

Transcript - Group 1 "Engineers", Interview 4

I ... Interviewer (BLINDED)

B ... Expert

(Unv.)... Incomprehensible passage

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

() ... Comment

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

Transcript

1 1. I: Aufnahme läuft. #00:00:03-8#

2 2. B: Okay. #00:00:04-8#

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

4 [REDACTED]
5 [REDACTED] Möchten Sie sich vielleicht kurz vorstellen und Ihre Verbindung zu Verfahrenstechnik,
6 Prozessmodellierung erklären? Ich möchte Sie bitten, dass Sie dabei nicht Ihren Namen nennen,
7 sondern nur die folgenden Informationen. Und zwar, Ihre Berufsbezeichnung beziehungsweise
8 Umschreibung Ihres Arbeitgebers, Basis der Expertise zum Forschungsthema, Ausbildung
9 beziehungsweise fachlicher Hintergrund und Berufserfahrung. #00:00:42-0#

10 4. B: Okay. [REDACTED]

11 [REDACTED]
12 [REDACTED]
13 [REDACTED]
14 [REDACTED]
15 [REDACTED]
16 [REDACTED]
17 #00:01:59-2#

18 5. I: Super. Dankeschön, das passt perfekt. Dann zur Einführung zum Thema. Unsere Forschung
19 konzentriert sich auf die Entwicklung einer Methodik, um kontinuierliche Prozesse in BPMN
20 darzustellen und sie in einer Workflow Engine ausführbar zu machen. Für diese Aufgabe haben
21 wir an BPMN-Erweiterungen für kontinuierliche Prozesse gearbeitet. Warum gerade
22 kontinuierliche Prozesse? Diskrete Prozesse können bereits in anderen/ wurden bereits in
23 anderen Forschungsarbeiten behandelt und bei ihnen wurden nicht die gleichen Schwierigkeiten
24 in der korrekten Darstellung mittels BPMN gefunden. BPMN ist bereits ein weit verbreiteter
25 Standard im Business Process Management und hat seinen Weg in die Fertigung gefunden.
26 Diskrete Fertigungsprozesse können bereits mit BPMN 2.0 modelliert werden. Im Grunde wollen
27 wir eine Methodik einführen, um solche Prozesse so darzustellen, dass sie von jeder Person in
28 einem Unternehmen vom Ingenieur bis zum Manager verstanden werden können. Dies könnte
29 durch die Verwendung von BPMN erreicht werden. Ein weiterer Vorteil wäre auch, dass es
30 bereits eine Reihe von Workflow Engines gibt. Das sind Anwendungen, die die Ausführung dieser
31 Prozess Modelle auf der Grundlage der für jedes Symbol implementierten Logik ermöglichen.
32 Wir arbeiten mit einer webbasierten Anwendung, die erweiterbar ist und mehrere
33 Kommunikationsschnittstellen implementiert hat. Also nicht nur HTTP, sondern zum Beispiel
34 auch OPC UA. Ein weiterer Vorteil ist daher die Interoperabilität in diesem Zusammenhang im
35 Vergleich zu anderen proprietären, starren Softwareanwendungen. Wir wollen herausfinden, ob
36 diese Technik auch für die Implementierung von digitalen Abbildern eingesetzt werden kann. Da
37 digitale Abbilder dazu dienen, ein physikalisches System oder einen Prozess in digitaler Form
38 darzustellen, meist anhand von Daten oder mathematischen Modellen, mussten wir einen Weg
39 finden, den Ablauf von kontinuierlichen Prozessen wie sie aus der Prozessindustrie bekannt sind,
40 darzustellen. Aus diesem Grund haben wir uns auf die Modellierung von Regelkreisen
41 konzentriert. Also nicht diskrete Steuerung, sondern Regelkreise. Die Prozessmodelle sollen
42 durch BPMN für Personen mit unterschiedlichem Hintergrund leicht verständlich sein. Die

Interviews werden geführt, um herauszufinden, wie Prozess- und Regelungstechnik und Techniken aus der Business Process Modellierung kombiniert werden können und wie erste Ergebnisse von Experten wie Ihnen wahrgenommen werden. Außerdem wollen wir herausfinden, ob es Schwachstellen gibt, die von Experten identifiziert werden können und wie wir diese auch beseitigen könnten. Dann kurz drei Begriffe, wie wir sie verstehen. Der Begriff digitales Abbild oder digitaler Zwilling. Es gibt verschiedene Methoden, Dinge aus der echten Welt, zum Beispiel echte Maschinen, zu simulieren. Teils merkt man aber, dass es mehr Parameter brauchen würde als bei normalen Simulationsmethoden, um eine Maschine vollkommen so abzubilden, wie sie sich in der Realität auch verhält. Bei einem digitalen Zwilling wird versucht, möglichst nahe an das reale Verhalten einer Maschine oder anderer Objekte heranzukommen. Das soll dazu führen, dass, wenn etwas getriggert wird bei einer echten Maschine, der digitale Zwilling sich im Grunde gleich verhält, das gleiche Verhalten zeigt oder möglichst ähnliches Verhalten zeigt.

Kontinuierliche Prozesse. Das möchte ich an Beispielen erklären. Wenn man Bierbrauen her nimmt, gibt es zwei Möglichkeiten. Es gibt einerseits das diskrete, also das nicht-kontinuierliche Bierbrauen und andererseits das, ja, das kontinuierliche Pendant. Das diskontinuierliche, also das diskrete Bierbrauen, wäre, wenn man in einem geschlossenen Kessel die Zutaten hinein gibt, zehn Liter Wasser und so weiter, und den Brauprozess einfach schrittweise ablaufen lässt. Also Batch-Prozess. Am Ende kommt dann eine begrenzte Menge an Bier heraus. Die andere kontinuierliche Variante wäre, wenn man keinen vollkommen abgeschlossenen Kessel hätte, sondern miteinander verbundene Kessel, bei denen immer wieder Zutaten zugefügt werden und auch immer wieder am Ende Bier entnommen wird. Das geht die ganze Zeit so, so dass man nicht nachvollziehen kann, welche 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. Geschlossene Regelkreise. Ein geschlossener Regelkreis, ist jene Logik in Form von Hardware oder Software, die das kontinuierliche Bierbrauen ermöglichen könnte. Wenn man einen Prozess wie das kontinuierliche Bierbrauen hat, muss man schauen, wie man schlechtes Bier vermeidet, während der Prozess läuft. Man möchte also die Qualität auf einem gewissen Punkt halten. Beim schrittweisen Bierbrauen hat man nur diese zehn Liter, bei denen etwas schiefgehen kann und mit den nächsten zehn Litern muss man die initialen Parameter einfach besser anpassen. Aber was ist, wenn die Brauanlage dauernd läuft und wenn man ständig Bier heraus bekommt? Dann muss man währenddessen den Prozess überprüfen und schauen, dass man die gute Qualität des Bieres erhält. Das heißt, man testet oder misst Werte, die die Qualität beschreiben, überprüft, wie sich diese Werte von optimalen Werten unterscheiden und reagiert entsprechend. Stimmt etwas zum Beispiel beim Zucker- oder Alkoholgehalt nicht, muss das Mischverhältnis geändert werden oder Zeitparameter oder wie auch immer angepasst 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. Gut, das ist mal zur Begriffsklärung. Aus Informatik-sicht bestehen kontinuierliche Prozesse aus einer sich ständigen wiederholenden Abfolge von Zustandsabfragen beziehungsweise Messungen 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. Dann kommen wir halt zu Frage Eins. Zur ersten gleich. Würden Sie die Merkmale für wichtig oder für unwichtig einstufen, die in dieser Tabelle hier aufgelistet sind? Ich würde Sie bitten, dass Sie kurz mit Ja oder Nein, also wichtig oder unwichtig, antworten und vielleicht kurz auch begründen, warum Sie so antworten. Merkmal Eins. Verschiedene Zustandsabfragen und Regulierungen, Kombinationen daraus, sind unabhängig und können parallel ablaufen. Das heißt, wenn man in einem Prozess mehrere Messungen durchführt und entsprechend mehrere Regelkreise hätte und die würden parallel zueinander laufen. Ist das wichtig oder ist das unwichtig? Und warum würden Sie das sagen? #00:08:53-7#

6. B: Ich versuche gerade die Frage in die Praxis umzusetzen. Also ich könnte die (...) Für die Antwort Nein könnte ich jetzt keine praxisnahe Anwendung/ Also für mich ist das ganz wichtig. #00:09:19-3#

7. I: Okay, gut. #00:09:20-7#

- 98 8. B: Weil die einzelnen Regelkreise im kontinuierlichen System immer ineinander greifen. Und
99 man/ In den seltensten Fällen ist es so, dass der vorgängige Regelkreis, also der nachfolgende
100 Regelkreis vom Vorgänger nicht beeinflusst wird. Das ist im kontinuierlichen Prozess immer so.
101 Dass die abhängig sind, voneinander. #00:09:58-9#
- 102 9. I: Okay, perfekt danke. Also genau so eine Erklärung hilft mir halt, um nachher zu begründen,
103 warum wir diese Eigenschaften überhaupt behandeln, also so etwas ist eine gute Antwort.
104 Dankeschön. Okay, dann zweite Eigenschaft. Regulierungen. Also entsprechende Regelkreise,
105 also die Antwort darauf, folgen immer auf Zustandsabfragen, also auf einer Basis/ basierend auf
106 einer Messung kann man sagen. Ist das wichtig oder unwichtig und warum würden Sie das sagen?
107 #00:10:32-4#
- 108 10. B: Jeder Regelkreis braucht einen Ist-Wert. Sonst wäre es kein Regelkreis. Also in dem Falle
109 auch/ ist ganz wichtig. Es gibt keinen Regelkreis, der keine Messung, also ein Regelkreis, der
110 keine Messung hat. Weil ein Regelkreis ja immer auf den Fehler reagiert. Und der Fehler ist
111 Soll-Wert minus Ist-Wert. Und der Soll-Wert wird aus irgendeinem Prozess oder irgendeinem/
112 irgendein Wert errechnet oder bestimmt, was man halt glaubt, dass man dort haben sollte und
113 nach der Regelstrecken erhält man den Ist-Wert. Und ja, es ist wichtig. #00:11:25-5#
- 114 11. I: Danke. Dritte Eigenschaft. Hier geht es eher um die Zeit und das Zeitverhalten. Die Dauer von
115 jeder Zustandsabfrage und entsprechenden Regelung. Die Dauer jeder Kombination ist
116 beschränkt. Würden Sie sagen, dass das wichtig oder eher unwichtig ist? #00:11:45-5#
- 117 12. B: Die Dauer von jeder Regelabfolge ist beschränkt. #00:11:50-9#
- 118 13. I: Oder nehmen wir statt beschränkt, definiert. #00:11:55-0#
- 119 14. B: Das ist unwichtig. #00:11:59-3#
- 120 15. I: Okay, warum? #00:12:00-6#
- 121 16. B: Weil (...) Ich versuche das Ganze gerade in die Praxis umzusetzen, weil so/ Was verstehen Sie
122 genau unter Dauer? #00:12:28-8#
- 123 17. I: Wenn ich mir jetzt vorstelle, dass ich so etwas wie eine kleinere Anlage hätte und vielleicht ein
124 Heizelement, das immer wieder auf einem gewissen Punkt gehalten werden muss. Also den,
125 keine Ahnung, die Flüssigkeit, die in dem Kessel sich befindet, auf einem gewissen
126 Temperaturpunkt dann halten muss. Dann würde ja immer wieder abgefragt werden und wenn
127 der kontinuierliche Zufluss zum Beispiel schwankende Temperatur hat, müsste das Heizelement
128 entsprechend natürlich gegen regeln. Und dann ist die Frage, wenn ich immer die Kombination
129 aus Messung und Regelung hab, wie lange darf das dann dauern, dass die Regelung auf die
130 Messung reagiert? Oder wie lange dürfen diese zwei // B: Ach so, okay. // Elemente auseinander
131 liegen. Also im Grunde, wenn man das dann auf eine SPS umlegen würde, wie lange dürfte dann
132 die Zykluszeit sein? #00:13:25-6#
- 133 18. B: Ja, das ist auch ganz wichtig, weil es sehr unterschiedliche Prozesse gibt. Unterschiedliche
134 Temperaturregelung in einem Behälter, wo der Behälter hundert Tonnen Wasser beinhaltet wird
135 die Dauer nicht so wichtig sein. Aber bei einer Dampfdruckregelung auf eine Dampfturbine ist
136 die Zykluszeit in Mikrosekunden. Also, ja und nein. #00:13:58-7#
- 137 19. I: Also für mein Verständnis, wäre es so // B: Also es ist wichtig/ //, dass es auf die Trägheit des
138 Systems auch ankommt natürlich. #00:14:06-0#
- 139 20. B: Genau, genau. #00:14:09-2#
- 140 21. I: Aber wenn man jetzt gar nicht die Möglichkeit hätte, eine zeitliche Bedingung für die Regelung
141 zu setzen. Dass man sagt, Okay, im Grunde könnte der Regelprozess ewig dauern. Das wäre

142 wahrscheinlich auch unpraktikabel. Aber ich will Sie hier nicht beeinflussen. Das ist nur so meine
143 Überlegung zu einem Thema. #00:14:29-9#

144 22. B: Naja, es kommt immer auf diesen Regelprozess an. Es gibt sicher in Astronomie Regelprozesse,
145 die mehrere Jahrtausende, Jahrmillionen dauern. Da ist es unwichtig. Aber es gibt
146 Regelprozesse, die sehr schnell ablaufen müssen. Ja, dann ist es wichtig. Ich sage einmal, ja, es/
147 wenn man diesen Regelprozess, ja sich anschauen muss und beurteilen muss, wie schnell muss
148 mein Regelprozess wirken, damit hinten das richtige richtige/ die richtige Konsistenz von Bier
149 herauskommt, oder richtiger Zuckergehalt. Also, sagen wir ja. #00:15:27-1#

150 23. I: Okay. Aber wie gesagt, ich wollte Sie nicht beeinflussen also. #00:15:31-4#

151 24. B: Ich habe am Anfang die Frage nicht ganz verstanden. Das mit Dauer kann man bei einem
152 Regelprozess so viel/ also man kann einiges falsch verstehen, sage ich mal. Aber wenn es die
153 Zyklusdauer von diesem Regelprozess ist, ja. Dann ist es schon verständlicher. Ja. #00:16:01-1#

