Nein, was für mich also irgendwo fehlt, für den Prozesstechnologen/ Er will ja jetzt  
1145 sehen, wofür die Pfade stehen. Und in der Beschreibung sieht man es irgendwo schon, oder?  
1146 #01:55:33-6#

1147 247. I: In den Labels. #01:55:34-5#

1148 248. B: In den Labels. #01:55:36-3#

1149 249. I: Ja, hier. Okay. #01:55:38-1#

1150 250. B: Genau, also. (...) Eigentlich (...) Eigentlich (unv.) muss er die Labels anschauen, oder? Also das  
1151 ist jetzt eine Beschreibung und da gibt es eine Input Reaktor In, irgendeinem Füllstand, einen  
1152 Operation-Modus und eine Energiesch/ ein Emergency Stop. Das sind meine Daten, die ich jetzt  
1153 einlese mit dieser Frequenz. Und spannenderweise ist die Frequenz immer die gleiche, oder?  
1154 (unv.) meistens typisch und dann ist ja die Frage, wieso, dass man diesen Parameter soviel mal  
1155 hat? Also wieso muss ich den gleichen Parameter, weil ich ihn bewusst gleich verwende, so  
1156 vielmals definieren? #01:56:39-4#

1157 251. I: Okay, ja. Jetzt weiß ich, was Sie meinen. #01:56:42-8#

1158 252. B: Also ein Trigger-Ereignis für Messen und dann parallel diese fünf Werte lesen hätte fast einen  
1159 größeren Charme. Und dann beim Controller. Ja, DER Controller hat jetzt einen Ausgang. Diesen  
1160 Heizwert. Und DEN sehe ich auch jetzt nur quasi in der Description. Und die Exceptions. Ja, da  
1161 sieht man auch, dass es viere sind und welche, sieht man in der Definition. Also ich/ Jetzt, bei der  
1162 Modellierung/ ich überlege halt immer jetzt im Vergleich. Wenn man einen Funktionsblock  
1163 darstellen würde, dann wäre es ein Block, hätte irgendwie fünf Inputs, vier Exception-Inputs und  
1164 ein Output. Also, es wäre ein Block mit diesen vier Inputs und den vier Ereignissen und einem  
1165 Output. Und eigentlich repräsentiert es ja nicht mehr und nicht weniger als das, (...) von der  
1166 Grafik her. Und dann ist es/ ist jetzt ein wenig unfair gewesen. Weil man hier schon die  
1167 Implementierung/ Woher kommen die Daten, hat man hier. Sie können ja theoretisch von über  
1168 verschiedene Transportmedien von unterschiedlichen Sensoren, Steuerungen und Systemen

1169 kommen. Und das setzen hier, ist allein schon irgendein Service, den man irgendwohin schreibt.  
 1170 Und die Exceptions definiert man auch woher, dass die kommen. Und auslösen/ Wenn sie was  
 1171 auslösen außer jetzt in den Stopp-Modus zu gehen, dann hatte man auch ein Script drinnen. (...)  
 1172 Als Erweiterungen wären eher, dass man diese verschiedenen Pfade, dass man die irgendwie  
 1173 beschriften kann. Dass es nicht zufälligerweise ein Text ist, auf dem man schauen muss. Sondern  
 1174 dass man sagt, in dem Pfad, der ist für das, der für das und das und das. #01:59:52-8#

1175 253. I: Also, dass man nicht nur unterscheiden kann, hier wird gemessen und hier ist der Controller  
 1176 oder die Regelungsaufgabe. Sondern dass man auch bei den verschiedenen Messsträngen  
 1177 unterscheiden kann, was eigentlich gemessen wird, aber nicht nur auf Basis des Labels.  
 1178 #02:00:15-2#

1179 254. B: Genau. #02:00:16-0#

1180 255. I: Okay. #02:00:17-0#

1181 256. B: So wie eine Überschrift, die man definieren kann. Und das wäre mehr/ also ist nur  
 1182 informationsmäßig. Oder vielleicht ist es einer/ Ja, vielleicht ist es ein Attribut dieser drei  
 1183 Elemente. Dass man die Elemente bezeichnen kann. Weil das ist ja das Charakteristikum, dass  
 1184 jeder Pfad mit einer dieser drei Measurement, Control oder Exception beginnt. Und diese drei  
 1185 könnten das als Attribut mitnehmen. Und dieses Attribut könnte man dann noch speziell oder  
 1186 mit Tooltip oder irgendwie noch bezeichnen. #02:01:05-0#

