

Transcript - Group 2 "Allrounders", Interview 1

I ... Interviewer (BLINDED)

B ... Expert

(Unv.)... Incomprehensible passage

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

() ... Comment

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

Transcript

1 1. I: Okay. Aufnahme läuft. (...) Gut. Hallo und danke, dass Sie sich Zeit nehmen um mit mir dieses
2 Interview durchzuführen. [REDACTED]
3 [REDACTED]. Möchten Sie sich vielleicht kurz vorstellen und Ihre
4 Verbindung zu BPMN beziehungsweise eventuell zur Verfahrenstechnik, Fertigungstechnik oder
5 Prozessmodellierung erklären? Ich möchte Sie nur bitten, dabei nicht Ihren Namen zu nennen,
6 sondern nur die folgenden Informationen. Und zwar Berufsbezeichnung und Umschreibung des
7 Arbeitgebers, Basis Ihrer Expertise zum Forschungsthema, Ausbildung beziehungsweise
8 fachlicher Hintergrund und Berufserfahrung. #00:00:51-6#

9 2. B: Ja, gerne. [REDACTED]
10 [REDACTED]
11 [REDACTED]
12 [REDACTED]
13 [REDACTED]
14 [REDACTED]
15 [REDACTED]
16 [REDACTED]
17 [REDACTED] . #00:02:07-4#

18 3. I: Okay, danke. Dann mal eine Einführung zu unserem Thema. Unsere Forschung konzentriert
19 sich auf die Entwicklung einer Methodik um kontinuierliche Prozesse in BPMN darzustellen und
20 sie in einer Workflow Engine ausführbar zu machen. Für diese Aufgabe haben wir an BPMN
21 Erweiterungen für kontinuierliche Prozesse gearbeitet. Warum gerade kontinuierliche Prozesse?
22 Weil diskrete Prozesse bereits in anderen Forschungsarbeiten behandelt wurden und nicht die
23 gleichen Schwierigkeiten bei der korrekten Darstellung mittels BPMN aufweisen. BPMN ist
24 bereits ein weit verbreiteter Standard im Business Process Management und hat seinen Weg in
25 die Fertigung gefunden. Diskrete Fertigungsprozesse können bereits mit BPMN modelliert
26 werden. Im Grunde wollen wir eine Methodik einführen, um solche Prozesse so darzustellen,
27 dass sie von jeder Person in einem Unternehmen, vom Ingenieur bis zum Manager, verstanden
28 werden können. Dies könnte durch die Verwendung dieser Notation erreicht werden. Ein
29 weiterer Vorteil ist auch, dass es bereits eine Reihe von Workflow Engines gibt. Anwendungen,
30 die die Ausführung dieser Prozessmodelle auf der Grundlage der für jedes Symbol
31 implementierten Logik ermöglichen. Wir arbeiten mit deiner web-basierten Anwendung, die
32 erweiterbar ist und mehrere Kommunikationsschnittstellen implementiert hat. Ein weiterer
33 Vorteil ist da ja die Interoperabilität in diesem Zusammenhang im Vergleich zu anderen
34 proprietären starren Software-Anwendungen. Wir wollen herausfinden, ob diese Technik auch
35 für die Implementierung von digitalen Abbildern eingesetzt werden kann. Da digitale Abbilder
36 dazu dienen ein physikalisches System oder einen Prozess in digitaler Form darzustellen, meist
37 anhand von Daten oder mathematischen Modellen, mussten wir einen Weg finden, den Ablauf
38 von kontinuierlichen Prozessen wie sie aus der Prozessindustrie bekannt sind, darzustellen. Aus
39 diesem Grund haben wir uns auf die Modellierung von Regelkreisen konzentriert. Die
40 Prozessmodelle sollen durch BPMN für Personen mit unterschiedlichem Hintergrund leicht
41 verständlich sein. Die Interviews werden geführt, um herauszufinden, wie Prozess- und
42 Regelungstechnik und Techniken aus der Business Process Modellierung kombiniert werden
43 können und wie erste Ergebnisse von Experten wie Ihnen wahrgenommen werden. Außerdem
44 wollen wir herausfinden, ob es Schwachstellen gibt, die von Experten identifiziert wurden und

wie wir diese beseitigen können. Ich möchte jetzt bevor wir mit den konkreten Fragen starten vielleicht noch ein paar Begriffe in diesem Zusammenhang klären. Und zwar der erste Begriff wäre der digitale 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 heran zu kommen. Das soll dazu führen, dass wenn etwas getriggert wird bei einer echten Maschine der digitale Zwilling das gleiche oder ein möglichst ähnliches Verhalten zeigt. Was verstehen wir unter kontinuierlichen Prozessen? Kontinuierliche Prozesse möchte ich vielleicht mit Beispielen erklären. Wenn man Bierbrauen her nimmt, gibt es zwei Möglichkeiten. Die diskrete, also nicht-kontinuierliche Variante, wäre wenn man in einem geschlossenen Kessel die Zutaten hineingibt, zehn Liter Wasser und so weiter, 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 vollkommenen abgeschlossenen Kessel hat, sondern miteinander verbundene Kessel, bei denen immer wieder Zutaten zugefügt werden und immer wieder Bier entnommen wird. Das geht die ganze Zeit so, sodass man nicht nachvollziehen kann, welcher Liter Wasser zu welchem Liter Bier gehört. Dabei läuft ein Teilprozess im ersten Kessel ab während gleichzeitig im letzten Kessel der letzte Prozessschritt stattfindet bevor das Bier fertig wird. Und zum Schluss noch 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 halt etwas schief gehen kann, und mit den nächsten zehn Litern macht man es dann einfach besser. Aber was ist, wenn man die Brauanlage dauernd laufen lässt und ständig Bier austritt? Dann muss man währenddessen den Prozess überprüfen und schauen, dass 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 entsprechend. Stimmt etwas zum Beispiel beim Zucker- oder Alkoholgehalt nicht, dann muss das Mischverhältnis geändert werden. Das heißt, in einem geschlossenen Regelkreis werden während der Prozess läuft gewisse Werte überprüft, diese werden mit optimalen Werten verglichen und je nach Abweichung reagiert das System darauf. (...) Okay, aus Informatik-sicht bestehen kontinuierliche Prozesse aus einer sich ständig wiederholenden Abfolge von Zustandsabfragen und 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 ausführen zu können, haben wir folgende Merkmale identifiziert. Ich würde Ihnen die Merkmale in der Tabelle, die Sie hier sehen, vortragen und würde Sie danach bitten, dass Sie mir kurz sagen, ob Sie die Merkmale für wichtig oder unwichtig einstufen. Und vielleicht könnten Sie auch eine kurze Begründung für Ihre Antwort dazu abgeben. Wir haben verschiedene Zustandsabfragen und Regulierungen, Kombinationen daraus sind unabhängig und können parallel ablaufen. Dann, Regulierungen folgen immer auf Zustandsabfragen. Nummer Drei wäre die Dauer von jeder Zustandsabfragen und Regulierungskombination ist beschränkt. Nummer Vier wäre wenn Zustandsabfragen gewisse Ergebnisse liefern, wird das System beendet. Nummer Fünf. Bevor das System beendet wird, muss es in einen konsistenten Zustand gebracht werden. Und Nummer Sechs. Das resultierende System soll für Menschen verständlich sein. Das sind die Merkmale. Beginnen wir mit dem ersten Punkt. Verschiedene Zustandsabfragen und Regulierungskombinationen sind unabhängig und können parallel ablaufen. Würden Sie sagen, dass das wichtig oder unwichtig ist? #00:08:54-8#

4. B: Das würde ich als wichtig einstufen. Weil ich denke, dass Parallelität ja das Ausschlaggebende ist für solche kontinuierlichen Prozesse. #00:09:06-1#

5. I: Nummer Zwei. Regulierungen folgen immer auf Zustandsabfragen. #00:09:14-7#

98 6. B: Ich meine, ist sicherlich auch wichtig, aber wahrscheinlich tendenziell eher unwichtig. (lacht)
99 (...) Ich meine, Regulierungen müssen ja nicht immer auf Zustandsabfragen folgen. Es kann aber
100 auch viele Zustandsabfragen in Folge geben, ohne dass eine Regulierung passiert, oder?
101 #00:09:42-1#