154 25. I: Okay, dann notiere ich mir das vielleicht, dass ich beim nächsten Interview das einfach genauer
155 umschreibe oder gleich ein bisschen anpasse. Die Fragestellung. Danke. Werde ich mir gleich
156 aufschreiben. Okay. Vierte Eigenschaft. Wenn Zustandsabfragen gewisse Ergebnisse, definierte
157 Ergebnisse liefern, wird das System beendet. Soll heißen, kann auch in einen anderen Zustand
158 übergeführt werden. Würden Sie sagen, dass das wichtig ist oder unwichtig? #00:16:38-7#

159 26. B: Bei einem kontinuierlichen Prozess eher unwichtig. #00:16:42-3#

160 27. I: Okay. #00:16:43-2#

161 28. B: Weil ein kontinuierlicher Prozess ist meistens so aufgebaut, dass es keine Zustände gibt.
162 Sondern eigentlich dann diese Zustände mit Regelkreisen abgefangen werden. Also in einem
163 Modell von einer kontinuierlichen Anlage/ ja, ist/ Schon. Also Zustände in dem Sinne gibt es in
164 einem kontinuierlichen Regelkreis oder in einem kontinuierlichen Prozess nicht, meiner Meinung
165 nach. #00:17:17-8#

166 29. I: Vielleicht ein bisschen weitergehend mit dem Verständnis. Was ist zum Beispiel, wenn wir
167 etwas wie Grenzwerte definiert hätten? Also Grenzwerte, wo man schon weiß, okay, es kann
168 sein, dass diese Fehler auftreten und entsprechende Messergebnisse dazu führen könnten, dass
169 das Heizelement zum Beispiel unkontrolliert hoch heizt. Könnte das eventuell ein
170 Anwendungsfall hierfür sein? #00:17:54-6#

171 30. B: Ja, aber die Grenzwerte/ also typisch jetzt irgendwelche High, Low-Alarme, die wirken ja nicht
172 auf den Prozess aus. Die wirken/ die sind eigentlich nur für den Operator als Hinweis oder für
173 den Bediener als Hinweis, hey, da passiert was. Mach was. Da. Aber so ein Prozess sollte ohne
174 Alarme, ohne Grenzwerte, stattfinden, ein kontinuierlicher Prozess. #00:18:31-9#

175 31. I: Natürlich mit dem Hintergrundgedanken, was ich vorhin gemeint habe. Wir wollen mit diesen
176 Modellen das Verhalten einer kontinuierlichen Anlage oder eines kontinuierlichen Prozesses
177 auch abbilden können. Das heißt, wir müssen auch irgendwie solche Verhaltensweisen, die zum
178 Beispiel durch einen Bediener abgefangen werden könnten, diese Meldungen, müssten wir auch
179 irgendwie darstellen können. Und deswegen ist das auch ein bisschen hier hereingekommen,
180 diese Überlegung. Also wenn es sich // B: Ja. // Entschuldigung, nur ganz kurz. Wenn es sich um
181 ein Prozessleitsystem handeln würde, wie es effektiv umgesetzt wird, verstehe ich natürlich,
182 dass man das anders machen würde. Aber wir zielen nicht darauf ab, eine Alternative unbedingt
183 für also eine komplette Orchestrierungsalternative oder so etwas anzubieten. Sondern eher mal
184 im ersten Schritt zumindest die Prozesse auch so abbilden zu können. Darum geht es uns auch.
185 #00:19:34-7#

- 186 32. B: Aber für die Prozessabbildung sind Grenzwerte meiner Meinung nach nicht relevant. Für eine
187 reine Prozessabbildung. So wie Sie es jetzt veranschaulichen wollen mit dem BPMN-Modell.
188 #00:19:55-1#
- 189 33. I: Okay, könnten Sie das noch kurz begründen? #00:19:59-9#
- 190 34. B: Weil Grenzwerte in einem Regelkreis oder in einem Zustandsregelkreis, so wie Sie es
191 beschreiben, in erster Linie sich nicht auswirken sollten. Sonst wäre es kein kontinuierlicher
192 Prozess. Man muss den kontinuierlichen Prozess so designen, dass es gar nicht zu diesen
193 Grenzwertüberschreitungen kommt. #00:20:35-4#
- 194 35. I: Aber könnte man argumentieren, dass man hiermit versucht, eigentlich den Idealfall
195 abzubilden und nicht vielleicht das reale Verhalten? Sagen wir mal eine Anlage zum Beispiel?
196 Weil natürlich, wird das dann eher ausgelegt darauf, dass die Anlage unter gewissen
197 Bedingungen mit idealen Parametern läuft. Und ein Regelkreis soll eigentlich dabei helfen,
198 diesen Zustand zu behalten. Und das sollte, wie Sie richtig sagen, also auch von meinem
199 Verständnis her, sollte es gar nicht soweit kommen, dass man in diese, dass man überhaupt in
200 die Nähe dieser Grenzwerte kommt. Aber bei realen Anlagen, oder bei realen Prozessen, kann es
201 ja trotzdem vielleicht durch irgendeinen Fehler oder irgendetwas passieren, dass es soweit
202 kommt. #00:21:35-4#
- 203 36. B: Genau. Natürlich macht man es ja, in der Praxis setzt man Grenzwerte. Damit, wie gesagt, der
204 Bediener dort händisch/ oder die vom Prozess nicht berücksichtigt werden. Der Prozess kann
205 durch so viel verschiedene Sachen gestört werden. Prozessbedingte Aktionen oder Störungen
206 oder je nachdem. Kann dieser gestört werden. Und da stellt man schon Grenzwerte ein und
207 reduziert die aber rein für eine kontinuierliche Anlage als Modell würde ich es jetzt nicht
208 unbedingt als wichtig kennzeichnen. #00:22:41-5#
- 209 37. I: Okay. Dann kommen wir gleich zur nächsten Eigenschaft. Und zwar, wenn man jetzt davon
210 ausgeht, dass das System in einen anderen Zustand übergeht oder herunter gefahren werden
211 muss oder wie auch immer. Eventuell beendet wird. Dann muss man es in einen konsistenten
212 Zustand bringen. Das heißt, könnte man sagen, in einen sicheren Zustand überführen, wenn eine
213 Fehlermeldung geschieht. Oder das soll/ Eigentlich könnte man sagen, dass wird die Aktion, die
214 der Operator dann vorgeschlagen bekommt. Die er dann machen soll. Würden Sie sagen, dass
215 das wichtig oder unwichtig ist und warum? #00:23:27-2#
- 216 38. B: Ja, das ist wichtig, dass der also/ Aber auch nicht in dieser Darstellungsart. Sondern eher als
217 Arbeitsanweisung oder als/ Es soll nicht Bestandteil dieser Dokumentation oder dieser
218 Darstellung sein. Es soll eher in einer anderen Form dargestellt werden, die mehr verbal oder
219 verbal technisch versiert niedergeschrieben ist oder dokumentiert ist, wie dieser Prozess in
220 einen sicheren Zustand abgefahren wird. Weil dieser sichere Zustand ist ja nicht Teil dieses
221 kontinuierlichen Prozesses. #00:24:28-7#
- 222 39. I: Stimmt ja. Okay, danke. Und dann noch die letzte Eigenschaft. Das resultierende System soll
223 für Menschen verständlich sein. Soll heißen, die Art und Weise, wie ein kontinuierlicher Prozess
224 dargestellt wird, von unserer Seite her, wenn wir wirklich von digitalen Abbildern sprechen, soll
225 für verschiedenste Personen verständlich sein. Würden Sie sagen, dass das wichtig für Sie als
226 jemand ist, der das wirklich in der Praxis ausführt? Oder damit arbeitet? #00:25:10-9#
- 227 40. B: Nein, weil der Informationsgehalt für einen Softwareentwickler ist wesentlich höher, wie für
228 das Management, wie für die Managementebene, der diese Dokumentation dann für eine
229 Präsentation oder vor dem Kunden, Lieferanten et cetera. präsentieren möchte. Also ich würde
230 das nicht, dieses Schema, dieses Modell, für extrem/ also die eine Extremseite für einen
231 Softwareentwickler und für die andere Extremseite das Management von dem Unternehmen
232 oder von der Anlage oder von dem Prozess bestimmen. #00:26:07-0#
-

- 233 41. I: Okay, gut. Natürlich ist es die Frage inwiefern, also, wie weit man nach oben gehen möchte bei
234 so einem Unternehmen, wenn es um die Verständlichkeit der eigentlichen Prozess geht, die
235 wirklich im eigenen Unternehmen durchgeführt werden oder ablaufen sollen. Ja. #00:26:31-5#
- 236 42. B: Es ist halt immer schwierig, da alle Persönlichkeiten und alle Wissens/ Alle Personen mit
237 unterschiedlichen Wissen, mit einem Abbild, mit einem Modell, auf den gleichen Stand zu
238 bringen. Also für einen Programmierer wird ein Modell, das das Management verwendet, für,
239 wie gesagt, irgendwelche Präsentationen et cetera wird nicht ausreichen, um diesen Prozess
240 lückenlos darzustellen. Und für das Management wird die Funktionsbeschreibung zum Beispiel
241 oder dieses Modell, das zum Beispiel auf eine Funktionsbeschreibung aufbaut, fürs Management
242 einfach zu umfangreich sein. Für eine Präsentation. #00:27:24-9#
- 243 43. I: Okay. Und was würden Sie sagen, wenn es die Möglichkeit gebe, die grundsätzliche
244 Prozesslogik in einem relativ einfachen Modell darzustellen und in diesem Modell aber die
245 Möglichkeit, auch hineinzubauen, ins Detail zu gehen, und sich wirklich einzelne Prozessschritte
246 noch detaillierter anschauen zu können, wenn man das möchte? Aber die grundsätzliche Logik
247 wäre auf der obersten Ebene recht einfach dargestellt. Wenn es sich hier um ein
248 mehrschichtiges Modell handeln könnte. Wie gesagt, mit Subprozessen, wie man sie aus BPMN
249 zum Beispiel kennt. #00:28:05-9#
- 250 44. B: Ja. (...) Wäre sicher möglich. Der Aufwand ist sicher enorm. Und das würden trotzdem noch
251 einige Informationen für den Softwareentwickler fehlen. Beziehungsweise es ist dann für/ also
252 ich habe die Erfahrung gemacht, dass es am besten ist, Funktionsbeschreibungen für solche
253 Prozesse zu verbinden. Beziehungsweise Prozessbeschreibung erstellen vom
254 Produktionsunternehmen oder vom Prozesstechniker erstellen zu lassen. Und nach diesen zu
255 programmieren. Und diese dann in die Logik zu implementieren. Weil solche Modelle dann/ also
256 manche Modellschritte können in den verschiedenen Leitsystemen nicht so dargestellt werden,
257 wie sie dann in dem Modell präsentiert sind. Und das ist halt ganz schwierig so etwas
258 abzudecken. Also ich bin nicht so der Freund von Modellen, die bis ins kleinste Detail den Ablauf
259 zeigen, also den Prozess abbilden. #00:29:50-4#
- 260 45. I: Dürfte ich nur fragen, weil Sie gerade erwähnt haben, wenn das möglich ist, dass so etwas
261 recht aufwendig sein kann, oder dass Sie hier eventuell schon, wie Sie gerade formuliert haben,
262 dass das eventuell schwierig sein könnte, hätten Sie da vielleicht ein Beispiel? Ein Prozessbeispiel?
263 #00:30:09-4#
- 264 46. B: Ja, ich habe (Jahr) in (Stadt) ein Fernwärmekraftwerk geplant. Also Kraftwerk. Einen
265 Fernwärmespeicher geplant. Dort habe ich auch Funktionspläne nach dem diesem VGB EA 170 C
266 Modell erstellt. Und diese dann an einen Programmierer weitergeben, der für die Firma (Name
267 der Firma), also fürs Leitsystem (Firma), was dort auf dem Kraftwerk, in dem Kraft schon
268 eingesetzt worden ist, übergeben. Und da waren so viele Fehler drinnen und so viele
269 Ungereimtheiten, dass wir mit diesem Programm ewig nichts anfangen haben können.
270 Mittlerweile verlange ich von den Kunden meistens nur Funktionsbeschreibungen weil die am
271 einfachsten für den Prozesstechniker zum Schreiben sind. Oder zum Erstellen sind. Und die
272 einzelnen Details, die werden dann in mündlicher oder in schriftlicher Form dann abgeklärt. Da
273 bin ich eher der Freund davon. #00:31:46-6#
- 274 47. I: Okay, ich will Sie jetzt nicht zu lange damit aufhalten, aber es ist halt jetzt interessant, dass wir
275 auf den Punkt gekommen sind. Weil im Grunde, wenn man jetzt sagt, man hätte also um solche
276 Fälle zu vermeiden, dass zum Beispiel Code sehr unübersichtlich gestaltet ist, nicht
277 nachvollziehbar, sehr individuell ausfällt. Dass man eventuell auf grafische Varianten umsteigen
278 würde. So etwas wie Funktionsblöcke beispielsweise, wo man ein definiertes Schema hat und
279 eigentlich nicht wirklich viel falsch machen kann. Weil man eventuell sogar nur Linien ziehen
280 muss und gewisse Parameter halt definiert. Aber im Grunde die eigentliche Aktion vordefiniert
281 ist. Die eigentliche Operation könnte man sagen. Ob das nicht dann sogar ein Vorteil wäre?
282 #00:32:46-5#
-

283 48. B: Das ist sicher ein Vorteil. Nur diese Funktionsbeschreibung, also Funktionsdiagramme. Nur
284 nach denen kann man auch nicht hundertprozentig programm/ Also man bekommt diesen
285 Informationsgrad nicht von einem Leitsystem. Oder man kann diese Funktionspläne nicht
286 hundertprozentig leitsystemunabhängig oder SPS-unabhängig aufbauen. Das geht nicht. // I:
287 Was würde/ // meiner Meinung nach. #00:33:21-6#

288 49. I: Was würden denn da für Informationen fehlen? #00:33:24-2#

289 50. B: Boah. Wo soll ich anfangen? Wo soll ich aufhören? Nein, es/ #00:33:28-6#

