

Transcript - Group 3 "Modelers", Interview 3

I ... Interviewer (BLINDED)

B ... Expert

(Unv.)... Incomprehensible passage

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

() ... Comment

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

Transcript

- 1 1. I: Okay. Aufnahme läuft. So. 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 zur Verfahrenstechnik, Fertigungstechnik oder
5 Prozessmodellierung erklären? Ich möchte Sie aber bitten, dabei nicht Ihren Namen zu nennen,
6 sondern nur die folgenden Informationen. Nämlich Ihre Berufsbezeichnung und Umschreibung
7 des Arbeitgebers, Basis Ihrer Expertise zum Forschungsthema, Ausbildung beziehungsweise
8 fachlicher Hintergrund und schließlich Ihre Berufserfahrung. Bitte. #00:00:49-2#
- 9 2. B: [REDACTED]
10 [REDACTED]
11 [REDACTED]
12 [REDACTED]
13 [REDACTED]
14 [REDACTED]
15 [REDACTED] Genau. #00:01:36-1#
- 16 3. I: Okay, wunderbar, danke. Dann möchte ich noch kurz bevor wir zu den Fragen kommen, eine
17 allgemeine Einführung zum Thema geben. Und zwar. Unsere Forschung konzentriert sich auf die
18 Entwicklung einer Methodik, um kontinuierliche Prozesse in BPMN darzustellen und sie in einer
19 Workflow Engine ausführbar zu machen. Für diese Aufgabe haben wir an BPMN-Erweiterungen
20 für kontinuierliche Prozesse gearbeitet. Warum kontinuierliche Prozesse? Weil diskrete Prozesse
21 bereits in anderen Forschungsarbeiten behandelt wurden und nicht die gleichen Schwierigkeiten
22 bei der korrekten Darstellung mittels BPMN aufweisen. BPMN ist bereits ein weit verbreiteter
23 Standard im Business Process Management und hat seinen Weg in die Fertigung gefunden.
24 Diskrete Fertigungsprozesse können bereits mit BPMN 2.0 modelliert werden. Im Grunde wollen
25 wir eine Methodik einführen, um solche Prozesse so darzustellen, dass sie von jeder Person in
26 einem Unternehmen, vom Ingenieur bis zum Manager, verstanden werden können. Dies könnte
27 durch die Verwendung dieser Notation erreicht werden. Ein weiterer Vorteil ist auch, dass es
28 bereits eine Reihe von Workflow Engines gibt. Anwendungen, die die Ausführung dieser
29 Prozessmodelle auf der Grundlage der für jedes Symbol implementierten Logik ermöglichen. Wir
30 arbeiten mit einer webbasierten Anwendung, die erweiterbar ist und mehrere
31 Kommunikationsschnittstellen implementiert hat. Ein weiterer Vorteil ist daher die
32 Interoperabilität in diesem Zusammenhang im Vergleich zu anderen proprietären, starren
33 Software-Anwendungen. Wir wollen herausfinden, ob diese Technik auch für die
34 Implementierung von digitalen Abbildern eingesetzt werden kann. Da digitale Abbilder dazu
35 dienen, ein physikalisches System oder einen Prozess in digitaler Form darzustellen, meist
36 anhand von Daten oder mathematischen Modellen, mussten wir einen Weg finden, den Ablauf
37 von kontinuierlichen Prozessen wie sie aus der Prozessindustrie bekannt sind, darzustellen. Aus
38 diesem Grund haben wir uns auf die Modellierung von Regelkreisen konzentriert. Die
39 Prozessmodelle sollen durch BPMN für Personen mit unterschiedlichem Hintergrund leicht
40 verständlich sein. Die Interviews werden geführt, um herauszufinden, wie Prozess- und
41 Regelungstechnik und Techniken aus der Business-Process-Modellierung kombiniert werden
42 können und wie erste Ergebnisse von Experten wie Ihnen wahrgenommen werden. Außerdem
43 wollen wir herausfinden, ob es Schwachstellen gibt, die von Experten identifiziert wurden und
44 wie wir diese beseitigen können. Noch zu Beginn möchte ich vielleicht drei Begriffe klären. Und

45 zwar. Was verstehen wir unter dem Begriff digitales Abbild oder digitaler Zwilling? Es gibt
46 verschiedene Methoden, Dinge aus der echten Welt, zum Beispiel echte Maschinen, zu
47 simulieren. Teils merkt man aber, dass es mehr Parameter brauchen würde als bei normalen
48 Simulationsmethoden, um eine Maschine vollkommen so abzubilden, wie sie sich in der Realität
49 verhält. Bei einem digitalen Zwilling wird versucht, möglichst nahe an das reale Verhalten einer
50 Maschine oder anderer Objekte heranzukommen. Das soll dazu führen, dass wenn etwas
51 getriggert wird bei einer echten Maschine, der digitale Zwilling das gleiche oder ein möglichst
52 ähnliches Verhalten zeigt. Zweiter Begriff. Was verstehen wir unter kontinuierlichen Prozessen?
53 Kontinuierliche Prozesse möchte ich vielleicht mit Beispielen erklären. Und zwar, wenn man
54 Bierbrauen her nimmt, gibt es zwei Möglichkeiten. Die diskrete, also nicht-kontinuierliche
55 Variante wäre, wenn man in einen geschlossenen Kessel die Zutaten hinein gibt, zehn Liter
56 Wasser und so weiter, und den Brauprozess einfach schrittweise ablaufen lässt. Am Ende kommt
57 dann eine begrenzte Menge an Bier heraus. Die andere kontinuierliche Variante wäre, wenn
58 man keinen vollkommenen abgeschlossenen Kessel hat, sondern miteinander verbundene
59 Kessel, bei denen immer wieder Zutaten zugefügt werden und immer wieder Bier entnommen
60 wird. Das geht dann die ganze Zeit so, so dass man nicht nachvollziehen kann, welche Liter
61 Wasser zu welchem Liter Bier gehört. Dabei läuft ein Teilprozess im ersten Kessel ab, während
62 gleichzeitig im letzten Kessel der letzte Prozessschritt stattfindet, bevor das Bier fertig wird. Und
63 der letzte Begriff. Geschlossene Regelkreise. Ein geschlossener Regelkreis ist jene Logik in Form
64 von Hardware oder Software, die das kontinuierliche Bierbrauen ermöglicht. Wenn man einen
65 Prozess wie das kontinuierliche Bierbrauen hat, muss man schauen, wie man schlechtes Bier
66 vermeidet, während der Prozess läuft. Man möchte die Qualität auf einem gewissen Punkt
67 halten. Beim schrittweisen Bierbrauen hat man nur die zehn Liter, bei denen etwas schief gehen
68 kann, und mit den nächsten zehn Litern macht man es dann besser. Aber was ist, wenn man die
69 Brauanlage dauernd laufen lässt und wenn ständig Bier austritt? Dann muss man
70 währenddessen den Prozess überprüfen und schauen, dass man die gute Qualität des Bieres
71 erhält. Das heißt, man testet und misst Werte, die die Qualität beschreiben, überprüft, wie sich
72 die Werte von optimalen Werten unterscheiden und reagiert dann entsprechend. Stimmt etwas
73 beispielsweise beim Zucker- oder Alkoholgehalt nicht, muss man das Mischverhältnis anpassen.
74 Das heißt, in einem geschlossenen Regelkreis werden während der Prozess läuft gewisse Werte
75 überprüft. Diese werden mit optimalen Werten verglichen und je nach Abweichung reagiert das
76 System dann darauf. Okay. Aus Informatiksicht bestehen kontinuierliche Prozesse aus einer sich
77 ständigen wiederholenden Abfolge von Zustandsabfragen und Regulierungen. Zustandsabfragen
78 und Regulierungen sind jeweils traditionelle Code-Stücke, die sich auf Sensoren oder Aktoren
79 beziehen. Um solche kontinuierlichen Prozesse konsistent formal zu beschreiben, zu modellieren
80 und in weiterer Folge ausführen zu können, haben wir folgende Merkmale identifiziert. Und jetzt
81 kommen wir auch zur ersten Frage. Wir haben hier eine Liste von Merkmalen, insgesamt sechs,
82 die ich kurz auflisten werde und ich würde Sie bitten, dass Sie die einzelnen Merkmale für
83 wichtig oder unwichtig klassifizieren, also einschätzen von Ihrer Seite aus und vielleicht eine
84 kurze Begründung dafür abgeben. Das erste Merkmal wäre, verschiedene Zustandsabfragen-
85 und Regulierungskombinationen sind unabhängig und können parallel ablaufen. Zweitens.
86 Regulierungen folgen immer auf Zustandsabfragen. Nummer Drei. Die Dauer von jeder
87 Zustandsabfrage- und Regulierungskombination ist beschränkt. Nummer Vier. Wenn
88 Zustandsabfragen gewisse Ergebnisse liefern, wird das System beendet. Nummer Fünf. Bevor
89 das System beendet wird, muss es in einen konsistenten Zustand gebracht werden. Und
90 schließlich Nummer Sechs. Das resultierende System soll für Menschen verständlich sein. Fangen
91 wir gleich mal mit Punkt Nummer Eins an. Verschiedene Zustandsabfrage- und
92 Regulierungskombinationen sind unabhängig und können parallel ablaufen. Würden Sie im
93 Kontext von kontinuierlichen Prozessen sagen, das ist wichtig oder unwichtig? Und warum?
94 #00:08:41-3#

- 95 4. B: Gut. Tja, das ist eine sehr, sehr gute Frage. Also ich würde schon sagen, dass zum Beispiel der
96 Zustand zu Beginn vielleicht dann doch wieder Einfluss auf einen späteren Zustand haben könnte.
97 Deswegen weiß ich nicht, wie sehr das dann quasi unabhängig voneinander ist. Also wenn ich
98 jetzt beim Bierbrauen irgendwie, wenn ich das richtig verstanden habe, quasi Zucker hinein gebe,
99 dann schaut das Endprodukt, ja/ wird ja dann dadurch beeinflusst, dass der Zuckergehalt
100 irgendwie wieder erhöht wird, richtig? #00:09:20-8#
-