102 7. I: Hm. (zustimmend) Genau. Also. Wir haben uns hier/ Es war eher so gedacht, das so zu
103 formulieren, dass Regulierungen nur passieren können, wenn man auch den aktuellen Zustand
104 des Systems kennt. #00:09:53-4#

105 8. B: Ach so, aber das ist dann schon wichtig. #00:09:56-7#

106 9. I: Hm. (zustimmend) So war es eher gedacht. Ja. Als Kontext. #00:10:00-5#

107 10. B: Kann ich alle Fragen als wichtig/ also kann ich alle Eigenschaften als wichtig einschätzen?
108 #00:10:08-1#

109 11. I: Also, es steht Ihnen offen. Es gibt natürlich keine Vorgabe von meiner Seite, aber es steht
110 Ihnen offen die Antwort dazu zu diskutieren, also/ #00:10:18-1#

111 12. B: Okay. #00:10:18-5#

112 13. I: Ihre Gedanken dazu mit mir zu diskutieren. #00:10:21-7#

113 14. B: Ja, (unv.) ich würde schon das (unv.) als wichtig erachten. Weil wenn man nicht weiß, was der
114 aktuelle Zustand ist, dann kann man ja nicht regulieren. Also ja, wichtig. #00:10:29-5#

115 15. I: Okay. Nummer Drei. Die Dauer von jeder Zustandsabfrage und Regulierungskombination ist
116 beschränkt. Vielleicht Kontext dazu. Wir gehen davon aus, dass das System in einem gewissen
117 Zeitfenster reagieren sollte, weil dann eventuell der letzte Wert, der gemessen wurde, nicht
118 mehr gültig sein könnte. #00:10:51-6#

119 16. B: (...) Ja, ist schon wichtig, aber das ergibt sich ja auch irgendwie Punkt Nummer Zwei, oder?
120 Also, dass ich immer eine Zustandsabfrage haben muss, bevor ich reguliere. #00:11:08-0#

121 17. I: Ja, aber wenn man konkret auch ein zeitliches Fenster oder quasi eine Range, einen gewissen
122 Zeitpunkt, angeben kann, in der die Regulierung geschehen sollte, hätte man dadurch natürlich
123 einen Mehrwert für den Informationsgehalt des jeweiligen Prozesses gewonnen. Also es geht
124 hier nicht nur darum, dass die Reihenfolge geregelt ist, sondern auch die zeitliche Abfolge.
125 #00:11:36-6#

126 18. B: Okay, also das heißt, dass die Zustandsabfrage zum Beispiel maximal fünf Sekunden her sein
127 darf, bevor ich reguliere. Sonst muss ich nochmal Zustandsabfragen? #00:11:47-1#

128 19. I: Zum Beispiel. Ja. #00:11:47-9#

129 20. B: Ich meine, ja, ist denke ich schon wichtig. Aber würde ich jetzt nicht als SO wichtig betrachten.
130 #00:11:58-5#

131 21. I: Darf ich fragen, warum? #00:12:03-1#

132 22. B: Das ist eine gute Frage. Ja ich meine, es kommt wahrscheinlich ganz darauf an, was für ein
133 Prozess das ist und wie wichtig das ist, wie lang das her ist. Aber ich schätze halt, dass es
134 wahrscheinlich oft nicht so wichtig ist, ob es jetzt fünf oder zehn Sekunden her ist. Aber ja,
135 wahrscheinlich ist es/ kommt es auf den Prozess darauf an. #00:12:28-3#

136 23. I: Okay, dann kommen wir zu Punkt Nummer Vier. Wenn Zustandsabfragen gewisse Ergebnisse
137 liefern, wird das System beendet. Das soll heißen, dass es/ Was ist der Hintergrund dazu? Was

138 ist der Kontext dazu? Das soll heißen, wenn die Zustandsabfragen gewisse Zustände belegen,
 139 oder wenn dadurch gewisse Zustände gemessen wurden, kann man diese natürlich so definieren,
 140 dass das System entsprechend beendet werden sollte, weil zum Beispiel sonst Gefahr droht oder
 141 eventuelle Schäden oder vielleicht sogar finanzieller Verlust. So etwas. Dass man einfach gewisse
 142 Zustände hat, die man nicht weiter erhalten lassen möchte oder weiterverfolgen möchte, mit
 143 dem System. Das ist der Hintergrund. #00:13:28-0#

144 24. B: Ja, ist definitiv wichtig. Eben eh aus den genannten Gründen. Dass es halt auch aus
 145 Sicherheitsaspekten sehr wichtig ist. Weil wenn man das dann eben wirklich in der Produktion
 146 einsetzt, muss es ja quasi so eine Art Notaus geben und so würde ich das jetzt mal sehen.
 147 #00:13:47-2#

148 25. I: Punkt Fünf. Bevor das System beendet wird, das basiert ein bisschen auf Punkt Vier, muss es in
 149 einen konsistenten Zustand gebracht werden. #00:14:00-9#

150 26. B: Ja, das ist definitiv auch wichtig, eh genauso wie Punkt Vier. Also das baut ja eben darauf auf,
 151 dass wenn es abgeschaltet wird, dass es in einem konsistenten Zustand abgeschaltet wird, ist
 152 sicherlich auch wichtig. Eben auch aus Safety-Aspekten. #00:14:17-5#

153 27. I: Und der letzte Punkt in dieser Tabelle wäre, das resultierende System soll für Menschen
 154 verständlich sein. Das ist ein bisschen darauf abzielend, dass man das Modell so modelliert, dass
 155 es für verschiedene User einfach verständlich gestaltet ist. #00:14:38-7#

156 28. B: Ja, finde ich definitiv auch sehr wichtig. Also ein Modell/ Der Sinn ist ja von einem Modell,
 157 dass es verständlich ist für Menschen. Und gerade BPMN ist ja eigentlich (...) eh gut verständlich
 158 und intuitiv. Aber/ Ja, das soll auch so bleiben. (lacht) Also definitiv wichtig. #00:15:01-8#

159 29. I: Okay, super. Dankeschön. Dann kommen wir zur zweiten Frage. Jetzt nach diesen Merkmalen,
 160 die ich aufgelistet habe in dieser Tabelle. Können Sie vielleicht grafische Eigenschaften nennen,
 161 die Sie für die Modellierung kontinuierlicher Prozesse wichtig finden? Und würde sich daraus
 162 vielleicht ergeben, dass wir auf Merkmale darauf kommen oder dass Sie jetzt Merkmale/ dass
 163 Ihnen welche einfallen, die wir eventuell in dieser Liste vergessen haben? #00:15:30-5#

164 30. B: Also grafische Eigenschaften, denke ich, ist auf jeden Fall diese Parallelität, die sehr wichtig ist.
 165 Dass halt gleichzeitig mehrere Dinge zum Beispiel gemessen werden. Dass man das gut
 166 darstellen kann. (...) Ja. Was vergessen worden ist, fällt mir jetzt nichts auf. Also mir würde jetzt
 167 nichts Zusätzliches einfallen. #00:16:03-9#

168 31. I: Okay, dann kommen wir zu Frage Drei. Wo liegen Ihrer Meinung nach, wenn man jetzt an
 169 kontinuierliche Prozesse denkt, die Herausforderungen bei der Modellierung? #00:16:18-5#

170 32. B: Also. Ich glaube das ist auf jeden Fall die Herausforderung, diese (...) diese Parallelität ist
 171 wahrscheinlich eine Herausforderung, dass es dann nicht zu zu großen, zu komplexen Modellen
 172 führt, die dann unübersichtlich werden und dann eben auch nicht mehr verständlich sind. Und
 173 wahrscheinlich ist auch bisschen dieses Problem, so, was ist jetzt genau eine Instanz? Also wie
 174 unterscheiden wir die unterschiedlichen Instanzen, sind wahrscheinlich dann auch eine
 175 Herausforderung, wobei das bei der Modellierung wahrscheinlich weniger herausfordernd ist als
 176 dann beim Ausführen. Oder bei der Analyse vielleicht sogar, noch mehr das Problem ist. //I: Ja/
 177 //Genau. #00:17:07-9#