290 51. I: Beispiele. (lacht) #00:33:29-8#

291 52. B: Beispiele. Conditional Alarming. Wenn jetzt eine Pumpe nicht läuft, soll auch der
292 Trockenlaufschutz dieser Pumpe stumm geschaltet werden. Diese soll nach erst 15 Sekunden
293 nach Pumpenstart dann aktiviert werden zum Beispiel. Also ich habe Funktionspläne bereits
294 gezeichnet. Aber so ins Detail nicht. Oder im Fehlerfall. Wenn der Messwert ausfällt, was soll
295 dann passieren? Wenn der Messumformer ausfällt, was soll dann passieren? Und ja, solche
296 Dinge. #00:34:33-3#

297 53. I: Das heißt, man könnte sagen, Sie versuchen dann/ Also, Sie bräuchten die Informationen,
298 wenn gewisse/ Wo haben wir das? (lacht) Ganz kurz. Wenn Zustandsabfragen gewisse
299 Ergebnisse liefern, wird das System beendet oder in einen anderen Zustand übergeführt. Also
300 quasi Reaktion auf eine gewisse Meldung. #00:35:02-8#

301 54. B: Auf/ Ja. #00:35:06-0#

302 55. I: Okay, ja. Also genau so etwas versuchen wir eigentlich auch in unseren Modellen darstellbar zu
303 machen. Ich sage jetzt nicht, dass BPMN die ultimative Alternative zu so etwas, zu wie zum
304 Beispiel SPS, die verschiedenen Standards wie man das programmieren kann, darstellt. Aber es
305 gibt halt auch die Möglichkeit, wie Sie gesagt haben. Im BPMN-Standard gibt es Timer Events,
306 die genau für solche Pausen zum Beispiel oder für solche zeitlichen Intervalle hergenommen
307 werden können, um so etwas in einem Prozess zu modellieren. Also nur so nebenbei gesagt. Es
308 gebe für solche Operationen die Möglichkeit, genau so etwas darzustellen in BPMN. By the Way
309 nur. Okay. Dann würde ich sagen, um ein bisschen Zeit zu sparen, springen wir gleich zu den
310 Extensions. Und zwar. Ich werde Ihnen Prozesse zeigen, die mit BPMN 2.0 und mit unseren
311 Erweiterungen modelliert wurden. Die Erweiterungen sollen zum einen vordefinierte
312 Modellierungskonventionen für in der Prozess- und Steuerungstechnik übliche Routinen
313 bereitstellen. Und zum anderen helfen, die Unterschiede zwischen den parallelen Pfaden in den
314 Prozessmodellen zu visualisieren. Die Prozesse werden in der [REDACTED], kurz
315 [REDACTED], modelliert. Und für diese, für das Verständnis der [REDACTED] Modelle müssen drei zusätzliche
316 Symbole erklärt werden. Nämlich HTTP Service Calls, Scripts und die Kombination aus beiden,
317 also HTTP Call mit Script. Diese habe ich Ihnen vor unserem Interview kurz gezeigt. Würden Sie
318 sagen, dass Sie dazu noch Fragen haben oder wissen Sie grundsätzlich, wie die drei funktionieren?
319 Also ist das verständlich gewesen oder wollen/? #00:36:59-9#

320 56. B: Ja, schauen wir mal. #00:37:01-2#

321 57. I: Okay, gut. #00:37:02-7#

322 58. B: Mir ist es schon verständlich ja. Aber ob ich es jetzt so in die Praxis umsetzen kann. Schauen
323 wir mal. #00:37:12-4#

324 59. I: Okay. Es geht darum, dass wir sie auch erwähnt haben. Aber wahrscheinlich ergibt sich das
325 dann eh auch durch die Prozessmodelle selber. Wir kommen jetzt aber zu den Erweiterungen,
326 die wirklich im Zuge auch meiner Dissertation, also aus unserem Projekt entstanden sind. Die
327 erste ist eine Unterkategorie von einem Gateway, eine Erweiterung. Und zwar das Closed Loop
328 Subsystem Gateway. Dieses Gateway ist eine Kombination aus einem inklusiven und einem

ereignisbasierten Gateway. Es enthält Verzweigungen, also beziehungsweise Kanten. Das heißt die Striche, die von diesem Gateway weggehen würden, die für die Zustandsabfragen und Regulierungsphasen des Zyklus ausgelöst werden. Sowie Verzweigungen, Kanten, die beim Empfang von Abbruchereignissen ausgeführt werden. Also das heißt, jede dieser Kanten wird durch ein gewisses Ereignis weiter abgearbeitet. Die Ereignisse und Tasks in den einzelnen Kanten sind unabhängig voneinander. Damit erfüllen wir das erste der oben genannten Features, also Unabhängigkeit, Parallelität, dass einzelne Verläufe unabhängig voneinander sind und sie parallel ausgeführt werden. Das Gateway ermöglicht außerdem die Definition der Intervalldauer jedes Zyklus sowie von Überschreitungsbedingungen. Das sind dann Bedingungen, die man modellieren kann, die zeige ich Ihnen dann gleich ein bisschen weiter unten. Mit warten beziehungsweise wait oder abbrechen beziehungsweise cancel. Und auch, was man auch definieren kann. Die Ausführungsreihenfolge für Zustandsabfragen und Regulierungen beziehungsweise Mess- und Steuerungsaufgaben. Hier sehen Sie zwei Beispiele, wie man das Ganze definieren kann. Also man kann einerseits bei Intervall Duration Overrun cancel oder wait definieren. Oder wenn man möchte parallel oder sequentiell. Zu cancel oder wait. Wenn wait gewählt wird, beginnt die nächste Iteration wenn alle Kanten, Verzweigungen, beendet sind. Das heißt, wenn der gesamte Prozess in diesem Strang abgeschlossen ist. Und die festgelegte Intervalldauer erreicht ist. Bei cancel definiert die Intervalldauer genau die Zeit, in der jeder Zweig zu beenden ist. Das heißt, wenn die Tasks in einem Zweig schneller beendet werden, wird der Zweig warten. Wenn noch nicht alle Tasks beendet sind, werden sie abgebrochen. Die andere Variante, das andere Feature, das man definieren kann oder das andere Attribut. Parallel oder sequentiell. Bei parallel werden die Tasks nach Measure und Control Events, also Zustandsabfragen und dann Regulierungen, parallel ausgeführt. Bei sequentiell werden die Tasks nach Control Events erst ausgeführt, nachdem alle Tasks nach Measure Events beendet sind. Das heißt, wenn ein Measure Event, eine Zustandsabfrage, getriggert wird, das heißt, hier wird jetzt gemessen. Dieser Prozess in diesem Strang wird abgearbeitet. DANN würde erst, wenn das abgeschlossen ist auch von/ würde erst das Ereignis der Control Events getriggert werden und dann würden erst diese Tasks abgearbeitet werden. Diese Events, von denen ich jetzt gesprochen habe, also Measure und Control, beziehungsweise Zustandsabfragen und Regulierungen. Hier haben wir auch eigene Timer Events definiert. Intermediate Catching Events. In einem Closed Loop Subsystem werden spezifische Ereignisse erwartet, die in einer der drei folgenden Kategorien fallen. Ereignisse für Zustandsabfragen oder Messungen. Ereignisse für Regelungen also beziehungsweise Regulierungen und Ereignisse für die Unterbrechung des Closed Loop Subsystems. Es gibt für jede Ereigniskategorie zumindest eine Kante, die vom Gateway ausgeht. Die Kanten zeigen an, welche Tasks nebeneinander laufen. Sobald diese Ereignisse eintreten, werden auch die Tasks, die Aufgaben, die in den Kanten danach angeordnet sind, ausgeführt. Hier sehen Sie ein Bild von einem Closed Loop Subsystem, das kommt hier gleich, in dem nur Ereignisse der drei Kategorien, ohne darauffolgende Tasks modelliert sind. Das heißt, das würde so grafisch ausschauen. Dieses Karo hier steht für das Closed Loop Subsystem. Also wir haben das Start-Event, dann haben wir unser Gateway. Hier unten wird es wieder zusammengeführt. Und hier haben wir das End Event. Wir haben ein Event, also hier den Gauge, für Measure Events. Wir haben ein Symbol für Control Events. Und wir haben eines eingeführt für Abbruch-Events. Die drei Ereigniskategorien, die wir definiert haben, sind wie folgt. Measure. Dieses Symbol empfängt Events für die Ausführung von Tasks in Messzyklen. Control empfängt Events für die Ausführung von Tasks in Regelzyklen. Und Cancel empfängt Events für das Abbrechen von Closed Loop Systems. Diese Symbole wie gesagt, geben den Zweck der nachfolgenden Aufgaben an. Diese Tasks werden nur ausgeführt, wenn die Ereignisse auch entsprechend ausgelöst werden. Das bedeutet, dass das Messereignis angibt, dass die nachfolgenden Symbole nur für Messabläufe gedacht sind, Zustandsabfragen. Das gleiche gilt auch für Regulierungs-, Kontroll- und Abbruchereignisse. Für Zustandsabfragen und Regulierungen können wir eine Zykluszeit definieren. Dadurch kann die Dauer von Anpassungen also von Regulierungen im System definiert werden. Je nachdem, ob das Closed Loop Subsystem einen parallelen oder sequentiellen, oder einen Wait oder Cancel-Ansatz verfolgt, kann man ja definieren, läuft die Ausführung unterschiedlich ab. In diesen Bedingungen kann man definieren inwiefern Anpassungen beim System auch wirklich erfolgen. Hier sehen Sie ein Closed Loop Subsystem mit einem Task, eine Aufgabe, für eine Messung. In diesem Fall wird das Ereignis für die Messung alle zehn Sekunden getriggert. Danach wird der Wert V 1 geholt beziehungsweise

386 gemessen. Wait bedeutet hier, dass ein neuer Zyklus erst startet, wenn die Messung, das heißt
387 der Prozess in dieser Kante, abgeschlossen ist. Mit Cancel wird nach zehn Sekunden automatisch
388 der neue Zyklus gestartet. Das heißt, hier sehen Sie mal das Ereignis an sich mit dem runden
389 Symbol markiert. Und sobald das eintrifft, also sobald diese zehn Sekunden entsprechend/
390 Sobald diese zehn Sekunden dann abgelaufen sind, also sobald der Zyklus wieder neu gestartet,
391 wird das hier weiter ausgeführt, entsprechend bei cancel. Also das ist dann wirklich die harte
392 Definition dafür, wie lange das dauern darf. Ansonsten wenn wir hier auf wait gehen, würde die
393 Kante hier wirklich würde das/ der neue Zyklus erst starten, also der neue Durchlauf in dem
394 Closed Loop System, wenn alle Kanten wirklich erledigt sind. Das heißt, wenn auch das Get
395 Request, HTTP Call, durchgegangen ist und entsprechend ein Ergebnis zurückgebracht hat. Wenn
396 dieser Prozess, wenn die Übertragung dieses Requests, aber länger dauert, dann würde hier der
397 Prozess entsprechend warten. Nur im Vergleich. So hier sieht man noch kurz, wie man das
398 definieren kann. #00:44:49-2#

399 60. B: Ganz kurze Frage. Wo würden Sie so einen Regelkreis einsetzen? #00:44:54-6#

400 61. I: Wie meinen Sie, wo? #00:44:59-3#

401 62. B: Die Praxis, in der Praxis. Also was/ Können Sie nochmal ein bisschen hinaufgehen? Was würde
402 das/ Ich meine, ich habe da jetzt eine Frequenz, 0.1 Hertz, dass ich irgendwo gemessen habe.
403 Nach der Messung warte ich X Sekunden. Und nachher soll mit diesem Control/ Was genau
404 macht dieses Symbol oder was passiert dort? #00:45:36-5#

405 63. I: Ach so, das Control. Also das, was hier jetzt derweil nur dargestellt ist, ist nur einmal das
406 Symbol. Es passiert noch nichts nach dem Control. Also das sind jetzt nur mal die Beispiele, dass
407 nur etwas hinter Measure passiert. #00:45:49-6#

408 64. B: Okay. #00:45:51-0#

409 65. I: Also das, was/ Da kommen noch weitere Beispiele, das sehen Sie hier im weiteren Verlauf. Das
410 ist jetzt nur die schrittweise Erklärung, wie das funktionieren soll. Und wegen dem Intervall. Ich
411 weiß nicht, was braucht denn so lange? Was wird denn nur alle zehn Sekunden überprüft? Ich
412 könnte mir vorstellen, dass ein Thermostat relativ träge ist, je nachdem, was für ein Heizsystem
413 verwendet wird und vielleicht da nur alle zehn Sekunden die Raumtemperatur überprüft wird.
414 #00:46:17-0#

415 66. B: Okay. #00:46:18-3#

416 67. I: Beispielweise. #00:46:19-4#

417 68. B: Meistens ist das irgendeine SPS, die mit einer bestimmten Zyklus Zeit X immer wieder abläuft.
418 Und diese Wartezeit kann ich ja nicht beeinflussen. Weil der fragt alle zehn Millisekunden die
419 Temperatur ab. Und macht je nachdem, ob es ihm zu kalt ist oder zu warm ist, macht er das
420 Heizventil/ oder die/ schaltet die Heizung ein oder schaltet sie aus. Aber ich kann diese
421 Wartezeit ja nicht wirklich beeinflussen. Weil die ja durch die SPS gegeben ist. #00:47:05-8#

422 69. I: Ja, genau. Wir wollen auch nicht beeinflussen, wir wollen es nur darstellen. #00:47:09-6#

423 70. B: Okay. #00:47:10-4#

424 71. I: Wie gesagt, es soll einfach nur ein Abbild des Prozesses sein. #00:47:14-1#

425 72. B: Okay. #00:47:15-3#

426 73. I: Also es soll nur zeigen, wir können grundsätzlich vielleicht auch solche Details eines Prozesses
427 // B: Das reicht dann. Okay. // darstellen. Aber wir wollen jetzt nicht die Zeit unbedingt
428 überschreiben oder so etwas. Also wenn, nämlich die Hardware dazu. Wie zum Beispiel halt ein

429 Thermostat, das gar nicht erst ermöglicht, dass man die Zykluszeit anpasst. Natürlich können wir
430 das dann auch nicht umsetzen. Also, das ist uns bewusst oder mir. (lacht) #00:47:41-8#