1187 257. I: Das heißt, Sie würden da eher auf die physikalische Größe gehen? Und zum Beispiel  
 1188 Temperatur oder so etwas hinschreiben oder? // B: Hm. (zustimmend) // Okay. Verstehe.  
 1189 #02:01:26-1#

1190 258. B: Ja, Sie haben es ja intuitiv bei Get gemacht, oder? Also. Da beim/ irgendwo steht immer beim  
 1191 Script, steht Get. #02:01:42-4#

1192 259. I: Meinen Sie das? #02:01:45-7#

1193 260. B: Ja. #02:01:46-2#

1194 261. I: Ja. #02:01:47-1#

1195 262. B: Da steht immer Get und eine Attribut dann, oder? // I: Ja, bei der Bezeichnung der Variablen.  
 1196 // Und das ist Ihre Variable zu wissen, was da im Script passiert, oder? // I: Ja. // Und  
 1197 spannenderweise kommt ja die/ also was gelesen wird, kommt ja nicht mal vom Script, sondern  
 1198 kommt vom Measurement als Attribut mit. #02:02:05-3#

1199 263. I: Stimmt. #02:02:10-0#

1200 264. B: Aber es könnten eben eins oder mehrere Argumente sein, wenn ich es richtig verstanden  
 1201 habe? #02:02:14-6#

1202 265. I: Ja. #02:02:15-3#

1203 266. B: Und dann wäre hier die Liste/ ich weiß ja nicht, ob es eine Liste gibt, aber da müsste man die  
 1204 drei Überbegriffe bauen können. Wenn es nur ein Attribut ist, dann ist es klar. Aber es so ein Set  
 1205 von Werten sind, die man einliest, dann müsste man so ein Topic oder ein, ja, Topic ist  
 1206 typischerweise so der Wert. Ja. Und dass man dann hier in Klammern das Topic hinschreibt. (...)  
 1207 Oder hier irgendwie vereinfacht. Und rein von der Darstellung ist eigentlich das Topic, was man  
 1208 hier machen will, ist viel wichtiger wie die konfigurierte Trigger-Zeit. #02:03:01-9#

1209 267. I: Vom Verständnis her, würden Sie sagen? #02:03:04-3#

---

1210 268. B: Verständnis eher. Also, ob Sie 200 Millisekunden/ also eine 0,1 Hz, 10 Sekunden sind, das will  
1211 man dann irgendwann schon wissen, aber ich glaub das ist für den Prozesstechnologen für einen  
1212 ersten Überblick ist das nicht so wichtig. Ich glaube, es ist wichtiger, welcher Prozesswert hier  
1213 eingelesen wird. #02:03:28-1#

1214 269. I: Ja, ich verstehe schon, was Sie meinen. Weil im Grunde kann man nur nachvollziehen, um  
1215 welche physikalische Größe es sich handelt, wenn man entsprechend die Variable richtig  
1216 benannt hat. Im Label. #02:03:49-8#

1217 270. B: Ja. #02:03:52-9#

1218 271. I: Ja, ist ein guter Punkt. Dankeschön. Okay. Ja, dann haben wir noch ein paar Sachen. Ich  
1219 bemühe mich jetzt nicht noch allzu lange dauern zu lassen. Ich hoffe, es geht für Sie noch.  
1220 #02:04:14-9#

1221 272. B: Ah ja, ist schon lange. Ja, geht schon. #02:04:20-2#

1222 273. I: Danke auf jeden Fall, dass Sie sich Zeit nehmen. Ja, eine Zwischenfrage bevor wir nochmal zu  
1223 einer Bewertung gehen. Wären Sie aufgrund dieser Erweiterungen bereit, diese  
1224 Modellierungsmethode mit den Erweiterungen in BPMN, in Ihrem Arbeitsalltag einführen zu  
1225 wollen, wenn Sie ein Modell eines kontinuierlichen Prozesses entwickeln oder generell  
1226 darstellen wollen würden? Modellieren wollen würden? Also wären Sie bereit, auf diese  
1227 Methode zurückzugreifen? #02:04:54-6#