- 101 5. I: Also, ja, durch den chemischen Prozess würde Zucker entstehen und man müsste dann am
102 Anfang von den Zutaten das Mischverhältnis anpassen. Genau. #00:09:29-2#
- 103 6. B: Genau. Das heißt, zum Schluss hinaus kann sich das ja dann schon irgendwie auch wieder
104 beeinflussen, richtig? #00:09:36-4#
- 105 7. I: Hm. (zustimmend) #00:09:36-8#
- 106 8. B: Das heißt, ich bin mir nicht so ganz sicher, ob es komplett unabhängig voneinander wäre. Die
107 Abfrage könnte aber sicherlich immer wieder parallel laufen. Also ich sehe da jetzt keinen Grund,
108 warum man quasi am Anfang des Prozesses und zum Ende des Prozesses nicht parallele
109 Abfragen machen könnte, wie denn aktuell der Zustand des Bieres wäre, oder? Also da wäre
110 meine Meinung, also/ Ist es dann wichtig oder unwichtig? Parallel wäre dann wichtig, dass man
111 es machen kann, die Abfrage. Unabhängig müsste man sich wahrscheinlich von Prozess zu
112 Prozess tatsächlich anschauen. Also könnte wichtig sein, könnte aber auch vielleicht weniger
113 wichtig sein, dass es jetzt unabhängig voneinander laufen kann. Genau. #00:10:35-3#
- 114 9. I: Also ganz kurz nur. Ich habe mich vorhin versprochen. Ich meinte natürlich, wenn man Zucker
115 am Anfang hinein gibt, entsteht dann aufgrund des chemischen Prozesses nachher der Alkohol
116 und das heißt natürlich entsprechend muss man es anpassen, das stimmt. Ganz kurz nur. Und
117 was hier gemeint ist, ist grundsätzlich, wenn man nicht nur einen Wert überprüft, sondern
118 mehrere verschiedene Werte überprüft. //B: Okay. // Dass die eigentlich unabhängig
119 voneinander gemessen oder erfasst werden können. #00:11:08-6#
- 120 10. B: Ach so, ja, das ist natürlich schon wichtig, ja. Genau. Das wäre schon wichtig, dass man das
121 dann wirklich zu jederzeit eigentlich und auch parallel eben abfragen kann, was gerade passiert.
122 Ja, genau. #00:11:21-0#
- 123 11. I: Okay, super, danke. Punkt Zwei. Regulierungen folgen immer auf Zustandsabfragen. Das ist, so
124 formuliert, ich weiß, ein bisschen schwierig zu deuten, aber wir haben es so gemeint, dass man
125 erst den Zustand erfassen müsste oder zumindest den Zustand wissen müsste, bevor man eine
126 Regulierung starten kann. #00:11:43-5#
- 127 12. B: Ja, der Meinung bin ich auch. Also, das wäre auf jeden Fall wichtig, finde ich, an der Stelle.
128 Weil wie kann ich wissen, was ich regulieren muss, wenn ich den aktuellen Zustand des Systems
129 ja nicht kenne? Das ist irgendwie schon klar, dass das wichtig ist, an der Stelle. Und dass es dann
130 auch immer darauf folgen sollte, ergibt sich dann logischerweise eigentlich. #00:12:08-9#
- 131 13. I: Okay. Nummer Drei. Die Dauer von jeder Zustandsabfragen- und Regulierungskombination ist
132 beschränkt. Man kann aber hier auch das Wort beschränkt durch definiert ersetzen, wenn Ihnen
133 das leichter fällt zum Deuten. #00:12:26-1#
- 134 14. B: Das heißt, sozusagen beschränkt. Ich habe nur zehn Sekunden Zeit, um dann zu reagieren,
135 oder? Okay. Ja, kommt wahrscheinlich auch auf die Art des Prozesses darauf an, oder? Also wie
136 schnelllebig ist dieser Prozess? Das heißt, wie schnell muss ich sozusagen in das System
137 eingreifen können und dann sollte die Dauer wohl schon beschränkt sein. Wäre dann wichtig.
138 #00:13:06-0#
- 139 15. I: Okay, super. Punkt Vier. Wenn Zustandsabfragen gewisse Ergebnisse liefern, wird das System
140 beendet. #00:13:15-2#
- 141 16. B: Bedeutet das, sozusagen, dass wenn was komplett aus dem Ruder läuft, dann habe ich immer
142 die Möglichkeit, den kompletten Prozess des Bierbrauens abubrechen? Verstehe ich das richtig
143 an der Stelle? #00:13:30-5#
- 144 17. I: Hm. (zustimmend) #00:13:32-0#
-

- 145 18. B: Ja? Okay. Ja, gut. Wäre wahrscheinlich schon irgendwie auch für die Produktion an sich
146 wichtig. Ist halt die Frage, ob das gewünscht ist. Weil dann könnte man ja das ganze Bier quasi
147 wegschmeißen. Aber wenn es für die Produktion wichtig ist, dass man quasi auch generell immer
148 komplett abbrechen kann, den ganzen Prozess, dann wäre natürlich vier, Punkt Vier, auch
149 generell wichtig, ja, für die Evaluierung. #00:14:00-1#
- 150 19. I: Im Kontext von kontinuierlichen Prozessen ist hier natürlich auch der Fokus darauf gelegt, dass
151 man aus dieser Kontinuität ausbrechen kann. Darum geht es auch. Also //B: Okay. // es heißt
152 eher, dass wenn man jetzt einfach nur einen kontinuierlichen Prozess modelliert hätte, dass man
153 zumindest aus dieser Loop ausbrechen kann. Darum geht das auch. #00:14:23-6#
- 154 20. B: Ja, dann wäre es schon wichtig. #00:14:26-3#
- 155 21. I: Punkt Fünf. Bevor das System beendet wird, muss es in einen konsistenten Zustand gebracht
156 werden. #00:14:35-8#
- 157 22. B: Ich denke auch, dass das wichtig ist. Gerade also aus Prozesssicht muss man ja dann auch
158 wissen, wie man weitermachen kann, wieder, wenn mal was abgebrochen worden ist. Deswegen
159 muss man ja den Zustand vielleicht auch kennen. Das sollte dann irgendwie auch konsistent im
160 Vergleich zum übergeordneten Prozess sein. Also. Genau. Also wäre auf jeden Fall wichtig, ja.
161 #00:15:02-0#
- 162 23. I: Und der letzte Punkt. Das resultierende System soll für Menschen verständlich sein.
163 #00:15:08-4#
- 164 24. B: Ja, finde ich auch sehr, sehr wichtig, dass das generell von Menschen interpretiert werden
165 kann. Ganz einfach, weil die ja sicherlich auch an so einem Prozess beteiligt sind und dann ja
166 vielleicht auch etwaige Fehler, die vielleicht aufgetreten sind, ja dann auch vielleicht
167 gegensteuern müssen, besondere Aktionen vielleicht nochmal treffen müssen. Genau. Ja, auf
168 jeden Fall wichtig, nachvollziehbar sein. #00:15:38-0#
- 169 25. I: Okay, super. Dankeschön. Dann würde ich gleich zu Frage Zwei überleiten. Und nämlich. Wir
170 haben jetzt diese Liste gesehen. Wir sind jetzt die einzelnen Punkte durchgegangen. Können Sie
171 jetzt grafische Eigenschaften nennen, die Sie für die Modellierung kontinuierlicher Prozesse
172 wichtig finden und ergeben sich daraus vielleicht Merkmale, die wir hier in dieser Liste vergessen
173 haben? Also, hier geht es wirklich um die grafische Modellierung. #00:16:09-5#
- 174 26. B: In Bezug auf BPMN, schon? #00:16:13-8#
- 175 27. I: Ja, BPMN und kontinuierliche Prozesse. Also, wenn Sie andere Möglichkeiten kennen,
176 kontinuierliche Prozesse zu modellieren, können Sie auf die gerne auch eingehen. Also es soll
177 eine eher allgemeinere Frage auch sein. Wenn Sie eher mit BPMN arbeiten, dann BPMN im
178 Fokus. #00:16:31-8#
- 179 28. B: Würde mir jetzt ad hoc nichts einfallen, was vergessen worden ist. (...) Naja, vielleicht müsste
180 man ja quasi auch/ man kann ja auch zwischen manuellen Tasks zum Beispiel unterscheiden in
181 BPMN. Vielleicht wäre so etwas wichtig. Dass man so etwas auch darstellen kann, von wem so
182 ein Schritt dann ausgeführt wird? #00:17:18-2#
- 183 29. I: Hm. (zustimmend) Also, Sie meinen eine bestimmte Rolle, vielleicht? #00:17:25-0#
- 184 30. B: Ja, genau. Eben auch gerade auf den Punkt Sechs zum Beispiel bezogen. Dass das ja für
185 Menschen auch verständlich sein soll, wenn ein Mensch eine bestimmte Aktion dann ausführen
186 muss, irgendwas messen muss zum Beispiel kontinuierlich oder ob das kontinuierlich eine
187 Maschine macht, wäre vielleicht dann auch schon ein Unterschied. Könnte auch interessant sein.
188 #00:17:48-3#
-

- 189 31. I: Hm. (zustimmend) Okay. Guter Punkt. Wenn wir dann weitergehen, wenn wir so über das
190 Modellieren von kontinuierlichen Prozessen nachdenken, würden Ihnen vielleicht
191 Herausforderungen einfallen, die man hierbei haben könnte, auf die man stoßen könnte, wenn
192 man kontinuierliche Prozesse modellieren möchte? Egal mit welcher Notation, also mit welcher
193 Möglichkeit Sie normalerweise arbeiten würden. #00:18:24-5#
- 194 32. B: Ja, wenn ich quasi den Zustand ja abfrage, eines Systems, dann muss ich ja auf Basis des
195 Zustands wieder eine Entscheidung treffen. Das heißt, irgendwo habe ich ja eigentlich immer
196 eine Entscheidung dann drinnen an solchen Punkten und das kann natürlich glaube ich dann
197 schon sehr komplex werden. Je nachdem, was man dann sozusagen als Aktionen machen muss
198 auf Basis des bestimmten Zustands des Systems. Und da immer wieder an den Punkt
199 zurückgehen zu können, stelle ich mir sehr schwierig vor. Auch bei der Modellierung. Und kann
200 wirklich sehr komplex werden, glaube ich. Also, es könnte auf jeden Fall eine große
201 Herausforderung sein. #00:19:06-1#
- 202 33. I: Also meinen Sie hier zum Beispiel auch die Reaktionen des Systems abzubilden, //B: Ja, genau.
203 // die verschiedensten Fälle? Hm. (zustimmend) #00:19:18-7#
- 204 34. B: Genau, weil das muss man ja eigentlich, wenn man Punkt Sechs quasi berücksichtigen möchte,
205 dass eben das System ja auch für die Menschen dann verständlich sein soll, muss ich das ja auch
206 irgendwie abbilden können vorher. Und wenn ich mir jetzt vorstelle, an jedem Punkt des
207 Prozesses frage ich einen bestimmten Zustand ab und muss dann wieder an einem bestimmten
208 anderen Punkt vielleicht zurück springen oder so, um gegensteuern zu können, dann, glaube ich,
209 ist das schon sehr herausfordernd. (lacht) #00:19:47-1#
- 210 35. I: Okay, ja. Kommen wir zu den Erweiterungen. Ich werde Ihnen Prozesse zeigen, die mit BPMN
211 2.0 und mit unseren Erweiterungen modelliert wurden. Die Erweiterungen sollen zum einen
212 vordefinierte Modellierungskonventionen für in der Prozess- und Steuerungstechnik übliche
213 Routinen bereitstellen. Und zum anderen helfen, die Unterschiede zwischen den parallelen
214 Pfaden in den Prozessmodellen zu visualisieren. Die Prozesse werden in der Cloud Process
215 Execution Engine modelliert. Und soweit ich weiß kennen Sie die [REDACTED] grundsätzlich.
216 #00:20:26-1#
- 217 36. B: Genau, aber es wäre vielleicht nochmal ganz gut, wenn wir einfach nochmal der Reihe nach
218 durchgehen. #00:20:32-7#
- 219 37. I: Okay, gut. Also wichtig für die [REDACTED] den Aufbau kennen Sie, die GUI (Graphical User Interface)
220 kennen Sie grundsätzlich. // B: Hm. (zustimmend) // Dass es die Möglichkeit gibt, Datenelemente
221 zu definieren //B: Genau. //, die Endpunkte, die für verschiedene Service Calls notwendig sind
222 und so weiter. Wir haben ein Fenster für den Prozessgraphen, der sich/ Also man kann die [REDACTED]
223 natürlich grundsätzlich nicht nur als Ausführungs-Tool nutzen, sondern auch als Editor. Man
224 kann in diesem Fenster auch einen komplett neuen Prozess erstellen, modellieren, und //B: Ja.
225 // hat dann die Möglichkeit, zu den einzelnen Elementen auch auf der rechten Seite dann die
226 Attribute zu definieren. //B: Ja. // Und diese ganze Darstellungsweise, die haben wir auch im
227 weiteren Verlauf des Leitfadens mit Screenshots mit einzelnen Fenstern, einzelnen Grafiken,
228 auch dargestellt. //B: Okay. // Und für das grundsätzliche Verständnis der Beispiele, die Sie
229 nachher sehen werden, die Prozessmodelle, die Sie dann sehen werden, ist es auch notwendig,
230 dass Sie grundsätzlich wissen, was Scripts sind, Service Calls und Service Calls mit Scripts. Also
231 Erweiterungen für BPMN, die bereits schon seit längerem implementiert sind und die in der
232 [REDACTED] genutzt werden. //B: Ja. // Und bei Scripts geht es grundsätzlich nur darum, dass
233 Code-Abschnitte ausgeführt werden, die hier eingefügt sind. Das heißt, im Grunde einfach
234 standardmäßig wie ein Script ausgeführt werden würde. Service Calls bieten die Möglichkeit,
235 HTTP Requests abzusetzen, HTTP Calls abzusetzen. Also Get Requests oder Post, Put, was auch
236 immer, über die Endpunkte, die definiert werden können in der [REDACTED]. Und dann gibt es noch die
237 Kombination aus beidem, also Service Calls mit Script. Dass man ein Script nutzen kann, um