178 33. I: Entschuldigung. Ich wollte Sie nicht unterbrechen. (lacht) #00:17:12-7#

179 34. B: Nein. Ich bin fertig mit meiner Antwort. #00:17:15-2#

180 35. I: Okay. Da Sie jetzt in diesem Zusammenhang Instanz auch erwähnt haben. Es wäre auch/ (...) Es
 181 wäre auch gut, um den Informationsgehalt des Interviews ein bisschen zu erweitern, wenn Sie
 182 kurz erklären könnten, was Sie mit Instanz meinen. #00:17:34-8#

183 36. B: Also, mit Instanz meine ich bei traditionellen Produktionsprozessen zum Beispiel gibt es ja
184 immer zum Beispiel ein Werkstück. Und das heißt eine Instanz wäre dann quasi ein Werkstück,
185 das halt zuerst bei der einen Maschine ist, dann bei der anderen Maschine ist. Das ist ja bei
186 kontinuierlichen Prozessen viel schwerer zu definieren weil wir ja diesen einen Liter haben, der
187 irgendwo sein kann. Also wir wissen nicht, wo dieser eine Liter ist und wenn der eine Liter die
188 Instanz ist, macht das dann quasi recht wenig Sinn. (...) Das meine ich hier mit Instanz.
189 #00:18:15-3#

190 37. I: Okay. Dankeschön. Danke für die Erklärung. // B: Gerne, gerne. // Okay. Dann kommen wir
191 zum nächsten Abschnitt. Ich würde gerne die Erweiterungen, die wir für diese Arbeit definiert
192 haben, erklären. Und zwar. Ich werde Ihnen Prozesse zeigen, die mit BPMN 2.0 und mit unseren
193 Erweiterungen modelliert wurden. Die Erweiterungen sollen zum einen vordefinierte
194 Modellierungskonventionen für in der Prozess- und Steuerungstechnik übliche Routinen
195 bereitstellen. Und zum anderen helfen, die Unterschiede zwischen den parallelen Pfaden in den
196 Prozessmodellen zu visualisieren. Die Prozesse werden in der [REDACTED],
197 kurz [REDACTED], modelliert. Nur ganz kurz zum Background. Sie sind mit der/ Sie kennen die [REDACTED],
198 grundsätzlich? #00:19:11-4#

199 38. B: Ja, genau. // I: Sie sind schon/ // Also ich habe schon öfter damit gearbeitet. #00:19:14-7#

200 39. I: Okay, Sie sind also damit vertraut. Sehr gut. #00:19:16-7#

201 40. B: Ja. #00:19:17-5#

202 41. I: Dann gehe ich gleich weiter zu unseren speziellen Erweiterungen. Das erste Symbol, das Sie
203 hier sehen, ist das sogenannte Closed Loop Subsystem Gateway. Das Gateway ist eine
204 Kombination aus einem inklusive und einem ereignisbasierten Gateway. Es enthält
205 Verzweigungen beziehungsweise Kanten, die für die Zustandsabfragen und Regulierungsphasen
206 des Zyklus ausgelöst werden, sowie Verzweigungen, die beim Empfang von Abbruchereignissen
207 ausgeführt werden. Die Ereignisse und Tasks in den einzelnen Kanten sind unabhängig
208 voneinander. Damit erfüllen wir das erste der oben genannten Features, dass einzelne Verläufe
209 unabhängig voneinander sind und sie parallel ausgeführt werden. Das Gateway ermöglicht
210 außerdem die Definition der Intervalldauer jedes Zyklus sowie von Überschreitungsbedingungen,
211 durch wait oder cancel definiert, und der Ausführungsreihenfolge für Zustandsabfragen und
212 Regulierungen, beziehungsweise kann man es auch als Mess- oder Steuerungsaufgaben
213 bezeichnen. Hier sehen Sie Ausschnitte aus der [REDACTED], wie man diese Eigenschaften definieren
214 kann. Einmal zum Beispiel das Attribut Interval duration overrun, mit wait oder cancel, und
215 andererseits Measure control cycle execution, einmal parallel oder sequentiell. Zum Unterschied
216 zwischen wait und cancel. Wenn wait gewählt wird, beginnt die nächste Iteration, wenn alle
217 Verzweigungen beendet sind und die festgelegte Intervalldauer erreicht ist. Bei cancel definiert
218 die Intervalldauer genau die Zeit, in der jeder Zweig zu beenden ist. Wenn die Tasks in einem
219 Zweig schneller beendet werden, wird der Zweig warten. Wenn noch nicht alle Tasks beendet
220 sind, werden sie abgebrochen. (...) Zu parallel und sequentiell. Bei parallel werden die Tasks
221 nach Measure und Control Events parallel ausgeführt. Zu diesen beiden Events kommen wir
222 gleich. Bei sequential werden die Tasks nach Control Events erst ausgeführt, nachdem alle Tasks
223 nach Measure Events beendet sind. (...) Jetzt kommen wir zu den Erweiterungen für
224 Intermediate Catching Events. In einem Closed Loop Subsystem werden spezifische Ereignisse
225 erwartet, die in eine der drei folgenden Kategorien fallen. Ereignisse für Zustandsabfragen,
226 Messungen, das wären Measure Events. Ereignisse für Regulierungen, das wären Control Events.
227 Und Ereignisse für die Unterbrechung des Closed Loop Subsystems. Es gibt für jede
228 Ereigniskategorie zumindest eine Kante, die vom Gateway ausgeht. Die Karten zeigen an, welche
229 Tasks nebeneinander ablaufen. Sobald diese Ergebnisse eintreten, werden auch die Tasks, die in
230 den Kanten danach angeordnet sind, ausgeführt. Hier sehen Sie gleich ein Bild eines Closed Loop
231 Subsystems, in dem nur Ereignisse der drei Kategorien, ohne darauf folgende Tasks modelliert
232 sind. Das heißt, wenn man das in der [REDACTED] modellieren würde, wäre das erste, was man sieht,
233 mit einem Closed Loop Subsystem diese Grafik, diese Darstellungsweise. Die drei

234 Ereigniskategorien, die wir definiert haben, sind wie folgt. Einmal Measure Events. Dieses
 235 Symbol empfängt Events für die Ausführung von Tasks in Messzyklen. Dann Control Events,
 236 empfängt Events für die Ausführung von Tasks in Regelzyklen. Und dann haben wir Cancel,
 237 empfängt Events für das Abbrechen von Closed Loop Systems. Diese Symbole geben den Zweck
 238 der nachfolgenden Tasks oder Aufgaben an. Diese Tasks werden nur ausgeführt, wenn die
 239 Ereignisse ausgelöst werden. Das bedeutet, dass das Messereignisse angibt, dass die
 240 nachfolgenden Symbole nur Messabläufe beziehungsweise Zustandsabfragen anzeigen. Das
 241 gleiche gilt für Regulierungs- oder Kontroll- und Abbruchereignisse. Für Zustandsabfragen und
 242 Regulierungen oder beziehungsweise auch Mess- und Steuerungereignisse könnte man sagen,
 243 können wir eine Zykluszeit definieren. Dadurch kann die Dauer von Anpassungen oder
 244 Regulierungen im System definiert werden. Je nachdem ob das Closed Loop Subsystem einen
 245 parallelen oder sequentiellen, oder einen Wait- oder Cancel-Ansatz verfolgt, läuft die
 246 Ausführung auch unterschiedlich. Mit diesen Bedingungen kann man definieren, inwiefern
 247 Anpassungen beim System erfolgen. Hier sehen Sie ein Closed Loop Subsystem mit einem Task
 248 für eine Messung. In diesem Fall wird das Ereignis für die Messung alle zehn Sekunden getriggert.
 249 Danach wird der Wert V 1 geholt beziehungsweise gemessen. Wait bedeutet hier, dass der neue
 250 Zyklus erst startet, wenn die Messung erfolgt, das heißt, der Prozess in dieser Kante
 251 abgeschlossen ist. Mit cancel wird nach zehn Sekunden automatisch der neue Zyklus gestartet.
 252 Das heißt, wir hätten jetzt genau einen Service Call nach dem Measure Event modelliert.
 253 #00:24:37-7#

