

## Transcript - Group 3 "Modelers", Interview 4

I ... Interviewer (BLINDED)

B ... Expert

(Unv.)... Incomprehensible passage

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

( ) ... Comment

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

### Transcript

1 1. I: Gut. Aufnahme läuft. Hallo und danke, dass Sie sich Zeit nehmen, um mit mir dieses Interview  
2 durchzuführen. [REDACTED]  
3 [REDACTED] Möchten Sie sich vielleicht kurz vorstellen und Ihre Verbindung zu BPMN  
4 beziehungsweise zur Verfahrenstechnik, Fertigungstechnik oder Prozessmodellierung erklären?  
5 Ich möchte Sie aber dabei bitten, nicht Ihren Namen zu nennen, sondern nur die folgenden  
6 Informationen. Und zwar Berufsbezeichnung und Umschreibung Ihres Arbeitgebers, Basis Ihrer  
7 Expertise, Ausbildung beziehungsweise fachlicher Hintergrund und ungefähre Berufserfahrung.  
8 Bitte. #00:00:42-4#

9 2. B: Ja, sehr gerne. [REDACTED]  
10 [REDACTED]  
11 [REDACTED]  
12 [REDACTED]  
13 [REDACTED]  
14 [REDACTED]  
15 [REDACTED]  
16 [REDACTED]  
17 [REDACTED]  
18 [REDACTED]  
19 [REDACTED]  
20 [REDACTED]  
21 [REDACTED]  
22 [REDACTED]  
23 [REDACTED]  
24 [REDACTED]  
25 [REDACTED]  
26 [REDACTED] #00:02:27-1#

27 3. I: Okay, wunderbar. Dankeschön. Dann möchte ich eine kurze Einführung zu unserem  
28 Forschungsthema geben. Und zwar, unsere Forschung konzentriert sich auf die Entwicklung  
29 einer Methodik, um kontinuierliche Prozesse in BPMN darzustellen und sie in einer Workflow  
30 Engine ausführbar zu machen. Für diese Aufgabe haben wir an BPMN-Erweiterungen für  
31 kontinuierliche Prozesse gearbeitet. Warum kontinuierliche Prozesse? Weil diskrete Prozesse  
32 bereits in anderen Forschungsarbeiten behandelt wurden und nicht die gleichen Schwierigkeiten  
33 bei der korrekten Darstellung mittels BPMN aufweisen. BPMN ist bereits ein weit verbreiteter  
34 Standard im Business Process Management und hat seinen Weg in die Fertigung gefunden.  
35 Diskrete Fertigungsprozesse können bereits mit BPMN 2.0 modelliert werden. Im Grunde wollen  
36 wir eine Methodik einführen, um solche Prozesse so darzustellen, dass sie von jeder Person in  
37 einem Unternehmen vom Ingenieur bis zum Manager verstanden werden können. Dies könnte  
38 auch durch die Verwendung dieser Notation erreicht werden. Ein weiterer Vorteil ist auch, dass  
39 es bereits eine Reihe von Workflow Engines gibt. Anwendungen, die die Ausführung dieser  
40 Prozessmodelle auf der Grundlage der für jedes Symbol implementierten Logik ermöglichen. Wir  
41 arbeiten mit einer webbasierten Anwendung, die erweiterbar ist und mehrere  
42 Kommunikationsschnittstellen implementiert hat. Also HTTP, OPC UA zum Beispiel, solche  
43 Sachen. Ein weiterer Vorteil ist daher die Interoperabilität in diesem Zusammenhang im  
44 Vergleich zu anderen proprietären starren Software-Anwendungen. Wir wollen herausfinden, ob

diese Technik auch für die Implementierung von digitalen Abbildern eingesetzt werden kann. Da digitale Abbilder dazu dienen, ein physikalisches System oder einen Prozess in digitaler Form darzustellen, meist anhand von Daten oder mathematischen Modellen, mussten wir einen Weg finden, den Ablauf von kontinuierlichen Prozessen wie sie aus der Prozessindustrie bekannt sind, darzustellen. Und aus diesem Grund haben wir uns auf die Modellierung von Regelkreisen konzentriert. Die Prozessmodelle sollen durch BPMN für Personen mit unterschiedlichem Hintergrund leicht verständlich sein. Die Interviews werden geführt, um herauszufinden, wie Prozess- und Regelungstechnik und Techniken aus der Business-Process-Modellierung kombiniert werden können und wie erste Ergebnisse von Experten wie Ihnen wahrgenommen werden. Außerdem wollen wir herausfinden, ob es Schwachstellen gibt, die von Experten identifiziert werden und wie wir diese beseitigen können. Und bevor wir zum Fragenteil kommen, möchte ich noch drei Begriffe vorab klären, was wir darunter verstehen. Nämlich, der erste Begriff digitaler Zwilling. Es gibt verschiedene Methoden, Dinge aus der echten Welt, zum Beispiel echte Maschinen, zu simulieren. Teils merkt man aber, dass es mehr Parameter brauchen würde als bei normalen Simulationsmethoden um eine Maschine vollkommen so abzubilden, wie sie sich in der Realität verhält. Bei einem digitalen Zwilling wird versucht, möglichst nahe an das reale Verhalten einer Maschine oder anderer Objekte heranzukommen. Das soll dazu führen, dass, wenn etwas getriggert wird bei einer echten Maschine, der digitale Zwilling das gleiche oder ein möglichst ähnliches Verhalten zeigt. Der zweite Begriff wäre kontinuierlicher Prozess. Kontinuierliche Prozesse möchte ich mit Beispielen erklären. Wenn man Bierbrauen her nimmt, gibt es zwei Möglichkeiten. Die diskrete, also die nicht-kontinuierliche Variante wäre, wenn man in einem geschlossenen Kessel die Zutaten hinein gibt, zehn Liter Wasser beispielsweise, und den Brauprozess einfach schrittweise ablaufen lässt. Am Ende kommt dann eine begrenzte Menge an Bier heraus. Die andere kontinuierliche Variante wäre, wenn man keinen vollkommen abgeschlossenen Kessel hat, sondern miteinander verbundene Kessel, bei denen immer wieder Zutaten zugefügt werden und immer wieder Bier entnommen wird. Das geht die ganze Zeit so, so dass man nicht nachvollziehen kann, welcher Liter Wasser zu welchem Liter Bier gehört. Dabei läuft ein Teilprozess im ersten Kessel ab, während gleichzeitig im letzten Kessel der letzte Prozessschritt stattfindet, bevor das Bier fertig wird. Und der letzte Begriff 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, die zehn Liter Wasser, bei denen etwas schief gehen kann und mit den nächsten zehn Litern macht man es dann besser. Aber was ist, wenn man die Brauanlage dauernd laufen lässt und ständig Bier am Ende 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 entsprechend. Stimmt etwas zum Beispiel beim Zucker- oder Alkoholgehalt nicht, muss das Mischverhältnis geändert werden. Das heißt, in einem geschlossenen Regelkreis werden während der Prozess läuft gewisse Werte überprüft. Diese werden mit optimalen Werten verglichen und je nach Abweichung reagiert dann das System darauf. Ja, kommen wir zum Fragenpart. Aus Informatiksicht bestehen kontinuierliche Prozesse aus einer sich ständigen wiederholenden Abfolge von Zustandsabfragen und Regulierungen. Zustandsabfragen und Regulierungen sind jeweils traditionelle Code-Stücke, die sich auf Sensoren oder Aktoren beziehen. Um solche kontinuierlichen Prozesse konsistent formal zu beschreiben, zu modellieren und in weiterer Folge ausführen zu können, haben wir eine Reihe von Merkmalen identifiziert. Und die erste Frage wäre/ ich würde Ihnen die Merkmale vortragen und //B: Okay. // würde Sie bitten, die Merkmale für wichtig oder unwichtig einzustufen, also Ihrer Meinung nach wichtig oder unwichtig. Und ich würde Sie auch bitten dafür eine kurze Begründung anzugeben. Ich lese ganz kurz die Merkmale vor. Und zwar einmal. Verschiedene Zustandsabfragen- und Regulierungskombinationen sind unabhängig und können parallel ablaufen. Nummer Zwei. Regulierungen folgen immer auf Zustandsabfragen. Nummer Drei. Die Dauer von jeder Zustandsabfrage- und Regulierungskombination ist beschränkt. Nummer Vier. Wenn Zustandsabfragen gewisse Ergebnisse liefern, wird das System beendet. Nummer Fünf. Bevor

101 das System beendet wird, muss es in einen konsistenten Zustand gebracht werden. Und  
102 Nummer Sechs. Das resultierende System soll für Menschen verständlich sein. #00:09:27-8#

103 4. B: Okay. #00:09:29-5#

104 5. I: Okay. Fangen wir mit Nummer Eins an. Verschiedene Zustandsabfragen- und  
105 Regulierungskombinationen sind unabhängig und können parallel ablaufen. Würden Sie das für  
106 wichtig oder unwichtig einschätzen? Und warum? #00:09:42-2#

107 6. B: Also, das ist sicher richtig, dass verschiedene von diesen Kombinationen unabhängig sind und  
108 parallel laufen können. Ob das jetzt wichtig oder unwichtig ist, würde ich fast sagen, dass das  
109 vom Regelkreis abhängt, also in manchen Regelkreisen ist das sicher wichtig, dass man möglichst  
110 viel parallelisiert und möglichst viel zur gleichen Zeit macht, weil das sonst extrem aufwendig ist.  
111 Bei anderen ist vielleicht der Aufwand zu hoch, um das wirklich parallelisieren zu können  
112 möglicherweise, und dann zahlt es sich vielleicht nicht aus. Aber generell würde ich schon sagen,  
113 natürlich, absolut richtig. Die können oft parallel laufen und wenn man sie parallelisieren kann,  
114 dann warum nicht? Da ist das sicher eine gute Sache und sicher richtig, dass so zu tun.  
115 #00:10:27-6#

116 7. I: Hm. (zustimmend) Danke. Nummer Zwei. Regulierungen folgen immer auf Zustandsabfragen.  
117 #00:10:34-8#

118 8. B: Also so wie ich jetzt diesen Regelkreislauf hier verstanden habe, ist das absolut wichtig. Weil  
119 man nimmt die Regulierung ja immer erst vor, weil man irgendeinen Status von den  
120 Zustandsabfragen bekommen hat. Also wenn wir jetzt nichts, keinen Grund dafür hat, irgendwas  
121 zu regulieren oder irgendwas zu ändern, dann gibt es auch keinen Grund dafür, diese  
122 Regulierung durchzuführen. Von daher sind ja eigentlich immer die Zustandsabfragen der  
123 Auslöser für die Regulierungen. Und daher würde ich sagen, ja, ist wichtig. #00:11:05-0#

124 9. I: Nummer Drei. Die Dauer von so einer Kombination, also Zustandsabfrage und Regulierung, ist  
125 beschränkt. #00:11:13-9#

126 10. B: Die Dauer ist beschränkt. Na ja, solange das Problem besteht, ist es wichtig, das Problem  
127 entsprechend zu beheben und zum Beispiel, keine Ahnung wie es im Beispiel vorher war, mehr  
128 Zucker dazu zu geben. Das heißt natürlich, ja, ist es beschränkt, dass/ irgendwann wird das  
129 Problem behoben sein. Dann wird vielleicht ein neues Problem auftreten. Wichtig oder  
130 unwichtig, tue ich mir da jetzt ein bisschen schwer. Ja, also es ist sicher eine Tatsache. Ich würde  
131 es jetzt aber nicht als besonders wichtig einstufen, weil im Prinzip ist das so eine Folge aus Punkt  
132 Zwei. Ich habe die Zustandsabfrage und wenn die nächste Zustandsabfrage daher kommt,  
133 irgendwer sagt, 'Ja, es passt wieder', dann höre ich mit der Regulierung auf. Aber jetzt so an sich  
134 würde ich das eher als unwichtig einstufen, (unv.). #00:12:04-2#

135 11. I: Nummer Vier. Wenn Zustandsabfragen gewisse Ergebnisse liefern, wird das System beendet.  
136 #00:12:14-5#

137 12. B: Ja, absolut richtig und wichtig. Das muss man halt auch dann korrekt definieren. Also wieder  
138 am Beispiel des Bierbrauens, mich daran einmal festzuhalten. Irgendwann ist das Bier einfach so  
139 kaputt, dass man es nur weg schütten kann. Und das ist dann quasi, nehme ich an, der Zustand,  
140 auf den hier Punkt Vier abzielt. Also im Prinzip so wie Punkt Zwei, immer wieder Regelungen  
141 triggert, so wird dann Punkt Vier einfach das Wegschütten oder das Beenden des Systems  
142 triggern, wo man dann einfach nur noch von Neu starten kann. Also von daher würde ich als  
143 absolut wichtig einschätzen. #00:12:48-6#