238 eventuell ein Ergebnis, dass man aus einem Service Call bekommt, das jeweilige Result, einfach
239 weiterzuverarbeiten. //B: Genau. // (unv.) #00:22:41-8#

240 38. B: Für den Fall wäre ja dann der letzte Punkt eigentlich sehr interessant. #00:22:45-5#

241 39. I: Hm. (zustimmend) #00:22:46-1#

242 40. B: Ja. Okay. #00:22:48-5#

243 41. I: Okay, passt. Ja, und dann kommen wir noch zu den Erweiterungen, die sich im Zuge DIESER
244 Arbeit ergeben haben. Fangen wir gleich an mit der ersten Erweiterung, ein neues Gateway, das
245 wir definiert haben. Und zwar das Closed Loop Subsystem Gateway. Das Gateway ist eine
246 Kombination aus einem inklusiven und einem ereignisbasierten Gateway. Es enthält
247 Verzweigungen beziehungsweise Kanten, Lanes, die für die Zustandsabfragen- und
248 Regulierungsphasen des Zyklus ausgelöst werden, sowie Verzweigungen, die beim Empfang von
249 Abbruchereignissen ausgeführt werden. Also Zustandsabfragen, Regulierungen,
250 Abbruchereignisse. Die Ereignisse und Tasks in den einzelnen Kanten sind unabhängig
251 voneinander. Damit erfüllen wir das erste der oben genannten Features, dass einzelne Verläufe
252 unabhängig voneinander sind und sie parallel ausgeführt werden. Das Gateway ermöglicht
253 außerdem die Definition der Intervalldauer jedes Zyklus sowie von Überschreitungsbedingungen.
254 Zum Beispiel dargestellt durch wait oder cancel. Und der Ausführungsreihenfolge für
255 Zustandsabfragen und Regulierungen, beziehungsweise könnte man hier auch sagen, Mess- und
256 Steuerungsaufgaben. Was kann man hier definieren? Die zwei wichtigen Attribute sind einerseits
257 Interval duration overrun. Hier kann man cancel oder wait definieren. Und andererseits Measure
258 control cycle execution. Hier kann man parallel oder sequentiell definieren. Wie funktionieren
259 die beiden? Wenn wait gewählt wird, beginnt die nächste Iteration, wenn alle Verzweigungen
260 beendet sind und die festgelegte Intervalldauer erreicht ist. Bei cancel definiert die
261 Intervalldauer genau die Zeit, in der jeder Zweig zu beenden ist. Wenn die Tasks in einem Zweig
262 schneller beendet werden, wird der Zweig warten. Wenn noch nicht alle Tasks beendet sind,
263 werden sie abgebrochen. Bei parallel werden die Tasks nach Measure und Control Events
264 parallel ausgeführt. Bei sequential werden die Tasks nach Control Events erst ausgeführt,
265 nachdem alle Tasks nach Measure Events beendet sind. Das heißt, erst nachdem alle
266 Zustandsabfragen erledigt sind, können dann die entsprechenden Regulierungen ausgeführt
267 werden. Zu den einzelnen Events. In einem Closed Loop Subsystem werden spezifische
268 Ereignisse erwartet, spezifische Events erwartet, die in eine der drei folgenden Kategorien fallen.
269 Events für Zustandsabfragen oder Messungen, Events für Regulierungen und Events für die
270 Unterbrechung des Closed Loop Subsystems. Es gibt für jede Ereigniskategorie zumindest eine
271 Kante, die vom Gateway ausgeht. Die Kanten zeigen an, welche Tasks nebeneinander ablaufen.
272 Sobald diese Ereignisse eintreten, werden auch die Tasks, die in den Kanten danach angeordnet
273 sind, ausgeführt. Sie sehen jetzt hier gleich ein Bild eines Closed Loop Subsystems, in dem nur
274 die Events, die Ereignisse der drei Kategorien ohne darauf folgende Tasks modelliert sind. Das
275 heißt, wenn man ein Closed Loop Subsystem Gateway erstellen würde in einem Graphen oder
276 einfügen würde in einen Graphen, würde standardmäßig ein Measure, Control und ein
277 Abbruch-Event hier in drei parallelen Strängen auch erscheinen. Die drei Ereigniskategorien, die
278 wir definiert haben, sind, wie bereits erwähnt. Wir haben Measure, empfängt Events für die
279 Ausführung von Tasks in Messzyklen, Zustandsabfragezyklen. Control, empfängt Events für die
280 Ausführung von Tasks in Regelzyklen, Regulierungen. Und dann schließlich Cancel, empfängt
281 Events für das Abbrechen von Closed-Loop-Systemen. Diese Symbole geben den Zweck der
282 nachfolgenden Tasks an. Diese Tasks werden nur ausgeführt, wenn die Ereignisse ausgelöst
283 werden. Das bedeutet, dass das Messereignis angibt, dass die nachfolgenden Symbole nur
284 Messabläufe beziehungsweise Zustandsabfragen anzeigen. Das gleiche gilt für Regulierungs-
285 beziehungsweise Kontroll- und Abbruchereignisse. Also, Regulierungen sind gleich
286 Kontrollereignisse. Für Zustandsabfragen und Regulierungen können wir eine Zykluszeit
287 definieren. Dadurch kann die Dauer von Anpassungen im System definiert werden. Je nachdem,
288 ob das Closed Loop Subsystem einen parallelen oder sequentiellen oder einen Wait- oder
289 Cancel-Ansatz verfolgt, läuft die Ausführung unterschiedlich. Mit diesen Bedingungen kann man
290 definieren, inwiefern Anpassungen beim System erfolgen. In den Grafiken hier sehen Sie ein

Closed Loop Subsystem mit einem Task für eine Messung. In diesem Fall wird das Ereignis für die Messung alle zehn Sekunden getriggert, angegeben in Hertz. Danach wird der Wert V 1 geholt beziehungsweise gemessen. Wait bedeutet hier, dass ein neuer Zyklus erst startet, wenn die Messung erfolgt. Das heißt, der Prozess in dieser Kante abgeschlossen ist. Mit cancel wird nach zehn Sekunden automatisch der neue Zyklus gestartet. Das heißt, wir hätten hier jetzt nur mal einen Service Call definiert, mit dem Label Get process value V 1, und einmal in wait, einmal in cancel dargestellt. Was kann man jetzt hier beim Measure definieren? Wir haben das Attribut Interval frequency in Hertz. Das heißt alle zehn Sekunden, also in Hertz der Kehrwert angegeben, und dann kann man hier auch angeben, welcher Wert, welcher Value sich ändern sollte, also Values expected to change. Das heißt, man kann hier auch beliebig viele Werte hinzufügen, je nachdem, wie viele Werte im Zuge dieses weiteren Verlaufes in dieser Lane, in dieser Kante, auch noch überschrieben oder neu abgefragt werden, kann man sagen. Mithilfe von Regelungsereignissen kann ferner festgelegt werden, welches Reglermodell verwendet wird. PID, PI, PD. Das sind Modelle, die kennt man aus der Regelungstechnik. 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 Mess-Tasks geschehen. Die könnte man zum Beispiel auch in diesem Zusammenhang als Datenerfassungs-Tasks bezeichnen. In der Grafik hier sehen Sie ein Prozessmodell mit einem Wert, der gemessen wird und einer darauffolgenden Regelung. Das heißt, wir haben wieder wie vorhin ein Measure Event, das ausgelöst wird. Wir haben dann einen Service Call, wo der Prozesswert V 1 gemessen wird. Und dann haben wir das Control Event. Hier wird erst einmal zum Beispiel in einem Script eine Berechnung durchgeführt, dass beispielsweise die Differenz vom optimalen Wert, also Differenz zwischen optimalen Wert und V 1, dem aktuellen Wert, ausgerechnet wird. Damit gehen wir dann in die nächste Berechnung des PID-Reglermodells und dann am Ende setzen wir einen Service Call ab, wo der Manipulating value ausgeschickt wird, an den jeweiligen Aktor. Also das Ergebnis, das dann sich durch die Regulierung ergeben hat, wird dann quasi an den jeweiligen Aktor ausgeschickt. Hier kann man dann entsprechend, wenn man das möchte, Datenelemente dafür definieren. Für Control kann man die folgenden Attribute hier definieren. Also wie bei Measure auch Interval frequency in Hertz. Der gleiche Wert in diesem Fall. Values expected to change kann man hier auch angeben. Und zwar einerseits geben wir natürlich auch den Control Type an, also welches Regelungsmodell wird hier verwendet. In dem Fall PID-Regler. Welcher Wert wird verändert? Und hier kann man dann auch noch eine Range angeben. In welchem Bereich darf sich dann dieser Wert auch bewegen? Quasi als Sicherheitsmaßnahme könnte man sagen. Das habe ich jetzt erwähnt, das habe ich erwähnt. Würde hier parallel verwendet werden, würde der letzte Wert von V 1 genommen werden, für den keine Zeitgarantie besteht. Also wenn wir hier sagen, wir nehmen nicht sequential, sondern parallel, würde parallel zu den Zustandsabfragen auch die Regulierungen, würden diese Regulierungen auch erfolgen. Das heißt, wir können hier nicht garantieren, wann genau oder ob im gleichen Zyklus auch der aktuelle Wert gemessen wurde. Warum haben wir uns das jetzt aufbehalten, wenn im Grunde wir schon sagen, 'Okay, Regulierungen müssen immer auf Zustandsabfragen erfolgen'? Es könnte vielleicht sogar Beispiele geben, wo Werte gemessen werden oder Zustände abgefragt werden und das parallel zu den jeweiligen Regulierungen erfolgt. Und dann muss einfach der zuletzt verfügbare Wert genommen werden. Wenn ich sage, sequential, die Regulierung muss wirklich mit dem letzten, mit dem aktuellsten Wert funktionieren, das kann man dann natürlich auch mit dieser Bedingung dann hier festlegen, dann würde ich sequential verwenden als Attribut für das Closed Loop Subsystem. Könnte hier sagen, 'Okay, wenn im aktuellen Zyklus erst die Zustandsabfragen erledigt sind oder die, die wir brauchen erledigt ist, dann kann ich erst die entsprechende Regulierung durchführen.' Und da kann ich auch nicht den Wert aus dem letzten Zyklus verwenden. Also, das ist hier der Unterschied. #00:33:16-0#