431 74. B: Okay. #00:47:42-7#

432 75. I: Nein, das ist jetzt nur einmal die Darstellungsweise. #00:47:45-5#

433 76. B: Nur Darstellung. Okay. #00:47:47-2#

434 77. I: Wir können dann auch noch definieren, wenn wir das wollen, welcher Wert grundsätzlich
435 überschrieben wird. Das heißt, die Process Engine, für die wir die Erweiterungen definiert haben,
436 ermöglicht auch das Definieren von globalen Variablen, die auch entweder für Default Values
437 hergenommen werden können oder die ganze Zeit über konstant bleiben. Zum Beispiel für
438 Grenzwerte, die die ganze Zeit über für diesen Prozess gelten oder einfach überschrieben
439 werden beispielsweise und die man dann in anderen Subprozessen oder an einer anderen Stelle
440 des Prozesses einfach nutzt. Ja, man kann die Frequenz definieren. So. Mit Hilfe von
441 Regelungsereignissen kann ferner festgelegt werden, welches Reglermodell verwendet wird.
442 Also hier können wir zum Beispiel, um halt nachvollziehen zu können, welcher Regler eingesetzt
443 wird, auch sagen, okay, PD, PI, PID, wie auch immer. Diese Regler werden in ihrer
444 mathematischen Form dargestellt. Die Tasks für sie sind im Grunde Berechnungen, die in festen
445 Teilprozessen dargestellt werden sollen. Das heißt, man könnte einfach sagen okay, ich habe
446 genau dieses Script, diese Berechnung oder Reihe von Berechnungen eines PID-Reglers oder PI
447 oder einfach nur ein P-Glied, wie auch immer, wenn es was ganz einfaches ist. Und das könnte
448 man halt wie wenn man es zum Beispiel in MATLAB simulieren wollen würde oder berechnen
449 wollen würde einfach die mathematischen Formeln dazu programmieren entsprechend. Nach
450 diesen Berechnungen kann der Benutzer Tasks zur weiteren Datenverarbeitung hinzufügen.
451 Wenn man das will. Dies kann auch nach Mess-Tasks geschehen. Das heißt, wenn man/ man
452 kann das natürlich auch machen, wenn man jetzt ein Messergebnis hereinkriegt und mit dem
453 noch umrechnen möchte. Und die Mess-Task könnte man auch noch als Datenerfassung-Tasks
454 bezeichnen, also auch einfach zum Datensammeln. Wir können zum Beispiel sagen, wenn wir
455 jetzt so ein Prozessmodell haben, und wir wollen jetzt aber nicht unbedingt mit einer Regelung
456 arbeiten, sondern wir wollen einfach nur zyklisch Daten sammeln. Dann würde man hier nach
457 dem Measure Event einfach nur entsprechende Service Calls hinein modellieren, die einfach nur
458 Werte holen. Das kann man auch machen. So, und jetzt haben wir hier im Prozessmodell, das
459 sehen Sie, mit einem Wert, der gemessen wird und einer darauffolgenden Regelung. Das heißt,
460 wir haben hier einen Wert, der wird gemessen. Dann wird auch das Control-Ereignis getriggert.
461 Und wir haben dann zum Beispiel das Script, das einfach die Differenz zwischen zwei Werten
462 berechnet. Wir können hier sagen, wir haben hier ein Script mit PID Code. Wir können aber auch
463 einen ganzen Subprozess definieren, wenn wir das wollen. Also wir müssen das nicht in einzelne
464 Schritte aufspalten. Also wir können auch sagen, okay, das hier oder das zusammen läuft in
465 einem vordefinierten Subprozess ab. Und der würde dann wieder in einem eigenen
466 Prozessmodell abgebildet werden. Und dann kann man/ Hier sehen Sie dann, wie das mit den
467 globalen Variablen ausschauen würde, mit den Datenelementen. Ich will jetzt nicht in die [REDACTED]
468 selber ins User Interface wechseln, aber man könnte hier zum Beispiel bei den End Points, wenn
469 man den Reiter einfach wechselt, die entsprechenden Adressen für die HTTP oder OPC UA Calls
470 oder was auch immer, definieren. Also man hat dann einen Identifier für die Adresse. Zum
471 Beispiel Get T 1 für Temperatur in Kessel 1 oder so etwas. Und dann hätte man die Adresse mit
472 einem Identifier, also Get Underline HTTP Bindestrich zum Beispiel und dann eigentlich die
473 Adresse, wo man es abgreift. Also das könnte dann zum Beispiel über einen Webserver HTTP
474 Doppelpunkt, Slash, Slash, irgendetwas, IP-Adresse, Port, und dann der restliche Identifier sein,
475 wie man das auch immer ansprechen würde. (...) Für Control kann man auch ein paar Attribute
476 setzen, wenn man das will. Man kann wie gesagt definieren, was für ein Controller Type das ist.
477 Welche Variable überschrieben werden soll. Man könnte aber auch zum Beispiel Grenzwerte
478 definieren, wenn man das will. Also da haben wir uns gedacht, eventuell wäre das sinnvoll, das
479 beim Controller einzusetzen, ja. Eventuell könnte man es aber auch an anderer Stelle definieren,
480 wie man das möchte. Wait bedeutet wieder, dass für den nächsten Zyklus auf das Beenden aller
481 Tasks gewartet wird. Wie schon oben erklärt. Auch auf die Regulierungs-Tasks. Und sequentiell

482 heißt, dass die Tasks dann nacheinander ausgeführt werden. Das heißt, es wird erst gemessen
483 beziehungsweise der Zustand des Systems abgefragt. Und mit diesem gemessenen Wert wird die
484 Regelung durchgeführt. Dabei wird vom optimalen Wert V_{opt} , der aktuelle Wert V_1 abgezogen
485 und mit dieser Differenz der neue Stellwert ausgerechnet. Dieser wird dann mit einem Service
486 Request an das entsprechende Stellglied, an den entsprechenden Aktor geschickt. Also ein
487 Element, das aktiv Einfluss auf den Prozess ausübt. Und wenn man möchte, kann man die
488 Differenzberechnung wie gesagt, die Regelungsbeschreibung, also die Berechnung des
489 PID-Gliedes, den Code, und das Aussenden des Befehls an das System in einem Subprocess
490 zusammenfassen. Also da könnte man halt auch wieder sagen, okay, ich definiere hier das
491 einfach vor für den User und der müsste das nur noch hier hineinziehen. Bei Control können
492 zusätzlich die Art der Regelung, wie eben gesagt, und der neue Stellwert und dessen Limits
493 eingetragen werden. Bei parallel, wenn wir das verwenden würden, würde der letzte Wert von V_1
494 genommen werden, für den dann aber keine Zeitgarantie besteht. Sondern das wäre einfach
495 der Wert aus dem letzten Zyklus. Zustandsabfragen und Regulierungen sollten in regelmäßiger
496 Frequenz ausgelöst werden. Abbruchereignisse natürlich werden nur durch ihre
497 Abbruchbedingungen ausgelöst, die der Benutzer definieren kann. Ein Beispiel für ein
498 Abbruchereignis wäre, wenn etwas den Abbruch eines Zyklus auslöst, also zum Beispiel bei einer
499 SPS vielleicht eine Watch-Dog-Funktion, oder es könnte ein Notstopp sein. Das ist eventuell dann
500 sinnvoller. Und hier sehen Sie jetzt ein Beispiel mit einem zu messenden Wert, eine Regelung
501 und eine Abbruchbedingung. Das heißt, man könnte hier definieren nebenbei, was die Kondition
502 dafür ist, dass im Grunde dieses Abbruchereignis getriggert wird. Ich gebe nur zu bedenken, hier
503 ist jetzt noch nichts nachher definiert. Also keine Routine, als Aufräumroutine oder so etwas
504 eingefügt. Sondern es würde einfach nur dieses Ereignis getriggert werden und damit würde
505 man aus dem Closed Loop System heraus springen. Und die Condition hierfür haben wir gesagt,
506 ist einfach einmal als Beispiel Emergency Stop Active ist gleich true. Wir würden natürlich, dass
507 der Prozess durchläuft, davon ausgehen, dass er eigentlich auf false gesetzt ist, also dass kein
508 Notstopp gerade aktiv ist. #00:55:09-4#

509 78. B: Wie sehe ich das, dass der Prozess dort beim Emergency Stop hinausgeht, aus dem Gateway?
510 #00:55:19-6#

511 79. I: Das wäre im Grunde der Sinn hinter diesem Symbol. Also man müsste wissen, wie dieses
512 Symbol dann arbeitet. #00:55:27-1#

513 80. B: Okay, das springt über diesen Gateway drüber. #00:55:29-5#

514 81. I: Genau. Das würde dann hinaus springen. Also man könnte hier, wie bei Control, könnte man
515 hier Scripts, also das, was hier dranhängt, hiernach auch noch einfügen, nach diesem Symbol,
516 wenn man das möchte. Und das sind dann Aufräumroutinen oder man verfährt in einen anderen
517 Zustand oder so etwas. Das kann man dann einmalig definieren. Und dann würde man aber aus
518 diesem kontinuierlichen Ablauf hinaus springen und der Prozess, der hier modelliert wäre,
519 würde beendet werden. Man kann aber natürlich auch nach dem unteren Symbol, das im
520 Grunde das Closed Loop Subsystem wieder zusammenführt, wo sich die Kanten wieder treffen.
521 Also nach dem bis zum End Event auch bis hierher auch noch etwas definieren, wenn man das
522 will. #00:56:15-7#

523 82. B: Wieso zeichnet man das symbolisch nicht so, dass dieser Abbruchbefehl nach dem Gateway,
524 wo es wieder zum Ursprung zurück springt, dass es dort hinein geht? Also dass das/ oder
525 woanders hineinspringt? #00:56:38-8#

526 83. I: Weil diese Bedingungen auch immer wieder überprüft werden zyklisch. Ich hoffe, das
527 beantwortet das jetzt halbwegs. Also warum gehen wir hier parallel hinein? Weil/ #00:56:50-5#

528 84. B: Nein, das ist mir schon klar, warum das parallel ist. Aber warum ist dieser Ausgang von diesem
529 Emergency Stop/ Warum geht der auf diese Closed Loop wieder zurück? Auf dieselbe Linie wie,
530 wo der PID-Regler mit seinem/ Was hat denn der? Send Manipulated Value, mit diesem A 4. Und
531 warum springt er nicht darunter hinein? Nach dem Closed Loop Gateway? Rein von der

532 Darstellungsart. Weil das schaut so aus wie wenn dieser Emergency Stop, 'Ja, ist okay. Aber ich
533 springe wieder oben hinein und messe wieder.' Verstehen Sie, was ich meine? #00:57:43-3#

534 85. I: Ja, jetzt verstehe ich, was Sie meinen. Ja, wir haben uns natürlich/ ich kann nur sagen, wir
535 haben versucht, aus den bestehenden Beispielen vom BPMN-Standard das möglichst ähnlich zu
536 halten. Und es gibt halt gewisse Vorgaben, wie diese Kanten dann wieder vereinigt werden. Das
537 heißt, man müsste dann eventuell sogar etwas Eigenes noch intern in diesem Closed Loop
538 System vorsehen, damit sich die Abbruchbedingung wirklich abhebt. Also es ist ein bisschen
539 schwierig. Es ist ein guter Tipp für die Usability, dass man das nicht gleich sieht. Das ist ein guter
540 Input für uns. Weil ich könnte mir jetzt/ ich müsste mir dann überlegen, wie man das eventuell
541 etwas sichtbarer darstellt. #00:58:26-9#

542 86. B: Es ist ja so, für diesen Closed Loop, das (unv. #00:58:30-9#) den Gateway zum Schluss, dass
543 Strang A und Strang B oder Strang C, also es ist diese Linie unten, der da in diesem Gateway
544 hineingeht, ist es ein And und ein Or mit dem Emergency Stop. Das ist es Oder verknüpft mit der
545 Measure Line und mit der Control Line, die miteinander wieder eben unverknüpft sind. Also, von
546 der Darstellungsart gefällt mir das nicht wirklich. #00:59:06-1#

547 87. I: Ja, ich verstehe schon, was Sie meinen. Ja, das ist ein guter Punkt. #00:59:12-1#

548 88. B: Man sollte diese Abbruchbedingung eher seitlich in diesem Gateway oder weiter unten
549 einführen. Als separate Linie, damit man erkennt, ob das jetzt und oder oder verknüpft ist. Weil
550 die Measure Line ist mit der Control Line ja unverknüpft. Das heißt, der springt wieder in den
551 oberen Gateway hinein, von diesem Closed Loop. Wenn die Messung erfolgt ist und dieser
552 Kontrollschritt erfolgt sind. Dann springt er in den neuen Zyklus hinein. Aber nicht, wenn der
553 Emergency Stop ausgeführt worden ist oder wenn der aktiv ist. #00:59:55-4#

554 89. I: Ja, das stimmt. #01:00:01-2#

555 90. B: Und der nächste Punkt ist dieser Emergency Stop müsste eigentlich ganz oben sein.
556 Hierarchisch gesehen noch über dem Measurement-Punkt. Weil der ja als erstes ausgeführt
557 werden muss. Weil ich muss ja nicht mehr messen, wenn ein Notstopp, ein Pilz-Relais oder je
558 nachdem, aktiviert worden ist. Rein jetzt von der Abarbeitung. Man liest immer von links oben
559 nach rechts unten. #01:00:42-1#

560 91. I: Also modelliert ist es grundsätzlich so, dass es parallel ablaufen würde. Aber man könnte
561 natürlich standardmäßig sagen, dass immer zuerst die Abbruchbedingung überprüft wird.
562 #01:00:54-8#

563 92. B: Genau // I: Das könnte man machen. // Das muss so sein. #01:00:57-7#

564 93. I: Okay, ja. #01:00:58-5#

565 94. B: Meiner Meinung nach. #01:01:01-1#

566 95. I: Genau solche Schwachstellen versuchen wir halt auch herauszufinden. Also es ist sehr guter
567 Input. #01:01:16-3#

568 96. B: So schaut es für mich aus, wie wenn der Measurement-Strang, der Kontrollstrang, und der
569 Emergency Stop ter ab und der Emergency Stop oder Abbruchstrang parallel gleichzeitig
570 ausgeführt werden müssen. #01:01:46-4#

571 97. I: Gut, dass wäre auch grundsätzlich die Idee dahinter. Aber/ #01:01:56-1#