144 13. I: Okay, danke. Nummer Fünf. Bevor das System beendet wird, muss es in einen konsistenten  
145 Zustand gebracht werden. #00:12:57-5#

---

- 146 14. B: Bevor das System beendet wird. Ja, konsistenter Zustand. Das heißt, man muss wieder in  
147 einen Zustand kommen, von dem man wieder starten kann, nehme ich an, ist dann gemeint.  
148 Richtig? #00:13:09-8#
- 149 15. I: Also, es ist gemeint, dass man den Zustand definiert, in dem man das System überführen  
150 möchte, bevor es abgeschaltet wird, könnte man sagen, oder beendet wird. #00:13:19-5#
- 151 16. B: Also dass man quasi den Zustand des Systems kennt, wenn man es herunterfährt //I: Genau.  
152 // und abdreht. #00:13:26-1#
- 153 17. I: Genau, ja. #00:13:26-9#
- 154 18. B: Ja, ich würde auch sagen, dass das sehr stark vom System abhängt. Natürlich ist es eine gute  
155 Sache, wenn es funktioniert und wenn man es noch in einen konsistenten Zustand bringen kann.  
156 Aber ich würde es jetzt nicht als Top-Priorität einschätzen. Aber ja, es kann natürlich auch  
157 Anwendungsfälle geben, wo es wichtig ist. Und teilweise auch Fälle, wo es vielleicht gar nicht  
158 möglich ist, das in den konsistenten Zustand zu bringen, je nachdem, wie eben die Parameter  
159 gerade sind, zum Zustand, wo man halt beschließt, jetzt abdrehen zu müssen. Wenn es sich um  
160 eine beabsichtigte Abschaltung handelt, warum auch immer, was offenbar bei den  
161 kontinuierlichen Prozessen nicht häufig vorkommt, dann absolut wichtig, sonst eher nicht so.  
162 #00:14:11-5#
- 163 19. I: Und der letzte Punkt zu Frage Eins. Das resultierende System soll für Menschen verständlich  
164 sein. #00:14:21-1#
- 165 20. B: Ja, das ist natürlich absolut wichtig, dass die Menschen dann auch wissen, womit sie es  
166 eigentlich zu tun haben und möglichst intuitiv verstehen, was denn die einzelnen Bausteine  
167 bedeuten. Also würde ich definitiv als wichtig einschätzen. #00:14:33-8#
- 168 21. I: Okay, super. Nach diesen sechs Eigenschaften, die wir jetzt durchgegangen sind, können Sie  
169 vielleicht grafische Eigenschaften nennen, die Sie für die Modellierung kontinuierlicher Prozesse  
170 für wichtig halten? Und ergeben sich daraus vielleicht Merkmale, die wir hier in dieser Liste  
171 vergessen haben? Also da geht es jetzt wirklich um die grafische Darstellung. #00:14:57-5#
- 172 22. B: Also, generell natürlich, dass es irgendeine gewisse Leserichtung gibt, in der man diese  
173 kontinuierlichen Prozesse lesen kann. Also entweder immer von oben nach unten oder immer  
174 von links nach rechts. Aber dass man halt nicht wild mit den Augen herum hüpfen muss bis man  
175 mal sich wieder zurechtgefunden hat, wo man sich jetzt befindet. Generell ist auch wichtig, dass  
176 wenn man jetzt mehrere Prozesse hat, die, keine Ahnung, im selben Unternehmen, zum Beispiel,  
177 zu finden sind, das die noch eine gewisse Konsistenz haben untereinander. Keine Ahnung. Dass  
178 ich, wenn ich mich schon begonnen habe, mit dem Prozess auseinander zu setzen, nicht immer  
179 wieder schauen muss, wo fängt denn der Prozess jetzt eigentlich an. Sondern ich weiß immer,  
180 okay, der fängt immer links oben an und geht dann generell eher nach rechts hinüber. Also, dass  
181 man da eine gewisse Kontinuität hat, zwischen den verschiedenen Prozessen, die es gibt. Und  
182 dann, ja, gut, natürlich die ganzen Dinge, die in BPMN eh schon abgehandelt sind, wie dass man  
183 eindeutige Grafiken, Icons, der Präsentation der einzelnen Schritte. Aber das ist eh etwas, was  
184 durchaus klar ist. Ja, so auf die Schnelle sind das mal die Punkte, die mir jetzt eingefallen sind.  
185 Ergeben sich daraus Merkmale, die wir vergessen haben? Ja, eben diese Kontinuität, dass man  
186 eben sich möglichst schnell wieder zurecht findet in den einzelnen Prozessen. Aber das ist im  
187 Prinzip ein Unterpunkt vom sechsten Punkt, meiner Meinung nach. #00:16:26-7#
- 188 23. I: Okay, verstehe, ja. Wenn man jetzt darüber nachdenkt, wie man kontinuierliche Prozesse  
189 modellieren würde, wo liegen denn Ihrer Meinung nach die Herausforderungen dabei?  
190 #00:16:40-8#
- 191 24. B: Wo liegen die Herausforderungen, jetzt speziell von kontinuierlichen Prozessen? Also/  
192 #00:16:47-2#
-

- 193 25. I: Bei der Modellierung kontinuierlicher Prozesse. Genau, ja. #00:16:49-8#
- 194 26. B: Bei der Modellierung, ja. Also natürlich vor allen Dingen eben die Möglichkeit zu schaffen,  
195 dass man diese ganzen Zustandsabfragen, Regulierungen irgendwie unter bekommt. Dass man  
196 halt eben immer sehen kann, okay, was sind denn meine Trigger anhand/ was sind denn die  
197 Gründe, weshalb ich etwas anders mache auf einmal, als ich es sonst immer mache. Also wenn  
198 ich jetzt keine Prozessänderungen habe, was wahrscheinlich bei kontinuierlichen Prozessen nur  
199 nicht so oft eine gute Idee ist, sondern halt immer im laufenden Prozess darauf reagieren  
200 möchte, dann brauche ich halt irgendwie eine Notation, um grafisch zu sehen, was ändere ich  
201 denn und was tue ich in welchem Zustand, und warum tue ich es. Also da ist, denke ich mir, halt  
202 wirklich eine Notation erforderlich um zu sehen, wo man denn/ wie man so etwas unterbringt.  
203 Und was weitere Details angeht jetzt, was kontinuierliche Prozesse betrifft, muss ich sagen, dass  
204 ich damit zu wenig zu tun hatte, um mehr dazu zu sagen. Aber es soll natürlich immer nur ganz  
205 wichtig sein, wenn es ein kontinuierlicher Prozess ist, wie Bierbrauen. Dass man halt dann immer  
206 sieht, wo ist der Anfang, wo ist das Ende. Und in der Modellierung, ja, eben eine Übersicht hat  
207 darüber, was für einzelne Schritte das sind. Viel mehr/ (unv.) fällt mir nicht viel mehr dazu ein.  
208 #00:18:15-9#
- 209 27. I: Okay, danke. Ich werde Ihnen Prozesse zeigen, Prozessbeispiele zeigen, die mit BPMN 2.0 und  
210 mit unseren Erweiterungen, die wir definiert haben, modelliert wurden. Die Erweiterungen  
211 sollen zum einen vordefinierte Modellierungskonventionen für in der Prozess- und  
212 Steuerungstechnik übliche Routinen bereitstellen, und zum anderen helfen die Unterschiede  
213 zwischen den parallelen Pfaden in den Prozessmodellen zu visualisieren. Die Prozesse werden in  
214 der [REDACTED], modelliert. Sie sind mit diesem Tool ja auch  
215 bekannt? Also, Sie kennen es bereits, die [REDACTED]. Soweit ich //B: Ja, absolut. // weiß. Super. Und,  
216 es ist nämlich auch so. In unseren Prozessmodellen werden auch zum Beispiel Scripts, Service  
217 Calls und Service Calls mit Scripts eingesetzt und sind auch ein Teil der Prozessmodelle. Und da  
218 wollte ich nur kurz nachfragen, ob Sie diese Symbole eh auch kennen und ungefähr wissen,  
219 wofür sie stehen, weil dann muss ich die natürlich jetzt auch nicht dazwischen erklären.  
220 #00:19:26-7#
- 221 28. B: Ja, mit denen kenne ich mich aus. #00:19:28-4#
- 222 29. I: Wunderbar, danke. Dann komme ich direkt zu unseren Erweiterungen, die sich im Zuge dieser  
223 Arbeit ergeben haben. Und zwar das Closed Loop Subsystem Gateway. Das Gateway ist eine  
224 Kombination, könnte man sagen, aus einem inklusiven und einem ereignisbasierten Gateway. Es  
225 enthält Verzweigungen beziehungsweise Kanten, die für die Zustandsabfragen- und  
226 Regulierungsphasen des Zyklus ausgelöst werden, sowie Kanten, die beim Empfang von  
227 Abbruchereignissen ausgeführt werden. Die Ereignisse und Tasks in den einzelnen Kanten sind  
228 unabhängig voneinander. Und damit erfüllen wir das erste der oben genannten Features, die  
229 erste Eigenschaft, dass einzelne Verläufe unabhängig voneinander sind und sie parallel  
230 ausgeführt werden. Das Gateway ermöglicht außerdem die Definition der Intervalldauer jedes  
231 Zyklus über Measure und Control, sowie von Überschreitungsbedingungen, also wait oder cancel,  
232 und der Ausführungsreihenfolge für Zustandsabfragen und Regulierungen. Beziehungsweise  
233 könnte man hier auch sagen, Mess- und Steuerungsaufgaben. Die wichtigsten Attribute, die man  
234 bei einem Closed Loop Subsystem Gateway definieren kann, sind Interval duration overrun und  
235 Measure control cycle execution. Interval duration overrun definiert/ oder wird definiert  
236 entweder durch cancel oder wait, und Measure control cycle execution durch parallel oder  
237 sequentiell, also parallel oder sequential. Ich möchte hier gerne den Unterschied jetzt noch  
238 erklären. Und zwar, wenn wait gewählt wird, beginnt die nächste Iteration im Closed Loop  
239 Subsystem, wenn alle Verzweigungen beendet sind und die festgelegte Intervalldauer erreicht ist.  
240 Bei cancel definiert die Intervalldauer genau die Zeit, in der jeder Zweig zu beenden ist. Wenn  
241 die Tasks in einem Zweig schneller beendet werden, wird der Zweig warten. Wenn noch nicht  
242 alle Tasks beendet sind, werden sie abgebrochen. Bei parallel oder sequentiell. Bei parallel  
243 werden die Tasks nach Measure und Control Events parallel ausgeführt. Bei sequentiell werden  
244 die Tasks nach Control Events erst ausgeführt, nachdem alle Tasks nach Measure Events beendet

245 sind. Also Measure und Control Events werden gleich auch erklärt. Das sind nämlich auch  
246 Erweiterungen, die wir hier eingeführt haben. #00:22:11-6#

247 30. B: Okay, also im Prinzip auf der linken Seite bei parallel bedeutet es einfach, dass die Dinge  
248 parallel ausgeführt werden //I: Genau. // können? #00:22:17-6#

249 31. I: Ja. #00:22:18-7#

250 32. B: Und bei der rechten Seite bedeutet es, dass sie nacheinander ausgeführt werden. So weit, so  
251 klar. Wait und cancel habe ich ehrlich gesagt noch nicht so ganz verstanden. Bei wait warte ich  
252 erst, ganz am Ende nach Measure Eins und Measure Zwei, beginnt die nächste Iteration, wenn  
253 alle Verzweigungen beendet sind. Da habe ich keine Verzweigung, da habe ich ja nur einen Pfad.  
254 #00:22:37-3#

255 33. I: Okay ja, das ist nur ein Beispiel, wie es ausgeführt werden würde. //B: Okay. // Aber wenn man  
256 im Closed Loop Subsystem Gateway mehrere Measure Events nebeneinander hätte, also in  
257 parallelen Kanten, dann/ mehrere Measures und mehrere Controls nebeneinander angeordnet  
258 hätte in parallelen Kanten, dann würde man bei wait warten, bis all diese Kanten fertig sind. Also,  
259 man würde warten, dass wirklich jeder Strang mit jeder Task vollständig ausgeführt wurde. Und  
260 dann geht erst das Closed Loop Subsystem in den nächsten Zyklus über. #00:23:17-1#