1228 274. B: Jetzt kommt es darauf an, ob die Bedingung ist, dass man eh schon in BPMN modelliert oder  
1229 weggeht von einer anderen Modellierungsmethode. Also, wenn man BPMN hat, dann glaube  
1230 ich ist es wichtig, dass man eine standardisierte Methode hat oder ein Meta-Modell, wie man  
1231 kontinuierliche Prozesse in die doch diskrete Modellierungsmethoden hineinbringt. Das ist  
1232 immer ein Thema. Also entweder, man ist kontinuierlich, hat man bei diskreter Darstellung ein  
1233 Problem oder umgekehrt. Und unsere Systeme haben halt beide Elemente beisammen. Darum  
1234 wenn in einem BPMN-System schon/ wenn jetzt die Frage ist eher, dass man in einem anderen  
1235 möglich Modellierungsmethode wird BPMN, also angenommen man hat  
1236 Funktionsblockprogramm #02:05:48-0#

1237 UNTERBRECHUNG bei Frage 6 des Leitfadens

1238 NEUE AUFNAHME

1239 275. B: für die Maschine und Symbole und alles abbilden, dann endet man mit hundert tausenden  
1240 von States. Und am Schluss muss man das/ gibt es verschiedene Methoden, wie man die  
1241 Zustandsmodelle dann segmentiert, entweder mit Substates oder gibt andere Methoden, sodass  
1242 eine Zustandsmaschine, die man auch im Überblick hat, hat, irgendwo fünf bis zehn Zustände.  
1243 Wenn es mehr sind, gibt es zu viele und dann wird die Komplexität zu groß in der grafischen  
1244 Darstellung. Dann muss man das reduzieren. #00:00:30-8#

1245 276. I: Ja. #00:00:31-7#

1246 277. B: Und diese Methodik muss man hier ja auch andenken, weil sonst, wenn es keine so Methodik  
1247 gibt zum Modularisieren, dann wird es in der Regel nicht funktionieren. Dann funktioniert es für  
1248 so einfache Modelle, wie man jetzt für das Besprechen natürlich nutzt. Aber in der Realität sind  
1249 die Modell/ also sind die Anwendungen sind dann komplexer und dann muss man die irgendwie  
1250 kaskadieren oder verschachteln oder irgendwie aufsplitten können. #00:01:03-0#

1251 278. I: Ja natürlich, das haben wir uns auch überlegt ja, das stimmt schon. Das war eigentlich auch  
1252 schon der letzte Punkt. Vielen Dank, dass Sie sich Zeit genommen haben. Danke, dass Sie das so  
1253 lange mit mir durchgehalten haben. #00:01:21-5#

---

1254 279. B: Gerne gerne. #00:01:22-9#

1255 280. I: Ich bin auf jeden Fall dankbar, für Ihr Feedback zu unseren Erweiterungen und wenn Sie auch  
1256 Feedback zum Ablauf des Interviews haben, können Sie mir das gerne auch natürlich sagen. Also  
1257 ob ich hier etwas vielleicht an der Fragestellung anpassen kann. Ob die Fragen einfach genug  
1258 gestellt waren und für Sie angenehm beantwortbar oder ob Sie vielleicht zu komplex waren.  
1259 Oder wie Sie die Dauer des Interviews empfunden haben. #00:01:49-5#

1260 281. B: Nein, war gut. Wenn die Zeit schnell durchgeht, dann ist es immer ein gutes Zeichen.  
1261 #00:01:56-2#

1262 282. I: Also wenn man gar nicht merkt, wie schnell, wie wie viel Zeit eigentlich/ Ja, ich verstehe. (lacht)  
1263 Also danke auf jeden Fall dafür. Ich werde jetzt auch/ #00:02:05-9#

1264 283. B: Man sagt immer, den Zeitstrahl könne man nicht verändern, oder? #00:02:08-6#

1265 284. I: Ja das stimmt. Aber eigentlich/ #00:02:11-0#

1266 285. B: Ab und zu geht es doch. #00:02:11-9#

1267 286. I: Ja, die Wahrnehmung ändert sich dann etwas. #00:02:14-2#

1268 287. B: Genau. #00:02:14-2#

1269 288. I: Ja, ich werde dann gleich auch die Aufnahme stoppen, dass Sie das sehen.

---