254 42. B: Aber darf ich kurz nachfragen? Hier läuft es aber parallel ab, oder? Also, hier wird gleichzeitig
 255 gemessen und das Kontroll/ (...) #00:24:47-5#

256 43. I: Genau/ #00:24:48-5#

257 44. B: Ereignis kann aber dazu parallel ablaufen? #00:24:51-0#

258 45. I: Genau, exakt. Ja. #00:24:52-6#

259 46. B: Okay. Und das Null Komma Eins (0,1), also das bezieht sich auf die Messung? Also/
 260 #00:25:09-8#

261 47. I: Genau, das wäre die Frequenz, in der/ Entschuldigung. #00:25:15-9#

262 48. B: Ist das dann so wie eine Abtastrate, oder so? #00:25:17-9#

263 49. I: Genau, ja. #00:25:19-4#

264 50. B: Okay. #00:25:20-2#

265 51. I: Also, die Frequenz wird für das Timer Event am Anfang der Kante definiert und im Grunde wird
 266 das hier mit einer Frequenz von 0,1, also alle zehn Sekunden, überprüft und ausgelöst. Und
 267 sobald es ausgelöst wird, werden auch die Tasks, die nachher dann in diesem Strang oder in
 268 dieser Kante modelliert werden, ausgeführt. #00:25:42-2#

269 52. B: Okay. #00:25:43-6#

270 53. I: Und eine Frequenz kann man auch bei Kontrollereignissen, also bei Control hier im zweiten
 271 Strang dargestellt, definieren. #00:25:51-9#

272 54. B: Passt. #00:25:58-4#

273 55. I: Für Measure Events hätte man hier diese Darstellung. Also, so ein Fenster. (...) Man kann
 274 Interval frequency in Hertz angeben. Und dann kann man auch den Wert angeben, der hier
 275 überschrieben wird beziehungsweise der geändert wird mit der Zustandsabfrage. Sollte man
 276 mehrere Zustandsabfragen in dieser Kante modelliert haben nacheinander, dann kann man hier

auch mehrere Werte auflisten, wenn man das möchte. (...) Mithilfe von Regelungsereignissen kann ferner festgelegt werden, welches Reglermodell verwendet wird. PID, PI, PD. Diese Regler werden in ihrer mathematischen Form dargestellt. Die Tasks für sie sind im Grunde Berechnungen, die in festen Teilprozessen dargestellt werden. Nach diesen Berechnungen kann der Benutzer Tasks zu weiteren Datenverarbeitung hinzufügen. Dies kann auch nach Messtasks geschehen. Diese könnte man zum Beispiel auch als Datenerfassungs-Tasks bezeichnen. Das heißt, man könnte hier auch einfach einen Task nach dem Measure Event modellieren und es einfach immer wieder zyklisch durchlaufen lassen und sagen, 'Ich möchte dieses Closed Loop Subsystem nur für die Datenmessung oder für die Datenerfassung nutzen.' Es muss nicht unbedingt ein Task nach dem Control Event modelliert werden. Hier sehen Sie ein Prozessmodell mit einem Wert, der gemessen wird, und einer darauf folgenden Regelung. Wir haben wieder V 1, der nach dem Measure Event gemessen wird. Dann wird das Control Event ausgelöst und wir können beispielsweise hier ein Script ausführen, in dem die Differenz zwischen zwei Werten berechnet wird. Dann wird das entsprechende mathematische Modell des Reglers angewandt, in dem Fall hier als Beispiel vom Bezeichner her PID Code. Und dann kann man, wenn man ein entsprechendes Ergebnis heraus bekommen hat, den Manipulating Value, also das Ergebnis, an den jeweiligen Aktor schicken, auch mit einem Service Call. (...) Man kann bei Control Events ähnliche Attribute definieren, sogar noch ein paar mehr als bei Measure Events. Wir haben hier wieder wie gesagt die Frequenz, die man definieren kann. Man kann aber auch den jeweiligen Control type hier definieren, also wie gesagt PID, PD, was auch immer angeboten wird. Man kann den jeweiligen Wert, der berechnet wird hier eintragen und man kann auch ein Upper und ein Lower limit für diesen Wert angeben, also eine gewisse Range auch. #00:28:53-5#

56. B: Das heißt, die Werte dürfen nur innerhalb dieser Range liegen? #00:28:58-6#

57. I: Genau. (...) Wait bedeutet wieder, dass für den nächsten Zyklus auf das Beenden aller Tasks gewartet wird. Auch auf die Regulierungstasks. Sequentiell heißt, dass die Tasks nacheinander ausgeführt werden. Das heißt, es wird erst gemessen beziehungsweise der Zustand abgefragt, und mit diesem gemessenen Wert wird die Regelung durchgeführt. Dabei wird vom optimalen Wert, wie gesagt V opt, der aktuelle Wert V 1 abgezogen, also die Differenz wird berechnet. Und darauf basierend wird dann der neue Stellenwert MV ausgerechnet. Dieser wird dann mit einem Service Request an das entsprechende Stellglied, also ein Element, das aktiv Einfluss auf den Prozess ausübt, geschickt. Wenn man möchte, kann man die Differenzberechnung, die Regelungsberechnung für das PID-Modell und das Aussenden des Befehls an das System in einem Subprozess zusammenfassen, für die bessere Übersichtlichkeit. Bei Control können zusätzlich die Art der Regelung sowie der neue Stellwert und dessen Limits eingetragen werden, wie ich es vorhin gesagt habe. Und würde hier parallel verwendet werden, würde der letzte Wert von V 1 genommen werden für den dann aber keine Zeitgarantie bestehen würde. Das heißt parallel würde heißen, dass gleichzeitig gemessen und reguliert wird. Das heißt, die Tasks nach Control müssen dann mit dem letzten verfügbaren Wert arbeiten und man könnte dann nicht genau sagen, in welchem Zeitfenster des letzten Zyklus das passiert ist. Also man weiß, das ist aus dem letzten Zyklus, wo der letzte Wert gemessen wird, aber es ist halt nicht der wirklich jüngste Wert, den man nehmen könnte. Das wäre dann eher bei sequentiell gegeben. Zustandsabfragen und Regulierungen sollten in regelmäßiger Frequenz ausgelöst werden. Abbruchereignisse hingegen werden nur durch ihre Abbruchbedingungen ausgelöst, die der Benutzer definieren kann. Ein Beispiel für einen Abbruchereignis wäre, wenn etwas den Abbruch eines Zyklus auslöst, wie beispielsweise eine Watchdog-Funktion zur Überwachung der maximalen Zykluszeit bei SPS'en oder es könnte ein Notstopp sein. Hier sehen Sie jetzt einen Prozess mit einem zu messenden Wert, einer Regelung und einer Abbruchbedingung. Das heißt, wir haben das letzte Modell von vorhin, das letzte Beispiel. Und haben hier aber auch schon eine Abbruchbedingung definiert, haben aber keine Tasks oder dergleichen nach dem Abbruch-Event modelliert. Das heißt, als Definition hier, als Abbruchbedingung haben wir Emergency Stop und wenn der active gesetzt ist, also wenn active auf true ist, dann wird das System abgebrochen. Und standardmäßig hätten wir für diesen Wert, wenn wir unter Data Elements schauen in der  eigentlich false. Weil wenn diese Bedingung natürlich von Anfang an true ist, hätten wir ein Problem, weil dann würde das System eigentlich gleich abrechen. (...) Nachdem das Abbruchereignis ausgelöst wurde, können Tasks zur Aufräumroutine abgearbeitet werden, bevor