261 34. B: Das heißt, wait wäre dann quasi nachdem alle Kanten zusammengeführt worden sind.  
262 #00:23:23-2#

263 35. I: Ja, genau. #00:23:24-6#

264 36. B: Okay, gut. Das war da jetzt nicht ganz klar ersichtlich. #00:23:27-3#

265 37. I: Okay. Gut, dass Sie nachfragen. Ich will natürlich, dass Sie das auch möglichst //B: Okay. Nein,  
266 ich kenne mich schon aus. // verstehen für die Beispiele. Okay, und bei cancel geht es darum,  
267 man kann bei Measure und Control jeweils für die Kante definieren, wie sich die Zykluszeit  
268 verhält. Also was für eine Frequenz/ in welcher Frequenz die Events ausgelöst werden. Und  
269 diese zeitliche Bedingung steht auch dafür, wie lange es dauern darf, dass dieser Strang beendet  
270 wird. #00:24:00-1#

271 38. B: Okay, das heißt also quasi danach wird dann noch das Control kommen und das ist halt da  
272 aber nicht drinnen. #00:24:08-9#

273 39. I: Da geht es im Allgemeinen um die Ausführung des gesamten Closed Loop Subsystems.  
274 #00:24:13-5#

275 40. B: Ja, ich glaube ich habe es. Okay. #00:24:14-2#

276 41. I: Okay, gut. Wir gehen jetzt eh schrittweise durch, was das alles auch heißt, welche Attribute es  
277 auch bei Control und Measure gibt. Und wenn sich nach der Introduction der Extensions noch  
278 Fragen ergeben, dann können wir das eh auch noch klären. Und es kommen dann auch noch  
279 zwei verschiedene Prozessbeispiele, wo die angewendet werden. #00:24:34-8#

280 42. B: Hm. (zustimmend) #00:24:35-6#

281 43. I: Okay, passt. In einem Closed Loop Subsystem werden spezifische Ereignisse erwartet. Die  
282 wurden jetzt schon erwähnt, Measure, Control, die in eine der drei folgenden Kategorien fallen.  
283 Wir haben einmal Ereignisse oder Events für Zustandsabfragen oder Messungen, Ereignisse für  
284 Regulierungen und Ereignisse für die Unterbrechung des Closed Loop Subsystems. Es gibt für  
285 jede Ereigniskategorie zumindest eine Kante, die vom Gateway ausgeht. Die Kanten zeigen an,  
286 welche Tasks nebeneinander ablaufen. Sobald diese Ereignisse eintreten, werden auch die Tasks,  
287 die in den Kanten danach angeordnet sind, ausgeführt. Hier sehen Sie ein Bild, da gehe ich ein

288 bisschen hinunter dann, eines Closed Loop Subsystems, in dem nur Ereignisse der drei  
289 Kategorien ohne darauf folgende Tasks modelliert sind. Das heißt, wenn man das in der [REDACTED]  
290 einfügen würde, das Closed Loop Subsystem, dann hätte man hier mal eine Edge, eine Kante, mit  
291 einem Measure Event, eine Kante mit einem Control Event und eine Kante mit einem Cancel  
292 Event. Diese drei Kategorien werde ich jetzt etwas genauer erklären. Und zwar, Measure  
293 empfängt Events für die Ausführung von Tasks in Messzyklen, Zustandsabfragezyklen. Control  
294 empfängt Events für die Ausführung von Tasks in Regelzyklen, Regulierungszyklen. Und Cancel  
295 empfängt Events für das Abbrechen von Closed Loop Subsystems. Und diese Symbole geben  
296 dem Zweck der darauf modellierten, also der darauf folgenden Tasks oder Aufgaben an. Diese  
297 Tasks werden nur ausgeführt, wenn die Ereignisse ausgelöst werden. Das bedeutet, dass das  
298 Messereignis angibt, dass die nachfolgenden Symbole für Messabläufe beziehungsweise  
299 Zustandsabfragen eingesetzt werden. Das gleiche gilt für Regulierungs- und auch  
300 Abbruchereignisse. Für Zustandsabfragen und Regulierungen können wir eine Zykluszeit  
301 definieren. Dadurch kann die Dauer von Anpassungen im System definiert werden. Je nachdem,  
302 ob das Closed Loop Subsystem einen parallelen oder sequentiellen oder einen Wait- oder  
303 Cancel-Ansatz verfolgt, das haben wir oben gesehen, läuft die Ausführung unterschiedlich. Mit  
304 diesen Bedingungen kann man definieren, inwiefern Anpassungen beim System erfolgen.  
305 #00:27:16-0#

306 44. B: Okay. #00:27:18-9#

307 45. I: Hier sehen Sie ein Closed Loop Subsystem mit einem Task für eine Messung. In diesem Fall  
308 wird das Ereignis für die Messung alle zehn Sekunden getriggert, also mit einer Frequenz von 0,1  
309 Hertz. Danach wird der Wert V 1, also hier Get process value V 1 geholt beziehungsweise  
310 gemessen, könnte man sagen. Wait bedeutet hier, dass ein neuer Zyklus erst gestartet wird,  
311 wenn die Messung erfolgt, das heißt, der Prozess in dieser Kante abgeschlossen ist. Mit cancel  
312 wird nach zehn Sekunden automatisch der neue Zyklus gestartet. Also hier haben wir jetzt nur  
313 nach dem Measure Event mal //B: Okay. // eine Task eingefügt. Also ein Service Call eingefügt.  
314 Und im Grunde geht es hier nur darum, dass dieser Wert gemessen, geholt, abgefragt wird. Das  
315 wäre unsere Zustandsabfrage. #00:28:14-2#

316 46. B: (Unv.) Okay, man sieht jetzt am Closed Loop Subsystem Gateway nicht direkt, ob es ein Wait-  
317 oder Cancel-System ist. Das sieht man nur, indem man den Text daneben liest, richtig?  
318 #00:28:28-3#

319 47. I: Genau, über die Labels seitlich. Ja. #00:28:30-0#

320 48. B: Okay, passt. #00:28:32-3#

321 49. I: Sie werden dann auch bei den späteren Beispielen sehen, dass da dann zwei Indikatoren  
322 stehen, eigentlich zwei Strings, nämlich einmal, ob es wait oder cancel ist, und einmal, ob es  
323 parallel oder sequential ist. #00:28:44-8#

324 50. B: Okay. #00:28:46-0#

325 51. I: Das kommt dann eh ein bisschen weiter unten. Die Attribute, die man definieren kann, für  
326 Measure mal, sind im Grunde einmal Interval frequency in Hertz. Also, das heißt, diese Zykluszeit,  
327 könnte man sagen. Beziehungsweise die Frequenz, in der das Event ausgelöst werden soll. Und  
328 man kann auch angeben Values expected to change, also welche Werte oder welche Variablen,  
329 wie man es auch immer bezeichnen möchte, würden hier überschrieben werden  
330 beziehungsweise was würde hier sich ändern. Und warum steht hier jetzt Add value? Man  
331 könnte zum Beispiel auch sagen okay, wenn ich jetzt in dieser Zykluszeit eine gewisse  
332 vordefinierte Sequenz haben möchte an Zustandsabfragen, also dass wirklich zuerst eine  
333 Zustandsabfrage kommt und dann erst eine nächste darauf, dann könnte man hier sagen, dass  
334 sich natürlich nicht nur einen Wert ändert, also nicht nur ein Wert abgefragt wird, sondern  
335 vielleicht sogar noch ein Zweiter. Also nicht nur V 1 sondern vielleicht sogar V 2, //B: Okay. //  
336 könnte man definieren, wenn man das möchte. #00:29:50-6#

---

337 52. B: Okay. Also, man sieht nur die Info momentan, da kommen natürlich noch keine Werte hinein.  
338 Da sage ich nur mal, was ich denn erwarte, was von den darauf folgenden Schritten daher  
339 kommt, //I: Genau. // früher oder später. #00:30:02-8#

340 53. I: Das ist als Indikator zusätzlich gedacht, //B: Okay. // dass man sieht, welche Werte das  
341 beeinflusst, ja. #00:30:07-5#

342 54. B: Okay. #00:30:08-9#

343 55. I: Mit Hilfe von Regelungsereignissen kann ferner festgelegt werden, welches Reglermodell  
344 verwendet wird. Also PID, PI, PD. Diese Regler werden in ihrer mathematischen Form dargestellt.  
345 Das sind Standardreglermodelle, könnte man sagen, die man aus der Regelungstechnik kennt.  
346 //B: Okay. // Die Tasks für sie sind im Grunde Berechnungen, die in festen Teilprozessen  
347 dargestellt werden können oder sollen. Nach diesen Berechnungen kann der Benutzer Tasks zu  
348 weiteren Datenverarbeitung hinzufügen. Dies kann auch nach Mess-Tasks geschehen, die man  
349 auch als Datenerfassungs-Tasks bezeichnen kann. Das heißt, man kann sich eventuell noch  
350 immer dafür entscheiden, dass man natürlich noch weitere Schritte modelliert. Das wissen //B:  
351 Okay. // Sie eh, wenn Sie die ■■■ kennen, dass man hier recht flexibel ist. Und wenn wir jetzt an  
352 unser Beispiel von gerade eben denken, könnten wir es ein bisschen erweitern. Das heißt, wir  
353 sagen, wenn wir das Measure Event haben, das regelmäßig getriggert wird, und hier wird immer  
354 wieder unser V 1 abgefragt, dann kommt in einer parallelen Kante ein Control Event. Wir haben  
355 darauf folgend, wenn es ausgelöst wird, hier jetzt mal ein Script eingefügt, in dem eine  
356 mathematische Operation, also man könnte sagen hier, eine Standardberechnung stattfindet,  
357 einmal die Differenz berechnen zum Beispiel von  $V_{opt}$  minus V 1. Und dann könnten wir mit  
358 dieser Differenz oder mit diesem Ergebnis weitergehen ins nächste Script, das vielleicht sogar  
359 vordefiniert ist. Also man könnte sagen, ein Standard-Script mit dem jeweiligen mathematischen  
360 Reglermodell. Also in dem Fall als Beispiel angegeben mit dem PID-Regler. Und der Wert, der  
361 sich dann daraus ergibt, also der Wert, den wir dann an die Aktorik weiterleiten möchten, den  
362 könnte man wieder über einen darauffolgenden Service Call an den jeweiligen Aktor weiter  
363 schicken. Und das ist hier dargestellt mit dem Label Send new manipulated value MV.  
364 #00:32:20-2#

365 56. B: Sprich, der PID-Code wandelt die Berechnung um in etwas, was nur vom Aktor verstanden  
366 wird? Oder der kümmert sich dann darum/ #00:32:30-1#

367 57. I: Das wäre der entsprechende Reglerausgang. Also, das wäre //B: Okay. // quasi/ die  
368 Reglermodelle sind dafür da, zu wissen, wie sich das System verhält oder wie es sich verhalten  
369 sollte, um auf einen gewissen Zielwert zu kommen. Das heißt, das ist quasi die mathematische  
370 Repräsentation des optimalen Verhaltens des Modells oder des, ja, des jeweiligen Prozesses,  
371 könnte man sagen. Also so werden sie zumindest in der Regelungstechnik eingesetzt. Und diese  
372 PID, PD, PI-Reglermodelle sind Standardreglermodelle, die jeweils, je nachdem um was für ein  
373 System es sich halt handelt, unterschiedliches zeitliches Verhalten haben oder zum Beispiel  
374 unterschiedliche Antworten haben, dass wenn man eine Temperatur regeln möchte, dass die  
375 entsprechende Aktorik schneller darauf reagiert, also schnell auf den Zielwert zwar kommt, aber  
376 dadurch natürlich, wenn es extrem hoch heizt, eventuell darüber schießen könnte. Also ich heize  
377 dann vielleicht am Anfang um zehn Grad zu viel und muss dann erst wieder kühlen, oder dass es  
378 eventuell etwas langsamer geht, aber dafür, dass ich nie diese zehn Grad überschieße. Wenn ich  
379 fünfzig Grad zum Beispiel haben möchte, nie auf die sechzig komme, sondern das ganz langsam  
380 halt hoch heize. Also dieses Verhalten wird durch die Reglermodelle abgebildet. #00:34:01-6#