572 98. B: Rein prozesstechnisch muss zuerst der Abbruchstrang abgerufen werden beziehungsweise
573 abgearbeitet werden, damit diese Regelung auch freigegeben wird. #01:02:08-3#

574 99. I: Also würde man das auch so wirklich programmieren in der Prozessleittechnik? #01:02:13-5#

575 100. B: Ja. #01:02:15-2#

576 101. I: Okay. #01:02:16-9#

577 102. B: Ein Interlock, eine Verriegelung, ist immer/ wird vorher abgearbeitet, bevor überhaupt der
578 Sollwert auf den jeweiligen Block, ob das jetzt ein Ventil ist oder ein Regelkreis ist, wird immer
579 vorher abgearbeitet. Weil der Regler, also, es ist eine sequentielle Abarbeitung. Der Prozessor
580 kann nicht alle Wege gleichzeitig abarbeiten. Oder alle Zustände. Das heißt, er muss je nachdem
581 es die Vorgabe von den Programmierern ist, zuerst die Messe, die Eingänge abarbeiten. Dann
582 muss er die Logik, ob es jetzt der PID-Regler ist oder ein Auf-Zu-Ventil, ganz egal, und zum
583 Schluss arbeitet da ja die Ausgabe. Das heißt, 'Fahr das Ventil jetzt auf', 'Stell den Reglerausgang
584 auf 92 Prozent'. Und zu diesen Eingangssignalen gehört immer die Verriegelung, der Notaus, je
585 nachdem. #01:03:43-2#

586 103. I: Aber das heißt, was natürlich zuallererst kommen würde, wäre die Zustandsabfrage. So oder
587 so. Natürlich. (...) Also Zustandsabfrage im Sinne von Überprüfen, ob der Notaus aktiviert wurde.
588 #01:04:03-1#

589 104. B: Genau. Aber das würde ich so verstehen, dass der Notaus zum Beispiel diesen Emergency
590 Stop darstellt. Und der muss für einen Europäer oben links sein. #01:04:21-6#

591 105. I: Okay, ja. (...) Weil bei Measure hätten wir es so vorgesehen, dass grundsätzlich auch solche
592 Sachen, weil das sind ja im Grunde auch einfach Inputs für das System, auch so abgefragt
593 werden. Was nur/ Vom Technischen her jetzt, wenn man das wirklich technisch umsetzen würde,
594 hätten diese Daten, die in der SPS beim Abbild gemacht werden. Ich weiß ja nicht, wie es bei
595 einer Sicherheits-SPS dann läuft. Aber hätten die irgendeine gewisse Priorität? #01:04:55-5#

596 106. B: Die Prioritäten werden durch die Blockabarbeitung der Script/ In einem Script ist das einfach.
597 In einem Script ist alles, was oben steht, wird zuerst abgearbeitet. Alles //I: Natürlich. //, was
598 unten steht, wird zum Schluss abgearbeitet. #01:05:08-8#

599 107. I: Okay. #01:05:10-0#

600 108. B: In einem Blockdiagramm wird/ kann natürlich die Abarbeitungsreihenfolge einstellen. Ob es
601 jetzt eine SPS ist oder ein Prozessleitsystem, ist das natürlich möglich. #01:05:27-3#

602 109. I: Das heißt, theoretisch könnte, wenn man jetzt einen Programmierer hat, der nicht so erfahren
603 ist oder der vielleicht Fehler macht. Könnte der mit einem Funktionsblockdiagramm diese
604 Reihenfolge auch falsch darstellen? Das könnte passieren. #01:05:41-5#

605 110. B: Natürlich. #01:05:42-4#

606 111. I: Okay, das ist gut zu wissen. Aber auch jetzt, bei solchen kritischen Sachen wie zum Beispiel
607 einen Notstopp, würde erst das Prozessabbild gemacht werden. Also man würde erst auf die
608 Inputs warten und damit natürlich weiterarbeiten. #01:05:57-5#

609 112. B: Genau. Man hat immer den Input, dann die Calculation. Und dann den Output. Und der
610 Output ist natürlich abhängig von dem, was der Input sagt. Und in dem Fall ist es der Emergency
611 Stop, ist ein Input. Weil ich sage dem Regelkreis, stoppe jetzt. #01:06:22-9#

612 113. I: Genau. Okay, das haben wir nämlich bei einem etwas längeren Beispiel aus den
613 Siemens-Unterlagen auch so dargestellt, aber da kommen wir eh gleich dazu. Okay, na gut. Also
614 für die Modellierung ist das auf jeden Fall gut zu wissen als Input, dass man das machen kann.
615 Oder so machen sollte. #01:06:42-5#

114. B: Und. Unten, wo dieser Kreis zusammengeführt wird mit diesem Gateway. Entweder der Notstopp oder er regelt. Rein jetzt vom Logischen betrachtet her. Entweder ich hole mir den Wert von der Messung. Und dann sage ich ihm, was der Regler machen soll. Oder 'Geh auf Notstopp'. Also so, das verstehe ich unter einem Regelkreis. #01:07:17-1#
115. I: Okay, ja. (...) Okay, gehen wir ein bisschen weiter. Nachdem das Ereignis für eine Abbruchbedingung ausgelöst wurde, können Tasks zur Aufräumroutine abgearbeitet werden, bevor der Zyklus beendet oder der Prozess vollständig beendet wird. Damit wird auch eines der Kriterien, dass wir oben genannt haben, für Aufräumprozesse erfüllt. Jetzt kommt ein Prozess, bei dem dieser Aufräum-Tasks definiert sind. Das heißt, hier hätten wir im Fall nach dem Emergency Stop zum Beispiel einen Service Call, der halt bestimmen soll, dass man eine gewisse Shutdown-Sequenz zum Beispiel initiiert für Vessel 1. Also für den ersten Tank oder den ersten Behälter beispielsweise. Und das wird halt auch nur einmalig ausgeführt, bevor wir aus diesem Closed Loop System hinaus springen. Warum definiert man das hier nicht zum Beispiel hier dazwischen? Weil das eine Sequenz sein soll, die speziell für diese Abbruchbedingungen definiert werden soll. Also man kann auch mehrere Abbruchbedingungen parallel im System haben, die natürlich dann auch alle überprüft werden. Also das heißt jetzt zum Beispiel nicht Emergency Stop 1, sondern zum Beispiel mehrere Emergency Stops, die man aktivieren kann. Also mehrere Notschalter oder was auch immer, andere Bedingungen, zum Beispiel oder zum Beispiel der Ausschalter. Ein. Aus. Dann könnte man einzelne Routinen danach auch noch starten und die würden natürlich alle aus diesem System heraus führen. 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. Das habe ich eh gemeint. Und damit erfüllen wir auch eines der oberen Features, was zu Verständlichkeit eigentlich auch gedacht war. Und jetzt kommen wir endlich zu den Beispielen für die Prozessmodelle. Ich werde Ihnen Prozessbeispiele zeigen, die mit den in unserer Arbeit vorgestellten Erweiterungen modelliert sind. Ich möchte, dass Sie sich die Modelle ansehen und mir sagen, was Sie aus Ihnen herauslesen können und ob die Modelle den notwendigen Informationsgehalt für die Modellierung der zugrundeliegenden Regelungsprozesse erfüllen. Vorab wird Ihnen zum jeweiligen Prozess erklärt, was abgebildet werden soll. Und ich würde Sie einfach bitten, dass Sie mir offenes Feedback zu den Modellen geben. Das Erste ist etwas einfacher. Ein einfacheres Beispiel, das auf Unterlagen von MathWorks, also von MATLAB, basiert. Im Grunde geht es hier um eine einfache PI-Temperaturregelung für einen Wärmetauscher. Wie gesagt basierend auf den Unterlagen von MathWorks. Die Temperatur einer Flüssigkeit in einem Rührkessel wird mittels Wärmetauscher geregelt oder beeinflusst. Der über den Wärmetauscher eingebrachte Wärmestrom wird über ein Ventil, also über einen Dampfstrom, gesteuert. Der zu beachtende störende Umgebungseinfluss ist die schwankende Temperatur der zugeführten Flüssigkeit von oben. Das heißt, es kommt noch ein zusätzlicher Inflow dazu. Wir gehen davon aus, dass der konstant ist. Der Tank ist also isoliert. Das heißt, wir haben jetzt keine störenden Einflüsse von außen von der Tankwand zum Beispiel. Wir haben eine kurze Darstellung. (...) Ja, ich nehme an, Sie können sich da jetzt eh etwas darunter vorstellen, hoffe ich halbwegs. Es ist ein einfacheres Beispiel. Wir haben wieder einige Datenelemente vorgegeben. Zahlenwerte, eventuell die für die Berechnung des PI-Reglers wichtig sind. Das kann man hier auch definieren, da gehe ich jetzt nicht im Detail mit hinein, weil die Werte, die Sie hier sehen, sich eher aus der mathematischen Berechnung ergeben. Also was man hier definieren muss. Dann haben wir hier ein Beispiel für die verschiedenen End Points, die man modellieren kann. Also zum Beispiel wie gesagt, HTTP Get, eine IP-Adresse. In dem Fall ist es halt Localhost. Ich könnte über den Port 8080 zum Beispiel einen Datenserver laufen lassen. Der gibt mir aus Prozess 1 aus den Parametern halt die Temperatur von dem einen Tank, wie auch immer. Das Prozessmodell sieht hier wie folgt aus. Wir haben einmal die Messung, Get für Temperatur 1. Dann könnten wir zum Beispiel noch eine Datenumwandlung oder eine Berechnung, was auch immer, einfügen, das ist halt hier jetzt mal gekennzeichnet durch Conversion to Process Value. Wir haben dann noch die Disturbance, die wir messen. Wir rechnen hier einfach damit, dass wir nur die Temperatur des Inflows, der von

672 oben kommt, messen. Wir haben dann die entsprechende Regelung. Wir haben einen
 673 PI-Controller hier drinnen. Wir haben eine Conversion, wenn wir das wieder wollen von
 674 PI-Controller, Manipulating Value to Output. Das heißt, im Grunde wird das dann in das Signal
 675 umgewandelt, das wir dann wirklich an den jeweiligen Aktor ausschicken können. Das heißt
 676 eigentlich an das Ventil für den Darmstrom. Und als Abbruchbedingung haben wir einfach nur
 677 Stop Activated. Also hier ist nicht so viel Detail aus der MathWorks-Bibliothek vorgegeben. Hier
 678 haben wir einfach diese Abbruchbedingung dazu gegeben nur als Beispiel, was man modellieren
 679 könnte. Und dann könnte ein Script gestartet werden, das zum Beispiel Execute Shutdown
 680 Sequenz, oder so etwas, heißt. Wie gesagt, man könnte dann das Script wie normalen Code
 681 einfach darstellen und hier den entsprechenden Wert ausrechnen lassen für den PI-Regler. Also
 682 das Proportional- und das Integral-Glied. Und die Konversion sind dann teilweise einfach nur
 683 Überschreiben in andere Variablen. Aber natürlich könnte man auch komplexere Versionen,
 684 komplexere Scripte, hier hinein modellieren. Also das ist wirklich ein eher einfacherer Prozess.
 685 Ich würde Sie jetzt bitten, das Modell nach folgenden Kriterien auf einer Skala von Eins bis Fünf
 686 zu bewerten. Also mit Eins, schlecht oder sehr schlecht quasi, und mit Fünf, sehr gut. Also sehr
 687 schlecht, nicht gut, weder noch, gut und sehr gut. Und zwar. Würden Sie sagen/ Sagen Sie mir
 688 einfach, wenn ich das Modell nochmal ins Bild holen soll. Würden Sie sagen, dass das Modell
 689 verständlich für Sie ist? Auf einer Skala von Eins bis Fünf. Für den Prozess, der dahinter abläuft.
 690 #01:14:30-2#

691 116. B: Na ja. Man sieht ja Tanktemperatur als Prozessgröße. Man sieht die Zuflusstemperatur
 692 eigentlich als Störgröße. Die ist aber auf dem PI-Controller ja nicht wirklich als Störgröße zu
 693 erkennen. #01:15:00-4#

694 117. I: Rein aus der grafischen Darstellung? #01:15:05-2#

695 118. B: Aus der grafischen Darstellung. Wo ist der Prozessgröße und wo ist die Störgröße? Das fehlt
 696 mir. (...) Was soll passieren, wenn die Messung ausfällt? Irgendwer hat das Kabel
 697 durchgeschnitten, abgerissen. Grundsätzlich ist es so, dass die Controller mehrere Modis haben.
 698 Also man kann den Controller ja ein- und ausschalten, im Sinne von, ich stelle ihn auf auto, ich
 699 stelle ihn auf manuell. Das sieht man da auch nicht, was für einen Modus dieser Controller hat.
 700 Also ich würde sagen, Drei. #01:16:03-4#

701 119. I: Okay. Sehr gut, das ist wirklich guter Input. Dankeschön. Übersichtlichkeit. Würden Sie sagen,
 702 können Sie das Gesamtsystem auf einen Blick erfassen? Wenn ich nochmal hochgehe.
 703 #01:16:21-5#

704 120. B: Eigentlich nicht, weil die vorhin genannten Punkte. Es sind Informationen aufgelistet, zum
 705 Beispiel, dass die Messung mit einem Hertz abgefragt wird, einmal in der Sekunde abgefragt
 706 wird. Ich sage mal, das brauche ich jetzt als Programmierer nicht unbedingt. Es würde
 707 ausreichen, wenn dieser Regelkreis, zum Beispiel bei dem Gateway, dort in ein/ Also dort die
 708 Frequenz von der, die Abtastrate, beziehungsweise Abarbeitungszeit von dem Block gezeichnet
 709 sein würde. Und nicht für jede Messung, und nicht für den Controller. Weil der Loop kann nicht
 710 schneller durchlaufen wie das langsamste Glied. Verstehen Sie was, ich/? #01:17:42-4#

711 121. I: Ja, ich verstehe schon. Sie würden das jetzt/ Wenn ich jetzt wieder auf die SPS-Beispiele
 712 hergehe, wäre das quasi in einem Zyklusmodell drinnen? Also es gibt ja auch bei, soweit ich weiß,
 713 bei der SPS die Möglichkeit, dass man verschiedene Zyklusdauern //B: Zyklus. Genau. In
 714 verschiedene Zeitscheiben legt, diesen Zyklus. // definiert. #01:18:02-9#