42. B: Ist das dann auch im Modell ersichtlich oder wie würde man das dann/? #00:33:20-9#

43. I: Da zeige ich gleich die Darstellung, //B: Parallel-Fall? // ja. Also, Sie meinen, dass man die Stränge dann sieht, eventuell, dass sich das dann verändert, die //B: Genau. // Lanes vielleicht? Also das einzige, was wir derzeit vorgesehen haben, wäre diese Darstellung, dass man diese beiden Attribute als Strings, als Labels, neben dem Closed Loop Subsystem dargestellt hätte.

346 Also das wäre die einzige Darstellung aktuell. Was aber auch angemerkt wurde von anderen
347 Interview-Gästen wäre, dass es praktisch wäre, wenn es wirklich parallel abläuft, dass man es auf
348 die gleiche Höhe ziehen könnte, im Prozessgraphen. Aber das ist derzeit noch nicht der Fall.
349 #00:34:07-3#

350 44. B: Ja, und in dem Parallel-Fall ist es dann auch für einen Nutzer, der sich das anschaut, das
351 Modell, dann klar, dass quasi dieser letzte Wert genommen wird, wie es gerade dort stand?// I:
352 Nur eigentlich/ // Ist das ersichtlich dann? #00:34:24-2#

353 45. I: Nur durch diesen Indikator. Also nur durch diesen String, den Sie gerade gesehen haben, das
354 Label. Das müsste man dann wissen, aber es ist nicht grafisch im Prozessmodell, im
355 Prozessgraphen, dargestellt. #00:34:41-8#

356 46. B: Okay. Verstehe. #00:34:47-0#

357 47. I: Wenn Sie aber meinen, dass das gut wäre, das darzustellen, wir gehen nachher auch die
358 Prozessmodelle noch durch und da wird zum Beispiel auch danach gefragt, ob Sie vielleicht, ob
359 Ihnen vielleicht etwas auffällt, was man noch verbessern könnte. Also das wird auch noch Thema
360 werden. #00:35:04-4#

361 48. B: Okay, gut. #00:35:06-1#

362 49. I: Ja. Wo waren wir? #00:35:10-4#

363 50. B: Ja, weil es ist ja vielleicht/ Also ich, nur ganz kurz, ich denke, das ist schon wichtig zu wissen,
364 welcher Wert denn jetzt wirklich genommen wird, zum Schluss heraus. Also das wäre vielleicht
365 schon irgendwie, ja, eben gut zu wissen. #00:35:25-7#

366 51. I: Ja, das stimmt schon, ja. #00:35:29-1#

367 52. B: Auch für die Erklärbarkeit des Systems. Vielleicht also, wenn ich mir da wieder Punkt Sechs
368 überlege von den Kriterien vorher. #00:35:37-2#

369 53. I: Also allein als Label wäre es wahrscheinlich nicht gut genug ersichtlich. #00:35:44-8#

370 54. B: Ja, wenn man jetzt ad hoc weiß, was das sozusagen für Konsequenzen nach sich zieht, dass da
371 parallel steht. Okay. Aber, also, ist ja die Frage, was hat der Nutzer dann gerade für ein Wissen
372 über das System. Wenn man jetzt neu damit modelliert, zum Beispiel, dann ist einem das
373 vielleicht nicht ganz bewusst. Vielleicht kann man da irgendwie/ Ja, müsste man dann in der
374 Dokumentation quasi nachlesen, was so was bedeutet eigentlich. Wenn ich mir das so in
375 Richtung Programmiersprache, zum Beispiel, überlege. #00:36:17-1#

376 55. I: Ja, das wäre dann natürlich auch nicht intuitiv, eigentlich. //B: Hm. (zustimmend) // Wenn man
377 extra nachschauen müsste. Ja. Das stimmt schon. Guter Input. Okay, machen wir noch die
378 Einführung fertig und dann kommen wir eigentlich schon zu Prozessbeispielen. //B: Ja. (lacht) //
379 Nein, passt schon. Sie können //B: Okay. // jederzeit etwas einwerfen und ich bin auch dankbar
380 dafür. Es ist besser, wenn Sie es gleich sagen und nicht, dass es dann später verloren geht. Das
381 wäre schade. #00:36:45-8#

382 56. B: Ja, ich habe mir noch eine andere Sache notiert, aber das können wir dann später nochmal
383 durch //I: Okay. // gehen. Das war schon weiter vorne, aber das kommt sicher bei den Modellen
384 nochmal. Ansonsten greife ich es nochmal auf zum Schluss. (lacht) #00:36:57-0#

385 57. I: Perfekt, danke. Okay, Zustandsabfragen und Regulierungen sollten in regelmäßiger Frequenz
386 ausgelöst werden. Abbruchereignisse hingegen werden nur durch ihre Abbruchbedingungen
387 ausgelöst, die der Benutzer definieren kann. Ein Beispiel für ein Abbruchereignis wäre, wenn
388 etwas den Abbruch eines Zyklus auslöst, zum Beispiel von außen das Signal, dass ein Stoppsignal

389 hereingekommen ist oder vielleicht sogar generell Notaus oder etwas in dieser Art und Weise.
390 Hier sehen Sie jetzt einen Prozess mit einem zu messenden Wert, einer Regelung und einer
391 Abbruchbedingung. Das heißt, wir hätten mal Measure. Es wird wieder unser Prozesswert V 1
392 gemessen. Es werden im Strang daneben unter Control Event wieder die gleichen
393 mathematischen Operationen durchgeführt in den Scripts und dann wieder der entsprechende
394 Wert ausgeschickt, der sich ergibt. Und wir haben jetzt bei Cancel im letzten Strang die
395 Bedingung definiert für den jeweiligen Abbruch. Also in dem Fall hier wäre es Emergency Stop,
396 wenn der aktiviert wird, dann wird aus dem Closed Loop System ausgebrochen. Standardmäßig
397 würden wir natürlich sagen, dass dieser Wert aber auf false gelegt ist, weil wenn er von Anfang
398 an ausgelöst wäre, dann würde das Closed Loop Subsystem eigentlich kein einziges Mal wirklich
399 durchlaufen, also es würde gleich die Abbruchbedingung ausgelöst werden und damit würde nie
400 eine Regulierung erfolgen oder der gleichen. Ja. Nachdem das Ereignis ausgelöst wurde, können
401 Tasks zur Aufräumroutine abgearbeitet werden bevor der Zyklus beendet beziehungsweise der
402 komplette Prozess vollständig beendet wird. Damit wird auch das fünfte Feature für
403 Aufräumprozesse, dass wir oben genannt haben, erfüllt. Wenn wir jetzt Aufräum-Tasks
404 definieren wollen, dann können Sie hier in der Grafik ein Beispiel sehen, dass wir zum Beispiel
405 noch einen Service Call definieren, der abgesetzt wird. Den nennen wir jetzt beispielsweise
406 Initiate shutdown routine for oder in vessel one, also im Kessel Eins beispielsweise, Behälter Eins.
407 Man könnte hier aber auch einfach einen Subprozess, wenn es ein bisschen ein komplexerer
408 Ablauf ist, noch definieren, bevor man aus dem Closed Loop Subsystem ausbricht. Das ist jetzt
409 hier speziell auch modelliert, weil das auf diese Abbruchbedingung auch bezogen wäre. Man
410 könnte aber auch generell sagen, wenn ich standardmäßig aus diesem/ wenn ich generell aus
411 diesem Closed Loop Subsystem ausbreche, könnte man natürlich noch hier zwischen dem
412 zweiten Teil des Closed Loop Subevents, also dem zweiten Symbol, und dem End Event noch
413 etwas einfügen. Also quasi etwas, was immer abgearbeitet wird, sobald ich aus dem Closed Loop
414 Subsystem ausbreche. #00:40:00-0#

415 58. B: Ist das dann diese Bedingung, dass es sozusagen in einen passenden Zustand wieder versetzt
416 werden soll, das System? #00:40:09-1#

417 59. I: Genau, das wären die Tasks, die //B: Genau. // dafür ausgeführt werden müssten. Ja.
418 #00:40:12-5#

419 60. B: Ja, okay. #00:40:13-7#

420 61. I: Und einmal halt generell und einmal spezifisch für die jeweilige Abbruchbedingung. Und später
421 in den Modellen werden Sie auch sehen, dass man natürlich mehrere Abbruchbedingungen
422 nebeneinander definieren kann. Also die können auch einzelne spezifische Aufräumroutinen
423 auch definiert haben, wenn das gewünscht ist. #00:40:32-1#

424 62. B: Okay. #00:40:33-4#

425 63. I: Die vorgestellten Erweiterungen sollen bei der Modellierung von kontinuierlichen Prozessen
426 helfen, indem Vorlagen für die Erstellung von Prozessmodellen vorgegeben werden und
427 andererseits durch die Darstellung als Closed Loop Subsystem mit eigenen Symbolen für
428 Zustandsabfrage-, Regulierungs- und Abbruchereignisse helfen, solche Prozesse leichter
429 nachvollziehen zu können. Hinzu kommt, dass man für eine übersichtlichere Darstellung des
430 gesamten Prozesses auch Subprozesse zur Unterteilung nutzen kann. Damit wollen wir zum
431 Beispiel auch ermöglichen das letzte Feature zu erfüllen, nämlich die verbesserte
432 Verständlichkeit der Modelle von kontinuierlichen Prozessen. Ja, ich werde Ihnen nun
433 Prozessbeispiele zeigen, die mit den in unserer Arbeit vorgestellten Erweiterungen modelliert
434 sind. Ich möchte, dass Sie sich die Modelle ansehen und mir sagen, was Sie aus ihnen
435 herauslesen können und ob die Modelle den notwendigen Informationsgehalt für die
436 Modellierung der zugrundeliegenden Regelungsprozesse erfüllen. Vorab werde ich Ihnen zum
437 jeweiligen Prozess natürlich erklären, was abgebildet werden soll. Und ich würde Sie einfach
438 bitten, offenes Feedback zu den Modellen in BPMN zu geben. Wir haben einmal ein etwas
439 einfacheres und einmal ein etwas komplexeres Modell. Fangen wir mit dem einfacheren an. Es