381 58. B: Hm. (zustimmend) Verstehe. #00:34:03-5#

382 59. I: Ja, und wie Sie es ja kennen, kann man Datenelemente in der ■■■ definieren. Also hier könnte  
383 man sagen, man hat für den optimalen Wert eine Zielvorgabe, also den Zielwert. Man kann aber  
384 auch sagen, okay, ich habe den Manipulated value, der dann überschrieben wird. Der startet  
385 vielleicht, keine Ahnung, mit einem bestimmten Wert, wahrscheinlich eher null, und wird dann



386 einfach im Laufe des Prozesses überschrieben. Aber die wichtigen Dinge, die man für Control,  
387 also für das Control Event definieren kann, sieht man in diesem Fenster hier. Wir haben nämlich  
388 wieder Interval frequency in Hertz, kann man wieder definieren. Wir haben wieder Values  
389 expected to change. In dem Fall, welcher Wert wird hier, also was ist quasi das Ergebnis oder  
390 was wird hier überschrieben. Man kann definieren, okay, welches Reglermodell habe ich hier? In  
391 unserem Fall haben wir jetzt PID angegeben. Und man kann auch eine Range für den Wert  
392 angeben, so quasi als Sicherheitsmaßnahme, es dürfte eigentlich nie außerhalb dieser Range sich  
393 bewegen. Oder wenn man zum Beispiel nur das Lower limit angibt, das darf halt nie unter  
394 diesem Wert sein, oder nie über dem Upper limit. Also so könnte man auch //B: Okay. // das  
395 Ganze einsetzen. #00:35:26-5#

396 60. B: Verstehe. #00:35:27-6#

397 61. I: Wait bedeutet wieder, dass für den nächsten Zyklus auf das Beenden aller Tasks gewartet wird,  
398 auch auf die Regulierungs-Tasks. Und sequential heißt, dass die Tasks nacheinander ausgeführt  
399 werden. Das heißt, es wird erst gemessen beziehungsweise der Zustand abgefragt und mit  
400 diesem gemessenen Wert wird dann die Regelung eigentlich durchgeführt. Und, ja. Das habe //B:  
401 Okay. // ich schon alles erklärt. #00:35:56-1#

402 62. B: Das haben wir schon. Ja. #00:35:57-1#

403 63. I: Das haben wir schon. Zustandsabfragen und Regulierungen sollten in regelmäßiger Frequenz  
404 ausgelöst werden. Das ist klar, ja. Abbruchereignisse hingegen werden nur durch ihre  
405 Abbruchbedingungen ausgelöst, die der Benutzer auch definieren kann. Also ein Beispiel für ein  
406 Abbruchereignis wäre, wenn man jetzt ein ganz banales Beispiel her nimmt, dass man von  
407 draußen den Befehl bekommt, den Prozess einfach abzuschalten, also dass man eine Anlage  
408 herunterfährt. Als Beispiel haben wir jetzt hier Emergency Stop zwar eingegeben, aber natürlich  
409 ist Emergency Stop ein sehr, also, es wurde in einem anderen Interview auch schon angemerkt,  
410 ein sehr extremes Beispiel, weil das ja wirklich für den Stillstand der Anlage dann steht, also  
411 wirklich für den Notstopp. Aber natürlich könnte man jetzt sagen, wenn das nicht bei der  
412 aktuellen Anlage, sondern vielleicht bei einer anderen Anlage irgendeine Routine auslösen soll,  
413 könnte man hier noch dann definieren, okay, ich möchte hier vielleicht doch noch einen Service  
414 Call absetzen, oder einfach an eine andere Anlage dafür eine Notification schicken oder wie auch  
415 immer. #00:37:12-0#

416 64. B: Okay. #00:37:13-3#

417 65. I: Das heißt, die Bedingung für den Abbruch würde hier direkt neben das Cancel Event  
418 eingetragen werden. Also das wissen Sie ja grundsätzlich eh, wie man das dann definieren kann.  
419 #00:37:25-0#

420 66. B: Genau. #00:37:26-2#

421 67. I: Ja, und unser Default-Wert wäre natürlich bei diesem Zustand, also wenn der auf true gesetzt  
422 wird, würde es halt abbrechen //B: Ja. // und unser Default, damit das natürlich überhaupt  
423 laufen kann, wäre false. Weil sobald das natürlich auf true wäre, hätten wir den Abbruch schon  
424 getriggert aus dem System. #00:37:42-0#

425 68. B: Schon klar. #00:37:43-1#

426 69. I: Okay, das habe ich erklärt. Nachdem das Ereignis ausgelöst wurde, können Tasks zur  
427 Aufräumroutine abgearbeitet werden, bevor der Zyklus oder der gesamte Prozess vollständig  
428 beendet wird. Damit würde auch das fünfte Feature für Aufräumprozesse erfolgen, also erfüllt  
429 werden. #00:38:05-5#

430 70. B: (Unv.) gehen möchte. #00:38:07-3#

---

431 71. I: Bitte? #00:38:08-4#

432 72. B: Das war dann der konsistente Zustand, in dem wir dann noch übergehen möchten //I: Genau.  
433 Ja. // mit dem/ Okay, passt. Kenne mich aus. Ja. // I: Das heißt/ // Super. #00:38:15-6#

434 73. I: Nachdem das Cancel Event getriggert wurde, dass man noch flexibel hier Schritte hinein  
435 modellieren kann, die erfolgen sollen, um das System in einen gewissen Zustand überzuführen.  
436 #00:38:27-8#

437 74. B: Okay. Alles klar. #00:38:31-2#

438 75. I: Und man könnte jetzt sagen, also das kommt jetzt eh auch gleich als Beispiel, hier dann nach  
439 der Cancel-Bedingung, also wieder das Beispiel, dass wir eh gerade gesehen haben, dass man  
440 hier dann danach einfach zum Beispiel einen Subprozess, einen Service Call, was auch immer,  
441 vielleicht sogar mehrere Schritte noch hinein modelliert. Also hier stellvertretend als Initiate  
442 shutdown routine in vessel one dargestellt. Dass man hier sagt, okay, spezifisch für dieses Cancel  
443 Event gilt jetzt genau diese Aufräumroutine. Aber man kann natürlich auch sagen, dass ich hier  
444 nach dem zweiten Symbol für das Closed Loop Subsystem, also wenn ich hier komplett austrete,  
445 dass ich dann auch sage, okay, ich habe generell, wenn dieses System, wenn es aus der  
446 Kontinuität ausbricht und vielleicht in einen diskreten Prozess oder in einen anderen  
447 kontinuierlichen Prozess übergeht, dass ich hier noch Zwischenschritte hinein modelliere, bevor  
448 ich zu meinem End Event komme. Das würde dann natürlich für den gesamten Prozess gelten  
449 und nicht mehr für die einzelnen Cancel Conditions. #00:39:37-6#

450 76. B: Schon klar. Ja. #00:39:38-7#

451 77. I: Okay. Die vorgestellten Erweiterungen/ #00:39:42-3#

452 78. B: (Unv.) Eine kurze Frage noch. Ohne dieses Cancel Event aufzurufen, kommt man da aber nicht  
453 wirklich heraus, außer ich rufe diesen (unv., End Script?) explizit auf, oder? Also der  
454 kontinuierliche Regelkreis wird solange ausgeführt, bis ich die, in diesem Beispiel, die Emergency  
455 Stop active true setze? #00:40:01-6#

456 79. I: Genau, ja. #00:40:04-2#

457 80. B: Passt, okay, //I: Also das soll/ //alles klar. #00:40:06-9#

458 81. I: Das soll auch ein bisschen den kontinuierlichen Charakter davon verdeutlichen, weil ein  
459 kontinuierlicher Prozess ja eigentlich nicht wirklich einen Anfang und ein Ende hat, sondern  
460 eigentlich die ganze Zeit ablaufen sollte. Also das Hochfahren und das Abfahren des Prozesses  
461 sind eigentlich wieder, könnte man sagen, diskrete Prozesse. Aber wenn man sich dann in der  
462 Kontinuität befindet, ja, sollte man eigentlich, wenn es jetzt nicht wirklich etwas/ wenn es sich  
463 nicht um einen Befehl handelt, der diesen Prozess unterbrechen würde oder eigentlich die  
464 Kontinuität durchbrechen würde, sollte man eigentlich in dieser Loop drinnen bleiben. Und dafür  
465 sind dann aber natürlich/ natürlich, wenn man jetzt so einen Prozess sich in der Realität  
466 anschaut, du wirst so oder so, man wird so oder so, einen kontinuierlichen Prozess irgendwann  
467 mal wahrscheinlich unterbrechen. #00:40:57-8#

468 82. B: Ja, klar. #00:40:58-7#

469 83. I: Vielleicht durch Wartungsarbeiten oder so etwas. Unter solchen Bedingungen oder um so  
470 etwas auch abbilden zu können, haben wir die Cancel Conditions, die Cancel Events hier  
471 eingeführt. #00:41:10-4#

472 84. B: Okay. Jetzt, wo ich das so verstanden habe, möchte ich meinen Punkt Fünf revidieren, weil  
473 dann ist das doch wichtig mit den konsistenten Zustand. Ich habe hier glaube ich gesagt, dass es

474 nur teilweise wichtig ist, aber jetzt nach der Erklärung eigentlich ist es eh immer wichtig, dass  
475 man da noch in einen konsistenten Zustand übergeht. #00:41:27-2#

476 85. I: Ja, es ist natürlich auch von der Reihenfolge her so gestellt, dass wir erst einmal eine  
477 unbeeinflusste Meinung von den Experten bekommen. //B: Ja. // Und ja, also wenn man so  
478 durchgeht und sich das überlegt, okay, natürlich. Wenn ich jetzt so einen Prozess in der Realität  
479 mir anschau, dann wird der sehr wahrscheinlich doch irgendwann mal unterbrochen werden  
480 und deswegen muss man natürlich etwas einführen, um das auch zu ermöglichen. #00:41:52-5#

481 86. B: Okay. #00:41:53-6#

482 87. I: Die vorgestellten Erweiterungen sollen bei der Modellierung von kontinuierlichen Prozessen  
483 helfen, indem Vorlagen für die Erstellung von Prozessmodellen vorgegeben werden und  
484 andererseits durch die Darstellung als Closed Loop Subsystem mit eigenen Symbolen für  
485 Zustandsabfrage-, Regulierungs- und Abbruchereignisse helfen, solche Prozesse leichter  
486 nachvollziehen zu können. Dazu kommt, dass man für eine übersichtlichere Darstellung des  
487 gesamten Prozesses auch Subprozesse zur Unterteilung nutzen kann. Und die Subprozesse sollen  
488 dann natürlich auch eher dafür dienen, dass man sagen kann, okay, wenn ich ins Detail hier  
489 hinein blicken möchte, wie verhält sich der Prozess in diesem Abschnitt. Dass ich sagen kann,  
490 okay, ich gehe in den Subprozess hinein, klappe quasi diesen Prozess auf und schaue es mir im  
491 Detail an. Und das soll dann auch zur Übersichtlichkeit, zur Verständlichkeit der Modelle  
492 beitragen. Ja, ich werde Ihnen nun Prozessbeispiele zeigen, die mit denen unserer Arbeit  
493 vorgestellten Erweiterungen modelliert sind. Ich möchte, dass Sie sich die Modelle ansehen und  
494 mir sagen, was Sie aus ihnen herauslesen können und ob die Modelle den notwendigen  
495 Informationsgehalt für die Modellierung der zugrundeliegenden Regelungsprozesse erfüllen.  
496 Vorab wird Ihnen zum jeweiligen Prozess erklärt, was abgebildet werden soll. Und ich würde Sie  
497 bitten, einfach offenes Feedback dazu abzugeben. //B: Okay. // Also natürlich ist mir bewusst,  
498 wenn Sie nicht unbedingt Bezug zu industriellen Prozessen haben oder generell zu  
499 Regelungsprozessen/ Sie müssen jetzt nicht unbedingt die technische Realisierung dazu  
500 vollkommen im Kopf haben, um mir Kommentare abzugeben. Sondern mir geht es darum, wie  
501 Sie es empfinden von der Charakteristik her, auch von diesen sechs Eigenschaften her, die wir  
502 oben kurz durchgegangen sind. Ob einfach der Charakter des Prozesses auch gut dargestellt wird,  
503 um nachvollziehen zu können, wie sich das System verhält. Darum geht es. #00:44:03-5#