715 122. B: Also es sind unnötige Informationen. Aber andererseits Informationen, also, nicht vorhandene
 716 Punkte, wie zum Beispiel, Modus von dem von dem Regler. Was soll passieren, wenn die
 717 Tanktemperatur ausfällt? Was soll passieren, wenn der Sollwert verstellt worden ist? Welche
 718 Sollwertrampen hat der Controller et cetera? Also da gibt es einige/ Ich sage einmal, ich könnte
 719 rein nach den Angaben keinen Regelkreis programmieren. #01:18:57-2#

720 123. I: Okay, gut. #01:19:01-3#

721 124. B: Also eher Zwei. #01:19:07-2#

722 125. I: Und von der Einfachheit her, das wäre der nächste Punkt. Könnte man das Modell eventuell
723 sogar noch einfacher darstellen? #01:19:16-9#

724 126. B: Ja, nach den alten, bewährten Regelstrukturen. #01:19:25-6#

725 127. I: Okay, als Beispiel, wie würden Sie das dann machen? #01:19:32-7#

726 128. B: Also ich würde das Ganze nicht in einem Flow Chart ablaufen lassen, sondern eher in einem
727 Regelschemata. Wo der Tank mit seinen Leitungen hier weiter angesiedelt sind, weiter unten.
728 Also jetzt grafisch dargestellt. Und die jeweiligen Größen als Störgröße beziehungsweise als
729 Prozesswert, die eben diesem Reglerblock zugeführt werden. #01:20:11-7#

730 129. I: Okay, verstehe ja. Von der Darstellungsweise her, ja. Also das wäre natürlich für
731 Automatisierungstechniker wahrscheinlich verständlicher, weil sie diese Darstellungsweise auch
732 schon gewöhnt sind. #01:20:30-2#

733 130. B: Natürlich auch. Aber wie gesagt, ich sehe in dem Diagramm hier, in dem Modell, bei Ihnen
734 nicht, welche Messung WAS beim Controller auswirkt. Auf ersten Blick. Ich müsste mich da in die
735 Syntax vom PID, also oder in dem Fall vom PI-Regler, hinein arbeiten, damit ich dann verstehe,
736 wie die Messungen auf den Regler einwirken. #01:21:06-0#

737 131. I: Okay, also man müsste das Script einfach lesen für die Verständlichkeit. #01:21:11-0#

738 132. B: Genau. #01:21:12-4#

739 133. I: Ja, verstehe. Logik. Wie klar war für Sie, was parallel und was sequentiell passiert?
740 #01:21:23-4#

741 134. B: Das ist ganz gut. Ja, sagen wir einmal Vier. Deswegen Vier, weil da dieser Abbruchbefehl
742 eigentlich sequenziell zu dem Mess- beziehungsweise zu den Control-Strang passiert.
743 #01:21:50-8#

744 135. I: Ja natürlich, das ist der Punkt, den wir eh schon besprochen haben. Das stimmt, ja.
745 Erweiterbarkeit. Könnte man dem Modell noch etwas hinzufügen, was den Informationsgehalt
746 verbessern würde? Im Grunde haben Sie das eigentlich eh schon erwähnt, dass da ein paar
747 Informationen fehlen, aber vielleicht konkret nochmal. #01:22:11-3#

748 136. B: Also von der Bewertung her eher Zwei. Mir fehlen, wie die Messgrößen auf den Regler
749 endwirken. Wie sich der Regler verhält. Ob es ein Direct Acting, Indirect Acting ist. Das heißt, ob
750 der Fehler Mit Minus Eins oder mit Plus Eins multipliziert wird. Das heißt, ist die
751 Regelabweichung positiv, geht der Reglerausgang in die Richtung Plus oder Richtung Minus? Das
752 müsste man auch natürlich aus der Syntax heraus lesen. Das ist ein ganz ein wichtiger Punkt bei
753 einem Regler. Zum Beispiel wenn man eine Regelklappe hätte, für diese/ Also jetzt als Bauweise,
754 müsste man dort Grenzen einführen. Für ein Ausgangs-High-Low begrenzen. Weil die
755 Regelklappe ja unter oder über einem bestimmten Wert nicht mehr den Durchfluss regeln kann.
756 Oder die Temperatur regeln kann, den Dampfdurchfluss. Das fehlt auch noch, sage ich einmal,
757 im ersten Blick. #01:23:57-8#

758 137. I: Das heißt, eigentlich die Werte/ eigentlich das Fenster für den Wert, in dem sich der Stellwert
759 bewegen darf. Habe ich das jetzt richtig // B: Genau. Genau. // verstanden? Okay. Ja, das hätten
760 wir eigentlich/ Es ist auf den ersten Blick nicht ersichtlich, das stimmt schon. Wo habe ich jetzt
761 das Beispiel? Einen Moment. Das wäre dieses Minimum und Maximum gewesen, also Upper
762 Limit und Lower Limit. Aber es ist natürlich nicht sofort ersichtlich. #01:24:28-9#

763 138. B: Na ja, ich meine Upper und Lower Limit. Verstehe ich jetzt, dass der Prozesswert begrenzt
764 wird. #01:24:38-9#

765 139. I: Ach so, nein, in dem Fall nicht, dass wir wirklich für den Regelwert. #01:24:43-9#

766 140. B: Also für den Reglerausgang? #01:24:45-5#

767 141. I: Ja ja, das wäre für den Regler eigentlich gedacht. Deswegen sind diese Werte nämlich auch
768 beim Regler. #01:24:51-0#

769 142. B: Okay. Ja, beim Regler kann man den Prozesswert Lower Limit, Upper Limit begrenzen. Man
770 kann auch den Sollwert Upper Limit, Lower Limit begrenzen. #01:25:02-8#

771 143. I: Okay. #01:25:04-4#

772 144. B: Und man kann den Ausgang natürlich auch auf Upper und Lower Limit begrenzen.
773 #01:25:08-3#

774 145. I: Dann ist das nicht eindeutig, ja. #01:25:12-2#

775 146. B: Ich sehe jetzt auch nicht auf Anhieb, wie groß der Sollwert im/ oder wovon der Sollwert
776 abhängig ist. Ist der Sollwert eine/ Kommt der Sollwert aus einer Berechnung? Zum Beispiel. Um
777 auf dieses Bierbrauen zurück zu kommen. Ist bei der Bierart A ist der Kühltank auf zwölf Grad,
778 bei Bier B auf 15 Grad. Und sind das jetzt fixe Werte oder kommen diese Werte aus irgendeiner
779 Berechnung, was irgendeine F von X Berechnung/ #01:26:04-4#

780 147. I: Also der Sollwert/ Wenn wir jetzt bei dem einfachen Beispiel davon ausgehen, dass der keiner
781 Berechnung folgt, sondern das wirklich einfach ein statischer Wert ist, dann würde der sich bei
782 den Datenelementen wiederfinden. Eigentlich. Also ich weiß nicht, ob man das jetzt/ Da bin ich
783 halt sehr schnell darüber gegangen, aber ja. #01:26:28-1#

784 148. B: Set Point 210. Ist das ein Sollwert aus einer Berechnung oder ist der ein fixer Wert?
785 #01:26:38-4#

786 149. I: In dem Fall wäre es ein statischer Wert. #01:26:39-9#

787 150. B: Okay. #01:26:44-2#

788 151. I: Also es gebe auch jetzt kein Script oder keine Task oder so etwas, keine Aufgabe, die hier
789 hinein modelliert wäre, um diesen Wert zu beeinflussen. #01:26:55-3#

790 152. B: Diese Conversion. PI to Output. Was soll das genau machen? #01:27:09-8#

791 153. I: Das ist einfach um darzustellen, dass hier/ Wie gesagt, aus der MathWorks-Bibliothek gibt es
792 da jetzt nicht so viele Vorgaben, was in diesem Prozess wirklich passiert. Hier soll das einfach
793 zeigen, dass bevor man ein Argument an den Aktor schickt, je nachdem, wie das Interface von
794 dem Aktor wirklich funktioniert, dass man das eventuell noch umwandelt. Also wenn ich sage,
795 ich kriege hier zum Beispiel, keine Ahnung, einen analogen Wert heraus und müsste den
796 eventuell noch in ein digitales Signal umwandeln oder dergleichen. #01:27:44-7#

797 154. B: Es gibt aber keinen PI-Controller, der ein digitales Ausgangssignal hat. #01:27:50-8#

798 155. I: Ja, okay, dann ist das in dem Fall einfach eine, ja, es ist ein schlechtes Beispiel. Aber es gibt
799 noch den Zwischenschritt, um zu zeigen, dass man etwas umwandeln kann, wenn man das
800 möchte. #01:28:03-4#

801 156. B: Man könnte eine Begrenzung einbauen, und das ist gleichzeitig der Lower Limit oder Upper
802 Limit. Man könnte sagen, dort statt dieser Conversion, UL ist gleich hundert Prozent, LL ist gleich
803 null Prozent. (...) Also ich aus Programmiersicht sehe es nicht als so übersichtlich.
804 Beziehungsweise es sind zu viele Informationen, die man nicht braucht auf dem Bild zu sehen
805 und zu wenige Informationen, die man braucht. #01:29:03-4#

806 157. I: Okay. (...) Dann kommen wir zum nächsten Modell, das wird jetzt etwas komplexer. Das
807 Modell basiert auf der Beschreibung des Heizprozesses, den wir aus Schulungsunterlagen der
808 Firma Siemens entnommen haben. Es handelt sich hier ebenfalls um eine Temperaturregelung
809 für einen Rührreaktor. Die Regelung wird in diesem Beispiel aber mit einem PID-Regler
810 durchgeführt beziehungsweise ist auch eine Handsteuerung vorgesehen sowie ein Pulsgenerator.
811 Die Heizung erfolgt nicht über einen Wärmetauscher, sondern über ein Heizelement. Und
812 weiters gibt es auch Verriegelungsbedingungen, die definiert sind. Als Basis für die
813 Prozessmodellierung wurden die Beschreibungen wie gesagt aus den Schulungsunterlagen für
814 Simatic PCS 7 herangezogen. Unser Prozessmodell wird mit einer automatischen Steuerung
815 modelliert, die mit Umschalten auf Handsteuerung aus dem Closed Loop System ausbricht. Das
816 heißt, wir gehen davon aus, dass dieses Closed Loop System jetzt nur für den automatischen
817 Betrieb gedacht ist. Weiters wird der Prozess für einen Reaktor, also nur für einen Reaktor,
818 abgebildet und nicht für einen zweiten Reaktor. Da gehe ich mal ein bisschen drüber. Hier haben
819 wir wieder die Datenelemente. Kann man in einem eigenen Fenster definieren. Die
820 Verriegelungsbedingungen, die vorgesehen sind, sind so etwas wie ein Mindestfüllstand. Das
821 haben wir hier zum Beispiel, also Level Reaktor Min. 200 Milliliter. Beziehungsweise eine
822 maximale Temperatur, also Temperatur Reaktor Max von sechzig Grad. (...) Es ist jetzt ein
823 bisschen ein komplexeres Modell. Das heißt, wir haben insgesamt eins, fünf Stränge für die
824 Zustandsabfrage, einen Regelungsstrang. Und insgesamt auch fünf Abbruchbedingungen. Wir
825 haben einmal die Messung von Temperatur Reaktor In. Also, Temperatur des Reaktors. Also was
826 im Reaktor aktuell der Zustand ist. Wir haben dann ein Script für die Datenumwandlung. Also
827 wieder, da kann man definieren, was auch immer man normalerweise für einen
828 Umrechnungsprozess durchführen würde oder wie man das auch immer programmieren würde.
829 Wir haben dann einmal die Überprüfung des Füllstandes im Reaktor. Wir haben die Überprüfung
830 des aktuellen Betriebsmodus. Wir haben dann auch die Überprüfung des Emergency Stops und
831 auch die Überprüfung des Hauptschalters. Die Regelung ist, in dem Fall wie gesagt, wird
832 durchgeführt mit einem PID-Controller. Wir haben dann noch eine Pulsweitenmodulation
833 angehängt. Und danach wird der entsprechende Output, das Output-Abbild geschrieben. Die
834 Abbruchbedingungen, die überprüft werden, sind ob der Main Switch, also der Hauptschalter,
835 auf ausgeschaltet steht. Ob, wie gesagt, der Emergency Stop aktiviert ist. Ob die Temperatur des
836 Reaktors über der maximalen Temperatur liegt beziehungsweise ob das Fülllevel des Reaktors
837 unter dem minimal Füllstand liegt und ob eventuell der Operationsmodus auf manuell, also auf
838 Handsteuerung umgestellt wird. Und entsprechend sind dann hier Service Calls mit Script
839 eingefügt, die wieder durch Subprozesse ersetzt werden können und die einfach für einmalige
840 Aufrämroutinen stehen, die man in diesem Prozess auch definieren kann. Wieder das Script für
841 den PID-Controller in dem Fall. Und im Grunde würde ich Ihnen hier die gleichen Fragen stellen
842 wie beim oberen Modell schon. Ganz kurz bevor wir aber anfangen, würde ich gerne noch
843 einmal die Aufnahme stoppen und eine neue starten. #01:33:30-0#

844 UNTERBRECHUNG (5 Minuten Pause)

845 158. I: Okay, nimmt wieder auf. Und wir sind jetzt gerade bei der Bewertung vom zweiten
846 komplexeren Modell. Bitte. #00:00:10-4#

847 159. B: Ja, wie gesagt. Ich finde da ist eine A 4 Seite/ Oder zur Verständlichkeit. Je komplexer die
848 Prozesse sind, desto unverständlicher ist es, meiner Meinung nach. Also vorher auf Drei, jetzt
849 eher auf Zwei. #00:00:32-8#