der Zyklus beendet oder der Prozess vollständig beendet wird. Damit wird auch das fünfte Feature für Aufräumprozesse erfüllt. Hier sehen Sie einen Prozess, bei dem Aufräum-Tasks definiert wurden. Das heißt, bei dem wir auch entsprechende Tasks nach dem Abbruch-Event eingefügt haben. Und zwar gleiches Beispiel wie vorher, nur dass wir hier jetzt einen Task eingefügt haben, der dafür steht, dass eine gewisse Shutdown-Routine für diesen Kessel oder diesen Behälter durchgeführt wird. Das kann jetzt natürlich irgendetwas sein, es kann ein Service Call, ein einfacher sein, der einfach von außen her etwas Neues los startet. Das kann ein Script sein, das etwas beispielsweise einfach berechnet, bevor das System beendet wird, wie auch immer. Oder es kann natürlich auch ein Subprozess sein, der ausgeführt wird. Die vorgestellten Erweiterungen sollen bei der Modellierung von kontinuierlichen Prozessen helfen, indem Vorlagen für die Erstellung von Prozessmodellen vorgegeben werden, und andererseits durch die Darstellung als Closed Loop Subsystem mit eigenen Symbolen für Zustandsabfrage-, Regulierungs- und Abbruchereignisse helfen, solche Prozesse leichter nachvollziehen zu können. Hinzu kommt, dass man für eine übersichtlichere Darstellung des gesamten Prozesses auch Subprozesse zur Unterteilung nutzen kann und damit erfüllen wir auch das letzte Feature, dass wir unsere Modelle möglichst verständlich für User machen wollen. Kommen wir zu den Prozessbeispielen. Ich werde Ihnen nun Prozessbeispiele zeigen, die mit den in unserer Arbeit vorgestellten Erweiterungen modelliert sind. Ich möchte, dass Sie sich die Modelle ansehen und mir sagen, was Sie aus ihnen heraus lesen können und ob die Modelle die notwendigen, Ihrer Meinung nach, notwendigen Informationsgehalt für die Modellierung der zugrundeliegenden Regelungsprozesse erfüllen. Vorab wird Ihnen zum jeweiligen Prozess erklärt, was abgebildet werden soll. Ich würde Sie bitten, offenes Feedback zu den Modellen in BPMN zu geben. Das erste Beispiel ist eine einfache PI-Temperaturregelung, für einen Wärmetauscher, basierend auf dem Beispiel aus der MathWorks-Bibliothek. Die Temperatur einer Flüssigkeit in einem Rührkessel wird mittels Wärmetauscher geregelt. Der über den Wärmetauscher eingebrachte Wärmestrom, also die Wärmeenergie, kann man sagen, wird über ein Ventil, das einen Dampfstrom kontrolliert, gesteuert. Das heißt, der Dampfstrom ist die eigentliche Wärmequelle. Der zu beachtenden störende Umgebungseinfluss, das heißt, das was das System immer wieder von der optimalen Temperatur abweichen lässt, ist die schwankende Temperatur der zugeführten Flüssigkeit. Der Tank ist als isoliert anzunehmen. Das heißt, wir gehen nicht davon aus, dass Wärme über die Kesselwand oder die Tankwand abgeführt wird. Hier sehen Sie noch das Flowchart des Prozesses. Das heißt, wir haben einen Rührkessel, wir haben einen Temperaturfühler, wir haben einen kontinuierlichen Zufluss, und einen Abfluss, und die Temperatur der Flüssigkeit hier drinnen wird immer wieder durch einen Wärmetauscher beziehungsweise durch den Dampfstrom, der hier durch dieses Ventil gesteuert wird, beeinflusst. Und wir versuchen, die Temperatur der Flüssigkeit in diesem Kessel auf einem gewissen Wert zu halten. So. Auf Basis der Unterlagen aus MathWorks haben wir hier auch einige Variablen definiert, die dann später für das mathematische Modell des PI-Reglers wichtig sind. Beziehungsweise haben wir auch mal die Variable für die Temperatur im Tank definiert und mal angelegt beziehungsweise auch für die Störgröße. Beispielsweise habe ich hier noch ein paar Endpunkte eingefügt. Sie kennen ja die . Sie wissen ja grundsätzlich, wie man das hier definieren kann und wie man das nutzen kann für den Prozess. Wir gehen davon aus, dass dieser Prozess mit wait und sequential arbeitet. Das heißt, es wird gewartet bis in jedem Strang die Tasks fertig sind beziehungsweise werden erst die Messtasks ausgeführt und dann wird entsprechend geregelt. Und hier sehen Sie den Graphen für den Prozess. Das heißt, wir haben einmal die Messung für die Temperatur des Tanks. Dann habe ich noch beispielhaft ein Script hier eingefügt. Man könnte zum Beispiel noch die Temperatur, die man bekommt, je nachdem welches Interface für den Sensor verantwortlich ist, könnte man hier halt noch eine Umwandlung des Prozesswertes veranlassen. Wenn man es nicht braucht, kann man das natürlich auch weglassen. Und dann wird aber auch gemessen, welche Temperatur als Störgröße durch den kontinuierlichen Zufluss eingebracht wird. Das heißt, diese zwei Zustandsabfragen haben wir, und dann die entsprechende Regulierung, ist ein PI Controller. Hier haben wir einerseits die mathematische Berechnung des PI-Modells. Dann könnte man noch sagen, dass der Ausgangswert, den hier heraus bekommen, auch noch eventuell umgewandelt werden muss, und der jeweilige Wert, sollte man die Umwandlung jetzt brauchen, also irgendeine Conversion oder irgendeine Umrechnung, der jeweilige Wert, der dann herauskommt wird dann entsprechend an einen Akteur weitergegeben. Also das heißt, dass der Wert, den man dann

389 bräuchte, um entsprechend das/ den Motor, den Antrieb des Ventils, so zu steuern, dass der
390 Dampfstrom entsprechend reguliert wird. In den MATLAB-Unterlagen, in den
391 MathWorks-Unterlagen, war jetzt keine Stoppbedingungen oder keine Abbruchbedingung
392 definiert. Das heißt, wir haben hier jetzt allgemein einfach als Bedingungen eingefügt Stop
393 activated, wenn der auf true gesetzt wird, dann wird die entsprechende Shutdown-Routine oder
394 Shutdown-Sequenz einfach ausgeführt. Script-Beispiele, wie das mathematisch Modell des
395 PI-Reglers aussieht, sehen Sie hier. Das heißt, wir bekommen dann einen gewissen Wert heraus,
396 und je nachdem, ob wir den dann noch umwandeln müssen in einem Zwischenschritt oder nicht,
397 kriegen wir dann noch einen Wert, den wir, wie gesagt, an den Antrieb des Ventils schicken
398 können. Das heißt, dass wäre mal das weniger komplexe Prozessmodell von beiden Beispielen.
399 Ich würde Sie jetzt bitten, mit dieser Tabelle, die Sie hier sehen, das Modell nach den Kriterien
400 hier auf einer Skala von Eins bis Fünf zu bewerten, wobei Eins sehr schlecht ist und Fünf sehr gut.
401 Das heißt, je mehr Punkte desto besser. Und die Kriterien, die wir hier haben, sind einerseits
402 Verständlichkeit. Das heißt, verstehen Sie oder haben Sie Ihrer Meinung nach verstanden, was
403 hier passiert? Ist das aus dem Prozessmodell ersichtlich? Dann Übersichtlichkeit. Können Sie das
404 gesamte System auf einen Blick erfassen? Einfachheit. Könnte man das Modell eventuell noch
405 einfacher darstellen? Logik. Wird klar, was parallel und was sequentiell passiert? Und auch noch
406 Erweiterbarkeit. Könnte man dem Modell noch etwas hinzufügen, was den Informationsgehalt
407 verbessern würde? Ihrer Meinung nach. Wenn ich noch einmal heraus scrollen soll, sagen Sie es
408 einfach. #00:40:23-1#

409 58. B: Ich habe das eh sonst parallel offen dann. #00:40:26-0#

410 59. I: Okay. #00:40:26-8#

411 60. B: Ja, zur Verständlichkeit würde ich, glaube ich, Fünf geben. Also, ich glaube, ich habe sehr gut
412 verstanden, wie dieser Prozess aussieht. Auch bei der Übersichtlichkeit würde ich sehr gut
413 sagen. Also das hängt, glaube ich, eh auch stark zusammen. (...) #00:40:56-4#

414 61. I: Könnte man das// B: Eig/ // Entschuldigung. Ich wollte nur überleiten zur nächsten Eigenschaft.
415 #00:41:03-1#