504 88. B: Hm. (zustimmend) #00:44:04-4#

505 89. I: Gut. Das erste Modell ist etwas einfacher, das zweite wird dann etwas komplexer. Beim ersten  
506 Modell handelt es sich um eine Temperaturregelung mit einem PI-Controller über einen  
507 Wärmetauscher, basierend auf dem Beispiel aus der MathWorks-Bibliothek, basierend auf  
508 einem Beispiel aus der MathWorks-Bibliothek. Die Temperatur einer Flüssigkeit in einem  
509 Rührkessel wird mittels Wärmetauscher geregelt. Der über den Wärmetauscher eingebrachte  
510 Wärmestrom wird über ein Ventil, das einen Dampfstrom kontrolliert, gesteuert. Das heißt, über  
511 den Wärmetauscher, über den Dampfstrom wird der Flüssigkeit im Rührkessel Energie zugeführt,  
512 also Wärme zugeführt. Der zu beachtende störende Umgebungseinfluss, also das, was dann  
513 dieses System immer wieder abweichen lässt von der optimalen Temperatur, der gewünschten  
514 Temperatur, ist die schwankende Temperatur der zugeführten Flüssigkeit. Das heißt, wir gehen  
515 hier davon aus, dass die Tankwand oder die Kesselwand isoliert ist und dass wir hier keine  
516 Wärmeverluste haben. #00:45:18-5#

517 90. B: Okay. #00:45:19-7#

518 91. I: Wenn man das Ganze jetzt als Flowchart darstellen würde, dann würde das ungefähr so  
519 aussehen wie hier in der ersten Abbildung. Und zwar haben wir einen Rührkessel. Wir haben ein  
520 Rührwerkzeug. Wir haben einen Zufluss und einen Abfluss. Das heißt, über den Zufluss, die  
521 Flüssigkeit, die hier hineinkommt, das ist dann hier der störende Einfluss von außen. Wir haben  
522 einen Temperaturfühler. Wir haben dann den Wärmetauscher, hier sehen Sie dann, dass von

523 links der Dampfstrom dem System zugeführt wird mit dem entsprechenden Ventil, das dann  
524 gesteuert werden kann. Das heißt, dass wäre dann die Flowchart-Darstellung. #00:46:06-4#

525 92. B: Okay. #00:46:07-5#

526 93. I: Und wenn wir jetzt das Ganze in der [REDACTED] mit unseren Erweiterungen modellieren wollen  
527 würden, hätten wir für die Berechnung des PI-Controllers auch einige Datenelemente definiert,  
528 die man eventuell brauchen kann für das mathematische Modell, wenn man schon weiß, wie  
529 sich der Regler verhält. #00:46:25-2#

530 94. B: Okay. #00:46:26-6#

531 95. I: Wir haben, das ist jetzt nur ein Beispiel für die Leute, die die [REDACTED] nicht so gut kennen, wie man  
532 Endpunkte definieren kann. Also wie dann die Service Calls, mit welchen Adressen die dann  
533 funktionieren würden, auch nur als Beispiel einfach mal eingefügt. #00:46:43-4#

534 96. B: Kenne mich aus. Ja. #00:46:44-9#

535 97. I: Ja, kennen Sie eh. Und dann hätten wir eigentlich auch schon das Prozessmodell, also den  
536 Graphen dazu. #00:46:50-6#

537 98. B: Okay. Das heißt, ich schaue mir das jetzt mal einfach von Anfang bis zum Ende an und gebe  
538 Kommentare dazu ab, was ich davon verstehe und wie ich das Ganze jetzt sehe momentan?  
539 #00:46:59-6#

540 99. I: Also, ich habe es mit den anderen Interviewgästen so gemacht, dass ich kurz einmal durch  
541 bespreche, was hier circa passiert und dass Sie dann mir erklären, ob Sie das genauso sehen oder  
542 ob Ihnen dann etwas nicht/ ob Ihnen etwas fehlen würde. Da gibt es dann aber eh auch eine  
543 Tabelle mit Kriterien, die wir durchgehen würden für die Bewertung der Graphen, //B: Passt.  
544 Dann machen wir das so. // oder der Modelle. Okay. Also, wir verfolgen hier einen Wait- und  
545 Sequential-Ansatz. Das heißt, wir gehen davon aus, dass jeder Strang beendet werden muss,  
546 bevor wir in den neuen Zyklus gehen. Sequential heißt, dass wir zuerst Measure und dann  
547 Control ausführen. Wobei, das Beispiel aus der MathWorks-Bibliothek ist im Grunde sehr einfach  
548 gehalten und wir haben hier kaum Bedingungen vorgegeben, wie das System erfolgen muss. Das  
549 heißt, man könnte hier eigentlich auch genauso parallel statt sequential modellieren. Also das ist  
550 hier keine Vorgabe, aber, ja, man könnte sequential durch parallel im Grunde ersetzen.  
551 #00:48:03-5#

552 100. B: Okay. #00:48:05-0#

553 101. I: Wir haben hier zwei Temperaturen, die gemessen werden. Einmal die Temperatur natürlich  
554 vom Tank Eins, dann haben wir/ also wir haben den entsprechenden Service Call dazu. Eventuell  
555 haben wir dann noch eine Umrechnung oder einen Subprozess, den wir starten möchten. Das ist  
556 das, was ich vorhin gemeint habe, dass man hier in der Modellierung noch recht flexibel ist. Wir  
557 haben dann noch einen zweiten parallelen Measure-Strang. Also hier wird dann noch die  
558 Temperatur der Störung, also der Störgröße gemessen. Deswegen haben wir hier jetzt auch ein  
559 kleines D verwendet für Disturbance T 2. Und dann kommen wir auch schon zum Control-Strang.  
560 Das heißt, wir haben mal das Control Event. Wir haben das mathematische Modell des  
561 PI-Controllers. Wir haben dann eventuell sogar noch eine Umrechnung, die als Einzelschritt  
562 modelliert wird, als Script hier beispielsweise. Und am Ende, wenn wir wieder den  
563 entsprechenden fertigen Wert haben, den wir an den jeweiligen Aktor schicken können, dann  
564 wird wieder ein Service Call abgesetzt mit dem entsprechenden Wert. Das heißt, in dem Fall  
565 wäre das dann das Stellglied, also der Antrieb des Motors für das Ventil, das den Dampfstrom  
566 kontrolliert. Und wir haben, wie gesagt, leider in der MathWorks-Bibliothek in diesem Beispiel  
567 keine genauen Bedingungen, wie sich das System auch verhalten würde, wenn man es  
568 abbrechen möchte oder wenn man es herunterfahren möchte. Und deswegen haben wir jetzt  
569 stellvertretend Stop activated modelliert, wenn das auf true gesetzt wird, dann wird hier

570 eventuell ein Subprozess ausgeführt. Das heißt, eine Shutdown-Sequenz wird vielleicht initiiert  
571 oder ausgeführt. Und dann würde man aus der Kontinuität des Systems ausbrechen.  
572 #00:49:54-5#

573 102. B: Okay. #00:49:55-4#

574 103. I: Ja. #00:49:56-7#

575 104. B: Eine Frage hätte ich gleich mal kurz zum Modell an sich. Sie haben gerade vorher gesagt, dass  
576 die im parallel zweiten Measure-Pfad/ Generell ist das Modell ja auf sequential. Das bedeutet ja,  
577 dass die einzelnen Pfade nacheinander abgearbeitet werden, wenn ich das richtig verstanden  
578 habe. #00:50:14-5#

579 105. I: Zuerst wird Measure abgearbeitet und dann Control. #00:50:19-1#

580 106. B: Das heißt aber, die Measure-Pfade sind generell parallel zueinander //I: Generell, ja. // und  
581 nur Measure und Controller sind nacheinander. #00:50:26-7#

582 107. I: Es ist immer so, dass die Measure parallel ausgeführt werden, aber das Control erst nach  
583 Measure dann erfolgen würde, wenn wir sequential hätten. #00:50:37-8#

584 108. B: Also eigentlich könnte ich die beiden ersten Pfade, die beiden Measure-Pfade, auch in einem  
585 Measure-Block stecken und dann parallel laufen lassen. Das wäre dann im Prinzip das gleiche  
586 von der Ausführungslogik her #00:50:48-8#

587 109. I: Also man könnte sagen, dass nach dem Measure Event, nach einem Measure Event, man  
588 eventuell dann zwei Kanten hätte, die dann //B: Genau. // parallel ausgeführt werden, ja.  
589 #00:51:00-2#

590 110. B: Okay, verstehe. Also das Sequential bezieht sich nur auf die drei Blöcke, dass die  
591 nacheinander ausgeführt werden, aber nicht auf die Pfade im Modell, wie sie jetzt momentan  
592 modelliert werden? #00:51:09-1#

593 111. I: Das Sequential bezieht sich auf den Unterschied zwischen Measure und Control. #00:51:14-9#

594 112. B: Okay, passt. #00:51:16-2#

595 113. I: Das heißt, dass in dem Fall immer Control, der Strang/ also alle Control-Stränge erst ausgeführt  
596 werden würden, wenn sämtliche Measure-Stränge oder die zugehörigen Measure-Stränge  
597 ausgeführt werden. Darum geht es. #00:51:29-5#

598 114. B: Gut. #00:51:32-3#

599 115. I: Okay, hier habe ich noch als Beispiel, wenn man halt in der [REDACTED] nicht weiß, wie ein Script  
600 ausschaue würde, als Beispiel angeführt, wie die mathematische Berechnung halt in Code  
601 dargestellt aussehen könnte. Also verschiedenste Möglichkeiten, auch nur ganz einfach, nur das  
602 Überschreiben von dem Wert einer Variablen auf eine andere. Also das ist wirklich  
603 stellvertretend dafür zu sehen, wie man eine Berechnung hier durchführen kann oder wie ein  
604 Script ausschaue könnte. #00:52:05-7#

605 116. B: Okay. #00:52:07-0#

606 117. I: Und die eigentliche Bewertung, also die Kriterien, nach denen ich Sie bitten würde, das Modell  
607 zu bewerten, sind hier in dieser Tabelle aufgeführt. Und zwar, bei den Kriterien geht es  
608 einerseits mal um die Verständlichkeit. Also würden Sie sagen, Sie könnten aus dem Graphen  
609 heraus lesen, aus dem Prozessmodell heraus lesen, was passiert? Übersichtlichkeit. Können Sie  
610 das Gesamtsystem auf einen Blick erfassen? Einfachheit. Könnte man das Modell noch einfacher

611 darstellen Ihrer Meinung nach? Also würde Ihnen da vielleicht etwas einfallen? Logik. Wird klar,  
612 was parallel und was sequentiell passiert? Also wir haben es ja jetzt nochmal erklärt, vielleicht, ja.  
613 Überlasse ich natürlich Ihnen, wie Sie es bewerten. Und dann schließlich Erweiterbarkeit. Könnte  
614 man dem Modell Ihrer Meinung nach vielleicht noch etwas hinzufügen, was den  
615 Informationsgehalt verbessern würde? #00:53:05-6#

616 118. B: Okay, //I: Und das Ganze/ // also fangen wir mal ganz oben an. Hm? (fragend) #00:53:09-1#

617 119. I: Entschuldigung. Und das Ganze halt diesmal mit einer Skala versehen. Und zwar Eins bis Fünf,  
618 wobei das eher ein Punktesystem ist. Das heißt, Eins wäre sehr schlecht. Fünf wäre sehr gut. Also  
619 je mehr Punkte wir erreichen, desto besser ist die Bewertung. #00:53:26-4#