850 160. I: Ja. #00:00:33-5#

851 161. B: Übersichtlichkeit. Für einen, so einen kleinen Regelprozess für einen kontinuierlichen Ablauf
852 braucht man da eine komplette A 4 Seite, wo noch einmal eine oder zwei A 4 Seiten,
853 Fragezeichen, danach auftauchen, würde ich das auch noch einmal eher auf sehr schlecht
854 bewerten. Man könnte es natürlich einfacher darstellen. Mit einem Blockdiagramm, oder dem
855 Regelschemata. (...) Zur Logik. Was parallel und was sequentiell abgearbeitet wird, würde ich
856 sagen, das wie vorher schon besprochen. Diese Abbruchbedingungen/ Also zum Beispiel. Wenn
857 man den Regler auf manuell schaltet, dann würde diese Closed Loop immer wieder anspringen
858 nach einer Sekunde Abarbeitungszeit würde er wieder oben hinein springen und schauen, ob
859 dieser Baustein oder dieser PID noch immer auf Man ist oder er die Regelung aufnehmen kann.
860 #00:02:20-1#

861 162. I: Also so würden Sie es jetzt verstehen, aus diesem ersten Blick heraus? #00:02:23-9#

862 163. B: Nein, so würde ich es eher zeichnen. #00:02:27-1#

863 164. I: Okay, ja. #00:02:28-0#

864 165. B: Das heißt, wenn Sie kurz hinauf gehen. Der/ Ein bisschen hinunter. Dieser letzte Strang mit der
865 Operation Mode Manuell. #00:02:39-2#

866 166. I: Ja? #00:02:39-8#

867 167. B: Das würde natürlich in dieses Closed Loop Gateway hinein springen. Jedoch der Level
868 beziehungsweise Temperatur Max wieder heraus springen aus diesem Gateway. Oder der Main
869 Switch. Weil der Modus muss jede Sekunde abgefragt werden oder in der Zykluszeit abgefragt
870 werden. Und wenn der auf Man steht, darf dann der Regler natürlich nicht wirken. Aber der
871 muss wieder nach einer Sekunde abgefragt werden. Wenn der Emergency Stop auf true geht,
872 also der Notschalter, Notaus, gedrückt wird, dann muss dieser Regelkreis sofort beendet werden.
873 Und das sieht man da überhaupt nicht. #00:03:35-3#

874 168. I: Okay, nur ganz kurz zu manual. Also wir gehen davon aus, dass dann das/ In dem Fall gehen
875 wir davon aus, dass wir einfach mit Automation (Automatic) Mode fahren. Und wenn er dann
876 auf manual schalten würde, würde dazwischen eventuell etwas passieren. Aber wir können ja
877 dann, wenn dieser Prozess beendet ist, im Grunde in ein ähnliches Modell wieder hineinspringen,
878 wo wir aber dann mit manual arbeiten. Also wir würden aus dieser Prozessdarstellung dann
879 einfach hinaus wechseln, weil sich eine der Bedingungen einfach ändert. Das ist jetzt nur in dem
880 Fall ein Beispiel. Dass es praktisch ein bisschen anders ablaufen würde, ja, okay. Aber im Grunde
881 würden wir eigentlich in ein ähnliches System wieder hineinspringen, wo genau wieder der
882 PID-Regler wirken würde. Das im Grunde ähnlich modelliert wäre. Nur dass hier dann im Manual
883 Modus gearbeitet wird. Aber vielleicht dann unter anderen Bedingungen. #00:04:32-2#

884 169. B: Wenn ich einen Regler auf Hand nehme, dann will ich als Bediener ja nicht, dass der Regler
885 etwas macht. Und solange der auf Hand steht, soll der Reglerausgang dort bleiben, wo er gerade
886 steht. #00:04:57-9#

887 170. I: Okay, ich habe es jetzt so verstanden, dass man einfach unter anderen Bedingungen mit dem
888 PID-Regler arbeiten würde. Dann habe ich das falsch verstanden. #00:05:07-5#

889 171. B: Ja, aber in dem Moment, wo der Mode in Manual wäre, dann würde er ja jede Sekunde
890 wieder hinaufspringen und nicht diesen Gateway verlassen und den Prozess beenden. Sondern
891 dieser/ Wann schalte ich einen Regler auf manual? Wenn zum Beispiel aus irgendeinem Grund
892 dieser ganze Regelkreis anfängt zu schwingen. Weil aus irgendeinem Grund, keine Ahnung, die
893 Temperatur aus den bestimmten Werten gefallen ist und, also die Heizkreistemperatur, und der
894 Regler nicht dazu ausgelegt ist oder die Regelparameter nicht dazu ausgelegt sind, dass der
895 Regler unter den Bedingungen arbeitet. Dann würde der Bediener den Regler auf Man nehmen,
896 kurz warten, oder die Störung beheben, und dann wieder auf Automatik schalten. Aber dann
897 würde er diesen Prozess wieder aufnehmen, diesen Regelprozess. Das heißt, dass der letzte

898 Strang eigentlich da richtig gezeichnet ist und die anderen Stränge sollten, so wie ich es vorher
899 erwähnt habe, nach dem Gateway eingeführt werden. Das heißt, die anderen Stränge beenden
900 diesen Closed Loop Strang. Das würde ich, den würde ich anders machen. // I: Okay, verstehe. //
901 Weil man muss da priorisieren zwischen einer Verriegelung und einer Modus-Umschaltung. Und
902 wo soll der Ausgang des Reglers hin springen, wenn der Main Switch zum Beispiel auf Off
903 geschaltet wird? Soll der auf 15 Prozent gehen? Soll der auf null Prozent, soll der auf hundert
904 Prozent gehen? Soll der dort bleiben, wo er ist? #00:07:24-8#

905 172. I: Das kann man modellieren, wenn man das möchte. Stellvertretend hierfür steht hier halt Send
906 Message to Operator. Dass eventuell hier ein Eingreifen notwendig ist. #00:07:36-3#

907 173. B: Das fehlt mir da ein bisschen auf diesem Schema. #00:07:51-5#

908 174. I: Also das könnte ich als Beispiel, damit das verständlicher ist, hier dazu modellieren, meinen Sie?
909 Dass es für andere Leute, die sich eher mit Steuerungstechnik auseinandersetzen, einfach
910 verständlicher wird, was man dann macht? #00:08:04-9#

911 175. B: Hm. (zustimmend) #00:08:05-5#

912 176. I: Okay, das ist gut. Als Beispiel. #00:08:08-4#

913 177. B: Gut. Wo waren wir jetzt? Beim dritten Punkt, oder? #00:08:20-5#

914 178. I: Einfachheit, //B: Einfachheit.// glaube ich, haben wir gerade gemacht. Ja. Logik war es. Wird
915 klar, was parallel und was sequentiell passiert? #00:08:28-9#

916 179. B: Jein (Ja + Nein). Wieder aus dem Grund, dass diese Abbruchbedingungen unten rechts stehen
917 und die eigentlich vor der Abarbeitung von dem Regler passieren sollten. Also. Je komplexer die
918 Darstellung wird, desto unklarer wird die Darstellung, also wird das Modell. Meiner Meinung
919 nach. Also, ich glaube vorher habe ich da gut bewertet. Dann eher eine Zwei, nicht so gut.
920 #00:09:25-8#

921 180. I: Und dann der letzte Punkt, Erweiterbarkeit. Könnte man im Modell noch etwas hinzufügen?
922 #00:09:31-8#

923 181. B: Ich glaube, das habe ich da vorhin auch schon erwähnt. Wie gesagt. Bezüglich dem Regler.
924 Direct Acting, Indirect Acting. Also je nachdem, ob der Fehler positiv ist oder negativ, der
925 Ausgang dann steigt oder fällt. In dem Fall ist der Modus auch angeführt worden. Dann. Welches
926 ist der Prozesswert, welches ist die Störgröße? Wie wird die Störgröße, mit welchem Kp-Wert
927 wird die Störgröße aufgeschaltet. Also wie welchem Gain. In dem Fall ist es ein Gain. Ja. Also. Für
928 mich würden da einige Punkte sicherlich noch fehlen, die/ Also eher nicht so gut. Zwei.
929 #00:10:43-1#

930 182. I: Okay. Das heißt, wenn man zum Beispiel im mathematischen Modell, das heißt den
931 Regelalgorithmus darstellen könnte im Script und hier der Gain-Wert, also das Kp, als eigener
932 Parameter ersichtlich wäre, dann wäre die Information zwar vorhanden, im Script, als Variable,
933 die man auch von Anfang an definieren muss, weil man kennt ja dann das Reglermodell. Aber es
934 wäre nicht gleich ersichtlich. Das ist glaube ich ein großer Nachteil. Weil wir haben ja auch beim
935 letzten Modell über das Script gesprochen. #00:11:22-3#

936 183. B: Genau. Also. //I: Okay. // Ich habe in meiner ganzen fünfzehnjährigen Laufbahn noch nie
937 einen PID-Algorithmus programmieren müssen. Weil das meistens Standardblöcke von den
938 SPS-Prozessleitsystem-Hersteller sind. Aber dafür Reglersinn sehr sehr oft umstellen müssen.
939 #00:11:50-0#

940 184. I: Das heißt, wäre es hier sinnvoller zu sagen, ich habe wie bei einem Funktionsblockdiagramm
941 den Subprozess oder den Funktionsblock für einen PID-Regler und habe diese Input-Parameter.

942 Die könnte ich dann wie zum Beispiel bei den Events die Attribute, also ich gehe da ganz kurz
943 hoch und zeige Ihnen, was ich meine. Dass ich es ähnlich darstelle, wie auf diese Art und Weise,
944 dass das seitlich als Fenster erscheint und dass ich hier sage, ich habe diese Input-Parameter, die
945 notwendig sind, um das zu definieren. #00:12:28-5#

946 185. B: Zum Beispiel. Ja. #00:12:32-4#

947 186. I: Das könnte man machen. Und der Funktionsblock, also das mathematische Verhalten, ist
948 grundsätzlich immer gleich. Der Algorithmus ist immer gleich. //B: Genau. // aber ich kann die
949 Parameter beeinflussen. //B: Genau. // Also DAS könnte dann helfen in dieser Darstellung.
950 #00:12:43-9#

951 187. B: Ja. Also. So wie bei einer objektorientierten Programmierweise. Da muss ich ja nicht wissen,
952 wie mein Regler abläuft, also wie der Algorithmus gebildet ist oder wie dieser Algorithmus
953 berechnet wird. Aber ich möchte wissen, wie zum Beispiel die Klassenvariablen, wie zum Beispiel
954 der Gain, wie die Vorhaltezeit, wie die Nachstellzeit-Konstante, Reglersinn, welchen Modus ich
955 zur Auswahl habe, was der Prozesswert ist. Wird der Prozesswert vom Soll subtrahiert oder wird
956 der Sollwert vom Prozesswert subtrahiert? Also der Reglersinn. Und so weiter. #00:13:36-7#

957 188. I: Das heißt, die einfache Script-Darstellung würde nicht genügen, sondern/ #00:13:42-8#

958 189. B: Naja, jetzt müsste ich im Script nachlesen, wie dieser Block, wieder dieser Regler arbeitet.
959 #00:13:51-7#

960 190. I: Ja. #00:13:52-2#

961 191. B: Oder zum Beispiel, der PID-Aufbau. Ist der D-Anteil nur auf dem Prozesswert oder ist der auf
962 dem Fehler? Ist der P-Anteil rein nur auf dem Fehler oder ist der PI-Anteil auf dem Fehler
963 bezogen? #00:14:11-3#

964 192. I: Das heißt, für die Verständlichkeit wäre es hier/ Also wenn man jetzt davon weggeht, dass man
965 das Ganze wirklich wie in einer Programmiersprache darstellt, also in einem Script. Würde es
966 vielleicht helfen, wenn man das Ganze so aufbaut wie von der grafischen Darstellungsweise wie
967 bei einem Blockdiagramm? Eventuell? #00:14:31-0#

968 193. B: Eher. #00:14:33-9#

969 194. I: Eher. Okay, das ist gut. Könnte //B:Weil/ // man machen. #00:14:39-0#

970 195. B: /ja ein kontinuierlicher Prozess ja keinen Anfang und kein Ende hat. #00:14:45-2#

971 196. I: Genau. #00:14:47-2#

972 197. B: Und in dem Modell möchte man unbedingt auf Biegen und Brechen einen kontinuierlichen
973 Prozess in einen diskreten Prozess hineinpacken. #00:15:04-3#

974 198. I: Okay, das wäre genau NICHT unser Ziel gewesen. Das ist interessant, dass Sie das so sehen,
975 aber das ist dann schlecht. Dann haben wir etwas falsch gemacht. (lacht) #00:15:12-2#

976 199. B: Ja, weil sehr viele Zusammenhänge nicht ersichtlich sind. #00:15:21-7#

977 200. I: Zum Beispiel, wenn ich nachfragen darf? #00:15:27-6#

978 201. B: Ja, zum Beispiel Reglersinn. (...) Ob der Ausgang jetzt bei einem positiven Fehler Richtung Plus
979 geht oder Richtung Minus geht. #00:15:43-9#

980 202. I: Also, Sie meinen für einen kontinuierlichen Prozess wäre charakteristisch, zu sehen auf einen
981 Blick, wie der Regler sich wirklich verhält? #00:15:58-6#

982 203. B: Genau. #00:15:59-3#

983 204. I: Okay, das ist gut, ja. Und die Darstellung hier wahrscheinlich, mit den Abbruchbedingungen
984 und dem Ganzen, führt eher wahrscheinlich zu Ihrer Aussage, dass Sie sagen würden, man
985 versucht das Ganze diskret abzubilden? #00:16:15-1#

986 205. B: Genau. #00:16:16-8#

987 206. I: Okay, ja, das kann ich nachvollziehen. Ja. #00:16:19-5#

988 207. B: Also, eine Abbruchbedingung. Oder eine Verriegelung hat in einem Regelschemata nicht viel
989 verloren. Weil ein Regelschemata soll sich ja nur auf den kontinuierlichen Prozess beziehen und
990 nicht auf die (unv.). Und in einem kontinuierlichen Prozess gibt es keine Events. #00:16:49-4#

991 208. I: Okay. Wenn man jetzt nur natürlich das mathematische Verhalten oder an sich die Regelung,
992 an sich darstellen möchte, ja, natürlich. Das kann ich nachvollziehen. Aber wenn man den
993 ganzen Prozess darstellen möchte, könnte es ja doch passieren, dass einmal irgendetwas dazu
994 führt, dass es leider nicht dazu kommt, dass die Anlage Twenty-Four-Seven //B: Ja, nur von der
995 Darstellungsart.// gleich läuft. Ja. Okay. Passt. Dann gehen wir weiter. So viel ist das nicht mehr.
996 (lacht) Wären Sie aufgrund dieser Erweiterungen bereit, diese Modellierungsmethode in Ihrem
997 Arbeitsalltag einzuführen? #00:17:42-2#