440 handelt sich hierbei um eine einfache PI-Temperaturregelung für einen Wärmetauscher
441 basierend auf einem Beispiel aus der MathWorks-Bibliothek, also MathWorks, MATLAB. Ich
442 nehme an, die Software wird Ihnen ein Begriff sein. Kennen Sie/ #00:42:18-8#

443 64. B: Ja, schon damit gearbeitet in meinem Studium. #00:42:21-5#

444 65. I: Super. Okay, gut, ja. Also hier gibt es die Möglichkeit, auch über Simulink, vielleicht kennen Sie
445 das Package, //B: Hm. (zustimmend) // auch verschiedene Sachen zu simulieren oder abzubilden,
446 zum Beispiel auch Regelungsaufgaben. Und hier wurde ein Beispiel als Basis herangezogen. Die
447 Temperatur einer Flüssigkeit in einem Rührkessel wird mittels Wärmetauscher geregelt. Der
448 über den Wärmetauscher eingebrachte Wärmestrom wird über ein Ventil, das den Dampfstrom
449 kontrolliert, gesteuert. Das heißt der eigentliche Wärmelieferant ist ein Dampfstrom, der über
450 den Wärmetauscher quasi dem Kessel zugeführt wird, dem Rührkessel, und damit die Flüssigkeit
451 im Rührkessel erwärmen soll und der Dampfstrom wird durch ein Ventil gesteuert. Der zu
452 beachtende störende Umgebungseinfluss, das heißt das, was eigentlich dem Kessel oder der
453 Flüssigkeit die Wärmeenergie entziehen kann, ist die, oder die Temperatur beeinflussen kann, ist
454 die schwankende Temperatur der zugeführten Flüssigkeit. Der Tank ist grundsätzlich als isoliert
455 anzunehmen. Das heißt, wir gehen davon aus, dass über die Kesselwand, über die Tankwand,
456 keine Wärmeenergie, kein Wärmestrom abgeführt wird. Wenn man das Ganze jetzt in einem
457 Flow Chart darstellen würde, würde man hier sehen, wie das Ganze aufgebaut ist. Das heißt, wir
458 hätten einen Rührkessel mit entsprechendem Rührwerkzeug. Wir hätten einen
459 Temperaturfühler. Wir hätten hier den Inflow, also die zugeführte Flüssigkeit. Wir hätten hier die
460 abgeführte Flüssigkeit. Und hier hätten wir entsprechend den Wärmetauscher und das Ventil.
461 Und der Dampfstrom wird dann hier über diese Leitung quasi zugeführt. Wenn wir jetzt in das
462 -Beispiel hineingehen, können wir bei den Datenelementen verschiedene Default Values
463 schon einmal definieren, die für die Berechnung des PI-Reglers notwendig sind, also genau für
464 dieses physikalische Phänomen das entsprechende Reglermodell. Ja, das kennen Sie eh. Dann
465 verschiedene Endpunkte als Beispiel, die man definieren kann. Das wissen Sie ja eh auch, wie
466 man das machen kann. Und wir sind jetzt hier, wir haben natürlich für dieses Prozessbeispiel
467 nicht so viele detaillierte Vorgaben aus der MathWorks-Bibliothek und müssen daher auch vom
468 Verhalten her ein bisschen annehmen, wie es sich verhält. Also wir haben jetzt hier keine
469 zeitlichen Vorgaben. Deswegen haben wir statt dem Cancel den Wait-Ansatz. Das heißt, hier
470 würde gewartet werden auf jeden einzelnen Strang bis er beendet wird. Und in dem Fall steht
471 jetzt zwar sequential, aber grundsätzlich könnten wir auch davon ausgehen, wie gesagt, weil wir
472 keine Bedingung hier gegeben haben, dass es grundsätzlich parallel auch laufen könnte. Wir
473 haben hier einerseits mal Measure. Hier wird einmal in der Sekunde die Temperatur des Tanks
474 gemessen. Dann, wie gesagt, nach dem Measure Event wird dann der Service Call dafür
475 abgesetzt und man könnte hier zum Beispiel noch ein Script einfügen, dass irgendetwas mit dem
476 jeweiligen Prozesswert noch gemacht wird. Man könnte natürlich auch hier die Kombination,
477 also Service Call mit Script, einsetzen oder einen Subprozess oder dergleichen. Also hier ist man
478 in der Modellierung recht freigelassen. Das wissen Sie eh grundsätzlich, was man hier alles noch
479 beliebig einfügen könnte, je nachdem, was halt notwendig ist. Entsprechend hätten wir auch
480 noch/ müssen wir natürlich die zweite Temperatur auch überprüfen. D steht hier für Disturbance,
481 also für die Störung, das heißt für die zugeführte Flüssigkeit von oben. Und dann hätten wir den
482 Control-Strang. Hier haben wir das mathematische Modell des PI-Controllers. Das heißt, wir
483 gehen hier mal hinein, rechnen uns entsprechend mit der Differenz des Sollwertes das jeweilige
484 Ergebnis aus, das wir aus dem PI-Controller einfach heraus bekommen. Wir können zum Beispiel
485 noch, wenn das notwendig sein sollte, vorher noch eine Data Conversion einfügen, als eigenes
486 Script, als vorgegebenes. Wenn wir dann den Wert umgerechnet haben oder umgewandelt
487 haben, können wir dann schließlich den Service Call absetzen. Das heißt wirklich das Stellsignal
488 für den Antrieb, also für den Motor des jeweiligen Ventils in dem Fall, aus schicken und damit
489 den Dampfstrom dann kontrollieren, über das Ventil. Und wir haben hier auch keine
490 Abbruchbedingung in der Vorlage vorgegeben, in der MathWorks-Bibliothek. Deswegen haben
491 wir hier einfach angenommen, dass wir ein Signal kriegen, quasi Stop activated, wenn das auf
492 true gesetzt wird, dann wird entsprechend eine Shutdown-Sequenz ausgeführt. Also als
493 Subprozess stellvertretend hier als Script dargestellt oder man könnte halt wieder einen Service
494 Call definieren oder was auch immer. Ja. #00:47:52-7#

- 495 66. B: Kann ich kurz eine Frage stellen? #00:47:54-8#
- 496 67. I: Sicher, bitte. #00:47:55-8#
- 497 68. B: Ich kenne mich jetzt in der Produktion und Regelungstechnik nicht so ganz gut aus, also gar
498 nicht. Wie ist das, wenn zum Beispiel aufgrund von einer vorherigen Messung eine andere
499 Messung dann angestoßen werden muss, nochmal? Also, wenn nochmal irgendwas überprüft
500 werden muss, weil die eine Messung quasi ergeben hat, dass der Parameter vielleicht nicht ganz
501 passende Zustand, nicht ganz in Ordnung ist, und dann muss man das nochmal gegebenenfalls
502 bei einer anderen Messung überprüfen. Wie könnte man es dann HIER modellieren, solche
503 Abhängigkeiten? Geht das? Kommt das überhaupt vor, der Fall, vielleicht? #00:48:30-2#
- 504 69. I: Also, ich kann mir schon vorstellen, dass der Fall vorkommen kann. Warum auch nicht? Also
505 das ist durchaus realistisch. Wir haben es jetzt hier in dem Prozessbeispiel nicht dargestellt, aber
506 wenn man wirklich zwei Messungen hätte, die auch zeitlich voneinander abhängig sind oder von
507 der Reihenfolge her voneinander abhängig sind, das heißt ich messe erst Wert A und kann
508 daraufhin entscheiden, ob ich Wert B noch brauche, oder Wert B ist dann noch notwendig, wenn
509 Wert A in einer gewissen Range liegt oder so etwas, dann würde in dem Fall hier die Möglichkeit
510 bestehen, noch diese Messung einzufügen. #00:49:08-8#
- 511 70. B: Okay. #00:49:10-1#
- 512 71. I: Also wenn ich sage, 'Hier wird Wert A gemessen und dann zum Beispiel ein Script eingesetzt,
513 das etwas auswertet.' und dann kann ich hier zum Beispiel sagen, 'Okay, wenn das in einem
514 gewissen Bereich liegt, dann muss hier noch ein Wert gemessen werden.' Das kann man einfach
515 einhängen, ja, einfügen. #00:49:30-6#
- 516 72. B: Hm. (zustimmend) Und könnte man dann auch nochmal dieses Gateway zum Beispiel
517 dazwischen schalten an der Stelle? #00:49:35-9#
- 518 73. I: Es ist grundsätzlich so gedacht, dass/ es ist hier sehr stark an die SPS-Programmierung
519 angelehnt. Das heißt, dass man ein Closed Loop Subsystem hätte, in dem man sagen kann, 'Okay,
520 ich habe hier verschiedene parallele Stränge.' Aber man kann hier definieren, dass die natürlich
521 mit unterschiedlicher Frequenz ablaufen. Das heißt, das Closed Loop Subsystem soll darstellen,
522 welche Dinge, welche Tasks, nebeneinander parallel immer wieder in einem Zyklus ausgeführt
523 werden auf kontinuierliche Art und Weise. Und man müsste hier eigentlich kein Closed Loop
524 Subsystem nochmal instanziiieren, weil man die Möglichkeit hätte, sowieso sehr viele Stränge
525 nebeneinander einzusetzen, außer natürlich es ändert sich etwas bei den Attributen, die dieses
526 eine System halt definieren, also bei wait oder cancel oder sequentiell oder parallel. Aber da
527 müssen wir uns eventuell auch noch Beispiele überlegen. Grundsätzlich ist derweil nicht
528 vorgesehen, soweit //B: Okay. // ich weiß. #00:50:50-7#
- 529 74. B: Verstehe. (...) Ja, gut, das war schon die Frage eigentlich. #00:50:59-3#
- 530 75. I: Es ist halt ein eher technischer, eine eher technische Herangehensweise, aber wenn man sich
531 jetzt/ wenn man den Anspruch erhebt, dass man nicht nur technische Prozesse,
532 Produktionsprozesse, abbilden möchte, sondern vielleicht sogar aus verschiedenen anderen
533 Branchen, also eher Management-lastigere Sachen abbilden möchte, dann wäre es interessant
534 zu wissen, ob das notwendig sein könnte, so etwas auch vorzusehen. Also, mir würde jetzt selber
535 nicht unbedingt ein Beispiel dafür einfallen, //B: Okay. // muss ich zugeben. Ja. #00:51:40-6#
- 536 76. B: Ja, wie gesagt, ich kenne mich da in der Produktion nicht aus. Deswegen halt meine Frage, ob
537 es so einen Fall prinzipiell geben könnte. Und, ja. #00:51:50-4#
- 538 77. I: Ja, das ist gut. Das ist eh genau die Sichtweise, die wir auch erfassen wollen. Also das ist eh
539 perfekt. Ja, hier sehen Sie halt noch die grafische Darstellung, wie die Scripte auch aussehen