620 120. B: Passt. Okay, super. Dann gehen wir es mal der Reihe nach durch. Punkt Eins, Verständlichkeit.  
621 Was passiert? Ja, finde ich eigentlich sehr, ja, klar was passiert. Man kann sich halt noch über die  
622 Werte, die auf der rechten Seite definiert sind, schön heraus lesen, was sich in den einzelnen  
623 Schritten tut. Und man kann das ganze Modell eigentlich von oben nach unten sehr schön lesen.  
624 Also ich würde hier mal fünf Punkte vergeben. Punkt Zwei, Übersichtlichkeit. Und hier das  
625 Gesamtsystem kann ich hier sehr schön auf einen Blick erfassen, was natürlich auch damit  
626 zusammenhängt, dass ich hier ein sehr kurzes Prozessmodell habe. Aber für dieses Beispiel  
627 würde ich auf jeden Fall auch da fünf Punkte vergeben. Einfachheit. Könnte man das Modell  
628 noch einfacher darstellen? Ich wüsste jetzt ehrlich gesagt nicht, wie. Ich meine, man könnte  
629 natürlich, was noch ein bisschen in die Logik hinein führt, die beiden Measure-Blöcke zu einem  
630 zusammenfassen. Das ist dann halt die Frage eben, was brauchen die Leute. Wollen die vielleicht  
631 sehen, dass der eine Measure-Block sich auf das eine Ding bezieht, der andere Measure-Block  
632 sich auf das andere, auf die Flüssigkeit bezieht vielleicht? Also von daher kann man durchaus  
633 argumentieren, dass diese Aufteilung so Sinn macht. Also würde ich auch sagen fünf Punkte  
634 dafür. Logik, was parallel und sequentiell passiert, darüber haben wir schon gesprochen. Ich sage  
635 mal so. Wenn man weiß, was die Labels beim oberen Gate bedeuten, dann ist auch durchaus  
636 klar, was hier parallel oder sequentiell ist. Weil es ist eine Notation, die man mal lernen muss,  
637 wo man halt wissen muss, was das bedeutet. Aber wenn man sich damit auskennt, dann ist  
638 durchaus klar, was da passiert. Also würde ich sagen, auch fünf Punkte dafür. Und  
639 Erweiterbarkeit. Könnte man dem Modell noch etwas hinzufügen, was den Informationsgehalt  
640 verbessern würde? Können wir bitte nochmal kurz hinaufgehen auf das Modell? (...) Aber ja, ich  
641 meine im Prinzip ist das hier jetzt alles sehr technisch. Aber die Leute, die das Modell sich  
642 anschauen werden, werden schon wissen, was die einzelnen Schritte bedeuten. Jetzt Get D T 2  
643 (Get d\_T2) zum Beispiel ist jetzt etwas, da muss man wahrscheinlich nochmal schauen. Vielleicht  
644 kann man da noch ein Kommentar dazu schreiben, was das bedeutet. Aber ich nehme an, dass  
645 sich das, diese Frage, jetzt auf das gesamte Modell bezieht und nicht nur auf diese eine Grafik  
646 hier. Von daher würde ich auch sagen, dass das hier fünf Punkte sind. Also man braucht natürlich  
647 die gesamte Information dazu, aber passt für mich. Finde ich sehr schön so. #00:56:07-4#

648 121. I: Okay, wunderbar. Dankeschön. Ich würde jetzt kurz vorschlagen, bevor wir zum zweiten  
649 Modell kommen, das ist doch ein bisschen komplexer, vielleicht fünf Minuten Pause. Weil wir  
650 sind jetzt schon/ ich schaue ganz kurz auf die Uhr, 54 Minuten haben wir schon. Ja. #00:56:23-7#

651 122. B: Ja, können wir gerne machen. #00:56:26-5#

652 123. I: Okay, dann höre ich kurz auf und stoppe kurz die Aufnahme. #00:56:29-9#

653 UNTERBRECHUNG - 5 Minuten Pause

654 124. I: Okay, Aufnahme läuft wieder. #00:00:03-8#

655 125. B: Okay. #00:00:05-4#

---

126. I: Gut. Kommen wir zum zweiten Modell. Das zweite Modell ist ebenfalls ein Heizprozess. Und zwar basiert dieses Beispiel auf Schulungsunterlagen der Firma Siemens. Es handelt sich hierbei ebenfalls um eine Temperaturregelung für einen Rührreaktor. Die Regelung wird in diesem Beispiel aber mit einem PID-Regler, einer Handsteuerung sowie einem Pulsgenerator realisiert. Die Heizung erfolgt nicht über einen Wärmetauscher, sondern über ein Heizelement. Weiters sind auch in den Unterlagen Verriegelungsbedingungen definiert. Und als Basis für die Prozessmodellierung wurden die Beschreibungen aus den Schulungsunterlagen für die Prozessmodellierung mit Simatic PCS 7 herangezogen. Unser Prozessmodell wird mit einer automatischen Steuerung modelliert, die mit Umschalten auf Handsteuerung aus dem Closed Loop System ausbricht. Wir gehen davon aus, dass das System bereits angelaufen ist und automatisch gesteuert wird. Das heißt, wir befinden uns im Automatikmodus. Und weiters wird der Prozess für nur einen Reaktor und nicht wie in den Unterlagen beschrieben für zwei Reaktoren dargestellt und modelliert. Okay, wir haben in diesem Fall nicht nur Datenelemente für die Berechnung des PID-Controllers, sondern wir haben hier auch Werte für die verschiedenen Verriegelungsbedingungen, die wir in den Unterlagen definiert haben. Das heißt, wir haben hier jetzt natürlich mehr Informationen als im ersten Modell. Wir haben zum Beispiel eine maximale Temperatur, die nicht überschritten werden darf mit sechzig Grad. Wir haben einen Mindestfüllstand im Reaktor, von der Flüssigkeit, mit 200 Milliliter, der nicht unterschritten werden darf. Weiters haben wir als Default-Wert, also als unsere Standardbedingung für diesen Prozess Operation Mode, Betriebsmodus, in automatic und der Hauptschalter wurde auch erwähnt. Das heißt, wenn der Main Switch natürlich nicht auf On, sondern auf Off geschaltet wird, würde natürlich der Prozess auch ausbrechen, entsprechend. Ja, wie Sie hier jetzt sehen, ist das Modell um einiges umfangreicher. Ich gehe es einmal schrittweise wieder durch, dass man die Details jetzt noch etwas besser erkennen kann und würde dann etwas heraus zoomen, dass man den Graphen oder das Modell selber etwas besser vielleicht sieht. Wir verfolgen jetzt, weil wir wissen es handelt sich hierbei um ein Continuous Function Chart, haben wir zeitliche Bedingungen. Das heißt, wir wissen, okay, wir arbeiten mit cancel. Und wir wissen in dem Fall auch wir arbeiten erst mit einem Prozessabbild. Das heißt, wir müssen auf sämtliche Inputs warten und mit denen dann erst in die verschiedenen Prozesse hinein gehen. Und deswegen ergibt sich hier auch ein Sequential-Ansatz. Das heißt zuerst werden alle Messungen durchgeführt und dann erst das Control Event ausgelöst. Wir haben die Messung wieder des Reaktors, also der Temperatur im Reaktor. Wir haben die Messung aber jetzt auch vom Füllstand im Reaktor, weil das natürlich eine der Verriegelungsbedingungen ist. Wir haben die Überprüfung, auf welchem Zustand sich unser Operation Mode befindet. Wir haben auch die Überprüfung, ob wir zum Beispiel einen Emergency Stop auf Active haben. Das kann man natürlich auch als Push Notification, das war der Input von einem anderen Interview, als Push Notification könnte man das eventuell auch darstellen, dass man es nicht als Zustandsabfrage modelliert. Und wir haben dann auch den Main Switch natürlich, den ich vorhin erwähnt habe. Das heißt eine Reihe von verschiedenen Zustandsabfragen. Wir haben dann Control wieder mit dem mathematischen Modell des PID-Controllers in dem Fall. Wir haben dann wieder eine Umrechnung, also in dem Fall die Pulsweitenmodulation als nächsten Schritt, bevor wir dann auf den Wert kommen, den wir an die Outputs weitergeben können oder den wir als Output halt an den jeweiligen Aktor an das System weitergeben können, wieder hier dargestellt als Service Call. Und die Cancel-Bedingungen sind dann entsprechend wie vorhin schon aufgelistet. Also sobald der Hauptschalter auf Off gehen würde, könnten wir hier einen Subprozesses anhängen oder irgendetwas noch daran modellieren, was natürlich das System in einen konsistenten Zustand überführt, bei einem normalen Abschalten oder Herunterfahren des Systems hier mal dargestellt mit Send message to operator. Dass dann entsprechend der Maschinenbediener oder Anlagenbediener zum Beispiel den Reaktor entleeren müsste oder vielleicht öffnen müsste oder dergleichen. Dann Emergency Stop wie vorhin auch schon erklärt. Also einmal das Bedingung hier, dass das auf true gesetzt wird, haben wir hier jetzt stellvertretend auch Send message to operator eventuell, dass hier eine Benachrichtigung einfach noch geschickt wird. Und was dann interessant ist, speziell für diesen Prozess, jetzt haben wir diese Werte, die wir oben auch definiert hatten. Wenn die aktuelle Temperatur des Reaktors über der Maximaltemperatur, also über den sechzig Grad liegt, müssen wir eine gewisse Routine natürlich starten. Wenn der Füllstand/ ich versuche das immer zu markieren, dass man sieht, wo ich bin. Wenn der Füllstand, der aktuelle, im Reaktor unter das

713 Mindestfüllniveau kommen würde, müssten wir natürlich auch eine gewisse spezielle  
714 Maßnahme fahren. Und dann haben wir noch die letzte Bedingung. Operation Mode, wenn der  
715 auf manuell, also manual, geschaltet werden würde, würden wir vielleicht einen Subprozess  
716 aufrufen, speziell für diese Cancel Condition, dieses Cancel Event. Und dann würde man aus der  
717 Kontinuität, aus diesem Closed Loop Subsystem ausbrechen. #00:06:42-8#

718 127. B: Okay. #00:06:44-5#

719 128. I: Und, ich würde Sie/ Also man sieht hier wieder das mathematische Modell des PID-Controllers.  
720 Das ist ein bisschen komplexer als der normale PI-Controller, weil es auch einen  
721 Differentialanteil hat. Aber ja, also nur als Beispiel, wie man es mathematisch darstellen kann.  
722 Und für die Bewertung des Modells würde ich Sie wieder bitten, genauso wie beim ersten  
723 Modell, beim weniger komplexen Modell, eine Bewertung zwischen Eins und Fünf abzugeben.  
724 Also inklusiv Eins, inklusiv Fünf, mit sehr schlecht bis sehr gut abzugeben. Wieder bezüglich  
725 Verständlichkeit, Übersichtlichkeit, Einfachheit, Logik und Erweiterbarkeit. #00:07:29-4#

726 129. B: Okay. Moment, ich nehme mir kurz diese Fragen heraus. Und können wir bitte wieder kurz auf  
727 das Modell gehen? #00:07:37-9#

728 130. I: Ja. #00:07:38-4#

729 131. B: Dann kann ich das mal Schritt für Schritt durchgehen. Also. Generell der erste Punkt, den ich  
730 mir bei dem/ ich gehe mal kurz nur allgemeine Anmerkungen vielleicht durch und, weil das  
731 bezieht sich eigentlich auf einige Punkte, die in den Bewertungskriterien drinnen stehen. Was ich  
732 mir recht oft gedacht habe, ist, hier mir schon klar jetzt das oft nicht der Fall sein wird und in  
733 vielen Bereichen sicher anders abläuft, wenn es viele Bedingungen gibt, die auf mehrere  
734 Abbruchbedingungen Einfluss haben. Aber in diesem Fall ist es ja so, dass ich ein Event habe,  
735 dass ich messe und dann hängt damit genau ein Abbruch-Event zusammen, mit dem ich dann  
736 das Ganze abreche und nachher noch aufräumen muss. Also zum Beispiel ich messe den  
737 Füllstand und wenn der einen gewissen Wert unterschreitet, dann muss ich abrechnen und das  
738 Ganze aufräumen. Hier ist es ja jetzt so, dass diese beiden Events ja sehr weit auseinander  
739 stehen und nicht besonders viel miteinander zu tun haben. Da wäre mal so eine erste Idee, ich  
740 meine mir ist schon klar, es gibt sicher viele Prozesse und viele Beispiele, wo man irgendwie drei  
741 verschiedene Pfade (unv., adjektieren?) muss und komplexe Berechnungen durchführt und am  
742 Ende kommt dann irgendeine Zahl heraus, die ich vergleiche. Und erst dann weiß ich, ob jetzt  
743 wirklich das Abbruch-Event getriggert werden muss oder nicht. Aber hier ist halt zu überlegen,  
744 ob man das irgendwie vielleicht besser heraushebt, was denn jetzt da zusammen gehört. Also ich  
745 habe zum Beispiel jetzt oben irgendwo Get main switch. Ich finde gerade die Zeile nicht. Also, ob  
746 ich jetzt den Main Switch/ Genau, da. Zeile eins hinunter noch. Oder da, genau. Und dann viel  
747 weil darunter nochmal das Ergebnis, was dann überhaupt passiert. Also. Ich kann mir vorstellen,  
748 dass es irgendwie vielleicht einfacher möglich sein muss, da die relevanten Blöcke  
749 herauszunehmen, die da irgendwie zusammenhängen. Also worauf hat die Messung jetzt  
750 Einfluss? Was mache ich mit der Messung nachher? Solche Dinge wären da glaube ich für die  
751 Übersichtlichkeit sehr wichtig. Weil so wie es momentan ist, ich meine, wir haben halt alles da,  
752 ist technisch sicher absolut korrekt modelliert. Als Benutzer habe ich schon oft versucht, die  
753 Zeile wiederzufinden. Okay, warte mal. Was ist denn jetzt das Ding, was hier jetzt zu Measure, ist  
754 gleich Main Switch, dazu gehört? Welches Ding ist das gerade? Wenn Sie es jetzt so wie gerade  
755 zu sehen ist, markiert haben, ist es natürlich klar, verständlich, man sieht es. Aber genau so eine  
756 Markierung oder High Lighting wäre glaube ich ganz gut auch für Benutzer, dass die eben  
757 nachschauen können, was passiert denn mit den Daten weiter. Also wo gehen die hin? Wofür  
758 brauche ich sie nachher noch? #00:10:24-2#