998 209. B: Es würde so viel Papier erzeugen, weil die Prozesse teilweise so komplex sind, dass/ Eher nicht.
999 #00:17:56-5#

1000 210. I: Okay, // B: Also, ich würde/ // aufgrund des Umfanges? #00:17:59-1#

1001 211. B: Des Umfanges. Sie haben da jetzt einen Tank, eine Heizungssteuerung von einem Tank, auf
1002 einer A4-Seite gezeichnet. Da gibt es wesentlich komplexere Prozesse. Also, ich würde sagen,
1003 dass das Problem ist, dass je komplexer dieser Prozess ist, desto mehr Papieraufwand muss da
1004 betrieben werden. Und desto Fehler/ oder desto breiter ist das Interpretationsfenster, das man
1005 dann von einem Management oder von einem Programmierer dort/ (...) Es wird auch falsch
1006 interpretiert werden. Je komplexer dieses Problem ist. Desto mehr Fehler würden auftauchen
1007 oder entstehen. #00:19:18-0#

1008 212. I: Okay, verstehe ich. Frage Sieben. Wie gut beschreiben diese Erweiterungen Ihrer Meinung
1009 nach ein Kontrollsystem für diese Beispiele, die ich angeführt habe? Also einmal das etwas
1010 einfachere Beispiel und einmal das mit den etwas komplexeren Bedingungen. Also Modell Eins
1011 und Modell Zwei. Wieder auf einer Skala von Eins bis Fünf, wobei Fünf sehr gut ist und Eins sehr
1012 schlecht. #00:19:44-8#

1013 213. B: Also // I: Modell Eins. // Modell Eins ist eher Drei weil es von der Komplexität nicht so
1014 umfangreich ist. Und Modell Zwei ist eher sehr schlecht. #00:20:03-1#

1015 214. I: Okay. Könnten Sie kurz sagen/ Also Sie meinen bei Modell Zwei, weil es doch etwas komplexer
1016 dargestellt ist und der Aufwand sich irgendwie dann auch nicht rechnet, es so darzustellen?
1017 #00:20:16-9#

1018 215. B: Genau. #00:20:17-7#

1019 216. I: Okay. Frage Acht haben Sie schon beantwortet, zu genüge, würde ich sagen. Das können wir
1020 überspringen. Wenn Sie Erfahrung in der Regelungstechnik haben, was würden Sie empfehlen
1021 um diese Erweiterungen zu ergänzen um sie für Ingenieure attraktiver zu machen? #00:20:33-8#

1022 217. B: Wie gesagt, eher die Regelung, also den kontinuierlichen Prozess, von den Events abgliedern.
1023 Und sagen, okay, auf einer Seite betrachte ich den kontinuierlichen Prozess, mit den Reglern in
1024 Blockdiagrammen. Und in der anderen Seite betrachten wir die Events, in Ablaufdiagrammen
1025 oder Funktionsblöcken. So wäre es sicher für einen Softwareentwickler einfacher. Und/ (...) Eher
1026 Zwei. #00:21:41-4#

1027 218. I: Okay. Frage Zehn. Ich möchte zum Schluss noch etwas genauer auf die Modelle mit unseren
1028 Erweiterungen eingehen und Sie bitten, nochmal kurz auf einer Skala von Eins bis Fünf die
1029 folgenden Fragen einfach zu beantworten mit einer Bewertung. Und zwar, wie einfach ist in den
1030 gezeigten Modellen nachzuvollziehen, dass die einzelnen Abläufe parallel und unabhängig
1031 voneinander laufen? Das sind jetzt im Grunde die Punkte, die wir ganz am Anfang
1032 durchgesprochen haben, ob die wichtig oder unwichtig sind. Die werden jetzt hier ein bisschen
1033 behandelt. #00:22:20-2#

1034 219. B: (...) Nicht so gut. Weil ja die, wie gesagt mit dem Abbruchkriterium, und dem/ also dass dieser
1035 vor gereiht werden müsste, dem eigentlichen Input, Output. Oder Input, Calculation, Output.
1036 #00:22:55-7#

1037 220. I: Wie einfach ist es, zu definieren, wann eine Anpassung am System erfolgt? Das heißt, wann
1038 eine Regelung wirklich passieren würde. #00:23:05-2#

1039 221. B: Ich verstehe die Frage nicht ganz. #00:23:13-7#

1040 222. I: Okay, also, soll heißen. Das ist das eher gemeint bezogen auf sequentiell und parallel. Also ob
1041 es/ auf welcher Basis, auf welchem Messergebnis basierend die Regelung abfolgen würde. (...) Oder
1042 erfolgen würde. #00:23:39-8#

1043 223. B: Nachdem nicht erkennbar ist, so bei Modell Eins oder bei Modell Zwei, welches die Störgröße
1044 ist, welches die Prozessgröße ist, würde ich auch eher nicht so gut, also eher Zwei definieren.
1045 #00:24:07-2#

1046 224. I: Wie einfach ist es, die maximale Dauer, also Zykluszeit einer Anpassung, zu definieren? (...) Das
1047 war das mit der Frequenz. #00:24:20-7#

1048 225. B: Naja, dieser Regelkreis ist genauso schnell wie das langsamste Glied. Das heißt, es bringt
1049 nichts wenn ich zehn mal in der Sekunde messe, aber der Regler nur einmal in der Sekunde
1050 durchläuft. #00:24:36-9#

1051 226. I: Okay, ja. #00:24:38-3#

1052 227. B: Und es bringt auch nichts wenn ich zehn mal in der Sekunde den Regler/ Also die Darstellung
1053 ist/ beziehungsweise wenn ich die Zeitscheiben ändern würde in dem Beispiel betrachtet von
1054 einer Sekunde auf eine Zehntelsekunde, also auf zehn Hertz, dann müsste ich jeden Strang
1055 einzeln ändern. #00:25:12-9#

1056 228. I: Genau. Das wäre es. Ja. #00:25:15-2#

1057 229. B: Und es ist sinnlos, jeden Strang einzeln zu ändern, weil so wie ich es vorher schon erwähnt
1058 habe, die einzelnen Stränge ja voneinander also miteinander in Abhängigkeit stehen.
1059 #00:25:30-7#

1060 230. I: Unter gewissen Bedingungen, ja. #00:25:36-5#

1061 231. B: Nein, es ist immer so. Es bringt nichts, wenn ich ihn zehn mal in der Sekunde den Messwert
1062 abfrage, wenn ich ihn nur einmal über den PID laufen lasse. #00:25:47-3#

1063 232. I: Man kann in diesem System natürlich auch einen zusätzlichen Regelkreis hinein modellieren
1064 und sagen, dass der etwas schneller funktionieren sollte. Also man könnte auch sagen, ich hänge
1065 hier zwei Stränge mit Control hinein und könnte dem eine andere Zykluszeit eigentlich geben.
1066 Wäre das dann vielleicht sinnvoller? #00:26:10-2#

1067 233. B: Das Problem ist, // I: Also derweil haben wir es so vorgesehen. // wie soll ich das dann auch
1068 am Papier darstellen? Da werden die Symbole und die Stränge so viel und so umfangreich, dass
1069 man da nichts mehr erkennen wird, auch wenn man es auf einem A0 (Papierformat) ausdrucken
1070 würde. Also dann würde die Komplexität so zunehmen und die Übersichtlichkeit von dem
1071 Schema rapide abnehmen. #00:26:47-0#

1072 234. I: Okay. Wie einfach ist es zu definieren, dass/ unter welchen Bedingungen sämtliche repetitiven
1073 Aufgaben beendet werden? Also die Abbruchbedingungen. (...) Die zu definieren. Wie würden
1074 Sie das empfinden? #00:27:13-2#

1075 235. B: Man sieht, dass es eine Abbruchbedingung gibt. Man kann auch erkennen, welche
1076 Abbruchbedingung da jetzt aufgelistet wird, aber nicht wirklich/ Also man müsste wirklich in den
1077 Syntax von diesem Block hineinschauen, um wirklich zu erkennen oder festzustellen, welche
1078 Bedingungen dann oder wie diese Bedingungen dann aufgebaut ist. Weil teilweise gibt es
1079 verschachtelte Bedingungen über verschiedene Ebenen mit Gattern zum Beispiel. Weiß ich nicht.
1080 Ich habe mehrere Zuflussventile, die dann noch unterschiedlich, also in unterschiedliche Parallel-
1081 beziehungsweise Serienschaltung. Und dann müsste man diese dann alle in eine Syntax
1082 verpacken und das wäre ziemlich komplex. Also für einfache Beispiele ist die Darstellung auch
1083 schon sehr sehr komplex. Und wenn dieses Beispiel noch um einiges komplexer wie es in der
1084 Realität ist, wäre, dann würde man das fast nicht mehr lesen können. Das menschliche Gehirn ist
1085 nicht wirklich, im Gegensatz zu einem Computer, leistungsstark was solche Probleme betrifft.
1086 Und dann muss man versuchen, die Probleme so klein wie möglich zu gestalten und die
1087 Darstellungen auch so klein wie möglich zu gestalten, damit man diese auch erkennt. Man muss
1088 jedes Problem in Kleinstprobleme aufteilen, dass diese vom menschlichen Gehirn so klar und so
1089 schnell wie möglich abgearbeitet werden. Und DAS ist in diesem Modell nicht der Fall.
1090 #00:29:25-2#

1091 236. I: Okay. #00:29:29-4#

1092 237. B: Also auch eher Zwei. #00:29:33-3#

1093 238. I: Wie einfach ist es zu definieren, dass nachdem so eine Abbruchbedingungen getriggert wird,
1094 so ein Ereignis getriggert wird, zu definieren, dass danach Aufräumaufgaben einmalig zu erfolgen
1095 haben? Das wären dann die Aufgaben nach den jeweiligen Abbruchbedingungen. #00:29:52-2#

1096 239. B: Da gilt auch wieder dasselbe. Man muss da wirklich in der Syntax dann nachschauen, was da
1097 genau passieren soll. Man sieht nicht auf einem Blick, was genau mit diesem Regler oder mit
1098 diesem Loop passieren soll. #00:30:11-3#

1099 240. I: Okay. Würde es helfen, wenn man diese Aufgaben, die nach so einem Ereignis kommen, in
1100 einen Subprozess verwandelt, den man dann quasi aufklappt und dann Schritt für Schritt sieht,
1101 was hier wirklich geschehen soll? In Form von Service Calls beispielsweise. Der Befehl geht
1102 dorthin. Nachdem man da eine Bestätigung zurück bekommen hat, geht der Befehl dorthin. Und
1103 dieses Script wird beispielsweise ausgeführt. Würde das vielleicht helfen? #00:30:40-2#

1104 241. B: (...) Ja. Also, so ein Script-Befehl mit einem Namen zu versehen und zu sagen, okay, dieser
1105 Script-Befehl macht das und das. Wenn dich das mehr interessiert, was dieser Script-Befehl
1106 macht, dann schau doch hinein. Das ist schon in Ordnung. Aber dann verläuft man sich so
1107 dermaßen in die Tiefe, dass man nicht mehr weiß was oben passiert. Je mehr Ebenen man hat,
1108 wo irgendwelche Scripts oder Subscripts verwendet werden, hat man das Problem, das man sich
1109 einfach dann/ nach kurzer Zeit nicht mehr weiß, was die übergeordnete Ebene dann gemacht
1110 hat beziehungsweise die oberste Ebene. Also mit den Ebenen muss man auch bisschen

1111 aufpassen, meiner Meinung nach. Dass man dann nicht zu sehr vom eigentlichen Ablauf sich
1112 wegbewegt. #00:32:02-3#

1113 242. I: Ja, das könnte dann schnell unübersichtlich werden. Ja. #00:32:06-8#

1114 243. B: Weil, ich sage einmal so. Kurzzeitgedächtnis. Wenn man von einem menschlichen Gehirn/
1115 wenn dieser/ dann zu viele Informationen in kürzester Zeit dort hinein gestopft werden, dann
1116 fallen die nächsten Informationen wie beim Fifo (First in, first out) einfach heraus. Und man
1117 muss immer wieder hinausgehen, in die erste Ebene. 'Was war da noch schnell nochmal? Okay,
1118 das. Und die nächste Ebene hat das und das gemacht. Okay. Und dann das und das. Aha, gut.
1119 Und was macht die Ebene? Ah, was war jetzt die erste Ebene?' So auf die Art. #00:32:51-3#

1120 244. I: Verstehe ich, ja. Das wäre natürlich einerseits eine Möglichkeit, um ein bisschen aufzuräumen,
1121 das gesamte Modell. Aber andererseits führt das natürlich wieder zu höherer Komplexität.
1122 #00:33:05-4#

1123 245. B: Genau. #00:33:07-1#

1124 246. I: Ja, damit kommen wir eigentlich eh zum letzten Punkt in dieser Liste. Und das haben Sie
1125 eigentlich eh auch schon ein bisschen beantwortet. Wie einfach ist es, komplexe Abläufe im
1126 Kontext von kontinuierlichen Prozessen mit diesen Erweiterungen zu beschreiben? Also im
1127 Grunde, wie einfach ist es, einen komplexen kontinuierlichen Prozess darzustellen mit diesen
1128 Erweiterungen? Das haben Sie jetzt eigentlich eh schon ein bisschen ausgeführt. Aber vielleicht
1129 sagen Sie da auch noch etwas ganz kurz dazu. #00:33:35-8#

1130 247. B: Ja, wie gesagt, ich würde eher diese kontinuierlichen Prozesse von den event-getriggerten
1131 Prozessen trennen. In der Darstellung, weil ansonsten man die Übersicht nicht behält, was da
1132 eigentlich passieren soll. Also würde ich auch eher eine Zwei abgeben. #00:34:06-9#

1133 248. I: Okay, danke das war es eigentlich eh mit den Fragen, auch von meiner Seite. Vielen Dank, dass
1134 Sie sich Zeit genommen haben. Also ich bin sehr dankbar für das Interview. Jetzt würde ich die
1135 Aufnahme beenden und würde Sie jetzt nochmal um Feedback vielleicht im Allgemeinen zum
1136 Interview bitten. Also ganz kurz, dass Sie das sehen. Aufnahme beendet.