540 könnten, also im Grunde die mathematischen Operationen, hier ganz einfache Beispiele, wo
541 einfach von einem Wert auf den anderen überschrieben wird. Aber nur, dass ersichtlich ist, was
542 man in den Scripten definieren könnte für Leute, die die ■■■ auch nicht kennen. #00:52:20-9#

543 78. B: Ja, das ist gut, dass so noch darzustellen. Das ist dann quasi ausklappbar oder ist dann immer
544 vorhanden? Kann es auch komplexer werden? Dieses Script-Teil, oder? #00:52:34-6#

545 79. I: Also die Scripts würden grundsätzlich in einfachster Art und Weise hier auf der rechten Seite
546 dann neben dem Graphen erscheinen. Wir hätten dann im Script halt in der Zeile natürlich auch
547 nichts stehen, auch beim Label. Das müssten wir auch erst definieren, und man könnte dann hier
548 halt beim Script, wie gesagt, mehrere Zeilen Code einfügen und, ja, könnte dann entsprechend
549 damit arbeiten. #00:53:00-0#

550 80. B: Okay. #00:53:01-6#

551 81. I: Ja. Nachdem wir jetzt den ersten Prozess gesehen haben, möchte ich Sie bitten, das Modell
552 nach den Kriterien hier in dieser Tabelle zu bewerten. Und zwar auf einer Skala von Eins bis Fünf,
553 wobei Eins eher schlecht ist, also sehr schlecht ist, und Fünf sehr gut. Also es geht darum, je
554 mehr Punkte erreicht werden, desto besser ist es. Und die Kriterien, die wir hierfür haben, die
555 gehe ich auch nochmal ganz kurz durch. Wir werden die gleichen auch beim zweiten
556 Prozessbeispiel haben. Und zwar, einerseits Verständlichkeit. Würden Sie sagen, Sie wüssten
557 jetzt, was hier passiert, also Sie könnten nochmal wiedergeben, was in dem Prozess ablaufen soll?
558 Beziehungsweise zweiter Punkt, Übersichtlichkeit. Würden Sie sagen, Sie können das
559 Gesamtsystem auf einen Blick erfassen? Einfachheit. Könnte man das Modell, Ihrer Meinung
560 nach, vielleicht noch einfacher darstellen? Logik. Wird für Sie klar, was parallel und was
561 sequentiell passiert? Und letzter Punkt wäre Erweiterbarkeit. Könnte man dem Modell, Ihrer
562 Meinung nach, noch etwas hinzufügen, was den Informationsgehalt verbessern würde? Also
563 bitte auch so ein bisschen mit einer Begründung, dass man nachvollziehen kann, was Sie damit
564 meinen oder wie Sie darauf kommen. #00:54:35-3#

565 UNTERBRECHUNG - Verbindungsproblem

566 82. I: Hallo? #00:54:40-0#

567 83. B: Hallo? #00:54:41-6#

568 84. I: Ja. Hören Sie mich? #00:54:44-0#

569 85. B: Ja, jetzt wieder. Es war gerade irgendwie weg, die Verbindung. #00:54:49-3#

570 86. I: Okay, ja. Das kann vielleicht mein Internet gewesen sein. #00:54:52-5#

571 87. B: Ausgefallen. Ja. #00:54:54-0#

572 88. I: Ja, so. Einen Moment. Ich share nochmal meinen Screen, dass Sie den Leitfaden wieder sehen
573 können. #00:55:00-5#

574 89. B: Hm. (zustimmend) Moment. Jetzt. #00:55:06-2#

575 90. I: Okay, Sie können es wieder lesen? #00:55:11-5#

576 91. B: Ja, ich kann es wieder lesen. Genau. Also vielleicht der Punkt Erweiterbarkeit, das wäre ja
577 dann das, was wir vorhin schon mal kurz besprochen haben. Sequentiell und parallel, da könnte
578 man vielleicht den Informationsgehalt noch ein bisschen verbessern an der Stelle. Das heißt, ich
579 finde es prinzipiell schon gut. Aber eben diese Kleinigkeit würde ich mir vielleicht noch wünschen,

580 beim Model. Das heißt, ja, inzwischen, ja, eigentlich/ Ja, wie gesagt, es ist schon recht gut. Vier
581 würde ich da wohl vergeben. (...) Genau. Dann Richtung Logik. Darf ich das Prozessmodell
582 nochmal kurz sehen, bitte? #00:56:01-3#

583 92. I: Ja. Ich zoome vielleicht noch ein bisschen/ Ja, so. #00:56:08-4#

584 93. B: Ja, ich denke/ Ja, wie gesagt, halt auch wieder dieser Punkt mit, ob man sequentiell und
585 parallel noch implizit anders darstellen könnte. Also würde ich da wahrscheinlich auch wieder
586 eine Vier vergeben, an der Stelle. Was hatten wir noch? Übersichtlichkeit. Das Gesamtsystem auf
587 einen Blick erfassen. Genau. Man hat ja rechts quasi auch, ja, rechts, links das Prozessmodell,
588 genau, und rechts hat man ja noch diese Beschreibungen. #00:56:49-5#

589 94. I: Genau, das würde dann hier/ ja, also die Labels würden natürlich erscheinen und dann jeweils
590 auf welches Symbol man auch klickt, hätte man dann die Attribute ungefähr hier rechts davon
591 dargestellt. #00:57:00-7#

592 95. B: Genau. Also ich finde es prinzipiell dann sehr übersichtlich. Genau. Zur Einfachheit. Man kann
593 ja diese Elemente ein- und ausklappen, richtig? Das heißt, wenn man hier jetzt einfach nur
594 darauf klickt, dann sieht man noch diese zusätzlichen Attribute, dann auf der rechten Seite. //I:
595 Genau. // Man kann es aber auch sozusagen wieder wegklicken und somit nur das
596 Prozessmodell an sich anschauen, mit den Labels? #00:57:30-4#

597 96. I: Ich glaube, dass die GUI (Graphical User Interface) das grundsätzlich ermöglicht, ja.
598 #00:57:33-3#

599 97. B: Ja, das finde ich vielleicht gar nicht so verkehrt. Weil je nachdem, wer sich dieses
600 Prozessmodell anschaut, kann das auf der rechten Seite natürlich wichtig sein oder halt eher
601 auch vielleicht unwichtiger. Kommt dann vermutlich auf die Person an. Ich finde es aber sehr gut,
602 dass man da sozusagen die Wahlmöglichkeit hat, ob man das Modell vereinfachen kann oder
603 halt eben komplexer dann darstellen kann und so die Modellierung an sich finde ich prinzipiell
604 sehr gut. Also was Einfachheit betrifft. Aber das ist ja auch grundsätzlich, finde ich, bei der [REDACTED]
605 recht gut gestaltet. #00:58:12-6#

606 98. I: Und der letzte Punkt //I: So, was haben wir noch nicht? // wäre noch Verständlichkeit.
607 #00:58:19-8#

608 99. B: Genau. Was passiert? Kann ich das Prozessmodell nochmal sehen? #00:58:24-0#

609 100. I: Ja, klar. (lacht) #00:58:24-6#

610 101. B: Verständlichkeit. Genau, also man muss natürlich erst einmal die ganzen Symbole sich merken,
611 also was jetzt quasi Abbruchkriterium darstellt, und so weiter. Aber ich finde es gut, dass die
612 Labels mit dabei stehen, also ich würde auch (...), ja, zwischen Vier und Fünf würde ich da geben.
613 #00:58:51-9#

614 102. I: Okay, wunderbar, danke. Bevor wir dann zum zweiten Prozessmodell, zum zweiten Beispiel
615 kommen, würde ich vorschlagen, dass wir eine kurze Pause einlegen. #00:59:04-3#

616 103. B: Hm. (zustimmend) #00:59:05-2#

617 104. I: Okay. Und ich würde auch gleich die Aufnahme für den ersten Teil beenden. Einen Moment.
618 #00:59:10-6#

619 UNTERBRECHUNG - 5 Minuten Pause

105. I: Okay, Aufnahme läuft wieder. Kommen wir zum zweiten Prozessbeispiel, das wir modelliert haben. Das Modell basiert auf der Beschreibung wieder eines Heizprozesses, den wir aus Schulungsunterlagen der Firma Siemens entnommen haben. Es handelt sich hierbei ebenfalls um eine Temperaturregelung für einen Rührreaktor. Die Regelung erfolgt aber in diesem Beispiel mit einem PID-Regler, einer Handsteuerung sowie einem Pulsgenerator. Die Heizung erfolgt nicht über einen Wärmetauscher in dem Fall, sondern über ein Heizelement. Und weiters haben wir hier auch Verriegelungsbedingungen definiert. Als Basis für die Prozessmodellierung wurden die Beschreibungen aus 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 Subsystem ausbricht. In der Beschreibung ist es aber generell mit Handsteuerung auch kombiniert. Wir gehen davon aus, dass das System bereits angelaufen ist und automatisch gesteuert wird. Und weiters wird der Prozess für nur einen Reaktor und nicht wie in den Unterlagen für zwei Reaktoren beschrieben. Ja. Was sind unsere Verriegelungsbedingungen beispielsweise? Wir haben hier eine maximale Temperatur vorgegeben, nämlich sechzig Grad, die nicht überschritten werden darf. Beziehungsweise haben wir einen Mindestfüllstand von 200 Milliliter angegeben, der auch nicht unterschritten werden darf im jeweiligen Rührkessel. Und diese Maximalwerte beziehungsweise Minimalwerte können wir natürlich auch als Datenelemente für die Verwendung im Prozessmodell hier angeben. Weiters haben wir wieder Parameter für die Berechnung des Reglers und wir haben auch zum Beispiel den Default-Wert für den Operation Mode, in unserem Fall Automatic, beziehungsweise ist auch vorgegeben, dass der Hauptschalter natürlich auf Eingeschalten stehen muss, damit der Prozess überhaupt laufen kann. Ja, wir haben hier einen Cancel- und Sequential-Ansatz, weil wir hier wissen, dass das Programm für eine SPS, also für eine Steuerung in einem Prozess genutzt werden soll. Das heißt, ja, wir haben ein etwas komplexeres Prozessmodell, in diesem Fall auch mit mehreren Zustandsabfragen, die parallel zueinander laufen. Wir haben einmal die Abfrage der Temperatur im Reaktor, eventuell dann noch eine Datenumwandlung. Wir haben die Abfrage des Füllstandes im Reaktor. Wir haben die Abfrage des Operation Modes beziehungsweise auch die Abfrage, steht ein Emergency Stop beispielsweise an. Also hier gab es auch schon den Input, wir haben es halt auf diese zwei unterschiedlichen Weisen dargestellt. Man könnte einerseits abfragen, man könnte aber das Ganze auch als Push Notifications oder quasi, dass die Daten einfach an den Prozess selber übertragen werden auch einrichten. In dem Fall fragen wir halt ab. Beziehungsweise haben wir auch die Abfrage des Hauptschalters. Dann haben wir eine Regulierung, eine Regelung, hier modelliert in Form, wie gesagt, von einem PID-Controller. Wir haben dann noch in einem Script daran gehängt die Pulsweitenmodulation, also den Pulsgenerator, und dann haben wir wieder einen Service Call, der den jeweiligen Prozesswert, also den Wert, der ausgerechnet wurde, der an den Aktor geschickt werden soll, hier als Service Call eingeführt oder eingefügt. Die Abbruchbedingungen sind auch vorgegeben in diesem Beispiel. Wir haben hier auch mehrere. Und zwar einerseits, wenn der Hauptschalter auf Off geschaltet wird, wird zum Beispiel Service Call mit Script in diesem Beispiel, eine Benachrichtigung an den Maschinenbediener oder den Anlagenbediener geschickt. Wenn der Emergency Stop aktiviert wird, ebenfalls Nachricht, also natürlich, im Falle Emergency Stop sollte alles herunterschalten, aber es könnte dann noch eine Nachricht an den Operator geschickt werden. Wir haben hier aber auch die konkretere Bedingung in dem Fall jetzt mit, wenn die aktuelle Reaktortemperatur über der maximalen Temperatur liegt, muss etwas entsprechend ausgeführt werden beziehungsweise aus dem Closed Loop System ausgebrochen werden. Oder wenn der Füllstand, der aktuelle, unter dem Mindestfüllstand liegt und die letzte Bedingung, die wir noch haben, die wir auch definiert haben in diesem Kontext, war, wenn der Betriebsmodus, der Operation Mode, auf Manuel wechselt. Ja, also etwas komplexeres Modell. //B: Ja. // Wieder die Darstellung des Scripts für den PID-Controller. Und wir haben wieder eine ähnliche Bewertung oder die gleiche Bewertung für das Modell hier vorgesehen bezüglich Verständlichkeit, Übersichtlichkeit, Einfachheit, Logik und Erweiterbarkeit. Ja. #00:05:46-8#