416 62. B: Zur Einfachheit? Ja, das ist auch/ Also, ich meine, man könnte eventuell beim Kontrollstrang
417 die zwei Scripte und den Task zusammenfassen in einen Prozess, aber das ist glaube ich in
418 diesem Fall hier nicht nötig. Also ich würde hier auch Fünf hergeben. Also sehr gut. Logik. Wird
419 klar, was parallel und was sequentiell passiert? Ja, also die zwei Messprozesse, die laufen aber
420 schon parallel ab, oder? Die/ Also/ #00:41:47-8#

421 63. I: Grundsätzlich ist es so gedacht. Weil für beide auch die gleiche Frequenz grundsätzlich
422 definiert ist. Also, hier. Genau. Aber wenn es für Sie nicht wirklich ersichtlich ist, dann könnte
423 man eventuell noch etwas verbessern. #00:42:06-8#

424 64. B: Ja, ich/ Also. Für mich ist es halt so, mir ist klar, wenn sequentiell da steht, wird auf jeden Fall
425 zuerst gemessen, dann wird der Kontrollprozess ausgeführt. Und die Stoppbedingung ist sowieso
426 immer/ die steht halt immer. Aber für mich ist entweder/ entweder wird alles sequenziell
427 ausgeführt, also zuerst der erste Messprozess, dann der zweite, dann die Kontrollbedingung,
428 oder alles parallel. Also ist für mich nicht ganz klar, dass die ersten zwei parallel sind, aber
429 Kontrolle dann sequentiell ist. #00:42:43-9#

430 65. I: Das heißt, man könnte hier an der Darstellungsweise eventuell noch etwas verbessern?
431 #00:42:49-2#

432 66. B: Ja, zum Beispiel, dass es einfach auf derselben Höhe ist, also dass es nicht untereinander ist,
433 die zwei Messprozesse sind vielleicht auf derselben Höhe. Das würde vielleicht noch klarer
434 ausdrücken, dass das parallel abläuft. Wenn Sie verstehen, was ich sagen möchte. (lacht)
435 #00:43:10-3#

- 436 67. I: Ja, also, dass man beide Measure-Symbole beispielsweise hier auf die gleiche Höhe zieht.
437 #00:43:16-5#
- 438 68. B: Genau, ganz genau. Das meine ich. #00:43:18-9#
- 439 69. I: Verstehe. Ja. #00:43:23-5#
- 440 70. B: Also, da würde ich dann vielleicht eine Vier geben. Es ist schon gut, ich verstehe es schon.
441 Aber es kann vielleicht noch eine Spur klarer dargestellt werden. Bei Erweiterbarkeit, könnte
442 man dem Modell noch etwas hinzufügen, das den Informationsgehalt verbessern würde? Was
443 mir auch noch nicht ganz klar ist. Diese gesamte Sequenz wird mit der Frequenz von ein Hertz
444 wiederholt, oder wie oft wird diese gesamte Sequenz wiederholt? Also, der gesamte Prozess?
445 #00:44:14-5#
- 446 71. I: Der gesamte Prozess. Es gibt hier keinen Unterschied zwischen Measure und Control. Das heißt,
447 es wird alles im Ein-Hertz-Bereich, also einmal pro Sekunde, durchlaufen. #00:44:28-1#
- 448 72. B: Okay, und das wird so lange durchlaufen, wie wir definiert haben, in dieser Maximum time/
449 oder wie lange oder wie oft durchlaufen wir das oder ist das einfach durchgehend bis eine
450 Stoppbedingungen erfüllt wird? #00:44:44-3#
- 451 73. I: Das ist der Sinn hinter der Stoppbedingung oder der Abbruchbedingung. Dass der
452 kontinuierliche Prozess, der unseres Verständnisses nach ja auch nicht wirklich eine definierte
453 Start- und Endzeit hat, auf diese Stoppbedingung reagieren soll. Ja. #00:44:59-9#
- 454 74. B: Okay, passt. Dann habe ich es eh richtig verstanden. Also, nein, bei Erweiterbarkeit/ Da glaube
455 ich fehlt dann auch nichts. Ja, stimmt. Also eigentlich, also Stop activated wäre dann in dem Fall
456 einfach irgendwas Externes und dann drücke ich auf einen Stoppknopf, und dann stoppt der
457 Prozess, oder so. (lacht) #00:45:26-5#
- 458 75. I: Genau, ja. #00:45:27-7#
- 459 76. B: Nein, dann würde ich/ bei Erweiterbarkeit fehlt mir jetzt nichts. Also, das heißt, (...) sehr gut in
460 dem Fall. (lacht) #00:45:40-3#
- 461 77. I: Okay, wunderbar. Danke. #00:45:43-1#

462 UNTERBRECHUNG - 5 Minuten

- 463 78. I: Okay, Aufnahme läuft wieder. Gut. Frage Fünf. Das zweite Modellbeispiel. Das Modell basiert
464 auf der Beschreibung des Heizprozesses, die aus den Schulungsunterlagen der Firma Siemens
465 entnommen wurde. Es handelt sich hierbei ebenfalls um eine Temperaturregelung für einen
466 Rührreaktor. Die Regelung wird in diesem Beispiel aber mit einem PID-Regler, einer
467 Handsteuerung sowie einem Pulsgenerator realisiert. Die Heizung erfolgt nicht über einen
468 Wärmetauscher, sondern über ein Heizelement. Weiters sind Verriegelungsbedingungen
469 definiert, die wir als Abbruchbedingungen nutzen. Als Basis für die Prozessmodellierung, wie
470 gesagt, wurden die Beschreibungen aus den Schulungsunterlagen für die Prozessmodellierung
471 mit Simatic PCS 7 herangezogen. Unser Prozessmodell wird mit einer automatischen Steuerung
472 modelliert, die mit Umschalten auf Handsteuerung aus dem Closed Loop System ausbricht. Das
473 heißt halt, der Zustand des Systems würde sich ändern und würde in einen anderen Prozess
474 übergehen. Wir gehen davon aus, dass das System bereits angelaufen ist und automatisch
475 gesteuert wird. Weiters wird der Prozess für nur einen Reaktor und nicht wie in den Unterlagen
476 für zwei Reaktoren modelliert. Das heißt, wir haben hier wieder eine Liste von Datenelementen,
477 die definiert sind, wobei hier halt auch Default-Werte hergenommen werden beziehungsweise
478 auch Grenzwerte für Abbruchbedingungen. Wir haben hier zum Beispiel T Reactor Max, das