759 132. I: Okay, ich verstehe. //B: Also. // Ja. #00:10:26-8#

760 133. B: Da würden vielleicht auch nur einfach irgendwie ein Color Coding oder irgendwie ein, keine  
761 Ahnung, allein schon Trennlinien, horizontale, zwischen einzelnen Bereichen würde  
762 wahrscheinlich helfen. Dass ich mir die Zeile nicht so oft wieder suchen muss, wo denn jetzt was



763 ist. Verständlichkeit. Was passiert? Also das würde ich noch am absolut besten bewerten. Weil  
764 man geht das Ganze von oben bis nach unten durch und kann sich dann anschauen, was die  
765 einzelnen Elemente sind und hat damit dann eine gute Übersicht darüber. Also da würde ich fünf  
766 Punkte noch geben. Übersichtlichkeit sind für mich drei Punkte momentan. Weil, wie gesagt, da  
767 gibt es einfach zu viele Dinge, wo nicht ganz klar ist, wie die Dinge zusammenhängen für mich.  
768 #00:11:12-8#

769 134. I: Darf ich ganz kurz nachfragen? Man hat es jetzt nämlich akustisch ein bisschen schwieriger  
770 oder schlechter verstanden. Also zum ersten Punkt, Verständlichkeit, war es nochmal was?  
771 #00:11:23-3#

772 135. B: Da war es fünf Punkte für mich. #00:11:25-5#

773 136. I: Okay, sehr gut. Dankeschön. #00:11:26-9#

774 137. B: Übersichtigkeit, zwei bis drei. Eher drei, aber zwei bis drei, wenn es eine Möglichkeit ist.  
775 #00:11:34-9#

776 138. I: Ja. #00:11:35-7#

777 139. B: Einfachheit. Könnte man das Modell noch einfacher darstellen? Also wie gesagt. Technisch  
778 gesehen ist es so sicher korrekt. Wenn man was unter Einfachheit sagen kann, dann vielleicht,  
779 dass man eben diese Zusammenhänge irgendwie besser noch visualisiert. // I: Ja. // Was sind  
780 meine Inputs und was passiert nachher damit? Weil das hängt momentan an unterschiedlichsten  
781 Stellen im Modell und ich muss mir das erst einmal zusammen suchen, was ich überhaupt mache  
782 mit den Dingen. Also auch hier würde ich eher noch eine Drei vergeben momentan. Logik. Wird  
783 klar, was parallel und sequentiell passiert? Gut, ja. Das ist in dem Modell durchaus klar. Ich  
784 meine, das ist einfach dadurch, dass die Control-Blöcke und die Measure-Blöcke alle  
785 zusammenhängend sind, ist das sehr klar. Da kann man durchaus fünf Punkte dafür geben, weil  
786 dieser Aspekt ist klar. Weil eben leider auf Kosten der anderen Aspekten, mit, was von den  
787 Daten her zusammenhängt. Erweiterbarkeit. Könnte man dem Modell noch etwas hinzufügen,  
788 was den Informationsgehalt verbessern würde? Ja, das sind im Prinzip eh genau die Punkte, die  
789 ich schon erwähnt habe mit, irgendwie die Datenzusammenhänge ein bisschen besser darstellen.  
790 Vielleicht horizontale Linien einführen, dass man nicht so leicht die Zeile verliert, wenn man jetzt  
791 schaut, welches Element entspricht welchem Element. Vielleicht Dinge über Mouse over, dann  
792 halt die einzelnen, ja, Blöcke definieren oder Zusammenhänge darstellen. Vielleicht auch  
793 manche Blöcke einfach ausblenden. Dann müssten Sie jetzt überlegen, wie man das am besten  
794 angeht. #00:13:16-9#

795 140. I: Verstehe. #00:13:17-6#

796 141. B: Also, das Modell finde ich jetzt schon sehr grenzwertig, was die Übersichtlichkeit angeht,  
797 ehrlich gesagt. Aber technisch absolut korrekt. Also jetzt, von dem, wie man es hier darstellt.  
798 Und wenn man es von oben nach unten durchgeht, bekommt man auch einen guten Überblick  
799 darüber. Aber ich glaube für Detailabfragen ist es ein bisschen schwierig damit klar zu kommen.  
800 #00:13:41-0#

801 142. I: Verstehe ich. Okay. Danke. Okay. Und nachdem Sie jetzt diese Beispiele gesehen haben und  
802 halt auch eine kurze Einführung zu den Erweiterungen von mir bekommen haben, würden Sie  
803 sagen, wären Sie aufgrund dieser Erweiterungen bereit, diese Modellierungsmethode in Ihrem  
804 Arbeitsalltag einzuführen, wenn Sie kontinuierliche Prozesse modellieren müssten oder sollten?  
805 #00:14:13-9#

806 143. B: Also, ich muss sagen, ich habe in meinem Arbeitsalltag relativ wenig mit kontinuierlichen  
807 Prozessen zu tun. Aber rein aus BPMN und technischer Sicht, finde ich das ein sehr gelungenes  
808 Modell und einen sehr gelungenen Ansatz. Weil es im Prinzip die einzelnen Bausteine, die man  
809 jetzt dafür braucht, um kontinuierliche Prozesse mit den Messungen, mit den Regulatoren, die

810 man nachher ausführt, mit den ganzen Aufräumarbeiten, die erst getriggert werden sollen,  
811 wenn eine Abbruchbedingung definiert ist, gut beschreibt. Also so gesehen, finde ich es aus  
812 BPMN-Sicht sehr gut modelliert. Ja, wie gesagt, bei der Visualisierung könnte man noch ein paar  
813 Optimierungen durchführen, aber alles in allem könnte ich die Frage mit Ja beantworten.  
814 #00:15:03-3#

815 144. I: Okay, sehr schön. Danke. Wenn man es jetzt von der anderen Seite her betrachten würde, also  
816 wir haben jetzt die Prozesse/ Wir sind die Prozesse an sich durchgegangen, haben sie vorab  
817 erklärt und haben uns dann die Modelle dazu angeschaut. Von der anderen Blickrichtung her,  
818 würden Sie sagen, dass die Modelle die erklärten Kontrollsysteme gut beschreiben? Also wie gut,  
819 wieder auf einer Skala von Eins bis Fünf würden Sie sagen, beschreiben die Erweiterungen Ihrer  
820 Meinung nach ein Kontrollsystem für diese Beispiele? #00:15:46-0#

821 145. B: Wenn ich den Text nicht gekannt hätte und nur das Modell mit den Daten gesehen hätte,  
822 würde ich verstehen, worum es geht. Ist das die Frage? #00:15:55-4#

823 146. I: Ja, also wenn Sie jetzt den Hintergrund dazu wissen, okay, ich habe erst eine Zustandsabfrage  
824 und dann eine entsprechende Reaktion darauf, vom System. Also das System selber weiß durch  
825 das mathematische Modell des jeweiligen Reglers, wie es zu reagieren hat und führt dann eine  
826 entsprechende Reaktion aus. Und man kann aber aus diesem Zyklus, der immer wieder  
827 überprüft und dann reagiert, überprüft, reagiert, kann man aber auch ausbrechen mit  
828 Abbruchbedingungen. Würden Sie jetzt sagen, wenn man sich so ein Kontrollsystem in der  
829 Realität anschaut, denken Sie, dass diese Erweiterungen, wie gezeigt in diesen beiden Modellen,  
830 diese Logik gut beschreibt? #00:16:38-7#

831 147. B: Ja, ich denke schon. Also man hat dann ja wie gesagt eben die Visualisierungselemente oder  
832 diese Bausteine in BPMN für die einzelnen Blöcke und was sie bedeuten. #00:16:49-5#

833 148. I: Ja. #00:16:50-2#

834 149. B: Und die Kontroll/ und diese Systeme werden ja dadurch dann nun sehr schön modelliert, wie  
835 gesagt, eben. Man könnte vielleicht noch schauen, dass man eben gewisse Features besser  
836 zusammen zieht und dann schaut, dass man eben gewisse Blöcke näher aneinander stellt jetzt  
837 rein visuell. Aber von der Beschreibung her finde ich es durchaus gelungen. Also jetzt, was die  
838 dahinterliegenden Systeme angeht. Ja, also, ich würde sagen, im ersten Modell, auf jeden Fall  
839 fünf Punkte. Im zweiten Modell, eben doch dadurch, dass es sehr komplex wird und die  
840 Übersichtlichkeit nicht so hundertprozentig gegeben ist, eher vier Punkte, wenn man die  
841 Visualisierung mit einberechnet. Ohne die Visualisierung oder ohne der konkreten Visualisierung  
842 wie wir es hier jetzt gesehen haben, könnte man auch fünf Punkte geben, aber mit der jetzigen  
843 Visualisierung vier Punkte für Modell Zwei. #00:17:53-5#

844 150. I: Okay, danke. Würde Ihnen zu diesem Zeitpunkt vielleicht etwas einfallen, was für eine  
845 detaillierte Prozessbeschreibung noch fehlen könnte? Also ob jetzt hier, um die Prozesse, so wie  
846 sie sind, abbilden zu können, von der Logik her, ob hier eventuell noch mehr Details notwendig  
847 wären, die jetzt hier fehlen würden. #00:18:26-3#

848 151. B: Einen Moment. Ich mache das nochmal parallel dazu auf. (...) Gut. Na ja, was vielleicht ganz  
849 gut wäre. Ganz oben bei dem öffnenden Gate sozusagen habe ich ja cancel, sequential stehen.  
850 Das Cancel bezieht sich ja quasi darauf, dass man dann am Ende wieder über diese Blöcke  
851 aussteigen kann, über den Main-Switch-Block zum Beispiel, richtig? #00:19:02-2#

852 152. I: Das Cancel bezieht sich auf das Zeitverhalten. #00:19:06-4#

853 153. B: Genau, ja. Nein, lassen Sie mich noch kurz überlegen. Aber im Großen und Ganzen fällt mir,  
854 für eine detailliertere Prozessbeschreibung/ Ja, im Prinzip sind alle Informationen da, die man  
855 braucht. Man könnte vielleicht noch ein bisschen, die Schritte/ Ja, eigentlich fällt mir da jetzt auf  
856 die Schnelle nicht wirklich viel ein. (...) Man könnte ja/ (...) Eigentlich fällt mir jetzt spontan nicht

857 wirklich viel dazu, was man konkret auch brauchen könnte. Wie gesagt, visualisierungstechnisch  
858 noch ein paar Dinge, aber jetzt von den Daten, die man braucht, um den Prozess zu betreiben,  
859 ist ja im Prinzip alles da. Man hat die Variablen, die man braucht. Man sieht, was von wo herein  
860 kommt. Welches Service welche Daten liefert, welches Service/ wie die Daten dann verarbeitet  
861 werden über die Formeln und über die Systeme, die wir da drinnen haben. Man sieht die  
862 Bedingungen, die die ganzen Events triggern. Also von daher ist eigentlich aus meiner Sicht alles  
863 da, was eine detaillierte Prozessbeschreibung angeht. Also rein von den Daten ist eigentlich alles  
864 für mich vorhanden. #00:20:40-4#

865 154. I: Okay, Dankeschön. Frage Neun. Die zielt jetzt eventuell/ die geht ein bisschen in die Richtung,  
866 die Fragestellung, wenn man bereits mit Regelungstechnik in Berührung getreten ist. Ich stelle  
867 Ihnen die Frage und ich überlasse Ihnen, ob Sie sie beantworten möchten oder nicht. Wenn Sie  
868 Erfahrung in der Regelungstechnik haben, was würden Sie empfehlen, um diese Erweiterungen  
869 zu ergänzen, um sie für Ingenieure, Steuerungstechniker, Verfahrenstechniker attraktiver zu  
870 machen? #00:21:21-6#