106. B: Genau. Was mir jetzt bei so einem komplexen Modell auffällt, zwischen den Abbruchbedingungen/ also bei den Abbruchbedingungen jetzt zum Beispiel. Können diese Abbruchbedingungen auch eigentlich ja von dem abhängen, was ich gemessen habe, richtig? //I:

674 Genau, ja. // Sehe ich das korrekt mit der Reaktortemperatur? Wird das klar in dem Modell, was
675 von was abhängen kann, an der Stelle? Ist es dann, wenn ich quasi hinüber zähle, das
676 soundsovielte Task-Abbruch bezieht sich dann, also der dritte Abbruch-Task zum Beispiel bezieht
677 sich dann auf den dritten Mess-Task oder so? #00:06:34-4#

678 107. I: Ich weiß, was Sie meinen. Nein, das wird in dem Fall nicht ersichtlich. Das hängt nicht damit
679 zusammen. Die Möglichkeit, die hier jetzt gegeben wäre aus dem Prozessmodell herauszulesen,
680 welcher Wert für die jeweilige Abbruchbedingung notwendig ist, ist eigentlich nur über die
681 Labels gegeben. Beziehungsweise //B: Okay. // die Abbruchbedingung, die hier auch hinein
682 geschrieben wird. #00:07:02-4#

683 108. B: Okay. Ja, kann man sich vielleicht überlegen, ob man das noch anders darstellen könnte. Ich
684 wüsste zwar zwar ad hoc auch nicht unbedingt wie. Außer über die Labels, dass man es quasi
685 identifizieren kann. Aber könnte ja vielleicht für Nutzer sozusagen auch relevant sein, dass man,
686 wenn man das dann quasi modelliert, gleichzeitig, dass man schon gleichzeitig weiß, okay, diese
687 Abbruchbedingungen muss ich dann noch hinzufügen zum Beispiel, und das bezieht sich dann
688 auf diesen Wert, dass man quasi wieder den Konnex hat zwischen Abbruchbedingung, Wert
689 gemessen, und so weiter. #00:07:40-6#

690 109. I: Okay. #00:07:43-8#

691 110. B: Würde mir jetzt hier spontan auffallen. Genau. Also das wäre vielleicht zu // I: Können Sie das
692 lesen? // ja, Erweiterbarkeit, ja, würde das wahrscheinlich mit hineinfallen, oder? Um den
693 Informationsgehalt zu verbessern. #00:08:07-1#

694 111. I: Auf jeden Fall, ja. #00:08:10-4#

695 112. B: Und auch zur Übersichtlichkeit würde das vielleicht auch beitragen. Ich weiß nicht, ob es zur
696 Übersichtlichkeit beiträgt. Je nachdem, wie man es dann wahrscheinlich modelliert, ob es dann
697 nur komplexer wieder wird. Oder ob man sozusagen eine Art findet, wie man diese Dinge direkt
698 darstellen kann, dass zwei Sachen halt wirklich zusammenhängen an der Stelle und voneinander
699 abhängig sind. Würde dann natürlich auch wieder die Verständlichkeit vielleicht erhöhen, für die
700 Menschen, dass man quasi auf einen Blick erfassen kann, okay, ich habe jetzt den Fall., ich habe
701 jetzt quasi einen Wert gemessen, der sozusagen außerhalb des Toleranzbereichs lag und
702 deswegen ist abgebrochen worden. Oder was da mit dem System auch generell passiert,
703 vielleicht, beim Modellieren schon. Dass einem das bewusst wird, wenn dieser Wert nicht passt,
704 dann muss abgebrochen werden und dass man dann vielleicht darauf kommt, okay, wenn man
705 dann hier an der Stelle abbricht, müsste man noch zusätzliche Schritte zum Beispiel gehen
706 danach, damit es dann wieder in den konsistenten Zustand überführt werden kann. (...) Ja, das
707 heißt, ja, Bewertung ist halt schwierig. Das liegt natürlich auch daran, dass das Modell an sich
708 wahrscheinlich äußerst komplex ist schon mal. Also, ja, wahrscheinlich eine Drei oder so, bei
709 diesen Dingen. Ja. #00:09:43-0#

710 113. I: Bei einem konkreten Punkt oder generell bei/ #00:09:48-2#

711 114. B: Was ich jetzt so genannt habe, also bei //I: Also, Übersichtlichkeit/ // Erweiterbarkeit könnte
712 man das noch hinzufügen, was ich gerade gesagt habe. Und das würde gegebenenfalls, je
713 nachdem wie es modelliert ist, dann quasi den Informationsgehalt auch verbessern, würde ich
714 sagen. Und eben die Übersichtlichkeit kommt halt dann dadurch, dass man vielleicht den Konnex
715 bekommen zwischen bestimmten Dingen. Aber es kann natürlich auch dadurch wieder ein
716 bisschen komplexer werden, je nachdem, wie man es dann in der Modellierung versucht
717 umzusetzen. Da müsste man halt dann wahrscheinlich die konkreten Vorschläge nochmal
718 evaluieren, für das. Genau. Und bezüglich Einfachheit. Wenn ich jetzt/ Einfachheit. Könnte man
719 sich auch vielleicht überlegen, diese Kriterien, diese Abbruchkriterien, zum Beispiel einfach
720 zusammenzufassen und dann ausklappbar zu machen, wenn man genauer hineinschauen
721 möchte. Genau. So ganz links auch bei den Messungen. Dass man, wenn man wirklich so viele

722 unterschiedliche Tasks hat, die da passieren können, dass man das hat vielleicht eben kompakter
723 versucht anzuzeigen oder so. #00:11:16-7#

724 115. I: Okay. #00:11:20-8#

725 116. B: Ja, ich glaube, das war es, oder? #00:11:30-1#

726 117. I: Haben Sie zu Logik und zur Verständlichkeit konkret etwas gesagt? Da bin ich mir jetzt nicht
727 sicher. Ich meine Logik könnte man auch mit Erweiterbarkeit dann eigentlich und
728 Übersichtlichkeit ein bisschen zusammenlegen, ja. #00:11:46-6#

729 118. B: Ja, genau. #00:11:48-2#

730 119. I: Okay. #00:11:51-3#

731 120. B: Das es nochmal klarer wird, ja. #00:11:53-6#

732 121. I: Okay, passt. #00:11:57-0#

733 122. B: Gut. Ich hoffe, das ist in Ordnung so. #00:12:02-0#

734 123. I: Das passt schon. Ja, danke. //B: Das Feedback. Ja. // Wenn wir uns jetzt diese zwei Beispiele
735 angeschaut haben, wären Sie aufgrund der Erweiterungen, die hier eingefügt wurden oder
736 vorgestellt wurden, bereit, diese Modellierungsmethode in Ihrem Arbeitsalltag einzuführen,
737 wenn Sie Modelle für kontinuierliche Prozesse entwickeln müssten? Also wir haben ja jetzt hier
738 technische Beispiele, aber grundsätzlich können ja Zustandsabfragen und entsprechende
739 Regulierungen für verschiedenste Branchen angewendet werden. #00:12:37-8#

740 124. B: Also, ich denke schon, dass ich so etwas verwenden würde, weil ich die Idee halt gut finde,
741 dass diese drei Elemente, die ja wirklich miteinander einhergehen, dargestellt werden können
742 über dieses eine Gateway. Also. Dass man da quasi immer den Zusammenhang hat, deswegen
743 finde ich das schon gut eigentlich und würde es auch verwenden, wenn ich so etwas modellieren
744 müsste, ja. Weil ich dann halt direkt weiß, okay, ich kann auch nichts vergessen oder so, weil
745 man ja immer diese drei Elemente dann im Kopf hat. So könnte man vielleicht eine
746 Abbruchbedingung irgendwie versehentlich vergessen und wie kommt man dann wieder heraus
747 aus der Schleife. Also, das ist kompliziert, sonst glaube ich. Ja. #00:13:21-6#

748 125. I: Okay, danke. Wie gut würden Sie sagen, beschreiben die Erweiterungen Ihrer Meinung nach
749 die Regelungsprozesse für diese Beispiele? Also. Würden Sie sagen, Sie wüssten oder würden Sie
750 sagen, dass die Prozessmodelle grundsätzlich die Abläufe von den Beispielen, die ich
751 beschrieben habe, für Modell Eins und Zwei abbilden können? Dass sie sie gut beschreiben?
752 Wieder auf einer Skala von Eins bis Fünf. #00:13:59-7#

753 126. B: Ja, würde ich schon sagen. Also abbildbar sind sie meiner Meinung nach auf jeden Fall. Ist ja
754 dann eben nur diese Frage der Komplexität. Je nachdem, wie komplex dann das System auch an
755 sich wird, aber das ist ja generell die Frage bei der Modellierung, glaube ich, und sehr schwierig
756 im Allgemeinen bei Modellierung, wenn man schon einen komplexen Prozess hat, den dann
757 sauber zu strukturieren und das geeignet zu strukturieren, dass es eben auch einfach bleibt.
758 Generell schwierig, aber das kann abgebildet werden. Deswegen, ja. #00:14:30-1#