steht für die maximale Temperatur, die im Kessel herrschen darf, und das sind in dem Fall sechzig Grad. Beziehungsweise haben wir auch einen Mindestfüllstand, also Level Reactor Min vorgegeben, mit 200 Milliliter. Und dann haben wir wieder Default-Werte für gewisse Zustände oder gewisse Bedingungen. Wir gehen halt wie gesagt von Operation Mode Automatic aus, beziehungsweise gehen wir davon aus, dass der Hauptschalter auf Ein ist, und nicht Off. Und alle anderen Werte sind wieder zum Berechnen des PID-Reglers gedacht. Diesmal sind wir in cancel und sequential, also wieder sequential, aber diesmal mit cancel. Das heißt, würden die Prozesse oder würden die Tasks in einer Kante zu lange brauchen, würden sie abgebrochen werden. Dieses Prozessmodell ist, wie Sie sehen, um einiges umfangreicher und komplexer, natürlich, weil wir auch mehrere Infos jetzt definiert haben, die hier modelliert werden. Wir haben insgesamt eins, zwei, drei, vier, fünf Zustandsabfragen. Wir haben einen Regler, und wir haben insgesamt eins, zwei, drei, vier, fünf Abbruchbedingungen definiert. Was wird nun gemessen oder was wird abgefragt? Wir haben die Temperatur des Reaktors, wir haben den Füllstand des Reaktors. Wir haben den Operation Mode, der abgefragt wird. Emergency Stop eventuell auch, und Main Switch. Reglermodell, PID-Controller. Wir haben den Pulsgenerator, das heißt die Umwandlung des Signals. Und wir haben dann wieder den Service Call, der das entsprechende Ergebnis einfach an den Aktor schickt. Das heißt, an/ umwandelt in den Input-Wert für das Heizelement, des Rührkessels. Die Abbruchbedingungen sind, wenn der Main Switch, also der Hauptschalter, auf Off geht, also ausgeschaltet wird. Wir haben Emergency Stop wieder auf true. Das war auch eine Abbruchbedingung. Und diesmal haben wir zum Beispiel, dass wir einen Vergleich haben, zwischen Temperatur, also aktuelle Temperatur des Reaktors, und der maximalen Temperatur. Das heißt, wenn die Temperatur des Reaktors über der maximalen Temperatur darüber ist, muss das System heruntergefahren werden. Und das wird durch diesen Service Call mit Script hier realisiert oder dargestellt, Switch to safe shutdown routine. Eine ähnliche Bedienung, nur halt hier, dass wir ÜBER einem gewissen Wert bleiben müssen, ist der Füllstand des Reaktors. Also der aktuelle Füllstand des Reaktors darf nicht unter die Mindestfüllstandsmenge, also das Mindestfüllstandsvolumen kommen, nämlich 200 Milliliter. Und dann haben wir noch den Wechsel in einen anderen Betriebsmodus. Wenn Operation Mode of manual geht, müssen wir auch in einen anderen Prozess zum Beispiel hinein, oder wir müssen vielleicht, also beziehungsweise wir brechen aus diesen Closed Loop Subsystem aus, weil sich die Zustände ändern. Und wir rufen eventuell einen neuen Prozess auf oder führen einfach noch eine Aufräumroutine aus, bevor wir in ein neues Modell wechseln. Wir haben hier wieder das mathematische Modell des PID-Controllers, ein bisschen erweitert im Vergleich zum PI-Controller. Ja, und um einiges komplexer dargestellt, das ganze, weil wir halt natürlich auch mehrere Infos jetzt haben für das System. Ja. Für dieses Modell würde ich Sie wieder bitten, nach den Kriterien, die wir vorhin schon gesehen haben, eine Bewertung zwischen Eins und Fünf abzugeben, wieder Eins, sehr schlecht, Fünf, sehr gut. Es geht wieder um Verständlichkeit, Übersichtlichkeit, Einfachheit, Logik und Erweiterbarkeit. #00:05:54-2#

79. B: Hm. (zustimmend) #00:05:56-6#

80. I: Fangen wir mit Verständlichkeit an. #00:05:59-3#

81. B: Ja, da würde ich eh sehr gut sagen. Also ich glaube, ich habe verstanden, was passiert. Genauso eigentlich auch für Übersichtlichkeit. Also, obwohl es jetzt schon recht groß ist, habe ich das Gefühl, dass ich es noch gut verstehe. Also, da würde ich auch sehr gut sagen. Also, ich denke mal mehr als eine A4-Seite wird dann schon ein bisschen unübersichtlich werden, aber das finde ich, kann man noch gut überblicken. (...) Einfachheit. Kann man das Modell noch einfacher darstellen? Ja, was ich mich gerade gefragt habe, es gibt jetzt immer zum Beispiel diese Abfrage Operation Mode. Also, wir fragen quasi ab, in welchem Modus befinden wir uns gerade, oder? Und falls der Modus manuell ist, dann wird diese Abbruchbedingung (...) #00:06:57-8#

82. I: Getriggert. #00:06:58-5#

83. B: /auf true gesetzt und dann gestartet. Aber, also, muss das jedesmal abgefragt werden? Das fühlt sich irgendwie ein bisschen mühsam an, wenn wir das in jedem Zyklus wieder fragen

531 müssen. Könnte man das nicht mit irgendeiner Push-Bedingung oder so etwas lösen? Dass quasi
532 in der Abbruchbedingung das direkt aktiviert wird, dass nicht immer der Zustand abgefragt
533 werden muss. Ist das möglich? #00:07:25-2#

534 84. I: Ja, man müsste jetzt nicht unbedingt abfragen, aktiv. Also, man könnte es wahrscheinlich von
535 außen auch einfach kommen lassen und müsste nicht immer wieder aktiv danach fragen.
536 #00:07:37-4#

537 85. B: Genau, das dachte ich mir. Aber das kommt wahrscheinlich dann immer auf den Prozess
538 darauf an. Aber wenn es um so etwas ganz Allgemeines geht wie, irgendetwas ist angeschalten
539 oder nicht, ist das vielleicht umständlich, dass jetzt alle zehn Sekunden extra abzufragen.
540 #00:07:57-6#

541 86. I: Okay. #00:08:05-3#

542 87. B: Ja, also vielleicht würde ich dann bei Einfachheit Vier sagen. Bei der Logik ist mir glaube ich
543 klar jetzt, in welcher Reihenfolge das passiert. Also die Messungen werden quasi alle wieder
544 parallel ausgeführt, dann wird geregelt und die Abbruchbedingungen sind immer da und warten.
545 Ich meine, das ist das gleiche wieder wie vorher, dass es vielleicht noch eine Spur übersichtlicher
546 für mich zumindest wäre, wenn es auf derselben Höhe wäre, was parallel abläuft. Also, wenn
547 alle Messbedingungen auf derselben Höhe wären. Dann die Kontrollbedingung. Und dann
548 eventuell die Abbruchbedingungen auch auf derselben Höhe wie die Messbedingungen, weil es
549 ja quasi immer aktiv ist. Das wird wahrscheinlich wieder ein bisschen was/ Dann wird es
550 wahrscheinlich eine Spur unübersichtlicher werden. #00:09:08-4#

551 88. I: Vielleicht von den Labels her? Hier seitlich die Beschreibungen? #00:09:15-1#

552 89. B: Ja, genau das wird dann, das wäre das Problem. Das ist dann super/ das wird sich dann nicht
553 mehr gut ausgehen. Ja, das stimmt schon. Das ist dann wahrscheinlich eh so der beste Weg. Ja,
554 aber im Endeffekt ist es eh übers/ also man weiß es ja eh, wenn man einmal weiß, okay, alles
555 Messbare läuft parallel, alle Abbruchbedingungen laufen parallel, das ist eigentlich schon klar,
556 was wann passiert. Beziehungsweise steht ja oben, wenn da sequential da steht, wissen wir, es
557 wird zuerst gemessen, dann der Kontroll-Task ausgeführt. Nein, also passt schon. Was war jetzt
558 die Frage? Die Logik? Ja, ich denke, ich würde dann hier auch Fünf, sehr gut, sagen. #00:10:01-7#

559 90. I: Okay. #00:10:05-0#

560 91. B: Bei der Erweiterbarkeit. (...) Ja, vielleicht das Einzige eben, was ich vorher schon erwähnt habe,
561 dass man vielleicht so also Push-Nachrichten auch verwenden kann, für die
562 Abbruchbedingungen, dass nicht alles explizit abgefragt werden muss. Dass Sachen, die vielleicht
563 für den Kontroll-Task nicht relevant sind, sondern nur für die Abbruchbedingungen, dass das
564 vielleicht (...) ja, eher so mit Push-Mitteilungen zum Beispiel geregelt wird. Das würde vielleicht
565 dann auch noch alles ein bisschen übersichtlicher gestalten. (...) Ja, also würde ich hier gut sagen.
566 Vier. #00:11:01-1#

567 92. I: Okay, danke. Nach den Erklärungen und den beiden Beispielen. Was wäre so Ihr erstes
568 Empfinden? Würden Sie aufgrund/ oder wären Sie aufgrund dieser Erweiterungen bereit, diese
569 Modellierungsmethoden in Ihrem Arbeitsalltag einzuführen, wenn Sie ein Modell eines
570 kontinuierlichen Prozesses entwickeln würden oder modellieren würden? #00:11:31-2#

571 93. B: Ja, auf jeden Fall. Also, ich mag BPMN sehr gerne. Ich finde, das ist sehr übersichtlich und ich
572 finde jetzt auch diese Erweiterungen sehr übersichtlich. Also ich würde es auf jeden Fall
573 verwenden. Also für mich finde ich es auch übersichtlicher als so diese klassischen
574 Regelungsdiagramme. Aber das liegt wahrscheinlich auch daran, dass ich das nie wirklich gelernt
575 habe. Ja. #00:11:59-2#