871 155. B: Also, wie schon eingangs erwähnt, habe ich keine Erfahrung in der Regelungstechnik. Von  
872 daher kann ich nur meine persönliche Sicht dazu sagen, wie man das eben noch vielleicht an  
873 manchen Stellen verbessern könnte. Das habe ich eigentlich schon vorhin getan. Jetzt konkret,  
874 um es für Ingenieure attraktiver zu machen. Dazu kann ich leider nicht wirklich etwas sagen,  
875 nachdem ich in dem Bereich keine Erfahrung habe. #00:21:46-2#

876 156. I: Okay, das passt. Dankeschön. Dann kommen wir auch schon zur letzten Frage, die aber ein  
877 paar Unterfragen beinhaltet. Ich möchte zum Schluss noch etwas genauer auf die Modelle mit  
878 unseren Erweiterungen von eben eingehen und Sie bitten, diese nochmals zu bewerten oder  
879 besser gesagt, die Modellierungsmethodik noch einmal zu bewerten auf einer Skala von Eins bis  
880 Fünf. Und zwar wie eben, Eins, sehr schlecht, Fünf, sehr gut. Und hier geht es ein bisschen darum,  
881 wenn Sie jetzt in Begriff sind, diese Methodik anzuwenden, also die Fragen, die um die es geht.  
882 Wie einfach ist in den gezeigten Modellen nachzuvollziehen, dass die einzelnen Abläufe parallel  
883 und unabhängig voneinander laufen? Oder wie einfach ist es zu definieren, wann eine  
884 Anpassung am System erfolgt? Also eine Regulierung. Wie einfach ist es, die maximale Dauer  
885 einer Anpassung zu definieren? Wie einfach ist es zu definieren, unter welchen Bedingungen  
886 sämtliche repetitiven Aufgaben beendet werden sollen? Wie einfach ist es zu definieren, dass  
887 danach nach dem Beenden der repetitiven Aufgaben Aufräumaufgaben einmalig zu erfolgen  
888 haben? Und die letzte Frage. Wie einfach ist es, komplexe Abläufe im Kontext von  
889 kontinuierlichen Prozessen mit diesen Erweiterungen zu beschreiben? #00:23:18-8#

890 157. B: Okay. #00:23:20-8#

891 158. I: Gehen wir zur ersten Frage wieder nach oben. Wie einfach ist es, in den gezeigten Modellen  
892 nachzuvollziehen, dass die einzelnen Abläufe parallel und unabhängig voneinander laufen?  
893 #00:23:32-8#

894 159. B: Also, das war im ersten Modell natürlich deutlich einfacher als im zweiten Modell. Ich meine,  
895 ja, wenn man weiß, dass die verschiedenen Events, diese drei Events, die wir da definiert haben,  
896 generell parallel zueinander/ innerhalb der Events die Dinge parallel zueinander ausgeführt werden  
897 können, dann ist es durchaus verständlich. Wenn man das nicht weiß und sich mal nur so  
898 generell anhand des BPMN-Modells anschaut, wie denn das alles aufgebaut ist, dann ist es, muss  
899 ich sagen, nicht hundertprozentig intuitiv. Weil man dann denkt, okay, die Pfade sind alle  
900 nebeneinander. Das wird wohl bedeuten, dass die alle parallel zueinander sind. In Wahrheit ist  
901 es ja so, die ersten Measure-Blöcke sind vor den anderen Kontrollblöcken, vor den  
902 Beendigungsblöcken sozusagen. Also wenn man das weiß, ist es sehr gut nachvollziehbar. Wenn  
903 man es nicht weiß, ist es nicht hundertprozentig intuitiv für mich. Deshalb würde ich sagen, alles  
904 in allem ein Vierer. Weil wenn man sich auskennt, weiß man, was zu tun ist und worum es geht.  
905 Aber vielleicht könnte man das auch in gewissen Szenarien so BPMN-technisch untereinander  
906 dann definieren. Weil erst habe ich ja den/ Ja, wobei ist auch schwierig, jetzt wo ich technisch  
907 darüber nachdenke. Weil man ja jederzeit in den Block springen können muss. Von daher ist es

908       nacheinander auch nicht wahnsinnig/ Ja. Nein, also. Ja, technisch macht es absolut Sinn, so wie  
909       es definiert ist. Von der reinen Modellierung her ist es nicht hundertprozentig intuitiv. Aber  
910       wenn man es weiß, dann doch. #00:25:12-4#

911   160. I: Okay, also Sie meinen, es ist visuell nicht unbedingt intuitiv erkennbar. Aber wenn man sich  
912       mit der Erweiterung auseinandergesetzt hat, könnte man nachvollziehen, was gemeint ist.  
913       #00:25:24-4#

914   161. B: Absolut, ja. #00:25:25-9#

915   162. I: Okay. #00:25:27-1#

916   163. B: Also innerhalb der Definition ist es durchaus nachvollziehbar, wie es gemeint ist. Und wenn  
917       man weiß, was die einzelnen Elemente bedeuten, dann kennt man sich auch schon aus.  
918       #00:25:37-3#

919   164. I: Okay, danke. #00:25:40-7#

920   165. B: Wie einfach ist es zu definieren, wann eine Anpassung am System erfolgt? #00:25:46-9#

921   166. I: Genau. #00:25:47-9#

922   167. B: Gut, die Dinge sieht man hier sehr schön im Modell. Weil man sie ja so, mir fällt jetzt der  
923       richtige Name zu diesem Block nicht mehr ein, aber genau eben diese, helfen Sie mir kurz auf die  
924       Sprünge. Diese Kontrollblöcke sind das, glaube ich. #00:26:02-0#

925   168. I: Ja, Control Events. Ja. #00:26:03-2#

926   169. B: Genau. Ja, das ist wirklich sehr schön zu sehen. Sobald man ein Control Event sieht, weiß man,  
927       dass danach eine Anpassung erfolgt. Also diese Dinge finde ich sehr übersichtlich. Also hier  
928       würde ich sagen, sehr gut, fünf Punkte. Wie einfach ist es, die maximale Dauer einer Anpassung  
929       zu definieren? Gut. Die maximale Dauer. Wenn ich mir nochmal das Modell anschau, dann sind  
930       diese Hertzangaben ja das Intervall, in der wieder aufgerufen wird, richtig? Und //I: Genau. //  
931       nicht die Dauer, wie lange es maximal dauern kann. #00:26:39-8#

932   170. I: Nein, im Grunde/ Hertzangaben geht es ja/ ist ja im Grunde auch eine zeitliche Vorgabe, also,  
933       wie oft in der Sekunde. #00:26:49-0#

934   171. B: Genau, wie oft in der Sekunde. Aber ist das dann quasi auch die maximale Dauer. Weil die  
935       dürfen sich nicht überlappen, richtig? Weil im Prinzip ist das quasi dasselbe. #00:26:59-6#

936   172. I: Also je nachdem, wie man es definiert hat, im Closed Loop System, und wie die Measure und  
937       Control Events miteinander zusammenhängen, könnte man über diese Attribute die  
938       Kombination daraus herauslesen, wie lange es dauern sollte. Also kompliziert formuliert. Also  
939       wenn //B: (Unv.) // ich weiß, in einem Kontroll- oder in einem Measure-Strang, machen wir es  
940       ganz einfach, habe ich einen Task, der einfach einen Wert immer wieder messen soll. Und ich  
941       habe dann als Frequenz zum Beispiel Zehn angegeben, also zehn Hertz. Dann weiß ich, okay, das  
942       wird zehn mal in der Sekunde ausgeführt, oder dieses Timer Event wird zehn mal in der Sekunde  
943       getriggert. Das heißt, der darauffolgende Task sollte zehn mal in der Sekunde erfolgen. Und  
944       daraus ergibt sich dann natürlich auch ein gewisses Intervall für die zeitliche Dauer. Also ich weiß  
945       dann, okay, die Sekunde teilt sich gleichmäßig auf zehn Teile auf. Und damit habe ich, ja, im  
946       Grunde 0,1 Sekunden, die das halt dauern darf, also hundert Millisekunden. #00:28:11-9#

947   173. B: Also für die gesamten Blöcke finde ich das sehr einfach zu definieren. Weil, da hat man eine  
948       Anpassung und man sieht, wie oft sie ausgeführt worden ist und daraus ergibt sich dann die  
949       Dauer. Ja, passt. Dann sehr gut. #00:28:24-0#

---

950 174. I: Okay. #00:28:25-3#

951 175. B: Wie einfach ist es zu definieren, unter welchen Bedingungen sämtliche repetitive Aufgaben  
952 beendet werden sollen? Gut. Das ist im Prinzip dasselbe wie zwei Blöcke davor. Das sieht man  
953 dann auch sehr schön anhand des neu definierten Events, des Cancel Events, das hier definiert  
954 ist, wo man dann rechts daneben gleich die Regel definiert hat. Also auch hier finde ich es sehr  
955 schön übersichtlich, dass man eben direkt am Event stehen hat, was denn die Bedingung ist, um  
956 das auszulösen. Ja, fünf Punkte, sehr gut. (Unv.) #00:29:05-8#

957 176. I: Danke. #00:29:07-2#

958 177. B: Wie einfach ist es zu definieren, dass danach Aufräumaufgaben einmalig zu erfolgen haben?  
959 Ja, im Prinzip alle Punkte hier. Die Aufräumaufgaben sieht man ja direkt danach als  
960 Prozessschritte. Das ist (unv.) im BPMN-Syntax, wie es jeder kennen sollte, der mit dem Modell  
961 arbeitet. Dass sie einmalig zu erfolgen haben, ergibt sich auch logisch aus dem  
962 BPMN-Prozessmodell. Und daher würde ich auch sagen, ja, wenn man das Modell kennt und  
963 weiß, dass das eben dann die Abbruchbedingung ist, würde ich auch sagen, fünf Punkte dafür.  
964 #00:29:39-8#

965 178. I: Und der letzte Punkt. #00:29:43-4#

966 179. B: Genau. Wie einfach ist es, komplexe Abläufe im Kontext von kontinuierlichen Prozessen mit  
967 diesen Erweiterungen zu beschreiben? Ja, darauf bin ich im Prinzip schon vorher eingegangen.  
968 Komplexe Abläufe zu beschreiben, stelle ich mir damit ein bisschen schwierig vor. Technisch zu  
969 implementieren mit der ■■■■, das sieht man ja schon an Ihrem Prozessmodell, klappt sehr gut.  
970 Für die Beschreibung alleine, also jetzt auch für die Betreiber, speziell für Ingenieure oder für  
971 Endanwender, sehe ich hier wie gesagt noch ein bisschen ein Verbesserungspotential, gerade  
972 was Visualisierung angeht. Also hier würde ich jetzt momentan drei Punkte vergeben. Ja,  
973 technisch ist es gut umgesetzt und die ■■■■-Beschreibung ist soweit sehr gut für mich. Aber jetzt  
974 rein aus visueller Sicht glaube ich, dass je komplexer das Modell wird, desto schwieriger wird es  
975 im aktuellen Status hier auch wirklich den Benutzern klarzumachen, was dann hier alles  
976 zusammenhängt. Ja. #00:30:51-4#

977 180. I: Okay. #00:30:53-2#

978 181. B: Gut, das war es eigentlich jetzt aus meiner Sicht. #00:30:57-2#

979 182. I: Super. Dankeschön. //B: Sehr gerne. // Das war es auch vom Interview her. Ich habe keine  
980 Fragen mehr. Vielen Dank, dass Sie sich so viel Zeit für mich genommen haben. #00:31:07-6#

981 183. B: Sehr gerne. #00:31:09-4#

982 184. I: Und danke auch für das Feedback für die Erweiterungen und generell zum Thema. Ich bin  
983 natürlich auch dankbar dafür, wenn Sie eventuell noch Anmerkungen haben für die  
984 Fragestellung, ob die Fragen generell gut formuliert waren. Ob Sie sich wohl gefühlt haben, was  
985 auch Dauer eventuell anging. Wenn Sie da irgendetwas anmerken möchten, ich bin immer offen  
986 für Feedback und für/ Generell für den Inhalt würde ich jetzt auch die Aufnahme beenden. Einen  
987 Moment. Okay. #00:31:42-0#

---