759 127. I: Okay. #00:14:30-6#

760 128. B: Zwischen Vier und Fünf. #00:14:32-3#

761 129. I: Okay. Danke. Würden Sie vielleicht jetzt, nach den zwei Beispielen sagen, dass für eine
762 detailliertere Prozessbeschreibung noch etwas fehlt? #00:14:42-8#

763 130. B: Man hat ja immer die Möglichkeit, sich noch zusätzliche Details zu holen. Durch einen Klick
764 auf ein Element, rechts. Das heißt, ich glaube, detailreich genug könnte es schon sein. Es ist halt
765 dann nur quasi die Frage, ob man noch ein bisschen besser die Abstraktion vielleicht schafft in
766 die andere Richtung. Also dass es ein bisschen einfacher wird. Also ich glaube detaillierter muss
767 es nicht mehr unbedingt werden. #00:15:17-8#

768 131. I: Okay. Hier kommt eine Frage, das überlasse ich Ihnen, wie Sie die beantworten möchten. Und
769 zwar, wenn Sie Erfahrung in der Regelungstechnik haben, was würden Sie empfehlen, um diese
770 Erweiterungen zu ergänzen, um sie für Ingenieure attraktiver zu machen? Sie haben ja schon
771 anfangs erwähnt, Sie haben hier eigentlich weniger Zugang //B: Ja, genau. // eventuell.
772 #00:15:44-2#

773 132. B: Also, es ist glaube ich, sehr schwierig, die Frage hier sinnvoll zu beantworten auf Basis meiner
774 Erfahrungen. #00:15:53-2#

775 133. I: Okay, gut. Dann überspringen wir das, wenn Sie möchten. Wir kommen zur letzten Frage.
776 Frage Zehn, die etwas umfangreicher auch wieder ist. Wir haben wieder eine Tabelle mit
777 verschiedenen Fragen, wo ich Sie wieder bitten würde, eine Bewertung von Eins bis Fünf
778 abzugeben. Wieder Eins, sehr schlecht, und Fünf, sehr gut. Und zwar geht es im Allgemeinen um
779 die Modellierung an sich. Die Fragen, die wir haben, sind. Wie einfach ist in den gezeigten
780 Modellen nachzuvollziehen, dass die einzelnen Abläufe parallel und unabhängig voneinander
781 laufe? Dann, wie einfach ist es zu definieren, wann eine Anpassung am System erfolgt? Wie
782 einfach ist es, die maximale Dauer einer Anpassung zu definieren? Wie einfach ist es zu
783 definieren, unter welchen Bedingungen sämtliche repetitiven Aufgaben beendet werden sollen?
784 Wie einfach ist es zu definieren, dass danach Aufräumaufgaben einmalig zu erfolgen haben? Und
785 die letzte Frage. Wie einfach ist es, komplexe Abläufe im Kontext von kontinuierlichen Prozessen
786 mit diesen Erweiterungen zu beschreiben? Also das ist im Grunde wieder, die Frage, die ich
787 Ihnen gerade gestellt habe. Ob Sie sagen würden, dass die Abläufe hinter den
788 Regelungsprozessen für die zwei Beispiele, wenn man jetzt die realen Prozesse kennt, ob die
789 Erweiterungen das gut abbilden können. Und das ist jetzt die Frage dazu, wenn Sie genau vor
790 dieser Aufgabe stehen das abzubilden, ob Sie denken, dass das eventuell einfacher jetzt
791 durchzuführen wäre mit den Erweiterungen. Darum geht es. Wenn Sie das selber machen
792 müssten. Frage Eins. Wie einfach ist es in den gezeigten Modellen nachzuvollziehen, dass die
793 einzelnen Abläufe parallel und unabhängig voneinander laufen? #00:17:57-6#

794 134. B: Ja, schon gut bis sehr gut, würde ich da sagen. #00:18:07-6#

795 135. I: Okay. Wie einfach/ #00:18:11-3#

796 136. B: Weil man sieht ja die einzelnen Stränge sozusagen, die modelliert worden sind. Also, das ist
797 glaube ich schon recht klar. #00:18:19-6#

798 137. I: Entschuldigung. Ich will Sie natürlich nicht unterbrechen. #00:18:24-0#

799 138. B: Nein, passt schon. (lacht) Alles gut. #00:18:26-6#

800 139. I: Zweite Frage. Wie einfach ist es zu definieren, wann eine Anpassung am System erfolgt?
801 #00:18:34-2#

802 140. B: Anpassung am System bedeutet dann quasi der mittlere Teil, dass man eine Aktion trifft
803 sozusagen. #00:18:46-0#

804 141. I: Hm. (zustimmend) Genau. Die Regulierungen nach den //B: Ja, und/ // Zustandsabfragen, ja.
805 #00:18:51-2#

806 142. B: Der Zeitpunkt, wann (...) grafisch dann unten darunter/ #00:19:02-8#

807 143. I: Hallo? #00:19:06-5#

808 144. B: /getätigt, sozusagen. Ja? #00:19:08-0#

809 145. I: Sie waren ganz kurz unterbrochen. Entschuldigung. #00:19:10-1#

810 146. B: Ach so. Genau, also. Ja, also die Anpassung sieht man ja/ Schritte, links, wo gemessen wird.
811 Und dann sieht man ja, okay, auf Basis von dem würde dann quasi eine Entscheidungen, also,
812 eine Anpassung quasi getroffen. Und zu definieren ist es, glaube ich, auch recht einfach. Also ich
813 würde auch so zwischen Vier und Fünf wohl sag/. #00:19:49-7#

814 147. I: Zwischen Vier und Fünf. #00:19:55-9#

815 148. B: Ich glaube, die Verbindung war gerade wieder schlecht. #00:20:00-7#

816 149. I: Ja, ich höre Sie teilweise ein bisschen abgehackt. Einen Moment. Ich unterbreche kurz die
817 Aufnahme. #00:20:08-3#

818 150. B: Ja. #00:20:09-2#

819 PAUSE in Aufnahme, keine neue Datei

820 151. I: Wir waren //B: Bei Dauer, richtig? // genau. Wir hatten jetzt die Frage, wie einfach ist es, die
821 maximale Dauer einer Anpassung zu definieren? #00:20:21-2#

822 152. B: Hm. (zustimmend) #00:20:24-7#

823 153. I: Ja, nimmt wieder auf. #00:20:27-0#

824 154. B: Okay. Genau, zur Dauer. Kann ich die Elemente/ also, kann ich das noch mal sehen, wie das
825 dann genau aussehen würde? #00:20:42-3#

826 155. I: Hm. (zustimmend) Da muss ich ein bisschen weiter hinauf. #00:20:45-3#

827 156. B: Ja, genau. Das hatten wir eher eingangs des Interviews. #00:20:53-0#

828 157. I: Ja, jetzt. #00:20:57-5#

829 158. B: Ja, genau. #00:20:58-0#

830 159. I: Und auf ähnliche Art und Weise bei Measure. Also hier haben wir es bei Control. Und bei
831 Measure geht es auch über das gleiche Attribut, also mit dem gleichen Bezeichner. #00:21:09-7#

832 160. B: Genau, das würde ja quasi rechts dann wieder angezeigt und man sieht es ja eh auch hier. Ja.
833 // I: Genau. Wenn ich quasi/ // Ja, nein, das ist eigentlich sehr einfach zu definieren dann, würde
834 ich sagen. Ja. #00:21:20-6#

835 161. I: Okay, das heißt, Sie würden/ #00:21:25-4#

836 162. B: Ja, auch, wahrscheinlich sogar auch schon sehr gut, weil es ja wirklich in den Attributen direkt
837 drinnen steht und man das auch tatsächlich wieder gut auslesen kann dann. #00:21:35-7#

838 163. I: Okay. //B: Ja. // Wie einfach ist es zu definieren, unter welchen Bedingungen sämtliche
839 repetitiven Aufgaben beendet werden sollen? #00:21:46-7#

840 164. B: Das bezieht sich hauptsächlich dann auf die Abbruchbedingungen logischerweise.
841 #00:21:54-2#

842 165. I: Genau. #00:21:54-8#

843 166. B: Ja, also, ich meine, man hat ja diese Abbruchbedingungen und man kann ja dann auch wieder
844 zusätzliche Aufgaben dahinter schalten, also. Ich finde, das ist eigentlich dann auch sehr leicht zu
845 definieren. Und auch, ja, recht klar, wenn man sozusagen ja auch noch das mit diesem Konnex
846 wieder einbauen könnte, wenn man das gemessen hat und dann kann ja auch ein Abbruch
847 stattfinden auf Basis eines gemessenen Wertes, also. Würde ich dann hier quasi eine/ Zu
848 definieren ist es, glaube ich, recht einfach, also. Wenn es nur ums Definieren geht dann Vier bis
849 Fünf auf jeden Fall, ja. Genau. Und, ja, das geht ja eigentlich auch mit den Aufräumaufgaben
850 einher. Die sind ja dann auch recht leicht zu definieren gewesen. Ja. #00:22:50-3#

851 167. I: Okay, würden Sie sagen, auf gleiche Art und Weise einfach durchzuführen? #00:22:55-1#

852 168. B: Ja, ja. #00:22:56-3#

853 169. I: Okay, wunderbar. Ja, dann kommen wir eh schon zur letzten Teilfrage. Wie einfach ist es,
854 komplexe Abläufe im Kontext von kontinuierlichen Prozessen mit diesen Erweiterungen zu
855 beschreiben? Das zielt darauf ab, dass wenn Sie vor dieser Aufgabe stehen würden, wie Sie das/
856 wie einfach das für Sie jetzt wäre, durchzuführen? #00:23:22-3#

857 170. B: Also, wie vorher eigentlich auch schon gesagt, ich finde es gut, dass man diese drei Elemente
858 hat, die ja auch logisch irgendwie zusammengehören. Das heißt, ich messe etwas, ich mache auf
859 Basis von dem eine Aktion und ich kann zur Not auch den Prozess/ also ich kann auch
860 ausbrechen. Deswegen, ich finde, das vereinfacht schon, dass man möglichst nichts vergisst im
861 Prozess zu modellieren. Und deswegen finde ich die/ Ja, finde ich es gut und finde es dann auch
862 ein bisschen einfacher, diese komplexen Verläufe quasi beschreiben zu können, weil man eben
863 ein bisschen an die Hand genommen wird und vielleicht auch nichts vergessen kann. Genau.
864 #00:24:07-8#

865 171. I: Okay, wunderbar. Ja, danke. Das war schon meine letzte Frage. Vielen Dank, dass Sie sich Zeit
866 genommen haben. Und wenn Sie vielleicht Feedback hätten, also wenn Ihnen etwas aufgefallen
867 ist generell was den Leitfaden angeht, oder Dauer des Interviews oder Verständlichkeit der
868 Fragen. Ich bin immer dankbar für Feedback, also ist immer sehr willkommen. Und ich würde
869 dann auch gleich die Aufnahme beenden. Einen Moment. #00:24:38-5#

870 172. B: Okay. #00:24:39-8#