- 576 94. I: Frage Sieben. Wie gut beschreiben diese Erweiterungen Ihrer Meinung nach ein
577 Kontrollsystem für diese Beispiele? Also wie gut würden Sie sagen können Sie nachvollziehen,
578 wie so ein Regelungszyklus ablaufen würde, aufgrund dieser Erweiterungen? Für die beiden
579 Beispiele, die Sie jetzt gesehen haben. #00:12:25-0#
- 580 95. B: Ich denke, ich würde bei beidem sehr gut sagen. Also ich habe das Gefühl, ich habe beides
581 sehr gut verstanden und ich hätte es wahrscheinlich genauso modelliert. Oder genauso
582 dargestellt. #00:12:39-7#
- 583 96. I: Okay, würden Sie sagen, dass vielleicht noch etwas fehlt für eine detailliertere
584 Prozessbeschreibung? #00:12:50-5#
- 585 97. B: (...) Ich denke nicht. Also mir fällt jetzt nichts ein. Ich schaue mir gerade die Prozesse nochmal
586 an. Nein, also mir fehlt jetzt nichts. Mir würde jetzt nichts einfallen, was ich noch gerne dazu
587 modelliert hätte. #00:13:29-9#
- 588 98. I: Die nächste Frage ist ein bisschen, wenn Sie sich mit Regelungstechnik eventuell
589 auseinandergesetzt haben. Wenn Sie Erfahrung in der Regelungstechnik haben, was würden Sie
590 empfehlen, um diese Erweiterungen zu ergänzen, um Sie für Ingenieure attraktiver zu machen?
591 #00:13:50-5#
- 592 99. B: Ich glaube, dazu kann ich nicht wirklich etwas sagen. (lacht) Weil ich nicht viel Erfahrung in der
593 Regelungstechnik habe. Also, ich meine, ich persönlich würde es verwenden. Für mich bräuchte
594 es nicht attraktiver sein. (lacht) #00:14:07-9#
- 595 100. I: Okay, danke. Ich möchte zum Schluss noch etwas genauer auf die Modelle mit unseren
596 Erweiterungen von eben eingehen und Sie bitten, diese nochmals zu bewerten. Wir haben
597 wieder eine Skala von Eins bis Fünf, wobei wir hier jetzt statt Eigenschaften einige Fragen haben.
598 Und zwar eher allgemein gestellt. Die erste Frage wäre nach diesen Prozessbeispielen, wie
599 einfach ist in den gezeigten Modellen Ihrer Meinung nach nachzuvollziehen, dass die einzelnen
600 Abläufe parallel und unabhängig voneinander laufen? #00:14:46-5#
- 601 101. B: (...) Soll ich es gleich beantworten oder gehen wir zuerst alle Fragen durch? #00:14:56-3#
- 602 102. I: Ach so, wie Sie möchten. Gehen wir einmal alle Fragen durch. Stimmt, das habe ich ja bisher
603 auch so gemacht. Wie einfach ist es zu definieren, wann eine Anpassung am System erfolgt? Wie
604 einfach ist es, die maximale Dauer einer Anpassung zu definieren? Wie einfach ist es zu
605 definieren, unter welchen Bedingungen sämtliche repetitiven Aufgaben beendet werden sollen?
606 Wie einfach ist es zu definieren, dass danach Aufräumaufgaben einmalig zu erfolgen haben? Und
607 wie einfach ist es, komplexe Abläufe im Kontext von kontinuierlichen Prozessen mit diesen
608 Erweiterungen zu beschreiben? Das heißt, das zielt doch ein bisschen eher auf das komplexere
609 von beiden Beispielen ab. Wie gesagt, erste Frage wäre, wie einfach ist in den gezeigten
610 Modellen nachzuvollziehen, dass die einzelnen Abläufe parallel und unabhängig voneinander
611 laufen? #00:15:51-1#
- 612 103. B: Also sehr einfach, würde ich sagen. Das ist ja einfach die einzelnen Stränge, also Fünf. Sehr gut.
613 #00:15:59-8#
- 614 104. I: Zweite Frage. Wie einfach ist es zu definieren, Ihrer Meinung nach, wann eine Anpassung am
615 System erfolgt? #00:16:08-5#
- 616 105. B: Mit den Anpassungen am System ist jetzt diese Kontroll-Task gemeint, oder? Also diese
617 Kontroll/ #00:16:14-2#
- 618 106. I: Genau, ja. Das sind die Regulierungen, Anpassungen. Ja, genau. #00:16:17-6#
-

619 107. B: Naja, auch sehr einfach eigentlich. Man muss ja nur dann quasi die unterschiedlichen Tasks
620 definieren. Also halt quasi wann ein Script-Task ausgeführt wird, oder ein manueller Task. Also,
621 Fünf, sehr gut. #00:16:36-9#

622 108. I: Wie einfach ist es, die maximale Dauer einer Anpassung zu definieren? Also die Zykluszeit
623 quasi, die maximale. #00:16:49-3#

624 109. B: Die Zykluszeit ist definiert über die Messrate, oder? Also die/ #00:16:53-6#

625 110. I: Die Frequenz. #00:16:54-8#

626 111. B: Ja, das ist dann sehr einfach, weil im Endeffekt muss man ja nur die Zeitangabe ändern. Also
627 auch Fünf, sehr gut. #00:17:03-4#

628 112. I: Wie einfach ist es zu definieren, unter welchen Bedingungen sämtliche repetitiven Aufgaben
629 beendet werden sollen? #00:17:13-5#

630 113. B: Das wäre dann einfach nur das Hinzufügen von einer Abbruchbedingung. Also. Oder? Habe ich
631 das richtig verstanden? #00:17:21-9#

632 114. I: Ja. #00:17:22-7#

633 115. B: Ja, also dann auch Fünf, sehr gut. #00:17:25-0#

634 116. I: Dann, wie einfach ist es zu definieren, dass danach Aufräumaufgaben einmalig zu erfolgen
635 haben? #00:17:34-1#

636 117. B: Ja, genau so auch sehr einfach. Also da fügen wir dann einfach nur Tasks hinzu oder Events
637 hinzu. Also auch Fünf, sehr gut. #00:17:43-8#

638 118. I: Und dann die letzte Frage. Wie einfach ist es, komplexe Abläufe im Kontext von
639 kontinuierlichen Prozessen mit diesen Erweiterungen zu beschreiben? #00:17:55-0#

640 119. B: Also, ich glaube das Beschreiben ist einfach. Das ist nur die Frage, ab wann wird es nicht mehr
641 verständlich? Aber das ist halt, glaube ich, immer so bei Prozessen, jetzt, egal ob kontinuierlich
642 oder normal. Ja, ich glaube, da würde ich Vier sagen. Also ich glaube zu beschreiben ist es
643 einfach. Man muss wahrscheinlich nur darauf aufpassen, dass man Nutzer dann nicht
644 überfordert, mit einem zu komplexen Modell. Also da muss man wahrscheinlich schauen, was
645 man in Subprozesse unterbringt. Aber das Problem hat man ja eh immer bei der Modellierung.
646 #00:18:40-4#

647 120. I: Würden Sie sagen, dass das bei Ihren sonstigen Tätigkeiten auch immer wieder eine
648 Herausforderung ist? #00:18:48-7#

649 121. B: Ja, immer. Also, das ist immer die große Frage. Auf welcher Ebene, wie genau modelliert man
650 es? Was lagert man aus in Subprozesse? Ja, also im Endeffekt kann man hier auch Fünf, sehr gut,
651 nehmen, weil das ist ja eine Herausforderung, die allgemein bei der Prozessmodellierung
652 vorhanden ist, unabhängig jetzt von kontinuierlichen Prozessen. #00:19:17-1#

653 122. I: Okay, Dankeschön. Das waren meine letzten Fragen. Danke, dass Sie sich Zeit genommen
654 haben für dieses Interview. Und wenn Sie Feedback zum Leitfaden oder generell zur
655 Formulierung der Fragen haben und dergleichen, ich bin immer dankbar für jeweiliges Feedback.
656 Also. Vielen Dank. Danke, dass Sie sich Zeit genommen haben. Ich werde die Aufnahme gleich
657 beenden.
